ПРОЕКТИРОВАНИЕ ПРОЦЕССА ПРОЕКТИРОВАНИЯ

записки компьютерного эксперта

titula_Proektirov-processa.indd 1 31.10.2012 14:44:52

THE DESIGN OF DESIGN

Essays from a Computer Scientist

Frederick P. Brooks, Jr.
University of North Carolina at Chapel Hill

Addison-Wesley

titula_Proektirov-processa.indd 2 31.10.2012 14:44:52

ПРОЕКТИРОВАНИЕ ПРОЦЕССА ПРОЕКТИРОВАНИЯ

записки компьютерного эксперта

Фредерик П. Брукс, мл.

ВИЛЬЯМС

Москва · Санкт-Петербург · Киев
2013

titula_Proektirov-processa.indd 3 31.10.2012 14:44:52

ББК 88.5.75
Б89
УДК 681.3.07

Издательский дом "Вильямс"
Зав. редакцией *С.Н. Тригуб*
Перевод с английского и редакция *К.А. Птицына*
По общим вопросам обращайтесь
в Издательский дом "Вильямс" по адресам:
info@williamspublishing.com, http://www.williamspublishing.com

Брукс, Фредерик.

Б89 Проектирование процесса проектирования: записки компьютерного эксперта. : Пер. с англ. — М. : ООО "И.Д. Вильямс", 2013. — 464 с.: ил. — Парал. тит. англ.

ISBN 978-5-8459-1792-8 (рус.)

ББК 88.5.75

Все названия программных продуктов являются зарегистрированными торговыми марками соответствующих фирм.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотокопирование и запись на магнитный носитель, если на это нет письменного разрешения издательства Addison-Wesley Publishing Company, Inc.

Authorized translation from the English language edition published by Addison-Wesley Publishing Company, Inc., Copyright © 2010 by Pearson Education, Inc.

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise.

Russian language edition is published by Williams Publishing House according to the Agreement with R&I Enterprises International, Copyright © 2013

Научно-популярное издание

Фредерик П. Брукс

Проектирование процесса проектирования: записки компьютерного эксперта

Литературный редактор ***И.А. Попова***
Верстка ***Л.В. Чернокозинская***
Художественный редактор ***Е.П. Дынник***
Корректор ***Л.А. Гордиенко***

Подписано в печать 29.10.2012. Формат 70x100/16
Гарнитура Palatino Linotype. Печать офсетная
Усл. печ. л. 29,0. Уч.-изд. л. 24,25.
Тираж 1000 экз. Заказ № 0000

Первая Академическая типография "Наука"
199034, Санкт-Петербург, 9-я линия, 12/28

ООО "И. Д. Вильямс", 127055, г. Москва, ул. Лесная, д. 43, стр. 1

ISBN 978-5-8459-1792-8 (рус.)
ISBN 978-0-201-36298-5 (англ.)

© Издательский дом "Вильямс", 2013
© Pearson Education, Inc., 2010

Оглавление

I. Модели проектирования 31

Глава 1. Постановка задачи проектирования 33

Глава 2. Инженерный подход к проектированию — рациональная модель . . 43

Глава 3. Недостатки модели водопада 51

Глава 4. Требования, искушения и соглашения 69

Глава 5. Поиск лучших моделей процесса проектирования 81

II. Организация совместной работы в режиме непосредственного общения и в дистанционном режиме 91

Глава 6. Совместная работа в проектировании 93

Глава 7. Дистанционное сотрудничество. 121

III. Перспективы создания теории проектирования 137

Глава 8. Проявления рационализма и эмпиризма в проектировании 139

Глава 9. Модели пользователей — лучше неправильные, чем неопределенные 149

Глава 10. Лимитируемые ресурсы — дюймы, унции, биты, доллары 157

Глава 11. Полезные свойства ограничений 165

Глава 12. Эстетика и стиль в техническом проектировании. 177

Глава 13. Применение образцов в проектировании. 193

Глава 14. В чем могут ошибаться опытные проектировщики 209

Глава 15. Становление проектирования как самостоятельной научной дисциплины 219

Глава 16. Представление процессов и критериев проектирования 229

IV. Система проектирования домов, идеальная с точки зрения специалиста по информатике 247

Глава 17. Система проектирования домов, идеальная с точки зрения специалиста по информатике, — от замысла до компьютерного представления. 249

Глава 18. Система проектирования домов, идеальная с точки зрения специалиста по информатике, — от компьютерного представления до замысла. 267

Глава 19. Создание выдающихся проектов великими проектировщиками, а не участниками обычного процесса проектирования. 279

Глава 20. Предпосылки появления выдающихся проектировщиков. 293

Глава 21. Изучение практического примера. Дом на побережье “View/360” 311

Глава 22. Изучение практического примера. Пристройка крыла к дому. . . . 331

Глава 23. Изучение практического примера. Реконструкция кухни 351

Глава 24. Изучение практического примера. Архитектура System/360. 369

Глава 25. Изучение практического примера. IBM Operating System/360. 387

Глава 26. Изучение практического примера. Проектирование книги Computer Architecture: Concepts and Evolution 403

Глава 27. Изучение практического примера. Организация объединенного вычислительного центра. Вычислительный центр ассоциации трех университетов 411

Глава 28. Рекомендуемое чтение 423

Приложение А. Благодарности 427

Приложение Б. Библиография 431

Предметный указатель. 445

Содержание

Об авторе 21
Предисловие 23
Для кого предназначена эта книга 23
Зачем нужна еще одна книга по проектированию 24
К какому жанру относится эта книга? 26
Благодарности 28
Примечания 29

I. Модели проектирования 31

Глава 1. Постановка задачи проектирования 33
Прав ли Бэкон 34
Что такое проектирование 34
Место проекта в реальной жизни. Концепция проекта 36
Что является действительно значимым 38
Размышления о процессе проектирования 39
Разновидности проектирования 40
Сравнение системного и художественного проектирования 40
Массовое, адаптивное и оригинальное проектирование 40
Примечания и ссылки 40

Глава 2. Инженерный подход к проектированию — рациональная модель 43
Модель 44
Происхождение рациональной модели 46
Преимущества рациональной модели 47
Примечания и ссылки 48

Глава 3. Недостатки модели водопада 51
Отсутствие полного понимания конечной цели в начале проектирования 52
Построение дерева проектирования, как правило, осуществляемое в процессе поиска 53
Применение узлов, фактически являющихся не проектными решениями, а предварительными полными проектами 54
Отсутствие возможности оценивать функцию качества инкрементно 55
Отсутствие окончательно сложившегося перечня требований, весовые коэффициенты которых оставались бы неизменными 56
Продолжающееся изменение ограничений 57
Критический анализ других рациональных моделей 60
Итоговый критический анализ Шёна 62

Дальнейшее использование рациональной модели, несмотря на все ее недостатки и критические замечания 62
Что же из этого следует? Неужели столь важен сделанный нами выбор модели процесса проектирования? 63
Примечания и ссылки 65

Глава 4. Требования, искушения и соглашения 69
Фильм ужасов 70
Атмосфера закулисных интриг 71
Борьба с раздутыми и неконкретными требованиями 72
Искушение 74
Соглашения 75
Модель, основанная на заключении контрактов 75
Примечания и ссылки 77

Глава 5. Поиск лучших моделей процесса проектирования 81
Необходимость в определении доминирующей модели 82
Коэволюционная модель 83
Модель разработки Реймонда по принципу функционирования базара 84
Описание работы модели 85
Сильные стороны 86
Организация разработки по принципу функционирования базара 86
Спиральная модель Боэма 88
Модели процесса проектирования — сводка аргументов, приведенных в главах 2–5 89
Примечания и ссылки 89

II. Организация совместной работы в режиме непосредственного общения и в дистанционном режиме. 91

Глава 6. Совместная работа в проектировании. 93
Преимущества и недостатки организации работы на основе привлечения большого количества людей. 94
Коллективное проектирование как современный стандарт организации работы. 95
Причины перехода в организации технического проектирования от применения отдельных специалистов к коллективам 96
Затраты на организацию совместной работы 98
Достижение концептуальной целостности как основная цель 100
Возражения. 102
Способы достижения концептуальной целостности при коллективном проектировании 102
Современное проектирование как междисциплинарное согласование. 103

Системный архитектор . . . 103
Проектирование единообразного пользовательского интерфейса . . . 104
Условия успешной совместной работы . . . 105
Определение потребностей и пожеланий со стороны заинтересованных лиц . 105
Принципиально различающиеся подходы к концептуальным исследованиям . . . 106
Обзор проекта . . . 110
Отсутствие возможности обеспечения совместной работы непосредственно в самом проектировании . . . 112
Особая важность соблюдения принципа индивидуальной ответственности при концептуальном проектировании . . . 114
Отличительные особенности коллективов из двух человек . . . 114
Особенности организации труда программистов . . . 115
Примечания и ссылки . . . 116

Глава 7. Дистанционное сотрудничество . . . 121

Причины применения дистанционного сотрудничества . . . 122
Специализация . . . 122
Работа на дому . . . 122
Круглосуточная работа . . . 122
Денежные затраты . . . 123
Политика . . . 123
Распределение работ по территориально разрозненным подразделениям. Распределенная разработка семейства компьютеров IBM System/360, 1961–1965 гг. . . . 124
Ввод принципов дистанционного сотрудничества в действие . . . 126
Крайняя важность личных встреч! . . . 126
Качественные интерфейсы . . . 127
Технологии дистанционного сотрудничества . . . 128
Возможность успешного применения всего спектра средств организации связи . . . 128
Проведение видеоконференций . . . 130
Примечания и ссылки . . . 132

III. Перспективы создания теории проектирования . . . 137

Глава 8. Проявления рационализма и эмпиризма в проектировании . . 139

Сравнение рационализма и эмпиризма как систем взглядов . . . 140
Проектирование программного обеспечения . . . 141
Сам я — убежденный эмпирик . . . 141
Примеры применения подходов на основе рационализма, эмпиризма и проверки правильности в других областях проектирования . . . 143
Примечания и ссылки . . . 144

Глава 9. Модели пользователей — лучше неправильные, чем неопределенные 149
Явно заданные модели пользователя и использования 150
Это — вполне оправданный подход, но применяется ли он в действительности? 150
Коллективное проектирование 150
Организация проектирования в условиях отсутствия необходимых фактических данных 152
Руководствоваться предположениями! 152
Неправильная модель лучше, чем неопределенная! 153
Примечания и ссылки 154

Глава 10. Лимитируемые ресурсы — дюймы, унции, биты, доллары . . . 157
Общее определение понятия лимитируемого ресурса 158
Ресурсы, которые сложно оценивать в деньгах 158
Разновидности денежных затрат и других ресурсов, которые могут оцениваться в деньгах 159
Изменение значимости лимитируемого ресурса 160
Что из этого следует? 161
Явное определение состава лимитируемых ресурсов 161
Широкий контроль 162
Централизованное управление 162
Примечания и ссылки 163

Глава 11. Полезные свойства ограничений 165
Ограничения 166
Целесообразность ограничений 167
Парадокс проектирования — более высокая сложность проектирования продуктов универсального назначения по сравнению со специализированными 171
Сеть взаимосвязей 173
Примечания и ссылки 174

Глава 12. Эстетика и стиль в техническом проектировании 177
Эстетика в техническом проектировании 178
В чем выражается красота логических рассуждений? 179
Краткость 179
Структурная ясность 180
Непротиворечивость 181
Что представляет собой качественная архитектура вычислительной системы? . 181
Дополнительные преимущества единообразия 183
Стиль в техническом проектировании 183
Что такое стиль? 184
Свойства стилей 186

Необходимость документального описания стиля для достижения его единообразия 187
Способы достижения хорошего стиля. 189
Примечания и ссылки 189

Глава 13. Применение образцов в проектировании 193

Принципиальная новизна лишь немногих проектов 194
Функции образцов. 194
Характерные особенности проектирования аппаратного и программного обеспечения 195
Изучение используемых образцов 195
Изучение проектных требований к образцам 197
Компьютеры первого поколения 198
Компьютеры третьего поколения. 199
Виртуальная память 199
Революционные изменения, связанные с появлением мини-компьютеров . . 200
Принципиальные изменения, связанные с появлением микрокомпьютеров и архитектуры RISC 200
Возможности усовершенствования принципов проектирования на основе образцов в проектных дисциплинах 201
Коллекции образцов 201
Применение коллекции образцов 201
Особенности проектирования программного обеспечения 202
Преимущества образцов — возможность применить готовый материал, оценить оригинальность и проверить значимость достигнутого успеха 203
Несколько иной взгляд на вещи 203
Использование готовых материалов 204
Оригинальность и значимость достигнутого успеха 204
Примечания и ссылки 205

Глава 14. В чем могут ошибаться опытные проектировщики. 209

Ошибки 210
Самый неудачный из когда-либо созданных компьютерных языков 211
Краткая характеристика языка JCL 211
Основные недостатки языка JCL 212
Причины столь значительных недоработок при создании JCL. 214
Усвоение уроков. 216
Примечания и ссылки 216

Глава 15. Становление проектирования как самостоятельной научной дисциплины 219

Становление проектирования как направления деятельности, отличного от реализации и эксплуатации. 220
Причины становления проектирования как самостоятельного направления . 221

Отрицательные последствия отделения проектирования от других дисциплин . 221
Способы исправления существующего положения 222
Примечания и ссылки 226

Глава 16. Представление процессов и критериев проектирования 229
Введение 230
Линеаризация сети знаний 230
Сбор данных, касающихся процесса проектирования 232
Принятая нами методика изучения процесса проектирования дома 233
Определение понятия дерева проектирования 234
Новые данные, полученные в процессе проектирования 235
Проектирование должно служить не только выполнению известных, но и раскрытию до сих пор неизвестных требований 235
Проектирование — это не просто выбор среди альтернатив, но и осознание их существования 237
Изменение дерева проектирования в ходе развития проекта и способы представления этого изменения 237
Сравнение дерева решений с деревом проектирования 239
Сравнение модульных и тесно интегрированных проектов 240
Программа Compendium и альтернативные инструментальные средства 241
Программа Task Architect 241
Инструментальные средства управления проектированием 241
Система IBIS и ее потомки 242
Программа Compendium 242
DRed как одно из перспективных инструментальных средств 243
Примечания и ссылки 246

IV. Система проектирования домов, идеальная с точки зрения специалиста по информатике. 247

Глава 17. Система проектирования домов, идеальная с точки зрения специалиста по информатике, — от замысла до компьютерного представления 249
Постановка задачи 250
Мысленное представление 250
Последовательное приближение 250
Библиотека моделей 252
Потенциальные недостатки подхода, основанного на последовательном приближении 253
Применение способов визуального представления для ввода данных в компьютер 253
Ритмика глагола–существительного 254
Задание глаголов 255
Определение существительных 256

Определение текста 258
Определение наречий 259
Определение точки зрения и способа представления. 260
Виртуальные путешествия внутри объекта 260
Осмотр снаружи 262
Примечания и ссылки 264

Глава 18. Система проектирования домов, идеальная с точки зрения специалиста по информатике, — от компьютерного представления до замысла 267

Двухстороннее взаимодействие 268
Визуальные индикаторы — одновременное применение большого количества окон 268
Изображение координатной сетки и представление на экране чертежей . . . 268
Двухмерное контекстное представление. 269
Трехмерное представление. 270
Осмотр снаружи 271
Представление рабочей книги. 271
Представление спецификации 273
Звуковое сопровождение. 273
Восприятие с помощью осязания 274
Обобщения 275
Осуществимость. 275
Примечания и ссылки 276

Глава 19. Создание выдающихся проектов великими проектировщиками, а не участниками обычного процесса проектирования 279

Великие проекты и процессы разработки продуктов 280
Преимущества и недостатки сложившихся процессов разработки продуктов. . 281
В чем процессы разработки продуктов препятствуют созданию выдающихся проектов 281
Чем оправдано существование сложившихся процессов разработки продуктов. 283
Процедурные рамки сковывают творчество, но без определенной регламентации невозможно обойтись. Как выйти из этого положения? 287
Выдающиеся проекты создаются выдающимися проектировщиками. Найдите их!. 287
Необходимость в поддержке великих проектировщиков со стороны требующих инноваций авторитетных руководителей 288
Способы организации процесса, способствующего созданию выдающихся проектов 288
Обеспечение концептуальной целостности — проявление большего доверия главному проектировщику при создании проекта 289
Примечания и ссылки 290

Глава 20. Предпосылки появления выдающихся проектировщиков . . . 293
Выбор правильных способов обучения проектированию 294
Привлечение к работе преимущественно с учетом задатков в области проектирования 296
Целенаправленная подготовка будущих специалистов 296
Как сделать принцип "разных лестниц" действующим и признанным 297
Планирование формальных мероприятий по обучению 298
Планирование приобретения опыта работы во многих областях 299
Планирование творческих отпусков, проводимых за пределами организации 300
Творческий характер руководства работами творческого направления 300
Настоятельная необходимость во всем оберегать талантливых людей 301
Избавление талантов от всего постороннего 301
Защита проектировщиков от неумелых руководителей 302
Избавление проектировщиков от выполнения руководящих функций 303
Как развить свои способности в качестве проектировщика 303
Развитие способности представлять проекты в виде эскизов 304
Стремление к получению обоснованной критики собственных проектов . . . 304
Изучение образцов и прецедентов 304
Проект самообразования — поэтажный план дома площадью 1 000 квадратных футов 306
Примечания и ссылки 306

Глава 21. Изучение практического примера. Дом на побережье "View/360" 311
Основные факты и особенности 312
Вводные сведения и условия выбора местоположения 312
Местоположение 312
Владельцы 313
Проектировщики 313
Даты 313
Члены семьи, проживающие в этом месте, по состоянию на август 1972 года 313
Цели 313
Другие цели 313
Возможности 314
Ограничения 314
Законодательные и юридические требования 314
Проектные решения 315
Расчет оптимальной ширины фасада 318
Определение размеров дома в плане 320
Неудачные попытки 320
Изменения в проекте после проектных работ до начала строительства 321

Изменения в проекте после возведения каркаса дома и первоначального заселения . . . 322
Оценка достигнутых результатов (по прошествии 37 лет) . . . 323
Положительные стороны . . . 323
Полезность . . . 325
Прочность . . . 326
Если бы можно было начать сначала . . . 327
Общие усвоенные уроки . . . 328

Глава 22. Изучение практического примера. Пристройка крыла к дому . . . 331
Основные факты и особенности . . . 332
Введение и предыстория . . . 333
Местоположение . . . 333
Владельцы . . . 333
Проектировщики . . . 333
Строительная компания . . . 333
Даты . . . 333
Вспомогательный веб-сайт . . . 333
Предыстория . . . 334
Задачи . . . 335
Исходные задачи . . . 335
Дополнительно намеченные задачи . . . 337
Ограничения . . . 337
Параметры, не рассматриваемые как ограничения . . . 337
События . . . 338
Проектные решения и доработки . . . 338
Исследования . . . 338
Поэтапное решение задач проектирования . . . 340
Восточное крыло . . . 340
Расположение комнат в западной половине дома с учетом их назначения . . . 341
Изменение подхода. Отказ от рассмотрения бюджета в качестве одного из проектных ограничений . . . 341
Вновь обнаруженное требование. Увеличение площади шкафов для верхней одежды . . . 342
Окончательное решение по размещению комнат с учетом их функционального назначения . . . 343
Изменения, происшедшие в ходе строительства . . . 345
Оценка — успехи и неустраненные недостатки . . . 345
Новые функции . . . 347
Общие усвоенные уроки . . . 347
Примечания и ссылки . . . 348

Глава 23. Изучение практического примера. Реконструкция кухни . . . 351
Основные факты и особенности. 352
Общие сведения и предыстория 352
Местоположение. 352
Владельцы 352
Проектировщики 352
Даты 352
Предыстория 352
Задачи. 353
Возможности 354
Ограничения. 354
Регламентация проекта с учетом такого важного показателя, как ширина помещения 356
Необходимая ширина в направлении с севера на юг 356
Предварительное проектирование. 356
Альтернативные решения по проекту кухни с учетом ширины 357
Итоговый проект, рассчитанный на большую ширину 358
Распределение объектов в кухне с учетом длины помещения 358
Другие проектные решения 359
Освещение. 360
Оценка 361
Использование чертежей, средств автоматизированного проектирования, моделей, макетов и виртуальной среды в проектировании. 362
Данные, полученные в результате моделирования в виртуальной среде. 364
Общие усвоенные уроки 365
Примечания и ссылки 366

Глава 24. Изучение практического примера. Архитектура System/360 369
Основные факты и особенности. 370
Общие сведения и предыстория 370
Владелец 370
Проектировщики 370
Даты 370
Предыстория 370
Задачи. 373
Основные цели. 373
Другие важные цели 373
Открывающиеся возможности по состоянию на июнь 1961 г. 374
Сложности и ограничения 374
Наиболее значительные проектные решения 375
Наиболее важные этапы 379

Оценка 379
Устойчивость 379
Полезность — конкурентоспособность на любом рынке 380
Положительные итоги 381
Общие усвоенные уроки 384
Примечания и ссылки 385

Глава 25. Изучение практического примера. IBM Operating System/360 . . 387
Основные факты и особенности 388
Общие сведения и предыстория 389
Семейство компьютеров System/360 390
Контекст, в котором разрабатывалось программное обеспечение, по состоянию на 1961 г. 391
Новые направления, получившие дальнейшее развитие 393
Проектные решения 395
Структура системы 395
Оценка 398
Преимущества 398
Недостатки проекта 398
Недостатки в поддержке процессов 400
Проектировщики 400
Ключевые исполнители 400
Общие усвоенные уроки 401
Примечания и ссылки 401

Глава 26. Изучение практического примера. Проектирование книги Computer Architecture: Concepts and Evolution 403
Основные факты и особенности 404
Введение и предыстория 405
Авторы введения и основного текста 405
Даты 405
Предыстория 405
Задачи 406
Возможности 406
Ограничения 407
Проектные решения 407
Оценка 408
Усвоенные уроки 408

Глава 27. Изучение практического примера. Организация объединенного вычислительного центра. Вычислительный центр ассоциации трех университетов 411
Основные факты и особенности 412
Общие сведения и предыстория 413

Местоположение 413
Владельцы — резиденты штата Северная Каролина 413
Разработчики организационной структуры 413
Даты 414
Предыстория 414
Цели 414
Основные цели 414
Другие задачи 415
Возможности 415
Ограничения 416
Проектные решения 417
Правила голосования, предусмотренные для совета 417
Распределение полномочий 419
Оценка 419
Устойчивость 419
Положительные результаты 419
Усвоенные уроки 420
Примечания и ссылки 421

Глава 28. Рекомендуемое чтение 423

Приложение А. Благодарности 427

Приложение Б. Библиография 431

Предметный указатель 445

Посвящается всем,
кто принимал участие
в моих проектных авантюрах:
семье,
коллегам,
друзьям
и профессионалам
в области конструирования

Part01(Ch01-05).indd 19

ОБ АВТОРЕ

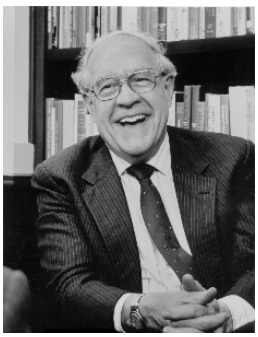


Фото: Джерри Маркатос (© Jerry Markatos)

Фредерик П. Брукс-мл. — заслуженный профессор информатики Северокаролинского университета в Чапел-Хилле. Он в большей степени известен как "отец компьютера IBM System/360", поскольку выполнял функции руководителя проекта его разработки, а в дальнейшем руководил проектом создания программного обеспечения Operating System/360 для этого компьютера на протяжении всей фазы его проектирования. За эту работу он, Боб Эванс (Bob Evans) и Эрих Блох (Erich Bloch) были награждены Национальной медалью США за технологию в 1985 году. Ранее Фредерик был системным архитектором компьютеров Stretch и Harvest в компании IBM.

В Чапел-Хилле доктор Брукс основал факультет информатики и возглавлял его с 1964 по 1984 год. Он также участвовал в работе Национального совета по науке и Совета по оборонным наукам США. В настоящее время он занимается преподавательской деятельностью и исследованиями в области архитектуры вычислительных систем, интерактивной компьютерной графики и виртуальных сред.

ПРЕДИСЛОВИЕ

Я пишу, чтобы призвать проектировщиков и руководителей дизайн-проекта к содержательным размышлениям о *процессе* проектирования предметов, особенно сложных систем. При этом я стою на позициях инженера, для которого важны не только полезность и производительность, но также эффективность и изящество[1].

Для кого предназначена эта книга

Моя книга [40] была рассчитана на "профессиональных программистов, профессиональных руководителей и особенно профессиональных руководителей-программистов". Я обсуждал потребности, трудности и методы достижения концептуальной целостности при создании программного обеспечения коллективами.

В настоящей книге предметная область существенно расширена и подытожены результаты, полученные мною в течение еще 35 лет. Опыт проектирования убеждает меня, что при всей неповторимости процессов проектирования в различных областях деятельности в них есть нечто постоянное. Поэтому моя целевая аудитория такова.

1. **Проектировщики самых разных направлений.** Систематическое проектирование, исключающее интуицию, порождает черепашьи темпы с бесконечными пересмотрами и остановками работ; интуитивное проектирование без систематизации подрывает веру в творческую фантазию. Как совместить интуицию и систематический подход? Как стать полноценным проектировщиком? Как организовать работу коллектива, занимающегося проектированием?
2. Безусловно, я стремлюсь охватить многие области, но в основном обращаюсь к читателям, занимающимся проектированием компьютерного программного обеспечения и оборудования, поскольку в большей степени готов обсуждать конкретные темы именно с ними. Поэтому некоторые приведенные мною примеры из этих областей будут содержать технические подробности. Другие читатели могут пропускать то, что их не заинтересует, без ущерба для общего понимания.
3. **Руководители дизайн-проекта.** Чтобы добиться успеха в своей работе, руководитель проекта должен тщательно регламентировать применяемый процесс проектирования на основе теоретических рекомендаций и уроков практического опыта, а не пытаться скопировать некоторую упрощенную академическую модель или сформировать этот процесс на скорую руку, не принимая во внимание теорию и не учитывая опыт других разработчиков.
4. **Исследователи проблем проектирования.** Настало время широкого исследования процессов проектирования; это означает, что появляются дополнительные возможности, но вместе с тем обнаруживаются определенные

сложности. Опубликованные исследования показывают, что рассматриваемые темы становятся все более узкими, а значительные проблемы обсуждаются менее часто. Создается впечатление, что из-за стремления к достижению максимальной строгости и созданию общей "науки проектирования" не поддерживается публикация чего-либо, кроме сугубо научных исследований. Я бросаю вызов теоретикам и исследователям в области проектирования и призываю их снова вернуться к рассмотрению вопросов фундаментального значения, несмотря на тот факт, что попытка применения методологических принципов социологии оказалась не совсем удачной. Полагаю, что эти ученые бросят мне ответный вызов и укажут на чрезмерную общность моих наблюдений и предвзятость высказанных мною мнений. Но я надеюсь внести свой вклад в развитие их научной дисциплины, поделившись некоторыми из полученных ими результатов с практиками.

Зачем нужна еще одна книга по проектированию

Создание ранее не существовавших предметов приносит большое удовлетворение. Дж. Р. Р. Толкиен (J. R. R. Tolkien) утверждает, что Бог наделил нас возможностью участвовать в создании как подарком, способным приносить подлинную радость[2]. В конце концов, "Мои все звери в лесу, и скот на тысяче гор ... Если бы Я взалкал, то не сказал бы *тебе*"[3]. Творчество само является источником удовольствия.

Процесс проектирования еще не изучен достаточно полно ни в психологическом, ни в практическом аспекте. Причиной этого не является недостаточное количество исследований. К тому же многие проектировщики поделились итогами собственных размышлений над применяемыми ими процессами. Но в основе многих исследований по-прежнему остается стремление преодолеть наблюдаемый в каждой дисциплине проектирования глубокий разрыв между продуктом высшего и среднего качества, а также между продуктом среднего качества и результатом применения недостаточно компетентного подхода. Значительную часть стоимости проектирования, которая нередко достигает одной трети, составляют переделки, связанные с исправлением ошибок. Несомненно, что посредственное проектирование влечет за собой бесполезные затраты дорогостоящих ресурсов, разрушает среду, отрицательно влияет на конкурентоспособность внутри страны и за рубежом. Проектирование важно; обучение проектированию еще важнее.

Как уже не раз было сказано, систематизация процессов проектирования должна привести в среднем к повышению качества готовой продукции, и практика это подтверждает. По-видимому, впервые попытки осуществить программу систематизации и стандартизации предприняли проектировщики-машиностроители из Германии[4].

В дальнейшем значительным стимулом к расширению исследований процессов проектирования явилось появление компьютеров, а затем искусственного

интеллекта. В частности, оправдались первоначальные прогнозы, которые так долго оставались неосуществленными и многими, в том числе и мной, считались неосуществимыми, о том, что методы искусственного интеллекта позволят не только перенести на компьютер большую часть тяжелой и рутинной работы по проектированию, но даже создать блестящие проекты, лежащие за пределами областей, обычно исследуемых людьми[5]. Возникла отдельная научная дисциплина — исследование проектирования, которой посвящаются специализированные конференции, выпускаются журналы и проводятся многочисленные исследования.

Итак, проблемы проектирования уже тщательно изучены и полученные результаты аккуратно систематизированы. Зачем тогда нужна еще одна книга?

Во-первых, процесс проектирования, начиная со Второй мировой войны, развивался очень быстро, и в него было внесено множество изменений, которые редко обсуждаются. При создании сложных артефактов общепринятой нормой становится коллективное проектирование. Коллективы часто географически рассредоточены. Проектировщики все в большей степени становятся изолированными и от сферы реализации, и от сферы эксплуатации; как правило, они больше не могут создавать своими собственными руками предметы, которые они проектируют. Все виды проектов теперь зафиксированы в компьютерных моделях, а не в чертежах. Формальные процессы проектирования все чаще преподносятся в качестве отдельной учебной дисциплины, а предприниматели, привлекая специалистов, требуют от них подготовки по этой дисциплине.

Во-вторых, остается много неизученного. Когда мы пытаемся преподать студентам знания о том, как качественно проектировать, становятся очевидными пробелы в нашем понимании этой темы. Найджел Кросс (Nigel Cross), пионер в исследовании проектирования, прослеживает четыре стадии развития исследований процесса проектирования.

1. *Предписание* идеального процесса проектирования.
2. *Описание* внутреннего характера проблем проектирования.
3. *Наблюдение* фактически происходящей деятельности по проектированию.
4. *Отражение* результатов в основных концепциях проектирования[6].

Мой опыт проектирования был накоплен в течение шести десятилетий и охватывает пять областей: архитектуру вычислительных систем, программное обеспечение, гражданское строительство, документацию и организационные структуры. В одних областях я главным образом выполнял функции основного исполнителя, а в других моя роль в большей степени сводилась к тому, чтобы просто быть сотрудником в коллективе[7]. Я в течение долгого времени интересовался процессами проектирования; еще в 1956 году я защитил диссертацию на тему "Аналитическое проектирование систем автоматической обработки данных"[8]. Возможно, теперь настало время обобщить результаты моих многолетних размышлений.

К какому жанру относится эта книга?

Я пришел к выводу, что практически во всех областях процессы проектирования весьма похожи, и это просто поразительно! Повсеместно происходят аналогичные мыслительные процессы, человеческие взаимодействия, повторы ранее выполненных этапов, проверки соблюдения ограничений, рабочие операции и т.д. В настоящих заметках отражены итоги размышлений над тем, есть ли во всем этом основополагающий инвариантный процесс и в чем он состоит.

Разумеется, архитектура вычислительных систем и архитектура программного обеспечения как области знаний имеют короткую историю, а накопленные результаты анализа применяемых в этих областях процессов проектирования остаются довольно скромными. С другой стороны, строительная архитектура и механическое проектирование имеют сложившиеся и развитые традиции. Эти области не испытывают недостатка ни в теориях проектирования, ни в теоретиках проектирования.

Я — проектировщик профессионального уровня в тех областях, которые имеют лишь скромные научные традиции, и любительского уровня в некоторых более обширных и глубоких областях. Поэтому я попытаюсь извлечь некоторые уроки из появившихся много ранее теорий проектирования и применить их к компьютерам и программному обеспечению.

При этом я полагаю, что часто провозглашаемая цель создания единой "науки о проектировании" недостижима и в действительности вводит в заблуждение. Мое скептическое отношение к предполагаемой формализации дает мне право рассуждать о применении в проектировании интуиции и опыта, включая опыт других проектировщиков, которые любезно поделились со мной своими мыслями на этот счет[9].

Таким образом, настоящая книга не является учебником или монографией с целостным изложением, а скорее представляет собой ряд отдельных заметок, которые носят отпечаток моего субъективного мнения. Разумеется, я постарался предоставлять по ходу изложения полезные ссылки и примечания, в которых приведены интересные дополнительные сведения, но рекомендую сначала быстро прочитывать каждую главу, игнорируя примечания и ссылки, а затем, по желанию, возвращаться и изучать вспомогательный материал. Поэтому указанная информация вынесена мною за пределы основного текста глав.

Кроме того, приведены практические примеры, содержащие конкретные данные, которые могут упоминаться в отдельных главах. Выбор этих практических примеров определялся не по их значимости, а по тому принципу, что с их помощью можно обобщить определенный опыт, на основании которого в настоящей книге сформулированы заключения и выражены мнения. Особое внимание я уделил описанию примеров функционального проектирования домов, поскольку эти примеры могут многое раскрыть для проектировщиков, дайствующих в любой информационной среде.

Я занимался функциональным проектированием (разработкой подробного поэтажного плана, чертежей системы освещения, электроснабжения, водопровода и канализации) для трех проектов домов в качестве главного архитектора. Сравнение и противопоставление осуществляемого при этом процесса с процессом проектирования сложного компьютерного оборудования и программного обеспечения помогли мне постулировать основы процесса проектирования, поэтому я использую некоторые выполненные мною проекты в качестве практических примеров, описывая указанные процессы более подробно.

Ретроспективный анализ показывает, что многие из рассматриваемых практических примеров имеют удивительный общий признак: *чем более смелыми являются проектные решения, кто бы ни был их автором, тем выше вероятность получения высокого качества результатов*. Иногда эти неожиданные решения были найдены благодаря предвидению, а в отдельных случаях возникли после отчаянных попыток выйти из безнадежного положения. Проектирование всегда немного сродни азартной игре, требующей дополнительных вложений в расчете на получение лучшего результата.

БЛАГОДАРНОСТИ

Я назвал эту книгу так же, как в свое время, отстоящее от нашего на целое поколения, назвал свою работу Гордон Глегг (Gordon Glegg), изобретательный проектировщик механических конструкций и очаровательный человек, который прочитал много увлекательных лекций в Кембриджском университете. Мне посчастливилось встретиться с Гордоном в 1975 году и найти в нем человека, так же, как и я, увлеченного проблемами проектирования. Название его книги идеально соответствует тому, что я пытаюсь донести до читателя, поэтому заимствую это название с благодарностью и уважением[10].

Я ценю поддержку Айвэна Сазерленда (Ivan Sutherland), который в 1997 году прослушал одну из моих лекций и предложил написать на ее основе книгу. Прошло немногим больше десятилетия, и вот я попросил Айвэна ознакомиться с черновиком этой книги, который он вначале подверг резкой критике, что послужило ее существенному улучшению. В целом интеллектуальные усилия, предпринятые мною в ходе написания книги, оказались очень полезными.

Эта работа стала возможной только благодаря трем исследовательским отпускам, предоставленным Университетом штата Северная Каролина в г. Чапел-Хилл и моими заведующими кафедрами, Стивеном Вейсом (Stephen Weiss) и Яном Принсом (Jan Prins). Мои усилия нашли горячее одобрение со стороны Питера Робинсона (Peter Robinson) из Кембриджского университета, Мэла Слейтера (Mel Slater) из медицинского колледжа Лондонского университета, заведующих кафедр этих учреждений и их коллег.

По инициативе помощника директора Национального научного фонда (США) Питера А. Фримена (Peter A. Freeman) в рамках программы проектирования технического управления по компьютерным наукам и информатике мне был выделен весьма своевременный грант на завершение этой книги и подготовку относящегося к ней веб-сайта. Такое финансирование позволило мне взять интервью у многих проектировщиков и сконцентрировать основные усилия в течение последних нескольких лет на этих заметках.

Я чрезвычайно обязан многим авторитетным проектировщикам, которые поделились со мной своими мнениями. Таблица с благодарностями, в которой перечислены специалисты, давшие мне интервью и участвовавшие в рецензировании, приведена в последней части. Несколько книг оказались особенно информативными и содержательными; они указаны в главе 28, “Рекомендованное чтение”.

Постоянным источником помощи и поддержки были моя жена, Нэнси (Nancy), которая являлась соисполнителем многих работ, описанных в книге, и мои дети, Кеннет П. Брукс (Kenneth P. Brooks), Роджер Э. Брукс (Roger E. Brooks) и Барбара Б. Ла Дайн (Barbara B. La Dine). Роджер провел необычайно тщательный анализ рукописи и сделал по каждой главе десятки замечаний, начиная с формулировки концепций и кончая расстановкой запятых.

К счастью, мне оказали мощную административную поддержку Тимоти Куигг (Timothy Quigg), Уитни Вон (Whitney Vaughan), Дарлин Фридмен (Darlene Freedman), Одри Рэбелэйс (Audrey Rabelais) и Дэвид Лайнс (David Lines) из Университета штата Северная Каролина. Питер Гордон (Peter Gordon), представитель издательства Addison-Wesley по подбору материалов для публикации, предоставил мне необычное содействие. Джули Нэхил (Julie Nahil), которая занимает в издательстве Addison-Wesley должность начальника производства, предоставляющего полный комплекс услуг, и Барбара Вуд (Barbara Wood), литературный редактор, проявили исключительные профессиональные навыки и терпение.

Джон Х. Ван Влек (John H. Van Vleck), физик, лауреат Нобелевской премии, был деканом отделения инженерного искусства и прикладной науки Гарвардского университета, когда я был аспирантом в лаборатории Эйкена (Aiken) этого университета. Ван Влек проявлял значительную заинтересованность в том, чтобы практика разработки была переведена на более надежную научную основу. Он прилагал энергичные усилия для переориентации американского технического образования от проектирования к прикладной науке. Но маятник качнулся слишком далеко; начинается пересмотр взглядов, и теперь отказ от обучения проектированию не считается таким уж бесспорным решением. Я благодарен тому, что трое из моих преподавателей в Гарвардском университете никогда не теряли из виду значимость проектирования и преподавали эту дисциплину: Филипп Э. Ле Корбейе (Philippe E. Le Corbeiller), Гарри Р. Мимно (Harry R. Mimno) и Говард Х. Эйкен (Howard H. Aiken), мой консультант.

Благодарность и хвала великому Создателю, который любезно предоставляет нам средства к существованию, ежедневную поддержку и радость участия в созидании.

Чапел-Хилл, штат Северная Каролина
Ноябрь 2009 года

Примечания

1. Рисунок на обложке взят из [193], где можно найти такие слова: "... Таким образом, Солсбери — единственный английский собор (кроме собора Святого Павла), в котором вся внутренняя конструкция была создана по проекту одного человека [или одного коллектива из двух человек] и закончена без перерыва".
2. См. [204].
3. Псалтырь 50:10,12. Курсив автора.
4. В разделе 1.2.2 книги [162] эта история прослеживается начиная с 1928 года. Оригинальный вариант этой книги, *Konstructionslehre*, выдержавшей семь изданий, является, возможно, самой важной попыткой систематизации. Я отличаю исследование *процесса проектирования* от правил проектирования

в любой конкретной информационной среде. Последние старше на тысячелетия.

5. Главной монографией, чрезвычайно влиятельной, была книга Герберта Саймона (Herbert Simon) (см. [192]).

6. См. [72].

7. Таблица конкретных событий проектирования включена в материалы приложения на веб-сайте [236].

8. Диссертация на соискание ученой степени доктора философии (см. [34]).

9. Таким образом, я не вношу свой вклад в достижение цели, поставленной методологами проектирования, как указано на веб-сайте [232].

Сложность состоит в том, что требуется преобразовать отдельные события, структуры и перспективы в общую, понятную и, что важнее всего, готовую для распространения область знаний. Виктор Марголин формулирует три причины, по которым эта задача окажется затруднительной, одной из которых является:

"...Отдельные исследования рассуждений о проектировании слишком сосредоточиваются на конкретных высказываниях, что приводит к выработке личной точки зрения, а не к созданию критической массы разделяемых всеми ценностей".

На это я обязан заявить: "Виновен по всем пунктам обвинения".

10. См. [102].

I

Модели проектирования

TheDesignofDesign.indb 31 30.10.2012 15:58:22

1

Постановка задачи проектирования

[Новые идеи обычно появляются] путем связывания и передачи наблюдений из одного искусства для использования в другом, когда восприятие нескольких загадок должно стать предметом рассмотрения разума одного человека.

СЭР ФРЕНСИС БЕКОН ([14], КНИГА 2, С. 10)

Лишь немногие инженеры и композиторы ...способны поддерживать взаимно полезный разговор о содержании профессиональной деятельности друг друга. Я предлагаю им начинать такой разговор с обсуждения вопросов создания внутренней структуры своих произведений, т.е. проектирования... [а затем] приступать к обмену опытом применения творческого профессионального процесса проектирования.

ГЕРБЕРТ САЙМОН ([192], С. 82)

◂ *Винтовая лестница. Корбис (Corbis)*

Прав ли Бэкон

Поставим перед собой цель проанализировать гипотезу сэра Фрэнсиса Бэкона, изложенную в эпиграфе. Обладает ли сам процесс проектирования инвариантными свойствами, иначе говоря, такими свойствами, которые сохраняются в широком диапазоне сред проектирования? Если этот вопрос имеет положительный ответ, то возникает возможность использования коллективных достижений проектировщиков из определенной сферы деятельности, которым удалось выявить некоторые из указанных принципов более определенно по сравнению с другими проектировщиками, поскольку им пришлось прилагать дополнительные усилия в своей информационной среде, необычайно трудной для овладения. Кроме того, некоторые информационные среды, такие как проектирование и возведение зданий, имеют более продолжительную историю проектирования и метапроектирования, иначе говоря, "проектирования самого процесса проектирования". Если сказанное выше соответствует истине (и если заключение Бэкона — верно), то проектировщики из различных сфер деятельности смогут узнавать нечто новое о том, как отшлифовать некоторые собственные творческие навыки, поделившись опытом и идеями с проектировщиками других специализаций.

Что такое проектирование

В оксфордском словаре английского языка (см. *Oxford English Dictionary*) приведено следующее определение глагола *проектировать* (design).

> *Формировать план или схему, мысленно приводить в порядок или вынашивать в уме замысел ...для выполнения в более позднее время.*

Основы этого определения — *план*, *в уме* и *выполнение в более позднее время*. Таким образом, проект (существительное) — это объект, созданный заблаговременно по отношению к проектируемому предмету и в связи с ним, но отличный от него. Дороти Сэйерс (Dorothy Sayers), английская писательница и драматург, в своей великолепной и пробуждающей мысль книге [184] дополнительно разбивает творческий процесс на три различных аспекта. Она называет их идеей, энергией (или реализацией) и взаимодействием[1], как указано ниже.

1. Формулировка концептуальных конструкций.
2. Реализация в реальных средах.
3. Обеспечение взаимодействия с пользователями в форме реального применения.

Согласно этой концепции, книга, компьютер или программа сначала возникает как идеальная конструкция, созданная вне времени и пространства, но в

основном завершенная в мозгу автора. В дальнейшем происходит реализация замысла во времени и пространстве с помощью пера, чернил и бумаги или кремния и металла. Создание завершается, когда кто-то читает книгу, использует компьютер или выполняет программу, вступая тем самым во взаимодействие с замыслом изготовителя.

В одной из своих вышедших ранее статей я разделил задачи создания программного обеспечения на *сущностные* (относящиеся к неизменным свойствам) и *акцидентные* (определяемые конкретными обстоятельствами)[2]. (Руководствуясь смыслом этих философских терминов, не следует считать, что задачи создания программ, обозначенные как акцидентные, являются менее значимыми, чем сущностные. В современном языке применяются более понятные термины — *существенные* и *второстепенные*.) Та часть создания программного обеспечения, которую я назвал *сущностной*, относится к оттачиванию в уме концептуальной конструкции, а часть, обозначенная как *акцидентная*, представляет собой процесс ее реализации. *Взаимодействие*, которое Сэйерс определила в качестве третьего шага, происходит при эксплуатации программного обеспечения.

Таким образом, проектирование — это мыслительная формулировка, которую Сэйерс называет "идеей", и должно быть выполнено до начала любых попыток осуществления. Ответ Вольфганга Амадея Моцарта на вопрос отца, будет ли готова через три недели опера, заказанная герцогом, и ошеломляет своей глубиной, и разъясняет концепцию:

Я уже все сочинил, осталось только записать.

ПИСЬМО ЛЕОПОЛЬДУ МОЦАРТУ [1780]

Для большинства изготовителей предметов, относящихся к роду человеческому, неполнота и несогласованность идей, которыми они руководствуются, становятся очевидными только во время реализации. Что же касается теоретиков, то неотъемлемыми составляющими их деятельности являются оформление в письменном виде, экспериментальная проверка и "выработка" концепций.

Этапы зарождения идеи, реализации и взаимодействия осуществляются рекурсивно. Реализация создает пространство, в котором должен быть выполнен еще один цикл проектирования. Так, Моцарт реализовал свою идею оперы пером на бумаге. Дирижер, взаимодействуя с созданием Моцарта, вынашивал идею относительно интерпретации, реализовал ее с помощью оркестра и певцов, а затем этот процесс завершился взаимодействием с аудиторией.

Проект — это созданный объект; с ним ассоциируется процесс *проектирования*, который я часто называю просто *проектированием*, без какого-либо уточнения. Осуществление самого проектирования описывается глаголом *проектировать*. Эти три термина глубоко связаны; я полагаю, что разрешать двусмысленность поможет контекст.

Место проекта в реальной жизни. Концепция проекта

Если множество людей пользуется одинаковым названием вещи, следует полагать, что они имеют также общее представление о вещи или ее форме; вы меня понимаете?

Понимаю.

Рассмотрим простой пример: есть кровати и столы, причем их очень много, не правда ли?

Да.

Но есть только два представления об этих вещах или их формах: одно представление — о кровати, а другое — о столе.

Верно.

И их производитель изготавливает для нашего использования кровать или стол в соответствии со своим представлением о них.

ПЛАТОН ([236], КНИГА X)

В книге [149] приведены материалы 7-го симпозиума по исследованию мыслительных процессов в проектировании, на котором каждый из докладчиков представил результаты проведенного им анализа одних и тех же четырех совещаний определенного коллектива проектировщиков[3]. Видеозаписи и расшифровки стенограмм совещаний были распространены заранее.

Рэйчел Лак (Rachael Luck) из Редингского университета выявила в дискуссиях о разработке архитектуры понятие, которое ни один из нас до сих пор не замечал, но сразу же признал существующим: понятие *концепции проекта*[4].

Вполне очевидно, что при обсуждении предстоящего задания и архитектор, и клиент время от времени ссылаются на это общее, но не осознаваемое понятие. И часто бывает видно, что не все жесты и слова экскурсоводов, рассказывающих о картинах, указывают на сами картины или на какие-то конкретные объекты, которые на них изображены. Но и в том и в другом случае присутствует стремление создать определенный образ и сохранить его концептуальную целостность.

Благодаря открытию Рэйчел Лак термин *концепция проекта* приобрел самостоятельное существование. То, что подразумевается под этим понятием, не раз проявлялось в работах, которые вошли в копилку моего опыта. Когда разрабатывалась единая архитектура семейства компьютеров-мэйнфреймов IBM System/360 (с 1961 по 1963 гг.), представление о концепции проекта всегда незримо присутствовало в группе создателей архитектуры, хотя об этом понятии никогда не шла

речь. На одного из разработчиков, Джерри Блау (Gerry Blaauw), по ходу работы сошло блестящее прозрение. Руководствуясь им, мы явно разделили действия по проектированию System/360 на *создание архитектуры*, *реализацию* и *осуществление*[5]. Основной замысел состоял в том, что должно появиться семейство компьютеров, все представители которого с точки зрения программиста выглядят одинаково (обладают одинаковой архитектурой), но имеют целый ряд параллельных реализаций, занимающих разные позиции на кривой "цена–производительность" (см. главу 24).

Все эти многочисленные реализации должны были создаваться практически одновременно, и данное требование, наряду с требованием о привлечении к работе наиболее выдающихся технических руководителей, обусловливало необходимость достижения предельной общности и простоты универсальной архитектуры и позволяло избежать многих мелких компромиссов, на которые иногда хотелось пойти ради сокращения затрат. Но каковы бы ни были движущие силы разработки, они лишь способствовали осуществлению сложившихся навыков и стремлений системных архитекторов, каждый из которых горячо желал получить в результате максимально совершенную машину[6].

По мере продвижения работ по проектированию архитектуры я заметил то, что на первых порах показалось весьма странным. Концепцией проекта, платоническим идеальным компьютером для коллектива разработчиков архитектуры была сама *реальная* система System/360. Механические и электрические компоненты изделий Model 50, Model 60, Model 70 и Model 90, которые проектировались в отделении разработки технических средств, были всего лишь отражениями идеализированной System/360. А самое полное и достоверное воплощение фактически разработанной System/360 было не в кремнии, меди и стали, а в тексте и диаграммах *IBM System/360 Principles of Operation*, справочника программиста по машинному языку[7].

Подобный опыт я приобрел при подготовке проекта дома на побережье с условным названием View/360 (см. главу 21). Как оказалась, концепция проекта этого дома стала реальностью задолго до того, как началось какое-либо строительство. Она сохранялась и после смены многочисленных версий чертежей и картонных моделей.

Достаточно интересно, что концепцию проекта семейства программного обеспечения Operating System/360 я никогда не воспринимал как подобную целостную сущность. Возможно, такое восприятие было у его архитекторов; возможно также, что я не приобрел достаточно близкого знакомства с концептуальными основами этого семейства. Можно предположить, что эта концепция проекта не сложилась у меня потому, что система OS/360 была фактически совокупностью четырех достаточно самостоятельных частей: супервизор, планировщик, система управления вводом-выводом и большой пакет компиляторов и утилит (см. главу 25).

Что является действительно значимым

Имеет ли положительное значение признание невидимой концепции проекта как реальной сущности при обсуждении проблем проектирования? Я считаю, что ответ на этот вопрос должен быть положительным.

Во-первых, выдающиеся проекты всегда имеют концептуальную целостность — унитарность, экономичность, четкость. Они не только работают, но и, как впервые открыто сформулировал Витрувий, *восхищают*[8]. Мы используем такие термины, как *изящный*, *качественный*, *прекрасный*, поддерживая разговор о мостах, сонатах, электронных схемах, велосипедах, компьютерах и коммуникаторах iPhone. Признавая концепцию проекта как сущность, мы получаем возможность искать ее целостность в собственных проектах, вступать для этого в сотрудничество при коллективном проектировании и учить этому тех, кто приходит нам на смену.

Во-вторых, чем чаще в обсуждениях затрагивается тема концепции проекта, тем проще становится общение в коллективе проектировщиков. Целью является обеспечение целостности концепции; она достигается только в результате самых широких обсуждений.

> *Обмен мнениями становится намного более конкретным, если в центре внимания находится непосредственно концепция проекта, а не производные представления или неполные детали.*

Так, например, чтобы обсуждение задач создания будущего кинофильма оставалось сосредоточенным на концепции проекта, а не на подробностях реализации, кинематографисты используют раскадровки (последовательности эскизов для визуального планирования сцен перед съемкой).

А дальнейшее уточнение, безусловно, позволяет выявлять конфликтующие версии концепции и принимать срочные меры по устранению противоречий. В качестве примера укажем, что для архитектуры System/360 требовался десятичный тип данных, который позволил бы упростить переход на нее тысяч существующих пользователей десятичных машин IBM. В применяемой нами развивающейся архитектуре уже было предусмотрено несколько типов данных, включая 32-битовые целые числа с фиксированной точкой в дополнительном коде и строки символов переменной длины.

Десятичный тип данных можно было бы создать как подобный тому или другому. Разработчики должны были найти ответ на вопрос: какой из этих вариантов в наибольшей степени соответствует концепции проекта System/360? В пользу той и другой версии были приведены весомые аргументы, но в конечном итоге значимость этих аргументов определялась тем, какой концепции проекта придерживается разработчик. Некоторые из архитекторов неявно придерживались концепции проекта, отражающей представление о созданных ранее компьютерах научного назначения; неявные концепции других более соответствовали существующим ЭВМ

для решения деловых или коммерческих задач. Но System/360 была специально предназначена для качественного обслуживания приложений обоих типов.

Мы решили промоделировать десятичный тип данных по принципу применения строки единичных символов, поскольку этот принцип был более знаком самому крупному из всех имеющихся сообществ пользователей десятичного типа данных, пользователям IBM 1401. И сейчас я бы поступил точно так же.

Размышления о процессе проектирования

Размышления о том, в чем состоит проектирование, имеют длинную историю и восходят, насколько нам известно, к Витрувию (умер приблизительно в 15 году до н.э.). Его *Десять книг об архитектуре* [216] — важный источник сведений о проектировании, начиная с классического периода. Очередными решающими этапами явилось творчество Леонардо да Винчи (1452–1529), с которым можно ознакомиться в издании [235], и Андреа Палладио (1508–1580), автора труда [231].

История размышлений о самом процессе проектирования уходит в прошлое на значительно более краткий период. Паль (Pahl) и Бейц (Beitz) прослеживают развитие процесса проектирования с 1852 года, начиная с достижений ученых из Германии, стимулированных развитием средств механизации; в том же году Фердинанд Редтенбахер (Ferdinand Redtenbacher) выпустил свою знаменитую книгу *Principien der Mechanik, Mannheim: 1852*[9]. Для меня главными этапами был выпуск книг Кристофера Александера [5], Герберта Саймона [192], Паля и Бейца *Konstructionslehre* (1977), а также основание "Общества исследования процессов проектирования" (Design Research Society) (1966 год) и начало издания журнала *Design Studies* (1979).

В книге [148] приведен отредактированный сборник, который включает примерно 23 статьи из журнала *Design Issues*, в основном посвященные критике и теории проектирования, с "эпизодическими экскурсами в философские проблемы, которые опираются на понимание проектирования" (стр. xi).

В моей книге [40] изложены размышления над процессом проектирования для системы Operating System/360 компании IBM, которая в ходе дальнейшего развития привела к созданию операционной системы MVS и последующих версий. В ней во главе угла находятся человек, коллектив и управленческие аспекты указанного проекта разработки и развития. Особую важность для раскрытия темы настоящей книги имеют главы 4–6 книги [40], которые посвящены описанию того, как достичь концептуальной целостности в коллективном проектировании.

Книга Блау и Брукса [25] включает исчерпывающее обсуждение вопросов проектирования архитектуры IBM System/360 (и System/370–390–z), а также показывает, как связаны между собой и на чем основаны десятки проектных решений. В ней не рассматривается процесс проектирования или, как таковые, аспекты проектирования, связанные с деятельностью человека. Тем не менее определенное отношение к данной книге имеет раздел 1.4 книги [25], в котором обсуждаются критерии качества проектов архитектуры компьютеров.

Разновидности проектирования

Сравнение системного и художественного проектирования

Эта книга посвящена описанию проектирования сложных систем с точки зрения инженера, основное внимание которого сосредоточено не только на обеспечении производительности и эффективности, но и на достижении рациональной организации и эстетической привлекательности.

С другой стороны, намечая в проекте свое будущее произведение, живописцы и писатели направляют основные усилия на достижение художественного впечатления и передачу смысла. Архитекторы и художники-конструкторы, разумеется, относятся к обоим лагерям.

Массовое, адаптивное и оригинальное проектирование

Многие считают, что проектирование мостов принадлежит к числу наиболее выдающихся областей искусства разработки, в которых новые достижения в постижении концепции или усовершенствовании технологии имеют принципиальные и весьма ощутимые последствия с точки зрения стоимости, функциональности и эстетической привлекательности.

Это действительно так, но значительная часть всех мостов на автомобильных и железнодорожных магистралях рассчитаны на предсказуемые условия и весьма невелики по длине, поэтому, например, разработка проекта для 15-метрового железобетонного моста — рутинный и легко автоматизируемый процесс. Для коротких мостов известны и внесены в руководства не только дерево проектных решений, но и ограничения и требования, поэтому инженерам-строителям остается лишь воспользоваться этими сведениями. Та же самая ситуация преобладает в части проектирования компиляторов для установленных языков на новых базовых системах. Такие примеры массового, автоматизируемого проектирования можно найти во многих областях.

Акцент в этой книге делается на оригинальном проектировании, в отличие от массовой разработки одного объекта за другим с учетом изменения параметров или даже адаптивного проектирования, которое представляет собой почти полную модификацию предыдущего проекта или объекта в соответствии с изменившимся назначением.

Примечания и ссылки

1. См. [184].
2. См. [42].
3. См. [149]. В этой книге содержатся отредактированные публикации симпозиума Design Thinking Research Symposium (DTRS7).

4. См. [143]; перепечатано в [149].
5. См. [24]. Блау дополнительно подразделяет понятие "энергии" по Сэйерс на *реализацию* и *осуществление*; я нахожу это уточнение весьма полезным.
6. В [128] утверждается, что качественно спроектированные предметы обладают индивидуальностью, и приведено обсуждение того, как добиться в проекте такой индивидуальности.
7. См. [125].
8. См. [216].
9. См. [162].

- **Цель**
- **Требования**
- **Функция полезности**
- **Ограничения, особенно в части лимитируемых ресурсов (возможно, не в денежном выражении)**
- **Дерево проектирования решений**

UNTIL (достижение “достаточно качественного решения”) или (исчерпан запас времени)

DO еще один проект (в целях улучшения функции полезности)

UNTIL проектирование закончено

WHILE проектирование остается осуществимым, разрабатывать очередное проектное решение

END WHILE

Прохождение по дереву проектирования в обратном порядке

Исследовать путь, не рассматривавшийся ранее

END UNTIL

END DO

Выбрать наилучший проект

END UNTIL

2

Инженерный подход к проектированию — рациональная модель

... [Поскольку] для теории проектирования это — общая теория поиска ...в больших комбинаторных пространствах.

ГЕРБЕРТ САЙМОН ([192], С. 54)

◂ ***Рациональная модель процесса проектирования***

Модель

Создается впечатление, будто инженеры обладают четкой, хотя обычно не формулируемой явно, моделью процесса проектирования. Это — упорядоченная модель упорядоченного процесса, поскольку она продумана инженерами до мелочей. Я проиллюстрирую эту мысль примером проектирования дома на побережье (который кратко описан в главе 21).

Цель. Прежде всего, следует начать с основной цели или задания: "Построить дом на побережье с учетом интенсивности ветра и волн, которая наблюдается на данном участке океанского побережья".

Требования. С этой основной целью связан целый ряд требований, или вторичных целей: "Дом на побережье должен быть укреплен так, чтобы он мог противостоять ветрам ураганной силы; в нем должно быть место для ночлега и питания по меньшей мере 14 человек; он должен гармонировать с великолепным ландшафтом окружающей местности" и т.д.

Функция полезности. Разрабатываемый проект необходимо оптимизировать в соответствии с некоторой функцией полезности или, позволяющей взвешивать заданные требования согласно их значимости. Насколько я могу судить, большинство проектировщиков считают, что в функции полезности должны линейно суммироваться веса самих требований, но каждую переменную, оценивающую качество, следует рассматривать не как изменяющуюся по линейному закону, а как приближающуюся по асимптотической кривой к насыщению. Например, как требование может рассматриваться установка окон большой площади, и это действительно важно при проектировании дома. Но по мере добавления каждого квадратного метра суммарной площади окон сверх необходимого происходит уменьшение значения функции полезности. То же касается количества электрических розеток. Тем не менее на первый взгляд кажется, что в функции полезности те переменные, которые зависят от площади окон и количества электрических розеток, просто суммируются.

Ограничения. В каждом проекте и в каждой задаче оптимизации необходимо учитывать ограничения. Некоторые из этих ограничений являются двухзначными — выполнимыми или нет: "Дом должен быть отодвинут от границ участка по меньшей мере на 3 метра". Другие ограничения могут характеризоваться большей гибкостью и даже предусматривать применение штрафов, резко увеличивающихся по мере приближения к определенному пределу; в качестве примера можно указать поставленное заказчиком категорическое условие, чтобы дом на побережье был готов к наступлению летнего сезона.

Некоторые ограничения, такие как нормативные пределы отступа здания, проще по сравнению с другими. Другие же скрывают под невинной формулировкой ужасающую сложность: "Дом должен соответствовать всем строительным нормам и правилам".

Распределения ресурсов, бюджета и критического бюджета. Многие ограничения определяются тем, что между компонентами проекта должен быть рационально распределен какой-то ограниченный ресурс. Наиболее часто в этой роли выступает общая бюджетная стоимость. Но ни в коем случае не следует считать, что это ограничение — единственное и что на нем должно быть сосредоточено все внимание проектировщика, разрабатывающего конкретный проект. Например, при подготовке поэтажного плана дома на побережье необходимо следить за тем, чтобы не был потерян ни один метр (и даже сантиметр) прибрежной океанской полосы, на которой расположен пляж. При проектировании архитектуры вычислительной системы критическим ресурсом может стать количество битов в регистре управления и в формате команды или пропускная способность оперативной памяти. Когда пришло время решать проблему 2000 года в программном обеспечении, критическим распределимым ресурсом стало количество рабочих дней в плане мероприятий.

Деревья проектирования. Итак, рациональная модель состоит в том, что проектировщик в первую очередь подготавливает проектное решение. Каким бы ни было это решение, в связи с его принятием пространство проектирования сужается, после чего проектировщик вырабатывает следующее решение, создавая тем самым очередной узел дерева решений[1]. В каждом узле может быть выбран один или несколько других путей, поэтому можно рассматривать процесс проектирования как систематическое исследование пространства проектирования с древовидной структурой (рис. 2.1).

С применением этой модели проектирование осуществляется очень просто (по крайней мере, концептуально). Достаточно выполнить поиск в пространстве проектирования с древовидной структурой, проверить осуществимость каждого варианта с учетом ограничений и выбрать такой вариант, который оптимизирует функцию полезности. Алгоритмы поиска известны и могут быть строго регламентированы.

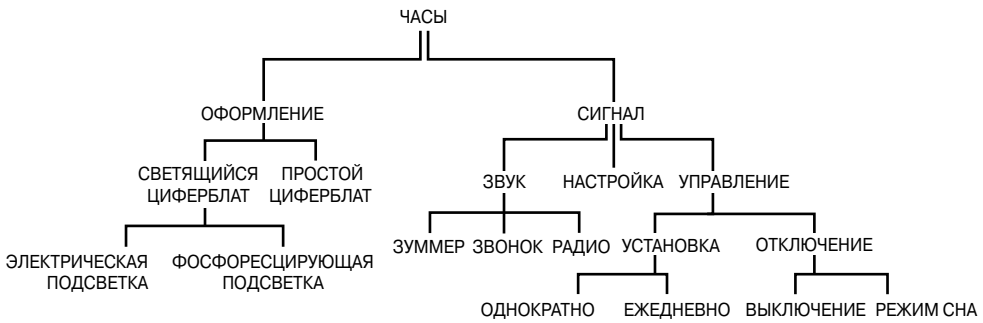


Рис. 2.1. Часть дерева проектирования будильника. Из книги [25], рис. 1-12, 1-14

Надлежащая строгость может быть достигнута, только если предусмотрен полный перебор всех путей для поиска действительно оптимального решения.

Проектировщики обычно довольствуются тем, что поиск проводится вплоть до нахождения "достаточно хорошего" решения[2]. Создается также впечатление, что многие инженеры применяют своего рода стратегию поиска в глубину, при которой в каждом узле выбирается наиболее перспективный или привлекательный вариант и исследование подобным образом проводится до конца. По достижении тупика происходит отслеживание в обратном порядке и выбор другого пути. При принятии решений о выборе варианта проектировщик неизменно руководствуется интуицией, опытом, стремлением к совершенству и эстетическим вкусом[3].

Происхождение рациональной модели

Представление о том, что процесс проектирования может быть смоделирован как систематический поэтапный процесс, по-видимому, впервые возникло в сообществе специалистов по машиностроению из Германии. Паль и Бейц представили описание этой модели, получившее наибольшую известность, в своей замечательной работе, которая выдержала семь последовательных изданий[4]. Эти авторы отметили, что систематический поиск альтернативных вариантов проектирования, который был выполнен на практике, хотя и не описан явно, показан в записных книжках Леонардо да Винчи [235].

Герберт Саймон (Herbert Simon) в книге [192] привел собственные доводы в пользу того, что проектирование представляет собой процесс поиска. Разработанная им модель и приведенное описание намного сложнее, чем в других указанных здесь источниках. Саймон выразил полную уверенность в том, что в свое время процесс проектирования станет предметом изучения в составе задач искусственного интеллекта (как только станут доступными достаточные обрабатывающие мощности), и счел себя обязанным точно изложить строго рациональную модель проектирования, *поскольку* именно такая модель должна лежать в основе средств автоматизации проектирования. Модель Саймона не утратила значимости и в наши дни, несмотря на то, что оригинальное проектирование теперь рассматривается как "дурная проблема"[5], являющаяся одним из наименее перспективных кандидатов для решения с помощью искусственного интеллекта.

В области разработки программного обеспечения Уинстон Ройс (Winston Royce) под впечатлением грандиозных провалов попыток создания крупных программных систем по принципу "что там думать, работать надо", независимо от других авторов предложил модель водопада (Waterfall Model), состоящую из семи этапов, чтобы хоть как-то упорядочить процесс проектирования; эта модель показана на фронтисписе следующей главы. В действительности Ройс описал модель водопада от имени воображаемого оппонента, против доводов которого в пользу этой модели он приводил возражения, а сам предложил более сложные модели, но многие его читатели встали на сторону приверженца модели водопада и согласились с аргументами этого воображаемого оппонента. Я сам, когда был моложе, допустил ту же ошибку, в чем в дальнейшем публично раскаялся[6].

Как это не забавно, критикуемая Ройсом модель с семью этапами следует считать одной из лежащих в основе рациональной модели проектирования.

Ройс подчеркивал, что описанные им семь этапов четко отличаются друг от друга, а их планирование и обеспечение ресурсами должно производиться поразному. Итерация предусмотрена, но тщательно регламентирована по своей области действия:

> *Упорядочение этапов основано на следующей концепции: в ходе выполнения каждого этапа и дальнейшей детализации проекта итерация допускается с охватом (непосредственно) предшествующих и следующих этапов, но редко распространяется на более отдаленные этапы в последовательности. ... Это позволяет эффективно формировать исходную позицию, допускающую в дальнейшем внесение исправлений, в ходе чего может быть на раннем этапе выполнен максимальный объем работы, обеспечивающей восстановление и сохранение*[7].

Представление о том, что пространство проектирования может быть представлено как дерево, основано на идеях Саймона. Мы вместе с Джерри Блау (Gerry Blaauw) описали и проиллюстрировали этот подход в нашей книге [25][8]. Далее в ней приведен пример строго иерархического упорядочения проектных решений по разработке архитектуры процессора в виде гигантского дерева, состоящего из 83 связанных поддеревьев. Простой пример дерева проектирования, который описывает процесс выработки звонка будильника, приведен на рис. 2.1. На этом рисунке показаны два типа ветвей, обозначенных открытыми и закрытыми корнями. В ветви первого типа, которая относится к событию "Звонок", показано дополнительное разделение; каждая ветвь соответствует отдельному проектному атрибуту, который должен быть задан. Такая ветвь называется *ветвью атрибута*. В *альтернативной ветви*, относящейся к событию "Звук", перечисляются альтернативы, одна из которых должна быть выбрана.

Преимущества рациональной модели

Какой бы ни была систематизация процесса проектирования, она является большим шагом вперед по сравнению с подходом по принципу: "Хватит рассуждать, давайте работать (программировать, строить...)". Систематизация позволяет наметить четкие этапы, позволяющие планировать процесс проектирования. В результате появляется возможность сформулировать критерии для составления плана и контроля за его соблюдением. Складывается полное представление о том, как организовать проектирование и какие ресурсы для этого необходимы. Взаимодействие представителей коллектива проектировщиков упрощается, поскольку все работы по проектированию регламентируются в рамках общего процесса. Руководитель коллектива, обмениваясь информацией с членами коллектива, имеет полное представление о ходе работ и может со знанием дела общаться со всеми лицами, заинтересованными в успехе проекта. К разработке могут быть

в любой момент подключены новые сотрудники. При этом человек, впервые приступающий к работе над проектом, видит, с чего он должен начать.

На этом преимущества рациональной модели не исчерпываются. Она позволяет с самого начала сформулировать цели, определить вторичные требования и ограничения, поэтому проектировщики не блуждают вслепую, а в составе единого коллектива решают общие задачи. Планирование всего процесса проектирования еще до начала разработки программного обеспечения или выполнения строительных чертежей позволяет избежать многих задержек в работе и бесполезных затрат усилий. Если процесс разработки организован как систематический поиск в пространстве проектирования, то проектировщики получают возможность выйти за рамки своих повседневных обязанностей и понять, в чем их личный опыт работы нуждается в обогащении.

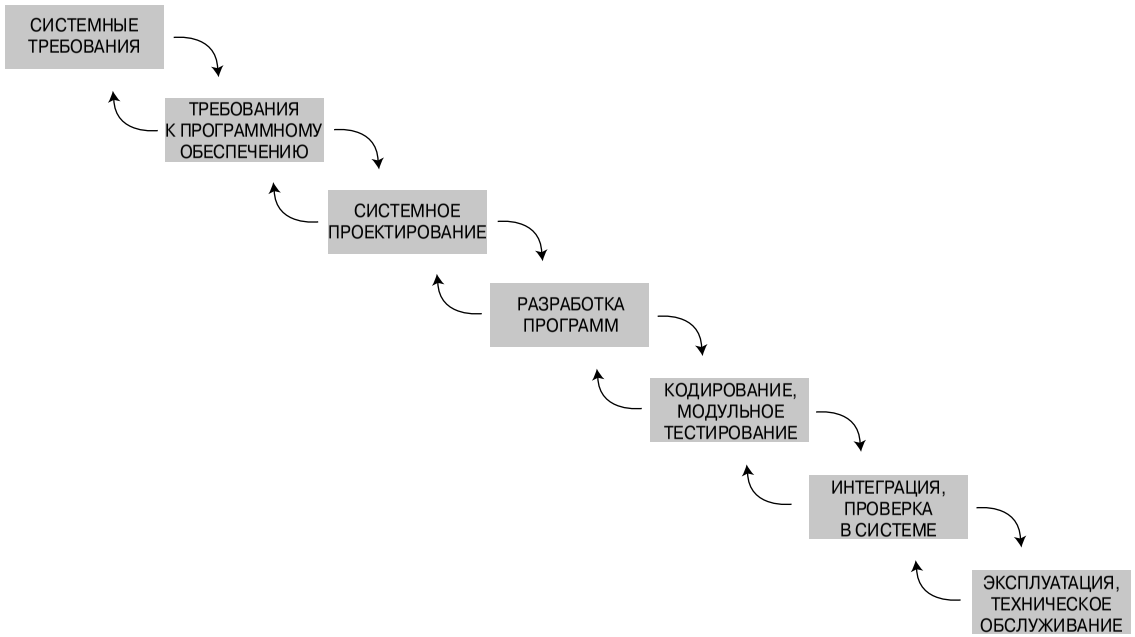
Но рациональная модель слишком проста, даже в той версии, которая была существенно усовершенствована Саймоном. Поэтому мы обязаны исследовать ее недостатки.

Примечания и ссылки

1. Следуя подходу, принятому Саймоном в книге [192], в данной книге я использую слово *человек* как общее существительное, относящееся к обоим полам, так же, как и местоимения *он*, *ему*, *его* и т.д. Я считаю более правильным решение продолжать давнюю традицию обращения в научной литературе к женщинам и мужчинам с применением этих общих местоимений вместо создания более неуклюжих, поэтому сложных для восприятия словесных конструкций.
2. В книге применяется выражение *достижение удовлетворительных результатов* для указания на то, что создаваемый продукт должен быть сделан достаточно качественным, но не обязательно оптимизированным (см. [192], с. 64).
3. Но при этом следует учитывать сказанное в книге [4], что по данным протоколов симпозиума DTRS7 (Design Thinking Research Symposium 7) архитекторы-строители на каждом уровне дерева проектирования производят поиск среди нескольких равнозначных альтернатив, а технические проектировщики в основном полагаются на поиск в глубину, руководствуясь начальным предложением по выбору решения.
4. См. [162].
5. В книге [177] этот термин определен формально. Эта тема подробно обсуждается на веб-странице `http://en.wikipedia.org/wiki/Wicked_problem`.
6. См. [40], с. 265.
7. См. [180], с. 329.
8. См. [25].

TheDesignofDesign.indb 49 30.10.2012 15:58:23

TheDesignofDesign.indb 50



30.10.2012 15:58:23

3

Недостатки модели водопада

Иногда проблема состоит в определении того, в чем заключается проблема.

СМ. КНИГУ ГЛЕЙТА [102]

Предметы создает проектировщик. ...Как правило, осуществляемый им процесс создания является сложным. Ему приходится рассматривать больше переменных (возможных движений, норм, взаимосвязей и т.д.), чем может быть представлено в конечной модели.

СМ. КНИГУ ШЁНА [185]

◂ *Разработка программного обеспечения на основе модели водопада. См. [180] и [28]. Royce © 1970 IEEE. Boehm © 1988 IEEE.*

В действительности каждый проектировщик рассматривает рациональную модель только в качестве идеальной. Иногда приходится слышать, как, по мнению проектировщика, должен осуществляться процесс проектирования, но не приходится видеть в реальности, что проектировщик проводит разработку так, как описывает.

Истина состоит в том, что не каждый инженер даже признается, что в глубине души он является сторонником столь примитивной и идеалистической модели. Но я полагаю, что большинство из нас действительно таковы; я сам в этом отношении не отличался от других в течение долгого времени. Поэтому мы должны подвергнуть рациональную модель проектирования жесткому критическому анализу, чтобы точно узнать, в чем она в наибольшей степени расходится с реальностью.

Отсутствие полного понимания конечной цели в начале проектирования

Самый серьезный недостаток рациональной модели обусловлен тем, что перед проектировщиком часто встает неопределенная, не полностью указанная цель, или основное задание. В этих случаях имеет место следующее.

Самая сложная задача проектирования состоит в определении того, что должно быть спроектировано.

Еще в студенческие годы я потратил целое лето, работая в большой ракетостроительной компании, где мне поручили спроектировать и создать небольшую систему базы данных для отслеживания около 10 000 чертежей радарной подсистемы и состояния обновления каждого из этих чертежей.

Прошло несколько недель, и я получил рабочую версию. Гордясь своим достижением, я представил типичный выходной отчет своему клиенту. Ответ был таковым: "Это прекрасно, но я хотел бы вас спросить: могли бы вы изменить этот отчет так, чтобы... ?"

Каждое утро в течение следующих нескольких недель я показывал своему клиенту выходной отчет, исправляемый снова и снова согласно указаниям, полученным накануне. Каждое утро клиент изучал результирующий отчет и просил выполнить еще одну доработку системы, применяя ту же вежливую формулировку.

Это была простая система (реализованная на машинах, обрабатывающих перфокарты), а сами доработки не отличались концептуальной сложностью. Наиболее трудоемкое изменение состояло в том, что я должен был представить список чертежей в отсортированном виде и предусмотреть в нем отступы, показывающие иерархические зависимости между чертежами, с учетом того, что уровень иерархии обозначен отдельными цифрами на карте от 0 до 9. К другим уточнениям относились многоуровневые промежуточные итоги, разумеется,

с исключениями, и автоматическое маркирование звездочками различных значений, требующих внимания.

Через некоторое время у меня возникло раздражение: "Почему клиент никак не может решить, что он хочет? Почему он не высказывает мне сразу все требования, а растягивает это на целые недели?"

Затем я начал постепенно понимать, что самая важная услуга, которую я оказываю своему клиенту, состоит в предоставлении ему возможности *решить*, на чем он в действительности хочет остановиться.

Да, в наши дни профессия разработчика программного обеспечения стала намного более сложной. Общепризнано, что быстрая разработка прототипа является основным инструментальным средством формулировки точных требований. Итерационным стал не только сам процесс проектирования, но и процесс постановки задач проектирования.

Но, несмотря на эти усовершенствования в разработке программного обеспечения, в литературе не исчезли и даже заметно не уменьшились в количестве упоминания о том, что уточнение требований к продукту является обычной составляющей процесса проектирования. Действительно, на полном основании можно утверждать, что заблаговременное определение всех требований к продукту — весьма редкое исключение, а не норма:

> *Основная услуга, которую проектировщик оказывает клиентам, состоит в определении того, что же действительно они хотят получить в результате проектирования.*

Концепция быстрой разработки прототипов приобрела свое имя и общепризнанное значение в основном лишь в области разработки программного обеспечения, а что касается конструирования компьютеров и архитектурно-строительного проектирования, то она не всегда имеет такой же статус. Тем не менее заметно, что и в этих областях проектирования постепенно происходит переход к использованию методов итерации не реальных, а виртуальных целей. Проектировщики все чаще создают эмуляторы для компьютеров и маршруты для мультипликационной демонстрации зданий в качестве быстро разрабатываемых прототипов, чтобы сходимость, конвергенция целей стала полностью управляемой. Это означает, что итерацию целей следует считать неотъемлемой частью процесса проектирования.

Построение дерева проектирования, как правило, осуществляемое в процессе поиска

Что касается первоначального проектирования сложных структур, таких как компьютеры, операционные системы, космические корабли и здания, то ни один крупный конструкторский прорыв не обходится без существенных новаций в следующих областях.

- Цель.
- Требования и функции полезности.
- Ограничения.
- Доступные технологии производства.

Поэтому проектировщику редко удается просто занять свое рабочее место и заранее составить дерево проектирования.

Кроме того, в сфере высокотехнологичного проектирования лишь немногие проектировщики обладают достаточными знаниями для того, чтобы вычертить основное дерево решений для своих областей. Разработка проектов часто продолжается два года или больше. При этом сами проектировщики могут переходить на другую работу или увольняться. Следовательно, на протяжении своего трудового стажа лишь немногим проектировщикам доведется принять достаточно полное участие более чем в 100 проектах. Это означает, что отдельный проектировщик за свою жизнь не успеет изучить все ветви даже самого упрощенного дерева проектирования в своей области. Для технических проектировщиков, в отличие от ученых, характерно то, что они редко исследуют альтернативы, в отношении которых нет полной ясности, лежат ли они на пути к решению[1].

Вместо этого проектировщики изучают дерево проектирования по мере осуществления текущей работы — принимают решение, после чего наблюдают за тем, какие альтернативы открываются и закрываются применительно к следующему по порядку проектному решению.

Применение узлов, фактически являющихся не проектными решениями, а предварительными полными проектами

В действительности само дерево решений представляет собой лишь упрощенную модель процесса поиска в дереве. Как показывает рис. 2.1, на практике возникают параллельные ветви атрибутов, а также альтернативные ветви. Варианты в одной ветви связаны с вариантами в другой отношениями исключения, сходства или сравнения. Весьма объемное дерево проектирования, приведенное нами в книге [25], слишком упрощено; вся "номенклатура компьютеров и компонентов", приведенная в той работе, необходима для описания того, как происходит связывание решений[2].

Это означает, что в каждом узле дерева проектирования приходится сталкиваться не с простым однозначным выбором среди вариантов одного проектного *решения*, а с однозначным выбором среди многочисленных предварительных полных *проектов*.

Кроме того, многое зависит от упорядочения решений в дереве проектирования, как отметил Парнас (Parnas) в своей классической статье [164][3].

Эти усложнения в древовидной модели возрастают по принципу комбинаторного взрыва настолько резко, что поражают воображение. (Эта ситуация напоминает развертывание дерева ходов в шахматах.) Указанное затруднение рассматривается более подробно в главе 16.

Отсутствие возможности оценивать функцию качества инкрементно

В рациональной модели предполагается, что проектирование сводится к поиску в дереве проектирования, в связи с чем должна быть предусмотрена возможность оценивать функцию качества нескольких нисходящих ветвей.

Но это фактически невозможно сделать, не исследовав все нисходящие ветви вплоть до их листьев, поскольку многие меры качества (например, производительность и стоимость) существенно зависят от результатов последующего детализированного проектирования. Поэтому, хотя оценка качества в принципе возможна, на практике снова приходится сталкиваться с комбинаторным взрывом количества альтернатив.

Что же должен делать проектировщик? Разумеется, проводить оценку, формально или неформально. *Обязательно* усекать дерево проектирования при переходе на каждый нижележащий уровень.

Накопление опыта. В этом процессе интуиция может опираться на многие вспомогательные средства. Одним из них является опыт, и собственный, и заимствованный: "Проектировщики OS/360 предоставили для всеобщего ознакомления подробные описания форматов общесистемных управляющих блоков в документации Operating System/360, и это в ходе эксплуатации обернулось страшной неразберихой. Тогда мы оформили эти форматы в виде объектов". "В семействе компьютеров Burroughs B5000 уже давно используются результаты исследования архитектуры вычислительной системы на основе дескрипторов. Применение такой архитектуры по самой своей сути приводит к существенному снижению производительности, поэтому мы не занимались исследованием этого поддерева". Разумеется, теперь в технологических разработках приходится искать другие компромиссы, но все равно урок, полученный в ходе этого эксперимента, остается поучительным. Наиболее весомая причина для изучения истории проектирования состоит в том, что она позволяет понять, *какие* направления бесперспективны и *почему*.

Простые формулы оценки. Проектировщики обычно используют простые формулы оценки на ранних этапах исследования дерева проектирования. Архитекторы-строители, получив контрольные параметры бюджета, применяют грубую оценку стоимости квадратного метра, выводят целевое значение стоимости и используют полученные исходные данные для последующего сокращения дерева проектирования. Разработчики архитектуры компьютерных систем используют данные о составе команд для первоначальной грубой оценки производительности компьютера.

Разумеется, при этом возникает опасность, связанная с тем, что применение грубой формулы оценки может привести к отказу от исследования ветвей дерева проектирования, которые фактически являются перспективными, но рассматриваются как не подлежащие просмотру из-за примитивности самой аппроксимации, выраженной в формуле оценки. Ошибки могут выражаться и в другом. Я сам был свидетелем того, как архитектор назначил высокую цену за проект возведения дополнительной стены в здании, строительство которого было полностью закончено, руководствуясь исключительно рутинной формулой оценки стоимости квадратного метра. В действительности основная часть расходов на строительство здания была уже подсчитана и оплачена, поэтому стоимость дополнительной работы должна была быть весьма невелика.

Можно вытянуть без труда и рыбку из пруда, если перед этим ты хорошо потрудился, чтобы ее вырастить.

Отсутствие окончательно сложившегося перечня требований, весовые коэффициенты которых оставались бы неизменными

Доналд Шён (Donald Schoen), покойный ныне профессор кафедры городских исследований и образования Массачусетсского технологического института и теоретик проектирования, сказал:

[Когда проектировщик] формирует ситуацию в соответствии со своим начальным представлением о ней, а ситуация изменяется иначе, чем он предполагал, ему приходится корректировать свое начальное представление.

Если процесс проектирования хорошо организован, такое управление ситуацией происходит бесперебойно. В ответ на непредвиденное изменение ситуации проектировщик размышляет и действует, формулируя задачу, намечая стратегию действий или составляя модель обнаруженного явления, поскольку стало явным то, что до сих пор в его деятельности было неявным[4].

Кратко можно отметить, что проектировщик постоянно занимается поиском компромиссных решений, в ходе чего формируется более глубокое понимание общей проблемы проектирования как учета взаимодействия факторов, связанных друг с другом сложными зависимостями. Наряду с этим происходит пересмотр весовых коэффициентов, которые присвоены требованиям. Аналогичная ситуация возникает, когда заказчик проекта (если проект делается по заказу) приходит к лучшему пониманию того, какой результат он желает получить, и приобретает более полное представление о том, как будет использоваться проектируемый продукт.

Например, когда я разрабатывал проект своего дома на побережье (см. главу 22), с самого начала упустил простой вопрос, который после выполнения мною

значительной части работы стал очевидным в тот момент, как мы с женой, рассматривая эскизные наброски, стали думать о том, как будем жить в новом доме: "А где приходящие к нам гости будут вешать свою верхнюю одежду?" Это на первый взгляд второстепенное требование выросло в значительную проблему, для решения которой пришлось перенести спальню хозяев из одного конца дома в другой.

Кроме того, если проектирование и изготовление осуществляются разными исполнителями, как в примерах со строительством дома и производством компьютеров, то проектировщику приходится взаимодействовать с изготовителями, приобретая растущее понимание взаимосвязи между теорией (проектированием) и практикой (производством). Поэтому в ходе работы часто происходит сдвиг сроков и уточнение многих требований и ограничений. Может также измениться технологическая и техническая база производства, что особенно часто наблюдается в области конструирования компьютеров.

Кроме того, многие требования (такие как быстродействие) оцениваются по весовым коэффициентам на основе значимости и соотношения издержек производства, поэтому иногда возникает еще одно явление. По мере осуществления проектных работ обнаруживается возможность достичь значительного усовершенствования по одному из критериев качества за счет крайне низких маргинальных затрат. В связи с этим появляется потребность расширить первоначальный список требований, а само вновь введенное требование часто приобретает такое значение, что резервирует за собой необходимость сохранения в более поздних вариантах проекта.

Только после того, как по заказу факультета информатики Университета штата Северная Каролина было спроектировано, построено и сдано в эксплуатацию здание Ситтерсон-Холла, заказчик обнаружил, что ряд помещений, состоящий из нижнего вестибюля, верхнего вестибюля, факультетского актового зала, аудиторий и тамбуров, может быть наилучшим образом приспособлен для использования в качестве гостевой площадки, вполне подходящей для организации конференций с количеством участников до 125 человек, которые могут проводиться без каких-то особых помех в работе для остальной части здания. Это открытие оказалось счастливым озарением, поскольку такая функция не была предусмотрена в исходной архитектурной программе. Тем не менее вновь приобретенное дополнительное назначение здания окончательно вошло в состав целей проекта высокого уровня: при любых будущих реконструкциях Ситтерсон-Холла, безусловно, останется неизменным стремление сохранить возможность эксплуатировать здание по этому назначению.

Продолжающееся изменение ограничений

Даже если цель была установлена и утверждена, все требования перечислены, точно известно дерево проектирования и однозначно определена функция качества, проектирование все равно приходится организовывать по итерационному принципу, поскольку ограничения продолжают изменяться.

Часто изменения происходят в среде: муниципалитет утверждает новые требования к отступу отбрасывающего тень здания от границ участка; происходит ежегодное обновление электротехнических правил и норм; микросхемы, намеченные для использования, снимаются поставщиком с производства. Мир вокруг нас продолжает изменяться, невзирая на то, что мы для успешного проектирования нуждаемся в стабильных условиях.

Ограничения изменяются также в связи с тем, что в процессе проектирования или на этапе изготовления открывается нечто непредвиденное: строители во время рытья котлована натыкаются на твердую скалу; анализ работы готового устройства показывает ухудшение условий охлаждения важнейших микросхем по сравнению с намеченными и т.д.

Безусловно, не все изменения ограничений являются неблагоприятными. Иногда случается даже так, что ограничения теряют свою значимость. Обнаружив изменение органичений, которое может пойти на пользу делу, хотя и не было результатом целенаправленных усилий, квалифицированный проектировщик оценивает новые возможности и, руководствуясь гибкой методикой проектирования, сразу же применяет их для усовершенствования проекта.

Однако мы должны сделать оговорку. Не все методики проектирования являются достаточно гибкими. Чаще всего после достаточно значительного продвижения в процессе проектирования разработчики перестают следить за изменениями исходных условий, поэтому теряют возможность возвращаться к проектным альтернативам, исключенным ранее по причине их неосуществимости.

Важно перечислить известные ограничения явно в начале процесса проектирования, на этапе работ, который архитекторы называют составлением *программы проектирования.* Программа проектирования — это документ, подготавливаемый вместе с клиентом, в котором формулируются цель, требования и ограничения. Пример такого документа приведен на сопровождающем веб-сайте к этой книге. Программа проектирования — это *не* формальное изложение требований, которое обычно имеет договорную силу при определении приемлемости готового проекта.

Явное перечисление ограничений позволяет учитывать их заблаговременно и избегать неприятных неожиданностей. Кроме того, если проектировщик постоянно сверяется с подобным списком, то шансы своевременного обнаружения того, что какое-то из ограничений потеряло свою актуальность, существенно увеличиваются.

Сам автор неуклонно проводит свою работу по проектированию на базе ограничений и пришел к выводу, что этот процесс пробуждает немалую изобретательность и вынуждает исследовать обычно редко посещаемые уголки пространства проектирования. В таком подходе заключается значительная часть творческого удовлетворения, которое приносит проектирование, и вместе с тем значительная часть сложностей.

Изменение ограничений вне пространства проектирования. Но иногда существенное продвижение в работе достигается после полного выхода за пределы пространства проектирования и осуществления деятельности по преодолению ограничений совсем в другой среде. Занимаясь проектированием главного крыла своего дома (глава 22), я долгое время безуспешно пытался преодолеть ограничение, связанное с необходимостью обеспечения нормативного отступа отбрасывающего тень здания от границ соседнего участка, и вместе с тем выполнить требование к площади музыкальной комнаты (в которой должны были стоять два рояля и орган, находиться квадратная сцена для струнного октета, а также зал для зрителей, отделенный рампой шириной 50 см). Один из вариантов проекта с учетом ограничений показан на рис. 3.1.

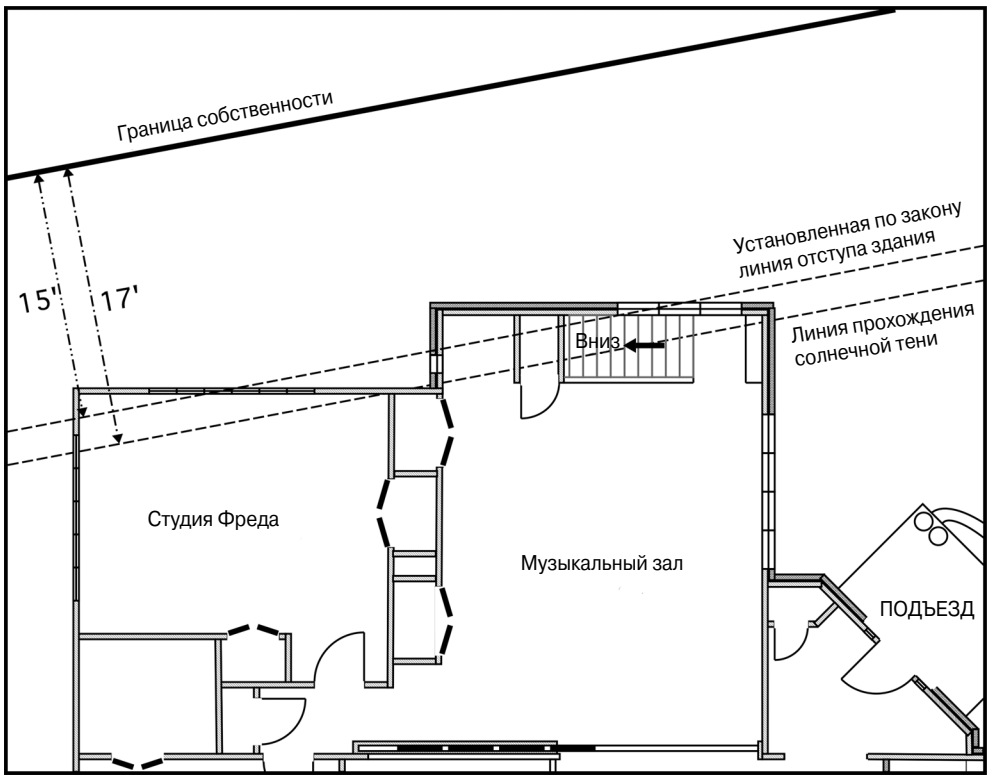


Рис. 3.1. Вариант проекта с учетом ограничений

Эта сложная задача проектирования была наконец-то решена полностью вне пространства проектирования, поскольку я купил полутораметровую полосу земли, затеняемую проектируемым зданием, у своего соседа. Очевидно, что такое решение было реализуемым с меньшими затратами и за более короткое время по сравнению с другим вариантом внепроектного подхода — попыткой получения в муниципалитете разрешения на сокращение отступа отбрасывающего тень

здания от границ участка. В результате ослабли также ограничения, касающиеся других частей проекта, в особенности размещения северо-западного угла студии Фреда (см. рис. 3.2).

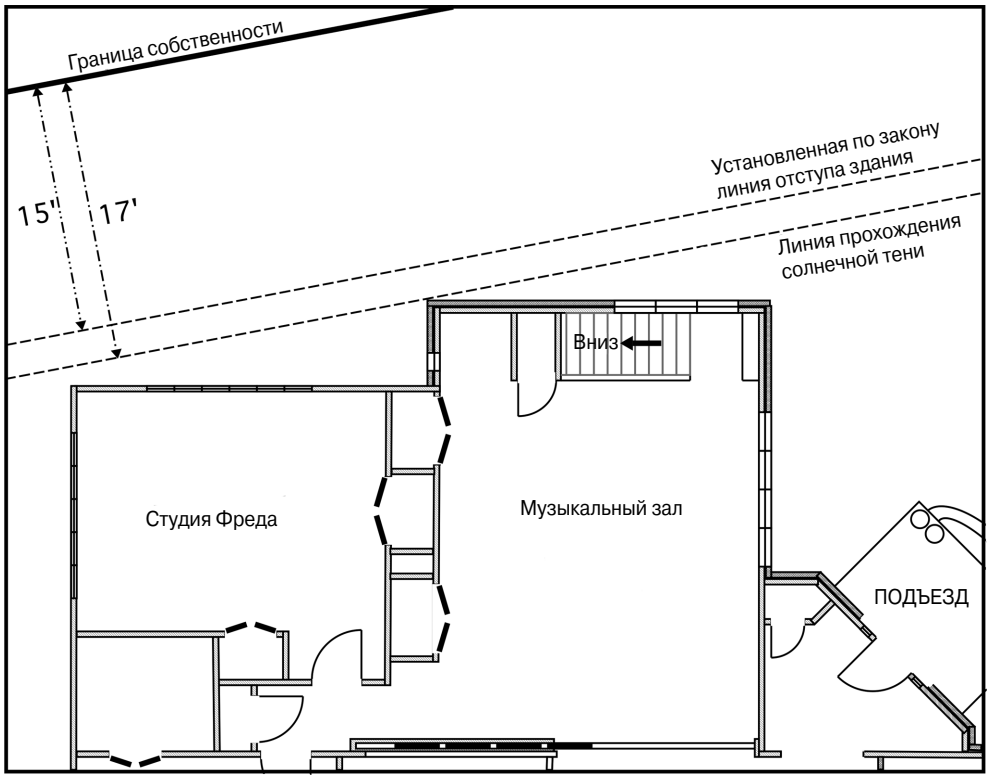


Рис. 3.2. Результат ослабления ограничения

И в данном случае помог все держать под контролем явный перечень известных ограничений в программе проектирования. Проектировщик может периодически просматривать такой список, выясняя: "Нельзя ли теперь исключить то или иное ограничение, потому что мир изменился? Можно ли его полностью обойти, предприняв какие-то действия вне пространства проектирования?"

Критический анализ других рациональных моделей

Естественная модель. Рациональная модель, в том виде, в каком она была представлена и подвергнута критике выше, может показаться примитивной. Но для людей весьма естественно размышлять в рамках именно этой модели. Эта естественность основательно подтверждается тем, что по такому же принципу были независимо созданы другие версии рациональной модели: модель Саймона, модель водопада и модель Паля и Бейца. Тем не менее рациональная модель с

самого начала была подвергнута убедительному критическому анализу в сообществе проектировщиков[5–7].

Проектировщики просто не работают по такому принципу. По-видимому, самым решающим аргументом против рациональной модели, хотя и сложнее всего поддающимся обоснованию, является то, что большинство опытных проектировщиков в действительности не практикуют ее в своей работе. Безусловно, в опубликованных материалах критического анализа редко приходится встречать констатацию факта, что рациональная модель просто не отражает профессиональную практику, равносильную откровению — "а король-то голый", но во всех детализированных исследованиях явно проглядывает признание этой модели как неудовлетворительной, вытесняющее все прочие чувства[8].

Лишь Найджел Кросс (Nigel Cross), хотя и не изменяет своей интеллигентной манере, высказывается более определенно по сравнению с другими. Цитируя много исследований, он прямо заявляет:

> *Обычно принято освещать процесс решения задач в строго определенном ключе, но создается впечатление, что опытные проектировщики часто действуют при этом совсем по-другому. Следует также учитывать, что проектирование во многом отличается от обычного решения задач. ...мы должны соблюдать исключительную осторожность при осуществлении попыток переноса моделей поведения проектировщиков из других областей. Эмпирические исследования деятельности в области проектирования часто показывали, что "интуитивно понятные" черты способностей к проектированию являются наиболее эффективными, присущими самой изначальной природе проектирования. Но в некоторых подходах к созданию теории проектирования предпринимались попытки разработать противоречащие здравому смыслу модели и предписания для регламентации процесса проектирования [выделено мной][9].*

Кроме того,

> *В большинстве моделей процесса проектирования игнорируется противоречивый характер мыслительной деятельности, осуществляемой в ходе проектирования. Модели, предусматривающие достижение согласования в процессе проектирования, наподобие провозглашенной в [213] ...предлагают, чтобы проектирование осуществлялось в виде последовательности стадий. ...С другой стороны, на практике проектирование, по всей видимости, осуществляется путем прямого и обратного переключения между областями промежуточных задач и промежуточных решений, а также путем декомпозиции основной задачи с последующим объединением частных решений[10].*

Я нахожу и сами эти рассуждения, и эмпирическое свидетельство в их пользу весьма убедительными. Такое периодическое переключение с одного вида деятельности на другой действительно было характерным для всего моего опыта проектирования. Типичным примером может служить требование, выраженное

в виде вопроса: "А где мы будем вешать верхнюю одежду?", которое было выявлено лишь после достаточно значительного погружения в процесс проектирования нашего дома, и последующая его реализация.

Критический анализ модели водопада, выдвинутой Ройсом. Ройс в своей оригинальной статье так построил описание модели водопада, чтобы ему было проще подчеркнуть ее недостатки[11]. По существу, он утверждает, что модель не работает даже с обратными стрелками, описывающими противоток между смежными полями в водопаде. Однако его предписание состоит просто в том, чтобы модель была дополнена стрелками противотока, которые возвращаются на два поля. Но это — паллиатив, а не настоящее лечение.

Итоговый критический анализ Шёна

[Саймон] указал на разрыв между профессиональными знаниями и реальными потребностями практики. ...Саймон предлагает преодолеть этот разрыв... путем ввода в действие науки проектирования, но подобная наука может быть применена только к полностью сформированным задачам, уже извлеченным из практических ситуаций.

Если модель технической рациональности ...не способна учитывать практическую пригодность в "расходящихся с действительностью" ситуациях, тем хуже для модели. Поэтому взамен мы должны приступить к поиску практической эпистемологии, скрытой в творческих, интуитивных процессах, которую многие практики способны пустить в ход в ситуации неопределенности, неустойчивости, неповторимости и конфликта ценностей[12].

Дальнейшее использование рациональной модели, несмотря на все ее недостатки и критические замечания

Часто люди, которые первыми стали сторонниками некоторой теории (или метода), оценивают ее перспективы, требования и наиболее подходящую для нее область более ясно, чем те, кто следует за ними. Последователи знают меньше, но проявляют больше энтузиазма, и сам этот энтузиазм становится причиной однобокости, неправильного использования и упрощения.

К сожалению, в наши дни этим характеризуются многие попытки использования рациональной модели. Еще сравнительно недавно, в 2006 году, исследователь в области проектирования, Кис Дорст (Kees Dorst), был вынужден констатировать в своей статье следующее:

Оригинальная работа в области методологии проектирования, написанная Гербертом Саймоном, в которой рассматриваются процесс решения задач и природа плохо структурированных задач, по-прежнему остается констатацией угрожающего

положения в этой области, хотя с тех пор было проведено много успешных разработок. Принципы рационального решения задач, которые основаны на концептуальной структуре, введенной Саймоном, все еще остаются доминирующими в сфере проектирования [13].

Действительно так! В области разработки программного обеспечения все мы слишком часто продолжаем рабски следовать модели водопада, которая представляет собой наше собственное воплощение рациональной модели.

Стандарт Ассоциации немецких инженеров (Verein Deutscher Ingenieure) VDI-2221 (см. [213]). Ассоциацией немецких инженеров в 1986 году в качестве официального стандарта для германского машиностроения была принята рациональная модель, почти полностью в том виде, как было сформулировано Палем и Бейцем[14]. Мне пришлось на деле обнаружить, в каких жестких рамках оказываются те, кто вынужден подчиняться этому решению. Но и сам Паль с трудом подбирал слова, пытаясь объяснить способы смягчения требований этого стандарта следующим образом:

Процедуры, приведенные в стандарте VDI-Richtlinie 2221-2223 и описанные в книге [162], следует рассматривать не как "однозначно заданную последовательность", но скорее лишь как направляющие рамки для базового целеустремленного действия. В реальных ситуациях полезный подход может состоять в том, чтобы применять любой итерационный метод (который предусматривает применение "прямых и обратных" шагов) или проводить разработку, последовательно переходя на более высокие информационные уровни и повторно выполняя необходимые действия[15].

Стандарт Министерства обороны США 2167A. Аналогично с этим, Министерством обороны США в 1985 году была официально принята для руководства как единственно применимая модель водопада в виде документа DoD Standard 2167A[16]. И только в 1994 году этот правительственный орган под влиянием авторитета Барри Боэма (Barry Boehm) расширил круг своих интеллектуальных приобретений, допуская применение других моделей.

Что же из этого следует? Неужели столь важен сделанный нами выбор модели процесса проектирования?

Стоит ли придавать такое значение выбору модели этого процесса? Влияет ли на результаты проектирования то, какую модель мы вместе с другими разработчиками используем для организации процесса проектирования? По моему мнению, ответ на этот вопрос является положительным.

Отсутствие единства мнений по проблеме выбора модели проектирования среди теоретиков. Профессор Кен Уоллис (Ken Wallace) из Кембриджа,

который перевел три издания работы Паля и Бейца на английский язык, полагает, что важнее всего выбрать не какую-то конкретную модель, а любую модель, которую можно легко объяснить участникам разработки и проводить в ее рамках обмен информацией. Он указывает, что принятие на вооружение определенной модели особенно важно для начинающих проектировщиков. Модель Паля и Бейца дает новичку точку отсчета, с которой он может начать работу над проектированием, поэтому, по крайней мере, ему не приходится беспорядочно хвататься то за одно, то за другое. "Я вывожу на экран диаграмму Паля и Бейца (рис. 1.6, приведенный в их книге [162]) и объясняю ее. А затем открываю свой следующий слайд, на котором сказано: «Но проектировщики в действительности так не работают»"[17].

Спасибо за откровенное признание! Но я сомневаюсь в том, что не столь маститые преподаватели с меньшим личным опытом проектирования всегда смогут в этом признаться.

Сюзанна и Джеймс Робертсон, консультанты с международной практикой и авторы превосходных фундаментальных работ по формулировке требований, также считают, что недостатки рациональной модели не имеют особого значения. "Люди, которые понимают, что такое проектирование, знают лучше, что к чему"[18].

Однако я полагаю, что нами выдвинута неадекватная модель, и рабское следование ей приводит к появлению раздутых, громоздких, перенасыщенных функциями продуктов, а также к нарушениям в планировании, распределении бюджета и обеспечении производительности.

Проектировщики с правополушарным мышлением. Проектировщики — главным образом люди с правополушарным мышлением, которые хорошо ориентируются в двухмерном и трехмерном пространствах. Действительно, когда я проверяю наличие таланта у потенциальных претендентов на звание проектировщика, то в числе решающих вопросов задаю следующий: "Когда наступит следующий ноябрь?" Если мой собеседник смотрит на меня с недоумением, я уточняю, "Вы держите в уме пространственную модель календаря? Таковая есть у многих. Не могли бы вы описать мне свою модель, если она у вас есть?" Перспективные кандидаты почти всегда имеют таковую; сами модели чрезвычайно разнообразны.

Точно так же группы проектировщиков программного обеспечения, совместно работая на виртуальных досках, неизменно рисуют схемы, а не пишут слова или код. Архитекторы считают эскизы, выполненные фломастером на кальке, незаменимым инструментальным средством общения, но еще в большей степени ценят их как основу для собственных размышлений.

Таким образом, проектировщики — это люди преимущественно с пространственным восприятием действительности, поэтому в глубине своих мыслей моделируют процессы с помощью объемных схем, будь то вытянутый вверх прямоугольник Паля и Бейца, дерево Саймона или даже водопадная модель, которую Ройс сам же и предложил, и раскритиковал. Применяемые визуальные средства в значительной степени влияют на ход наших размышлений, пусть даже это не воспринимается на сознательном уровне. Поэтому я полагаю, что несовершенная

модель процесса мешает нашей продуктивной работе в таких аспектах, о которых мы не можем все знать, а чаще всего нам остается лишь догадываться об их существовании.

Один из очевидных отрицательных результатов, возникающих в результате принятия на вооружение рациональной модели, состоит в том, что мы неправильно обучаем своих последователей. Мы учим их способам организации работы, которым сами не следуем. Поэтому мы оставляем их лишенными помощи в выработке их собственных применимых на практике способов организации работы.

По-видимому, такое расхождение теории с практикой не свойственно более опытным преподавателям, особенно тем, которые специализируются в области промышленного проектирования. Сами же проектировщики со стажем отчетливо понимают, что модели — это намеренные упрощения, которые должны помочь нам справиться с реальными задачами, которые пугающе усложнены. Поэтому мы предупреждаем наших студентов, что "карта — *не* местность", а модель — не полное изображение; любые средства абстрактного представления могут не просто неполностью отражать моделируемый объект, но даже быть весьма неточными.

В практике разработки программного обеспечения можно сразу же обнаружить еще одну причину затруднений, вызываемых рациональной моделью, — эта модель, в любой из ее форм, вынуждает нас заблаговременно формулировать требования к проекту. Это невольно наталкивает на мысль, что таковые действительно могут быть сформулированы. Поэтому мы становимся вынужденными идти на взаимные компромиссы, поскольку иного не позволяет сложившаяся традиция. Применение более реалистичной модели процесса позволило бы повысить эффективность проектирования, устранить многие причины конфликтов с заказчиками и уменьшить объем переделок. Проблема формулировки требований сформулирована более точно в главах 4 и 5.

Модель водопада — неправильная и вредная; мы должны ее перерасти.

Примечания и ссылки

1. Инженеру требуется удовлетворительное решение; ученый нуждается в открытии, и таковое позволяет сделать более широкое исследование.
2. См. книгу [25], с. 26–27, 79–80.
3. В книге Парнаса [164] процесс проектирования явно трактуется как обход дерева. Парнас приводит убедительные аргументы в пользу того, что процесс проектирования должен быть настолько гибким, насколько это возможно. Он доказывает, что для обеспечения этого необходимо помещать решения, в наименьшей степени подверженные изменениям, как можно ближе к корню дерева. Гибкость проектирования — важная цель. Задача ее обеспечения всегда является первостепенной в разработке программного обеспечения, будь то объектно-ориентированное проектирование или методология динамической разработки.

4. См. [185], с. 79.

5. Я нашел настолько мало критических замечаний в отношении формулировки рациональной модели Пала и Бейца и настолько много в отношении формулировки Саймона, что это вызывает удивление. Паль и Бейц сами признали неадекватность своей модели: в выпускаемых одно за другим изданиях их работы модель этих ученых (рис. 3.3, 4.3 во втором и третьем английских изданиях) включает все больше и больше явно заданных итеративных шагов (см. [162]). В трех изданиях книги [192] Саймона не отражены какие-либо предложенные изменения в модели, хотя в личной беседе со мной в ноябре 2000 года Саймон сказал, что в его собственном понимании модели достигнут определенный прогресс, но он не имел возможности заново продумать и внести соответствующие изменения в книгу.

В книге Виссера [215] имеется превосходный раздел 9.2, "Последующие уточнения позиций Саймона", в котором анализируется эволюция взглядов Саймона, нашедшая отражение в более поздних работах. Виссер разделяет мое удивление по поводу того, что эта эволюция не нашла своего отражения в более поздних изданиях книги [192].

6. В своей книге Холт писал [121]:

Существуют две различные интерпретации процесса инженерного проектирования. Подход, в основе которого лежит процедура решения задач, популярный во многих третьестепенных организациях, сосредоточен на решении структурированных, вполне определенных задач с использованием стандартизированных методов. В качестве истоков этого подхода можно рассматривать так называемую "формальную" школу системного анализа. С другой стороны, в творческом подходе к проектированию объединяются аналитическое и системное мышление с трактовкой человеческих факторов в инженерном проектировании для создания и использования открывшихся возможностей служения обществу В этой работе обсуждаются ограничения подхода на основе решения задач при попытке его применения по отношению ко многим реальным проблемам.

7. Критические замечания Кросса (Cross) имеют эмпирические истоки, а критика Шёна (Schoen) направлена против философских взглядов, лежащих в основе рациональной модели. Он указывает, что рациональная модель в том виде, в каком она изложена Саймоном, является естественным продуктом намного более распространенного философского мышления, которое он называет мышлением в рамках технической рациональности и определяет как унаследованное от дискредитировавшего себя позитивизма. Он указывает, что сама эта основополагающая философия является совершенно неподходящей для понимания того, что такое проектирование, даже притом, что она возведена в ранг ведущей большинством профессиональных проектировщиков:

С точки зрения технической рациональности повседневная работа профессионала представляет собой процесс решения задач. Задачи... решаются путем

выбора среди доступных средств того, что в наибольшей степени подходит для достижения поставленной цели. Но, делая такой упор на решении задач, мы уделяем меньше внимания постановке самих задач, т.е. процессу, позволяющему определить, какое решение должно быть принято, какие цели должны быть достигнуты и какие средства могут быть выбраны. В повседневной работе задачи не предстают перед разработчиком в готовом виде. Задачи приходится конструировать из материала проблемных ситуаций, которые являются обескураживающими, беспокоящими и неопределенными. ...Для этого разработчик должен провести определенную работу. В частности, ему необходимо извлечь смысл из неопределенной ситуации, которая первоначально кажется лишенной всякого смысла. ...Причем именно с такими ситуациями профессиональным специалистам приходится сталкиваться все чаще и чаще в своей работе. ... Без определенной договоренности в отношении намеченных целей подход на основе технической рациональности становится неосуществимым.

8. Ярким примером, подчеркивающим это утверждение, может служить такая мысль, которую высказал Сеймур Крэй (Seymour Cray) в 1995 г.: "Я считаю себя представителем науки, но при принятии наиболее важных решений в большей степени опираюсь на интуицию, чем на логику". `http://www.cwhonors.org/archives/histories/Cray.pdf`; в книге взята за основу версия этои страницы от 14 сентября 2009 г.

9. См. [69], с. 27.

10. В работах Кросса [69] и Дорста [85] приведено особенно содержательное обсуждение заочной дискуссии Саймона и Шёна. Эти работы перепечатаны также в книге [73]. Дорст также показал, что применительно к протоколам Delft II модель Шёна намного более точно соответствует наблюдаемому поведению проектировщика.

11. См. статью Ройса [180].

12. См. [185], с. 45–49.

13. См. [84].

14. См. [213].

15. См. [163].

16. В документе DoD-STD-2167A предпринята попытка исправить эту ситуацию, но, к сожалению, диаграмма модели водопада оставлена на видном месте, а все прочее сохранилось в том же виде, как и было. В документе MIL-STD-498, заменившем документ 2167A, проблема выбора модели решена. В дальнейшем Министерством обороны США документ 498 был заменен в результате принятия промышленных стандартов IEEE/EIA 12207.0, IEEE/EIA 12207.1 и IEEE/EIA 12207.2.

17. По материалам личного общения [2008].

18. По материалам личного общения [2008].

TheDesignofDesign.indb 68 30.10.2012 15:58:23

4

Требования, искушения и соглашения

Любая попытка сформулировать все возможные требования в начале проекта потерпит неудачу и вызовет значительные задержки.

СМ. КНИГУ ПАЛЯ И БЕЙЦА [162]

Комитет полагает, что для определения четких и полных требований системного уровня необходимо взаимодействие с потенциальными подрядчиками между этапами А и Б.

ДЖЕЙМС ГАРСИЯ (JAMES GARCIA).
МАТЕРИАЛЫ К ДОКЛАДУ [2]

◂ *Вертолет Boeing-Sikorsky RAH-66 Comanche, который первоначально носил обозначение LHX. Авиатехника Сикорского. Фото: Ричард Зеллнер (Richard Zellner). Сайт Wide World Photos издательства AP*

Фильм ужасов

Генерал всю жизнь прослужил в морской авиации и хорошо разбирался в вертолетах. Мы были отправлены с ним в самые недра Пентагона в составе подкомиссии "Совета по оборонным наукам". Нам пришлось внимательно выслушать доклад одного полковника о ходе проектирования изделия LHX (Comanche) — легкого штурмового вертолета следующего поколения, на который должны быть потрачены миллиарды долларов и от которого будут зависеть жизни многих солдат. Этот вертолет должен был заменить четыре разных типа современных вертолетов, каждый из которых имел свое отличное от других назначение.

Полковник выделил перечисленные ниже требования, которые были разработаны межгрупповым комитетом, представляющим несколько групп пользователей.

"Крейсерская скорость такая-то, дальность полета такая-то. Вооружение X; количество внешних подвесок Y; запас боеприпасов Z; перевозка полностью экипированных бойцов W, не считая экипажа.

Полет на низкой высоте, вне зоны действия радаров. Возможность передвижения темной ночью и в непогоду; быстрый набор высоты для обхода препятствий. Стремительный подъем для ведения огня; стремительный спуск для уклонения от ответного удара".

Затем, не меняя интонации и без тени смущения, он сказал: "К тому же этот вертолет должен быть способен самостоятельно перелететь через Атлантику". (А это намного превышает обычный оперативный диапазон, установленный для вертолетов.)

Мы с генералом не могли скрыть своего недоумения. Докладчик быстро отреагировал на это: "Я хочу сказать, что такой дальний перелет должен осуществляться после выгрузки всего вооружения и боеприпасов, а также заливки всех баков топливом". Но мы не могли поверить в осуществимость такого проекта, поскольку хорошо знали, какими возможностями действительно обладают современные вертолеты. Даже я, разработчик компьютеров, знал, какую цену приходится платить за каждое амбициозное требование и что речь идет не о долларах, а о необходимости смириться с ухудшением каких-то других характеристик.

"Для чего нам нужно обеспечивать самостоятельный перелет через океан? Ведь если вертолет не будет сбит на поле боя, то ему придется совершить такой полет всего лишь два раза за весь его срок службы".

"У нас не хватает транспортных самолетов C-5 для доставки всех вертолетов к месту их боевого применения".

"А почему бы не направить часть предполагаемых затрат по программе разработки LHX на закупку дополнительного количества самолетов C-5, вместо того, чтобы ставить под угрозу проект LHX?"

"Это невозможно".

Наше искреннее удивление по поводу того, насколько неосуществимыми являются эти экстремальные требования, усугублялось тем, что наш оппонент

не выражал ни малейшего беспокойства в ответ на все выраженные сомнения. И здесь мы поневоле стали думать, что, наверное, поддались первой реакции и не продумали все до конца. Возможно, в конечном итоге вертолет, способный доставить сам себя так далеко, не будет стоить слишком дорого. Возможно, в проекте заняты такие одаренные разработчики, которые уже давно справились с этой проблемой.

Но наш дальнейший разговор принес лишь одни разочарования. Насколько я понял, в комитете по выработке требований к проекту LHX не было ни одного авиаинженера, ни одного пилота вертолета, а заседали главным образом чиновники, закаленные в борьбе за интересы своих групп в межгрупповых согласованиях[1].

Атмосфера закулисных интриг

Можно, наверное, позавидовать многим читателям, которые не имели "счастья" присутствовать на заседаниях, подобных тем, которые собирал комитет по выработке требований к проекту LHX, а мы были вынуждены находиться на таких заседаниях.

Каждый участник подобных мероприятий втайне намечает то, чего он хочет добиться, исходя из стоящих перед ним целей и опираясь на свой личный опыт. Каждый утвердил себя как личность и завоевал репутацию успешного или неудачливого игрока, которая показывает, насколько его цели оказались реалистичными. Естественным образом возникает стремление к созданию коалиций как способа упрощения решения собственных задач: "Я не буду мешать вам добиваться намеченного, если вы не будете мешать мне".

Кто же в таком случае выступает в процессе выработки требований к продукту в роли защитника интересов создания самого продукта — достижения его концептуальной целостности, эффективности, экономичности, надежности и т.д.? Чаще всего никто. Слишком часто архитектор или инженер может выразить только мнение, основанное на интуиции и инстинктивном понимании, но еще не подтвержденное фактами[2]. В классическом процессе разработки, основанном на модели водопада, требования задаются еще до того, как начинается проектирование.

Результатом, безусловно, становится чрезвычайно отягощенный набор требований, в котором воплощены стремления всех причастных к их выработке, но не учтены ограничения[3]. Причем обычно требования в списке не расставлены по приоритетам и не обозначены весами. Из-за стремления сохранить миролюбивую атмосферу на заседаниях по выработке требований доминирует стремление избежать любых неприятных конфликтов, которые могут возникнуть даже по поводу определения того, какое требование является более весомым, не говоря уже о расстановке требований по приоритетам.

Но в конечном итоге наступает такой момент, когда приходится согласовывать список требований с ограничениями. На практике проектировщики товарной

продукции неявно сверяют предложенный им набор требований со сложившимся у них представлением о конкретном пользователе[4]. Но во многих случаях из-за того, что требования не обозначены как более или менее значимые, проектировщики не могут воспользоваться теми глубокими знаниями о пользователе и приложении, которыми действительно обладают некоторые опытные постановщики задач.

Выработка требований комитетами, по самому своему характеру, становится причиной создания продукции, которая слишком перегружена функциями (автомобили производства детройтских фирм, громоздкие программные комплексы, практически нереализуемые системы, задуманные в Налоговом управлении США и в ФБР). Возможно, крупные программные системы, на которые возлагались слишком большие надежды, так подвержены катастрофическим сбоям именно потому, что технические задания на них разрабатывались комитетами, в основном состоящими из чиновников.

С началом создания операционной системы OS/360 в компании IBM требования были первоначально установлены большим комитетом, состоящим из специалистов отделения маркетинга; при этом происходил точно такой же процесс выработки и формирования результатов, как было описано выше. Я, выполняя функции руководителя проекта, был вынужден полностью отклонить предложенную документальную формулировку требований как нереалистичную и привлечь весьма небольшой коллектив системных архитекторов, специалистов по маркетингу и конструкторов для решения задачи получения минимально необходимого набора требований.

Борьба с раздутыми и неконкретными требованиями

Против быстрого увеличения количества требований необходимо бороться, для чего следует привлекать способы предотвращения дальнейшего добавления требований и способы исключения требований, распознанных как ненужные. Обе эти категории способов описаны в весьма интересном отчете Air Force Studies Board [2] подразделения Национального научного совета (National Research Council)[5].

Мое знакомство с работой комитета по выработке отчета "Pre-Milestone A and Early-Phase Systems Engineering" началось с другого фильма ужасов. Продолжительность разработки основных военных систем тридцать лет тому назад составляла приблизительно пять лет. В наши дни время от запуска программы до развертывания системы увеличилось в два-три раза, несмотря на резкое ускорение технологических процессов разработки и темпов возрастания новых угроз.

Программы, которые чрезвычайно успешно разрабатывались в прошлом, как правило, базировались на постановке нескольких четких наиважнейших заданий и составлении планов, предусматривающих неукоснительное выполнение. Эти проекты начинались с небольшого количества требований верхнего уровня.

По мере дальнейшей разработки эти требования подразделялись на более конкретные вспомогательные требования с учетом ключевых показателей производительности, а реализация проектов велась под управлением авторитетных и способных руководителей, которые непрерывно следили за тем, чтобы реализация функций системы осуществлялась без выхода за пределы срочного графика и намеченных затрат.

Любые попытки выхода за пределы установленных требований, предпринимаемые пользователями или собственными разработчиками, проявляющими инициативу, безжалостно пресекались, для чего лучшим средством были ссылки на невозможность нарушения графика. (Для меня, когда речь шла о создании системы, это также было излюбленным оружием.) Члены указанного комитета пришли к выводу, что службы закупок Министерства обороны США утратили всепроникающее чувство безотлагательности предстоящих решений и взамен стали вводить все новые и новые организационные уровни предотвращения "оплошностей", чтобы избежать любых ошибок. Подобная эволюция нередко происходит и в технических корпорациях.

> *Комитет рекомендует, чтобы организационные мероприятия по разработке систем начинались задолго до того, как будет развернута целенаправленная подготовка технологических основ для новой системы. Но рекомендуется также не формулировать техническое задание с изложением требований до начала этапа А, а подготавливать начальное изложение детализированных требований во время разработки самой системы, между этапами А и Б.*

Если на этапе А будут четко сформулированы ключевые показатели производительности, а на этапе Б — однозначные требования, которые могут остаться неизменными в ходе достижения исходных эксплуатационных характеристик, *то появится возможность эффективно реализовать этап разработки* (курсив мой)[6].

В таком случае проблему неуправляемого изменения требований удается решить заранее наиболее рациональным способом. Комитет выработал перечисленные ниже рекомендации первостепенной важности: "С самого начала подбирать опытных и зрелых руководителей, хорошо знающих предметную область, с которыми можно будет реализовать всю программу разработки, вплоть до начала поставки системы. Наделить этих руководителей правами приспосабливать к своим нуждам стандартизированные процессы и процедуры так, как они сочтут необходимым"[7].

Комитет настаивает также на использовании таблицы определения требований для обеспечения того, чтобы каждое требование было описано, сформулировано и изложено с той точки зрения, в чем оно способствует достижению одного или нескольких первоначальных общих требований, поскольку это позволяет избежать проникновения в состав целей, намеченных для реализации, того, что продиктовано пожеланиями какого-то представителя пользователя или стремлением разработчика создать нечто оригинальное, новое и даже, возможно, полезное[8].

Искушение

Рассмотрим приведенный ниже сценарий.

- Некий заказчик никогда не проявляет скупости, но всегда готов платить привлеченным к работе системным архитекторам и программистам справедливую цену за их опыт и труд (возможно, потому, что он весьма заинтересован в успехе разработки или желает в дальнейшем снова прибегнуть к их услугам).
- Он привлек к работе системного архитектора, который в любых обстоятельствах считает себя подлинным представителем заказчика и жаждет приложить все свои таланты и профессиональные навыки, чтобы в полной мере раскрыть и обслужить истинные потребности заказчика.
- Он заключает контракт с программистом, который руководствуется принципом "всегда производить высококачественную продукцию с наилучшим соотношением затрат и стоимости в рамках бюджета и графика" и неизменно следует ему.
- Все участники заключенных при этом договоренностей (заказчик, системный архитектор и программист) являются честными и правдивыми, и между ними налажена превосходная связь.

В таком случае автор утверждает следующее.

- Имеющееся соглашение о расчетах и платежах с учетом фиксированной прибыли позволит заказчику получить наибольшую отдачу от денежных затрат.
- Проектирование и создание в рамках одного договора становится самым быстрым способом реализации проекта.
- Применение явно определенного процесса разработки по спиральной модели (см. следующую главу) позволит заказчику получить продукт, в наибольшей степени соответствующий его потребностям.

Возникает вопрос: если последнее утверждение справедливо, то чем объяснить применение из года в год модели водопада, несмотря на то, что в течение уже более четверти столетия проявляется бо́льшая адекватность спиральных и коэволюционных моделей?

На этот вопрос можно ответить так, что виной всему является искушение, которое выражается как самоуверенность, жадность и лень. Все те, кто имеет опыт проектирования, расценят описанный выше пример отношений заказчика, системного архитектора и программиста как слишком идеализированный. А кое-кто при чтении этого описания может даже фыркнуть: "Очень мало шансов на то, что действительно сложатся подобные условия!" Поскольку люди еще во времена Адама и Евы впали в искушение и совершили грех, никто не может верить в бескорыстность побуждений друг друга. Люди греховны, и такое общение между ними, в котором ничто не остается скрытым, невозможно.

Соглашения

По этим причинам всегда остается актуальной рекомендация: "Все договоренности следует оформлять в письменном виде". Нам нужны письменные соглашения, чтобы в процессе взаимодействия не возникали неопределенности; нам нужны неукоснительно соблюдаемые контракты для защиты от попыток нарушить их со стороны других людей и искушения выйти за их рамки со стороны нас самих. Подробные и неукоснительно соблюдаемые контракты становятся еще в большей степени необходимыми, если участники соглашений представляют организации с большим штатом, а не выступают от собственного лица. Дело в том, что четкого соблюдения договоренностей можно скорее ожидать от отдельных представителей организации, чем от всей организации в целом.

Необходимость в заключении соглашений очевидна, независимо от того, охватывают ли они представителей одной организации или разных организаций, поскольку лишь это позволяет с самого начала определять обязательства в рамках достижения целей, а также добиваться соблюдения требований и ограничений. При этом никто не оспаривает тот факт, что такие соглашения в дальнейшем непременно подвергнутся пересмотру. (Хотя при этом открываются новые возможности для злоупотреблений: "Не придавайте слишком большого значения этому контракту; составьте его как открытый для изменений".) Таким образом, создается впечатление, что потребность в заключении контрактов лучше всего объясняет жизнестойкость модели водопада, неизменно применяемой при проектировании и построении сложных систем.

Модель, основанная на заключении контрактов

В конечном итоге неуклонное стремление составить полный и согласованный набор требований диктуется желанием заключить контракт на получение определенного продукта по установленной цене (причем это желание чаще всего выражается на уровне руководства организации). Тем не менее это стремление сразу же упирается в тот непреложный факт, который доказан в главе 3, что по существу невозможно определить полный и точный набор требований для любой сложной системы, кроме как в рамках итерационного взаимодействия в процессе проектирования.

Как же это противоречие разрешается в области строительства зданий и сооружений, история которой измеряется веками? По существу, в этой области применяется совсем другая модель заключения контрактов. Рассмотрим обычный процесс проектирования зданий.

- Заказчик разрабатывает программу строительства здания, а не технические требования.
- Он заключает с архитектором контракт на услуги проектирования, обычно с почасовой или процентной оплатой, а не контракт на получение определенного продукта.

- Архитектор, проводя совещания с заказчиком, пользователями и другими заинтересованными лицами, составляет наиболее полную программу строительства, которая, тем не менее, не рассматривается как окончательно утвержденная спецификация готового продукта, подлежащая закреплению в контракте.
- Архитектор выполняет концептуальное проектирование, в ходе которого выполняется приближенное согласование программы строительства с бюджетными ограничениями, графиком и строительными нормами и правилами. Этот концептуальный проект становится первым прототипом, по которому проводится проверка соответствия первоначальному замыслу всеми заинтересованными лицами.
- По результатам этой проверки архитектор разрабатывает строительный проект, в ходе чего часто подготавливаются более подробные чертежи, трехмерные масштабные модели, макеты и т.д. Еще раз проведя согласование со всеми заинтересованными лицами, архитектор разрабатывает строительные чертежи и сметы.
- Заказчик использует эти чертежи и сметы для заключения по твердой цене контракта на создание готового продукта.

Заслуживает внимания то, что в этой модели, прошедшей долгое развитие, контракт на проектирование отделен от контракта на строительство. Даже если оба эти контракта осуществляются одной и той же организацией, благодаря указанному разделению удается внести ясность во многие вопросы, которые могли бы стать предметом разногласий.

Безусловно, эта модель не является также полностью последовательной. Любой, кто принимал участие в реализации даже самых скромных строительных проектов, знает, что практические проблемы строительства и высказываемые на последних этапах пожелания заказчика внести изменения или пересмотреть проект становятся причиной проведения корректировок проекта, которые выражаются в форме распоряжений об изменении контракта.

Описанный выше процесс строительства под управлением архитектора является классическим. Он имеет свои недостатки, причем не последним из них можно назвать большую продолжительность графика работ. Строительные проекты с

- тесными доверительными отношениями между заказчиком, архитектором и подрядчиком,
- хорошо изученные задачи проектирования или
- обусловленная неустранимыми факторами спешка, оправдывающая более высокие риски,
- часто становятся причиной того, что обычный процесс превращается в параллельный, конвейерный процесс проектирования и строительства в рамках одного договора. Архитекторы организуют свою работу так, чтобы

подробные строительные чертежи в первую очередь подготавливались для тех частей здания, которыми подрядчик должен заниматься с самого начала: заблаговременное возведение стальных конструкций, подготовка строительной площадки, закладка фундамента.

Системные проекты, которые соответствуют условиям в приведенном выше списке, по-видимому, также можно было бы реализовывать по принципу проектирования и строительства в рамках одного договора. В таком случае разработчики аппаратных средств и программного обеспечения должны будут определить последовательность создания систем и выяснить, какие компоненты должны быть подготовлены заблаговременно.

В этой области необходимо провести большой объем теоретических исследований. Я призываю все сообщество специалистов по информатике принять участие в этой работе[9,10].

Примечания и ссылки

1. История разработки программы "RAH-66 Comanche" изложена в статье Википедии (Wikipedia [2002–2009]). Приведенное в этой статье описание свойств вертолета подтверждает мои воспоминания о заявленных требованиях:

   Очень сложные системы обнаружения и навигации вертолета Comanche был предназначены для обеспечения его применения ночью и в плохую погоду. Корпус вертолета Comanche был спроектирован в целях более удобного размещения в транспортном самолете или на транспортном судне по сравнению с вертолетом Apache, что позволяет быстро доставлять его в горячие точки. Перегоночная дальность полета вертолета Comanche составляет 1260 морских миль (2330 км), что позволяет при отсутствии подходящих транспортных средств отправлять его к месту боевого применения своим ходом.

   В рассматриваемом случае на основе легкого штурмового вертолета LHX был создан разведывательный вертолет Comanche, показанный на фронтисписе главы. Проекты создания обоих вертолетов были успешно завершены, но в дальнейшем программа их развития подверглась пересмотру в связи с передачей разведывательных функций беспилотным летательным аппаратам.

2. В книге [195] приведены результаты исследования того, как правительство США осуществляет закупки продуктов инновационных технологий. "Всю эту книгу пронизывает мысль о том, что для достижения успеха необходимо предоставить проектировщикам возможность неизменно придерживаться требований технической целостности продукта" (комментарий рецензента Мэри Шоу (Mary Shaw)). Автор книги Сквайрс (Squires) убеждает проектировщиков быть преданными сторонниками идеи обеспечения целостности продукта:

Ученый или инженер, работающий в прикладной области, должен полностью посвятить себя совершенствованию проектируемого объекта с момента выдвижения его концепции и до момента ввода в действие для эксплуатации.

3. Некоторый анонимный рецензент обоснованно отметил, что определенное требование, обусловленное одним заинтересованным лицом в качестве необязательного, для другого часто может оказаться крайне необходимым. Но, по моему мнению, из этого следует, что при всех обстоятельствах определенная часть будущих пользователей категорически требует реализации каких-то конкретных функций. С другой стороны, хотя вряд ли кто-то будет возражать против того, чтобы создаваемый продукт был быстродействующим, малогабаритным, надежным и удобным в использовании, подобные требования не так уж часто предъявляют в категорической форме на этапе разработки технического задания, и это в значительной степени обусловлено тем, что на столь раннем этапе невозможно предсказать, не повлияет ли попытка достижения этих требований на успешную реализацию основных функций продукта.
4. В главе 9 описано, какими моделями взаимодействия с пользователями руководствуются проектировщики.
5. См. доклад [2].
6. См. доклад [2], с. 4. Но заслуживает также внимания с. 50 доклада, содержимое которой может быть неправильно истолковано:

   Необходимо четко установить полный и стабильный набор требований системного уровня и выяснить состав продуктов на этапе А. Безусловно, непостоянство состава требований представляет собой реальную проблему, которая должна быть решена, но остается необходимым обеспечение определенной адаптивности состава требований, позволяющей учесть результаты тестирования и эксплуатации, касающиеся осуществимости и практической применимости... Безусловно, необходимо обеспечивать стабильность состава требований, но не следует ее абсолютизировать.

   В беседе со мной председатель комитета, доктор Пол Камински (Paul Kaminsky), и ответственный сотрудник Национального научно-исследовательского совета (США), Джеймс Гарсиа (James Garcia), разъяснили, что цели комитета сформулированы в абзаце на с. 4 доклада. Джеймс Гарсиа указал:

   Цель комитета — следить за тем, чтобы четкие ключевые показатели производительности были разработаны на этапе А, а четкие и полные требования — на этапе Б, как указано в резюме и в главе 4. Комитет полагает, что для выяснения состояния разработки четких и полных требований системного уровня необходимо взаимодействие с потенциальными подрядчиками, осуществляемое при переходе от этапа А к этапу Б.

7. Джон Макманус (John McManus), обладатель сертификата MBCS, является одним из ведущих специалистов в области методов руководства проектами и разработки программного обеспечения. Доктор Тревор Вуд-Харпер (Trevor Wood-Harper) — профессор системотехники в Солфордском университете, отметил: "Жизненно важными точками отсчета для ИТ-проектов являются техническое задание к проекту, в котором изложены причины выдвижения этой новой инициативы и описаны ожидаемые результаты, а также выбранные руководителем проекта пути решения предстоящей задачи" (см. [150]).
8. В книге [30] описан проведенный с участием студентов эксперимент, в котором одна группа подготавливала проект на основе тщательно выверенной спецификации проекта, а вторая группа выполняла ту же работу исходя непосредственно из требований к проекту. Система, предложенная первой группой, отличалась таким недостатком, как наличие многих ненужных функций, поскольку проектировщики не могли отказаться от добавления все нового и нового, чтобы "дополнить" проект или добиться его единообразия. Таким образом, в создании "раздутых" проектов виноваты не только те, кто подготавливает списки пожеланий, но и сами проектировщики. Я сам в свое время допустил такую ошибку при создании компьютера Stretch компании IBM.
9. В статье [129] рассматриваются новаторские схемы заключения контрактов, которые использовались в рамках развития партнерства между частным капиталом и общественностью при проведении общественных работ в Великобритании.
10. В статье [157] рассматриваются материалы одной из конференций с предложениями об обеспечении наиболее тесного взаимодействия заказчиков и подрядчиков в целях минимизации непредвиденного риска при заключении контрактов между ними.

Совокупная стоимость
Поэтапное развитие
Оценка альтернатив, определение и устранение рисков
Разработка и проверка продукта следующего уровня
Определение целей, альтернатив, ограничений
Планирование следующих фаз
Обзор
Распределение ответственности
Анализ риска
Прототип 1
Концепция эксплуатации
Планирование требований
Планирование жизненного цикла
Прототип 2
Требования к программному обеспечению
Прототип 3
Макеты, модели, эталонные тесты
Эксплуатационный прототип
Проектирование программного продукта
Детализированное проектирование
Кодирование
Модульное тестирование
Интеграция и проверка
Приемочное испытание
Эксплуатация
План разработки
Проверка требований
План интеграции и проверки
Проверка и подтверждение правильности проекта

5

Поиск лучших моделей процесса проектирования

Общепризнано, что творческое проектирование состоит не в том, чтобы вначале формулировать задачу, а затем находить удовлетворительную концепцию решения; вместо этого оно, по-видимому, в большей степени сводится к одновременному поиску и уточнению не только формулировки задачи, но и идей для ее решения, при постоянной итерации процессов анализа, синтеза и оценки с охватом двух "пространств" — задачи и решения.

См. [71]

◂ *Спиральная модель Боэма (см. [28]). Boehm © 1988 IEEE*

TheDesignofDesign.indb 81 30.10.2012 15:58:24

Необходимость в определении доминирующей модели

В настоящее время и практические потребности проектирования, и потребности обучения будущих проектировщиков обязывают нас срочно заняться поиском решения следующей задачи:

- если рациональная модель действительно неверна,
- если применение неправильной модели существенно влияет на организацию работ по проектированию и
- неправильная модель все еще остается жизнеспособной по глубоким причинам,

то необходимо безотлагательно перейти к использованию лучших моделей, которые

- ставят во главу угла последовательное раскрытие и развитие требований к проекту,
- могут быть описаны настолько наглядно, чтобы они легко усваивались в процессе изучения, были вполне доступными для понимания членов коллектива проектировщиков и заинтересованных лиц и
- позволяли быстро перейти к их использованию даже сторонникам другой, неправильной модели.

Все модели по определению представляют собой упрощенные абстракции действительности. Это означает, что может быть предложено много применимых моделей развития жизненного цикла процесса проектирования, отличающихся тем, что в них одни аспекты подчеркиваются, а другие исключаются. Майк Пик (Mike Pique) снял видеофильм, резко подчеркнувший эту особенность, в котором показано приблизительно 40 различных графических компьютерных моделей белка бычьей супероксиддисмутазы: каркасные, ленточные, твердотельные, подвижные модели и др.[1]

Поэтому можно не без оснований утверждать, что поиск доминирующей модели для процесса проектирования — бессмысленное занятие. Так почему бы не принять на вооружение лозунг: "Пусть расцветают 100 моделей"? Каждая из них добавит новые краски.

Я категорически с этим не согласен. Сам факт повсеместного применения модели водопада в разработке программного обеспечения, несмотря на многие критические замечания и значительные потери, вызванные ее чрезмерным упрощением, позволяет убедиться в том, что ее жизнестойкость обусловлена необходимостью организовать общение на единой основе и самим характером теоретического обучения, т.е. потребностью в наличии пусть даже не совсем удачной, но единой, доминирующей модели процесса проектирования. Этим обусловлена

безотлагательная необходимость полностью заменить все прочие модели наиболее продуктивной моделью, а не просто продолжать сложившуюся практику усовершенствования существующих моделей. И действительно, что касается более широкой сферы проектирования, то я был свидетелем того, как применение модели решения задач Саймона приводило к большим затратам усилий для преодоления возникающих при этом тупиков теми, кто пытался лучше изучить и усовершенствовать проект.

Коэволюционная модель

Маэр (Maher), Пун (Poon) и Буанже (Boulanger) предложили формальную модель, которую я нахожу полезной, — коэволюционную модель[2,3]. Кросс (Cross) и Дорст (Dorst) описывают эту модель следующим образом:

> *Создается впечатление, что творческое проектирование не сводится к первоначальному определению задачи, а затем поиску удовлетворительной концепции решения. Творческое проектирование, по-видимому, в большей степени представляет собой процесс разработки и усовершенствования одновременно и формулировки задачи, и идеи ее решения, при котором постоянно происходит итерация этапов анализа, синтеза и оценки результатов в двух абстрактных пространствах проектирования — в пространстве задач и в пространстве решений. Модель творческого проектирования, предложенная Маэр и др. [146], основана на такой "коэволюции" пространства задач и пространства решений в процессе проектирования: пространство задач и пространство решений эволюционируют одновременно друг с другом в условиях непрерывного обмена информацией (рис. 5.1).*[4]

В этом контексте термин "эволюция" трактуется в широком смысле. Эволюционность модели определяется тем, что изучение задачи и разработка решения происходят в виде поэтапных приращений и поэтапно осуществляемых оценок результатов.

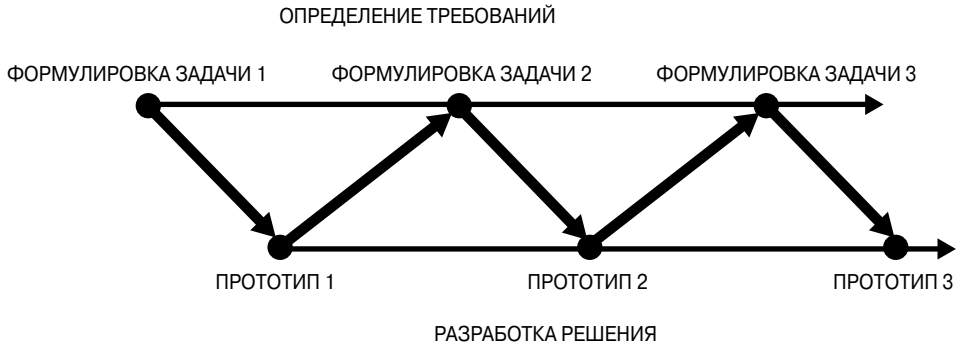


Рис. 5.1. Коэволюционная модель проектирования Маэр, Пуна и Буанже (см. [146])

Недавно в трудах по философии, посвященных анализу научно-технического прогресса, подробно рассматривался вопрос о том, насколько точно процесс новаторской деятельности человечества коррелирует с ходом биологического развития. В книге [230], многие главы которой написаны специалистами по различным дисциплинам, приведена содержательная сводка уровня развития этой области, достигнутого учеными, по состоянию на 2000 год[5]. В этой книге выдвинуты аргументы и в пользу того, что биологическая эволюция достаточно качественно моделирует научно-технический прогресс, поскольку и в том и в другом случае происходят случайная выработка новых признаков и естественный отбор, и в пользу противоположной точки зрения, поскольку новаторская деятельность осуществляется в форме целенаправленного проектирования (по-видимому, авторы этой книги полагают, что к эволюции это не относится).

Коэволюционная модель определенно подчеркивает роль последовательного уточнения и формулировки требований. Эту модель можно представить в виде графического изображения, легко доступного для запоминания. Ее не следует рассматривать как всеобъемлющую, поскольку в ней не предпринимаются попытки охватить все аспекты процесса проектирования, создания, тестирования, внедрения и сопровождения. Более того, графическое изображение модели не наталкивает на мысль, что представляемый ею процесс является сходящимся. Насколько я могу судить, эта модель еще не получила существенного последующего развития, а в своей первоначальной формулировке не содержит средств определения этапов и договорных обязательств. Безусловно, эта модель является привлекательной и лучшей по сравнению с рациональной моделью, но, по моему мнению, она недостаточно развита.

Модель карусели. Тем не менее пока мы не можем исключить из рассмотрения ни одной модели. Дело в том, что другие предложенные модели, пусть даже и очень развитые, являются настолько сложными, что затрудняют понимание, почти не поддаются запоминанию и не позволяют легко проводить продуктивное обсуждение. Примером может служить модель карусели Хиклинга (Hickling), с циклами и эпициклами[6].

Модель разработки Реймонда по принципу функционирования базара

Реймонд (Raymond) в своей великолепной книге [174] утверждает, что применявшаяся ранее организация процесса проектирования, которую он сравнивает с возведением собора, вышла из моды с того времени, как стали применяться разработки с открытым исходным кодом, организованные по принципу функционирования базара; в частности, подобный процесс оказался плодотворным и эффективным при создании Linux, на удивление многофункциональной и надежной

операционной системы. Аргументы, приведенные Реймондом, являются убедительными и четко сформулированными. Он утверждает, что процесс разработки с открытым исходным кодом может стать наиболее эффективным способом создания прикладных программ всех типов, а также операционных систем. Он иллюстрирует это утверждение на примере программы Fetchmail — его собственного создания[7].

Описание работы модели

Реймонд описывает процесс, организованный по принципу функционирования базара, так: какой-либо представитель сообщества пользователей/разработчиков обнаруживает определенную потребность, создает некоторый программный модуль в целях ее удовлетворения, а затем безвозмездно передает этот модуль в общее пользование. Сообщество, сосредоточенное вокруг операционной системы Linux, оценивает применимость модуля и встраивает его в систему, причем последняя задача значительно упрощается благодаря модульной структуре Linux, особенно тем, что в этой системе широко применяются такие механизмы, как каналы и фильтры. Тот же процесс применим при исправлении ошибок. Один из членов сообщества обнаруживает программную ошибку в используемом им модуле, находит ее причину, устраняет ее в целях успешного продолжения своей работы, после чего безвозмездно передает всему сообществу предложенное им исправление.

Очевидно, что люди пишут новые модули и устраняют программные ошибки прежде всего для того, чтобы можно было самим выполнять собственную работу. Что же их побуждает раздавать бесплатно свои результаты, притом что для проведения необходимого тестирования, документирования и публикации требуется существенная дополнительная работа?[8] Реймонд на этот вопрос дает такой ответ, который мне кажется отвечающим действительности, что стимулом и наградой является престиж[9].

Часто сообществу предлагают многие варианты модулей или многие предложения по исправлению одной и той же программной ошибки. Реймонд утверждает, что при этом работает механизм рынка (применительно даже к бесплатным товарам). Чем лучше инструментальное средство и чем лучше исправление, тем большее распространение оно получает, и пропорционально с этим его автор приобретает больший престиж.

Поэтому при разработке, организованной по принципу функционирования базара, на виртуальном рынке присутствует много "продавцов", предлагающих свои программные товары в электронном виде. "Покупатели" разбирают эти товары и платят своими голосами, внося свою долю в повышение престижа автора, о чем разносятся сообщения с помощью электронных средств по всему международному сообществу.

Сильные стороны

На этой основе развилась удивительно мощная культура, в которой наградой становится престиж! Насколько это не похоже на тупую работу за деньги и отстаивание прав на личную интеллектуальную собственность! Какие широкие перспективы открывает эта новая модель для любой другой социальной активности!

Кроме того, Реймонд убедительно доказывает, что системные продукты, произведенные в процессе, организованном по принципу функционирования базара, в целом являются технически более совершенными по сравнению с теми, которые произведены в процессе, организованном по принципу строительства собора. Во-первых, применение рыночного механизма способствует отбору по эволюционному принципу тех модулей, которые спроектированы лучше других. Во-вторых, новый модуль подвергается одновременному тестированию с участием сотен пользователей, что приводит к более быстрому выявлению ошибок и получению более надежного продукта. В-третьих, более качественно происходит устранение программных ошибок, поскольку рынок выбирает лучшие среди исправлений.

Таким образом, процесс разработки, организованный по принципу функционирования базара, выдвигает на передний план полностью новую модель как для создания продуктов, так и для обеспечения сотрудничества среди взаимодействующих с помощью электронных средств, пусть даже разделенных многими часовыми поясами и незнакомых друг с другом коллективов.

Я настоятельно рекомендую читателям ознакомиться с указанной книгой Реймонда, причем не только с широко цитируемым эссе "Собор и базар", но и с другими главами. В ней много справедливых утверждений, много идей и мудрых мыслей.

Организация разработки по принципу функционирования базара

Тем не менее я вынужден сделать шесть кратких независимых замечаний в отношении процесса разработки с открытым исходным кодом. Этим я заменяю продолжительное обсуждение, поскольку не имею опыта работы по такому принципу.

- В действительности модель организации разработки по принципу функционирования базара является эволюционной. Укрупнение системы происходит путем добавления компонентов, каждый из которых удовлетворяет некоторую потребность (или, если можно так выразиться, соответствует определенному требованию), обнаруженную пользователем, выполняющим роль проектировщика.
- В экономических отношениях, в которых вознаграждением является повышение престижа, могут участвовать только люди, зарабатывающие себе на жизнь каким-то иным способом. Это означает, что предоставляемое

бесплатно в распоряжение сообщества программное обеспечение является побочным продуктом по отношению к другим плодам производственной деятельности, приносящим доход, часть которого идет на оплату труда создателей/дарителей бесплатного ПО.

- Итак, многие продукты, созданные таким образом, в действительности являются побочными продуктами, поэтому среди них чаще встречаются инструментальные средства, а не приложения. Конечные результаты не всегда в достаточной степени отшлифованы или отлажены, поскольку критерием оценки их качества служит лишь то, насколько хорошо они соответствуют потребностям создателя программы. А "рыночный" отбор фактически базируется на контроле качества.
- Безусловно, было очень много сказано об "открытости" и "бесплатности" программного обеспечения, созданного в рамках разработок с открытым исходным кодом, но даже если речь идет о всем "здании" операционной системы Linux, то его нельзя назвать иначе, как случайно сформированное нагромождение разрозненных частей, поскольку доминирующей интеллектуальной мощью, приводящей к достижению концептуальной целостности, обладал лишь создатель этой ОС, Линус Торвальдс (Linus Torvalds). Кроме того, при создании Linux была возможность воспользоваться готовой функциональной спецификацией, поскольку уже существовал полный проект равнозначной операционной системы, т.е. UNIX.
- Ключом ко всем процессам проектирования является раскрытие выражаемых пользователями потребностей, пожеланий и критериев. Заметный успех процесса разработки, организованного по принципу функционирования базара, в сообществе Linux, по моему мнению, непосредственно обусловлен тем фактом, что создатели программного обеспечения являются одновременно его пользователями. Эти пользователи сами выдвигают свои требования, руководствуясь собственными пожеланиями и потребностями своей деятельности. При этом лишь собственный опыт пользователей непроизвольно становится основой формулировки ими требований, критериев и оценок. Никто не формулирует задачи, стоящие перед всей системой в целом, поэтому и требования к системе остаются лишь подразумеваемыми. Я хочу выразить весьма серьезные сомнения в том, что разработки с открытым исходным кодом были бы такими же успешными, если бы создатели этих программ сами не были пользователями и имели бы лишь косвенное представление о потребностях пользователей.
- Следовательно, все еще остаются необходимыми организованные по принципу строительства собора процессы разработки программного обеспечения, тщательно спроектированные архитектором, неукоснительно управляемые и придирчиво тестируемые. Стали бы вы, например, использовать процесс разработки с открытым исходным кодом для создания новой общегосударственной системы управления воздушным движением?[10]

Спиральная модель Боэма

В 1988 году Барри Боэм (Barry Boehm) предложил использовать спиральную модель при создании программного обеспечения[11]. На фронтисписе этой главы приведена указанная модель в том виде, в каком она была предложена первоначально. Сама спираль, как геометрическая форма, наталкивает на мысль о развитии. В спирали связаны последовательные повторения аналогичных действий. Эта геометрическая форма легко доступна для понимания и запоминания. В модели сделан акцент на создании прототипов, начиная с прототипов пользовательского интерфейса, а тестирование с участием пользователей проводится задолго до того, как становится возможным подготовка полнофункционального прототипа.

Эта модель нашла широкое применение, а в департаменте снабжения Министерства обороны США даже разрешено ее применение вместо модели водопада[12]. Кроме того, эта модель также получила определенное развитие.

В статье [78] приведены критические замечания, касающиеся спиральной модели: "[Она представляет собой шаг вперед], но все еще сосредоточена на работе проектировщика и создании продукта, а не направлена на выполнение требований пользователя и обеспечение функционирования системы"[13]. В этой статье в довольно неформальном виде предложен процесс проектирования, направленный на обеспечение функционирования системы, в котором уделено достаточное внимание требованиям пользователя и моделям эксплуатации. Рекомендую ознакомиться с этой содержательной статьей.

Тем не менее модель развития используется преимущественно разработчиками, поэтому я считаю полностью правильным то, что она сосредоточена на работе по проектированию. Кроме того, предложенный в указанной статье подход к организации разработки не отвечает потребности в наличии легко запоминаемой геометрической модели и не может служить основой для формального согласования проектных решений. Поддерживая Боэма и выступая против Дэннинга и Дарган, я настаиваю на том, чтобы взаимодействие с представителями пользователей было частым, но не непрерывным, а также на том, чтобы основой этого взаимодействия служили последовательно разрабатываемые прототипы.

Спиральная модель в том виде, в каком она была предложена первоначально, допускает осуществление последовательного раскрытия требований, но не ставит этот процесс во главу угла и не требует неукоснительного согласования договорных позиций.

Я выражаю полную уверенность в том, что мы достигнем цели создания оптимальной модели проектирования, взяв на вооружение и доработав спиральную модель. Предлагаю также задавать на спирали метки, обозначающие явно заданные договорные позиции, дополненные четким описанием того, что подлежит согласованию, какими должны быть достигнутые результаты и в чем выражается явное распределение риска. Управление риском лежит в центре внимания многих последних работ Боэма[14].

Модели процесса проектирования — сводка аргументов, приведенных в главах 2–5

- Формальная модель процесса проектирования необходима для обеспечения организации работ по проектированию, оказания помощи в общении в ходе проектирования и обсуждения самих проектов, а также для обучения.
- Крайне важно иметь визуальное, геометрическое представление модели проектирования, поскольку проектировщики — люди преимущественно с пространственным мышлением. Проектировщикам гораздо легче изучать модель, раздумывать над ней, обмениваться мнениями и проводить обсуждение в терминах модели, имеющей наглядное геометрическое изображение.
- Для инженеров кажется естественным применение рациональной модели проектирования. В действительности эта модель была независимо и формально определена несколькими разными специалистами, например, Саймоном, Палем и Бейцем, а также Ройсом.
- Рациональная модель является линейной, ступенчатой, поэтому не отражает суть организации проектирования и вводит в заблуждение. Эта модель не способна точно отразить, какие работы фактически выполняют проектировщики, или передать то, что лучшие специалисты в области проектирования рассматривают как сущность процесса проектирования.
- При использовании неподходящей модели нельзя обойтись без потерь. Это приводит к преждевременному взятию на себя обязанностей по выполнению, возможно, неправильно понятых требований, что влечет за собой создание продуктов, перегруженных ненужными функциями, а также к нарушению графика работ, раздутию бюджета и снижению производительности.
- Но рациональная модель неизменно применяется на практике, несмотря на ее неадекватность и большое количество убедительных критических замечаний. Это связано с тем, что данная модель обладает соблазнительной логической простотой, а также позволяет согласовывать договорные позиции, без чего не могут обойтись разработчики программного обеспечения и заказчики.
- Было предложено несколько альтернативных моделей процесса проектирования. Я считаю наиболее перспективной спиральную модель Боэма. Мы должны продолжить заниматься ее развитием.

Примечания и ссылки

1. См. [169].

2. См. [146].

3. См. [147].

4. См. [71], [86].
5. См. [230].
6. См. [114].
7. См. [174]. Я подозреваю, что у Реймонда представление о "разработках по принципу строительства собора" сложилось под влиянием изучения фронтисписа к главе 4 книги [40].
8. См. [40], глава 2.
9. См. [174].
10. Ричард П. Кэйс (Richard P. Case) глубокомысленно заметил:

    Возможно, нам следует различать два аспекта проектирования с открытым исходным кодом — "любой может вносить в код изменения" и "любой может просматривать код". Если любой может вносить изменения, то невозможно с полной уверенностью утверждать, что не произойдут отклонения в худшую сторону (иногда даже фатальные), а также иметь хотя бы какую-то гарантию того, что изменения действительно привели к исправлению программной ошибки и сбой в работе больше не произойдет. Еще одна проблема заключается в обеспечении секретности. Если к реальному проектированию имеют доступ лишь несколько человек, то уменьшается вероятность, что недостатки будут обнаружены на раннем этапе и что удастся разработать и оценить альтернативные концепции. (По материалам личного общения [2009].)

11. См. [28].
12. В дальнейшем Министерством обороны США были приняты адаптированные должным образом промышленные стандарты разработки программного обеспечения IEEE/EIA 12207.0, IEEE/EIA 12207.1 и IEEE/EIA 12207.2. Эти стандарты допускают использование спиральных моделей.
13. См. [78], с. 110.
14. См. [188], где в главе 5 собраны наиболее важные статьи Боэма, посвященные изучению рисков.

II

Организация совместной работы в режиме непосредственного общения и в дистанционном режиме

6

Совместная работа в проектировании

Встречи с людьми — это спасение от "безысходности работы и одиночества размышлений"[1].

БЕРНАРД БАРУХ (BERNARD BARUCH),
ЦИТИРУЕТСЯ ПО КНИГЕ [176]

◀ *Мост Sunniberg архитектора Менна, 1998 год. Кристиан Менн, ETH Zuerich, ChristianMennPartners AG*

Преимущества и недостатки организации работы на основе привлечения большого количества людей

С начала XX века в организации проектирования произошли два значительных изменения.

- Проектирование стало осуществляться главным образом коллективами, а не отдельными людьми.
- В современных коллективах проектировщиков для организации совместной работы все чаще применяются средства передачи данных, а не непосредственное общение.

Следствием этих существенных изменений стало то, что в сообществе проектировщиков все большее внимание привлекают следующие актуальные темы:

- Совместная работа в дистанционном режиме.
- "Виртуальные коллективы" проектировщиков.
- "Виртуальные студии проектирования".

Все эти способы организации совместной работы стали возможными благодаря применению телефонной связи, сетевого взаимодействия, компьютеров, графических дисплеев и средств проведения видеоконференций.

Чтобы понять принципы организации совместной работы в дистанционном режиме, необходимо вначале разобраться в том, какова роль сотрудничества в современном профессиональном проектировании.

Вообще говоря, принято считать, что достаточно обеспечить сотрудничество, и дела сами по себе "пойдут как надо". Начиная с детского сада, воспитатели и преподаватели положительно отзываются о ребенке, который хорошо играет с другими. "Мы все вместе взятые умнее, чем каждый из нас в отдельности". "Чем больше людей участвует в проектировании, тем лучше". Но теперь настало время, когда эти привлекательные суждения кажутся не столь очевидными. Во всяком случае, можно утверждать, что они, безусловно, не являются универсально истинными.

Большинство великих произведений человеческого ума было сотворено одним или двумя людьми в тесном сотрудничестве. В частности, это относится к большинству выдающихся технических достижений конца XIX и начала XX столетий. Но в наши дни современным стандартом считается коллективное проектирование, и для этого есть серьезные основания. Безусловно, с этим связана также опасность нарушения концептуальной целостности конечного продукта, что действительно становится очень серьезной потерей. Поэтому при организации совместной работы приходится решать две важные задачи — обеспечить сохранение концептуальной целостности при осуществлении коллективного проектирования и вместе с тем добиться реализации действительно значимых потенциальных преимуществ сотрудничества.

Коллективное проектирование как современный стандарт организации работы

Коллективное проектирование стало стандартным способом организации совместной работы при создании всех современных продуктов, начиная от серийно выпускаемых изделий и заканчивая такими уникальными проектами, как здания или программные комплексы. В этом действительно состоит самое значительное изменение со времен XIX столетия. В памяти человечества навсегда останутся имена выдающихся технических проектировщиков, работавших в XVIII и XIX столетиях: Картрайт, Уатт, Стефенсон, Брунель, Эдисон, Форд, братья Райт и др. С другой стороны, рассмотрим атомную подводную лодку Наутилус (рис. 6.1). Мы знаем, что инициатором ее создания был адмирал Риковер (Rickover), финансирование проекта обеспечил Уилл (Will), но кто из нас может назвать имя главного проектировщика? Создание этой атомной подводной лодки стало плодом труда коллектива квалифицированных специалистов.

Рис. 6.1. Атомная подводная лодка Наутилус, по материалам Лаборатории по изучению плавания подводных лодок в водах Арктики ВМС США (U.S. Navy's Arctic Submarine Laboratory); сайт Wikimedia Commons

С другой стороны, многое было сделано великими творцами, которые брали на себя всю ответственность за достижение конечного результата:

- Гомер, Данте, Шекспир.
- Бах, Моцарт, Гильберт и Салливен.

- Брунеллески, Микеланджело.
- Леонардо да Винчи, Рембрант, Веласкес.
- Фидий, Роден.

Большинство великих произведений было создано одним человеком. В создании выдающихся творений человеческого ума нередко участвовали также двое. И действительно, два считается магическим числом, когда речь идет об организации совместной работы. На эту тему уже было сказано очень много; достаточно отметить такое блестящее изобретение человечества, как семья (создаваемая двумя людьми).

Причины перехода в организации технического проектирования от применения отдельных специалистов к коллективам

Усложнение технологий. Наиболее очевидным побудительным требованием к переходу на коллективное проектирование является существенное усложнение каждого аспекта разработки. Достаточно сравнить один из первых в мире железных мостов (рис. 6.2) с тем великолепным вариантом такого же сооружения, которое появилось много позднее (фронтиспис данной главы).

В первом случае проектировщику пришлось применять конструкции со слишком большим запасом прочности, т.е. строить тяжелый мост с большим расходом металла, который, тем не менее, выглядит достаточно изящным. Проектировщик не мог действовать иначе, поскольку знаний о свойствах металлических конструкций, а также о распределении статических и динамических напряжений в то время было недостаточно (хотя нельзя отрицать, что даже тот уровень знаний применен весьма успешно!).

Что же касается моста Менна, то это парящее над землей сооружение внешне кажется немыслимым, но вместе с тем является реальным результатом многолетней работы в области анализа и моделирования.

Вообще говоря, в свое время я убедился в том, что в современной науке и технике почти не осталось примитивных технологий, и это произвело на меня большое впечатление. Однажды мне была предоставлена честь совершить экскурсию по научно-исследовательской лаборатории компании Unilever в районе Порт-Санлайт, г. Мерсисайд, Великобритания. Я был очень удивлен, когда узнал, что прикладной математик со степенью доктора философии выполняет на суперкомпьютере расчеты по вычислительной гидрогазодинамике (Computational Fluid Dynamics — CFD), чтобы создать технологию оптимального смешивания шампуня! Он объяснил, что шампунь — это трехслойная эмульсия, состоящая из водных и масляных компонентов, и при создании технологии крайне важно добиться смешивания без трения.

Рис. 6.2. Железный мост Притчарда (Pritchard) и Дарби (Darby), 1779. (Шропшир, Великобритания), iStockphoto

Проектировщики комбайна для сбора хлопка из компании John Deere использовали метод CFD для оптимизации структуры потока воздуха, применяемого для транспортировки семенных коробочек хлопка. Современному фермеру приходится работать не только на тракторе, но и на компьютере, поскольку он должен подбирать удобрения, гербициды и инсектициды, определять наиболее подходящие семена, проводить анализ почвы и следить за журналом севооборота[2]. Шеф-повар в компании Sara Lee постоянно корректирует рецепты выпечки в соответствии с химическими свойствами поступающей муки; руководитель производства на бумагоделательной фабрике аналогичным образом вносит коррективы, учитывая изменяющиеся свойства перерабатываемой целлюлозной массы.

Невозможно найти ни одного аспекта современного производства, в котором не происходило бы существенное усложнение технологий, поэтому невозможно обойтись без привлечения специалистов различных профилей. К тому времени, как я поступил в аспирантуру в 1953 году, многие люди были способны в одиночку овладеть всеми компьютерными науками. Проводились всего две ежегодные конференции и выпускались два журнала, выходящие раз в три месяца. А в тот год, как я закончил аспирантуру, мне уже пришлось выбирать между отдельными направлениями в области компьютерных наук, с сожалением отказываясь от одних в пользу других, поскольку способности человеческого разума следить сразу за

всем были исчерпаны: математическая лингвистика, базы данных, операционные системы, научные вычисления, разработка программного обеспечения и даже архитектура вычислительных систем — вот мои первые излюбленные направления. Распределение по направлениям такого рода произошло во всех творческих науках, поэтому проектировщики современных изделий не могут обойтись без помощи со стороны разных специалистов.

Взрывообразный рост потребности в конкретных знаниях в области многих технологий был частично компенсирован благодаря тому, что столь же стремительно происходило накопление нужных знаний и совершенствование способов их получения с помощью документов, квалифицированных специалистов, программного обеспечения, анализа, а также поисковых серверов, которые позволяют находить нужные документы и потенциальных кандидатов, которые могут быть привлечены в качестве сотрудников.

Стремление как можно скорее выйти на рынок. Еще одна важная побудительная сила, которая вынуждает проектировщиков сплачиваться в коллективы, состоит в стремлении как можно быстрее вывести новый проект (новый продукт) на рынок. Согласно одному эмпирическому правилу, компания, которая первой вывела на рынок продукт нового типа, вполне может рассчитывать на долговременное обладание долей рынка в 40%, тогда как остальная часть будет неизменно распределяться между многочисленными, не столь проворными конкурентами. Кроме того, опередившая всех компания может собирать всю прибыль в тот период времени, пока конкуренты не могут предоставить свои аналоги. Встречаются и такие случаи, когда компания, впервые представившая на рынке определенный продукт, так и продолжает доминировать на нем. Эти законы рынка известны всем, поэтому соблюдение планов разработки проектов всегда находится в центре внимания руководства компаний. Коллективное проектирование становится необходимым, если оно позволяет ускорить поставку нового продукта в конкурентной среде[3].

Причины усиления конкурентной борьбы в части сроков выхода новых продуктов состоят в следующем. Системы связи и передачи данных охватили весь мир, а рынки стали глобальными, поэтому зародившиеся где бы то ни было привлекательные идеи в наше время распространяются очень быстро.

Затраты на организацию совместной работы

"Чем больше рук, тем **легче** становится работа". Так бывает часто.

"Но чем больше рук, тем **больше** становится работы". Так бывает всегда.

С первым утверждением согласится каждый. Это особенно справедливо в отношении задач, которые можно распределять по исполнителям. На каждого участника трудового процесса ложится меньшая нагрузка, поэтому продолжительность выполнения всей работы сокращается. Но возможности распределения задач проектирования по исполнителям ограничены, причем весьма невелико

количество таких задач, которые могут быть сведены к более мелким в достаточной степени[4]. Поэтому в связи с организацией совместной работы возникают дополнительные издержки.

Стоимость разделения на подзадачи. Само разделение задач проектирования на подзадачи представляет собой достаточно серьезную дополнительную проблему. Дело в том, что при этом должны быть четко и точно определены интерфейсы между подзадачами, а это связано с выполнением большого объема работы, требующей внимательного контроля. По мере выполнения проектных работ приходится непрерывно пересматривать интерфейсы, определяющие взаимодействие подзадач, независимо от того, насколько точно они были определены с самого начала. Не исключено и возникновение потенциальных нестыковок. В ходе проектирования неизбежно возникают несогласованности в определениях и конфликты в интерпретациях; все эти расхождения должны быть своевременно устранены.

В целях упрощения подготовки изделия к выпуску обязательно следует провести стандартизацию общих элементов по всем компонентам. Кроме того, необходимо добиться определенной общности стиля проектирования во всем коллективе.

А в конечном итоге отдельные части проекта должны быть собраны в единое целое, что становится окончательной проверкой правильности интерфейсов. В наши дни нельзя больше руководствоваться принципом, применявшимся при сборке судна на старинных верфях: "Отрезай по чертежу, подгоняй по месту"[5].

Стоимость изучения и обучения. Если в проектировании совместно участвуют `n` человек, то все они должны стремиться быстрее всего достичь поставленных целей, выполнить предъявленные требования, соблюсти ограничения и добиться оптимизации функции полезности. Вся группа проектировщиков должна иметь общее представление обо всех этих аспектах проектирования, т.е. хорошо представлять себе, что должно быть получено в конечном итоге. В качестве первого приближения можно утверждать, что если работа по проектированию, выполняемая одним человеком, состоит из двух частей (изучение `l` и разработка `d`), то общий объем работы `work` с привлечением `n` человек больше не измеряется формулой

```
work= l + d,
```

а составляет по меньшей мере

```
work= n*l + d.
```

Более того, определенная часть сотрудников, обладающая лучшим представлением о будущем проекте и бо́льшим объемом знаний, должна заниматься обучением других и в связи с этим не принимать участия в самом проектировании. Таким образом, остается лишь надеяться на то, что благодаря преимуществам специализации хотя бы некоторые из этих затрат окупятся.

Расходы на обеспечение связи и передачи данных в процессе проектирования. На протяжении всего процесса проектирования совместно работающие

проектировщики должны быть уверены в том, что создаваемые ими отдельные части в конечном итоге будут успешно состыкованы. Для этого необходимо организовать между проектировщиками обмен структурированными данными.

Управление изменениями. Должен быть введен в действие такой механизм управления изменениями, с помощью которого каждый проектировщик мог бы вносить только те изменения, которые, во-первых, затрагивают лишь порученные ему части проекта или, во-вторых, согласованы с проектировщиками частей проекта, затрагиваемых предлагаемыми изменениями. В действительности значительную долю затрат на проектирование составляют изменения и переделки, поэтому стоимость управления изменениями становится достаточно заметной. Но если формальный процесс управления изменениями не организован, то приходится нести гораздо бо́льшие издержки[6].

Достижение концептуальной целостности как основная цель

Одним из наиболее важных внешних признаков качественного проекта является целостность и единообразие его концепций. Рассмотрим в качестве примера созданный архитектором Реном (Wren) великолепный шедевр — собор Святого Павла (рис. 6.3).

Такая согласованность во всем конкретного проекта не только вызывает восторг, но и способствует удобству в изучении и простоте в использовании. Кроме того, собор Святого Павла как здание практического назначения полностью соответствует своим функциям. Я в своей книге [40] специально отметил, что концептуальная целостность является одним из наиболее важных требований в системном проектировании[7]. Иногда вместо термина *концептуальная целостность* применяются термины *согласованность*, *непротиворечивость* и даже *единообразие стилей*. Мы вместе с Блау (Blaauw) в другой книге довольно подробно рассмотрели понятие концептуальной целостности и указали, что ортогональность, уместность и общность являются основными принципами идентификации компонентов[8]. Работающие самостоятельно проектировщики или художники в своем деле обычно придерживаются принципов концептуальной целостности подсознательно; принимая решения, касающиеся каждой детали будущего произведения, они, как правило, делают выбор в пользу согласованного общего результата (если иного не требуют достаточно веские причины). Если творцу не удается добиться такой целостности, то зрители оценивают его работу как испорченную, но отнюдь не великую.

Даже в наши дни многие выдающиеся технические проекты являются плодом творчества одного или двух человек. Достаточно вспомнить о мостах, спроектированных архитектором Менном[9]. Замечательным примером являются также компьютеры, разработанные компанией Сеймура Крэя (Seymour Cray). Крэй проявил себя как всесторонне развитый специалист, способный в своих проектах сохранить контроль над разрабатываемой конструкцией в целом и над отдельными ее

компонентами, начиная от архитектуры и заканчивая электрическими схемами, компоновкой и организацией системы охлаждения, что позволяло ему принимать оптимальные решения, касающиеся всех основных и смежных областей проектирования[10]. При этом Крэй неизменно брал под свою ответственность только такие проекты, над которыми он мог сохранять полный контроль, несмотря на то, что в его распоряжении и под его руководством был целый коллектив. Крэй умел противостоять любым корпоративным и внешним факторам, под влиянием которых он должен был бы отвлечь свое внимание от проекта и заняться другими делами. Он не раз переводил свой коллектив проектировщиков из лабораторий, в которых раньше добивался успеха, полностью на новые площадки и считал, что лучше создать условия для проявления творческой индивидуальности, чем возможность совместной работы. Крэй гордился тем, что разработал компьютер CDC 6600 в коллективе из 35 человек, "включая швейцара"[11].

Рис. 6.3. Собор Святого Павла архитектора Рена, iStockphoto

Таким образом, как в этом примере, предпосылками создания поистине инновационных, а не клонированных продуктов являются физическая изолированность коллектива, небольшое количество сотрудников, полное сосредоточение на полученном задании и единоличное руководство со стороны выдающегося человека. В качестве примеров можно указать разработку самолета Spitfire автономным коллективом под руководством Джо Митчелла (Joe Mitchell), которая проводилась в Херсли-Хаус, величественном здании в Хэмпшире, Великобритания; работы секретного подразделения Skunk Works компании Lockheed, возглавляемые Келли Джонсоном (Kelly Johnson), в результате которых были созданы самолет-шпион U-2 и истребитель F-117, выполненный по технологии Stealth; деятельность секретной лаборатории IBM в г. Бока-Ратон, шт. Флорида, позволившая IBM успешного догнать Apple в области разработки персонального компьютера.

Возражения

Принципы создания новаторских продуктов, которые я изложил выше, признают не все. Некоторые мои оппоненты утверждают, что при проектировании необходимо также учитывать интересы потенциальных пользователей, т.е. предоставить им возможность участвовать в проектировании объектов, предназначенных для использования ими[12]. Но если дело касается зданий, то участие в проектировании многих людей, которые будут его занимать, может быть оправданным (а также позволит избежать возможных конфликтов и учесть обоснованные требования), если же речь идет о разработке товаров массового спроса, то неизбежно приходится ограничиваться лишь небольшой выборкой предполагаемых пользователей. В последнем случае мнение каждого пользователя становится более весомым, поэтому приходится контролировать представительность выборки, отдавая должное способности проектировщика предвидеть успех будущего изделия.

Некоторые утверждают, что я неправильно трактую факты, подчеркивающие важность вклада отдельных выдающихся личностей, и доказывают, что фактически коллективное проектирование всегда было нормой[13]. Читатель вправе сам сделать выводы из изложенного.

Способы достижения концептуальной целостности при коллективном проектировании

Даже если намеченное к выпуску изделие является очень крупным, технически сложным или предназначенным для создания в кратчайшие сроки, не допускается ни малейшего отступления от требования, чтобы это произведение конструкторской мысли многих участников разработки выглядело как концептуально целостное с точки зрения каждого пользователя[14]. Безусловно, такая целостность обычно становится естественным следствием проектирования, осуществляемого одним человеком, но для ее достижения при коллективном проектировании

приходится решать нетривиальные управленческие задачи, требующие большого внимания. Рассмотрим возможности организации работ по проектированию в целях достижения концептуальной целостности.

Современное проектирование как междисциплинарное согласование

На основе того наблюдения, что в современных условиях достигнута чрезвычайно высокая степень разделения труда по отдельным специальностям, многие ученые (главным образом, теоретики) пришли к выводу, что в наши дни природа проектирования изменилась: сегодня проектирование должно осуществляться по принципу междисциплинарного согласования (в рамках одного коллектива). Из того, что под взаимодействием подразумевается согласование, следует вывод о равноценности мнений всех членов коллектива и о необходимости учитывать мнение каждого. Но это неверно! Если конечная цель состоит в достижении концептуальной целостности, то, руководствуясь необходимостью согласовывать все предложения, считающиеся равноценными, неизбежно приходишь к созданию продукта, перегруженного лишними функциями! Результат такого проектирования можно сравнить с решениями комитета, на заседаниях которого никто не смеет возражать против чьих-либо предложений[15].

Системный архитектор

Наиболее важным и единственным способом обеспечения концептуальной целостности при коллективном проектировании является наделение полномочиями единовластного системного архитектора. Безусловно, этот человек должен разбираться во всех технологиях, относящихся к проектируемому продукту. Кроме того, он должен обладать опытом проектирования систем такого типа, которая ему поручена. Но важнее всего то, что он должен обладать четким представлением о самой будущей системе и ее назначении и действительно заботиться о достижении концептуальной целостности этой системы.

На протяжении всего процесса проектирования системный архитектор должен выступать в качестве агента, лица, принимающего решения, и защитника интересов будущих пользователей, а также всех прочих людей, для которых создается будущий продукт. Следует учитывать, что пользователем часто становится не тот, кто покупает готовый продукт. Это в наибольшей степени относится к закупкам военного назначения, при которых те, кто платят деньги (и даже те, кто задает характеристики проектируемого продукта), далеко отстоят непосредственно от самих пользователей. Мало того, одна и та же система военного назначения может иметь много разных категорий пользователей, которые соприкасаются с ней на уровне стратегического управления, решения задач воинского подразделения, а также на уровне отдельного бойца. Что же касается продуктов гражданского назначения, то при определении требований к проекту интересы покупателя представляют специалисты по маркетингу. Свои требования выдвигают и инженеры-

проектировщики. Учитываются также требования изготовителей будущего продукта. А от имени пользователей выступает только архитектор. Кроме того, идет ли речь о создании сложных систем или о строительстве обычных зданий, именно архитектор должен проявлять все свои профессиональные знания в части применяемых технологий в целях учета текущих и перспективных интересов пользователей. На основании сказанного можно вполне оценить, насколько ответственной является роль системного архитектора[16]. Я обсудил эту тему достаточно подробно в главах 4–7 своей книги [40].

Проектирование единообразного пользовательского интерфейса

Если создаваемая система достаточно велика, то к ее разработке приходится привлекать не одного главного архитектора, а целую группу архитекторов. В связи с этим снова возникает угроза нарушения концептуальной целостности. Дело в том, что в этом случае даже работу системных архитектуров приходится разделять среди многих людей, контролировать ее выполнение, а впоследствии соединять в единое целое. Таким образом, и в этой области приходится решать задачи обеспечения концептуальной целостности.

Если в системе должен быть предусмотрен пользовательский интерфейс, то он рассматривается как один из наиболее важных компонентов системы с точки зрения пользователя, поэтому его создание должно осуществляться под пристальным контролем одного человека. В некоторых коллективах проектировщиков проработку всех деталей пользовательского интерфейса может осуществлять главный архитектор. В качестве примера можно указать программы MacDraw и MacPaint, которые относятся к числу одних из первых инструментальных средств в операционной системе Mac и фактически были разработаны теми же людьми, которые их спроектировали. Если группа архитекторов достаточно велика, то круг обязанностей главного архитектора становится слишком большим для того, чтобы он мог сам заниматься созданием интерфейса. Тем не менее за решение этой задачи всегда должен отвечать один человек. Если архитектор в одиночку не может добиться создания целостного интерфейса, то в конечном итоге готовым интерфейсом не сможет овладеть и отдельно взятый пользователь. Например, в компании Google один из вице-президентов, Марисса Майер (Marissa Mayer), лично контролирует форматы страниц и следит за составом начальной страницы[17].

Проектировщик интерфейса должен не только обладать большим опытом эксплуатации интерфейсов и уметь прислушиваться ко всем пожеланиям, но и обязан прежде всего проявлять настоящий художественный вкус. Однажды я спросил Кеннета Айверсона (Kenneth Iverson), лауреата премии Тьюринга и создателя языка программирования APL: “Почему язык APL является настолько удобным?” Его ответ мог бы заменить целые тома рассуждений: “Он действует в полном соответствии с тем, что вы от него ожидаете”. Язык APL является воплощением принципа концептуальной целостности и иллюстрирует все составляющие этого

понятия — ортогональность, уместность и общность. Кроме того, в этом языке достигнута предельная лаконичность, поскольку для предоставления многих функций используются лишь немногие концепции.

Однажды я участвовал в обзоре архитектуры нового компьютерного семейства Future Series (FS), на которое разработчики корпорации IBM возлагали большие надежды как на преемника семейства S/360. Коллектив архитекторов состоял из выдающихся специалистов, опытных и изобретательных. Я с восторгом всматривался в раскрывающуюся передо мной грандиозную панораму. Так много в ней было продемонстрировано прекрасных идей! В течение часа один из архитекторов описывал мощные средства адресации и индексации. Еще один час другой архитектор отвел описанию порядка следования команд, организации циклов и средств ветвления. Затем другой специалист раскрыл подробности богатого набора операций, включая мощные новые операторы для структур данных. Еще один описал всесторонне развитую систему ввода-вывода.

Наконец, ошеломленный сложностью увиденного, я спросил: "Могу ли я поговорить с одним архитектором, который разбирается во всех этих вопросах, чтобы получить общее представление о будущей системе?"

"Такого у нас нет. Ни один человек не способен все это охватить своим разумом".

Тогда я понял, что проект обречен, — система рухнет под собственным весом. Получив справочник пользователя объемом в 800 страниц, я окончательно убедился в правильности своего предвидения будущего системы. Найдется ли в мире пользователь, способный овладеть таким интерфейсом программирования?[18]

Условия успешной совместной работы

В некоторых областях проектирования увеличение количества разработчиков непосредственно способствует повышению качества конечного продукта.

Определение потребностей и пожеланий со стороны заинтересованных лиц

Если исходить из того, что определение требований к проекту является наиболее сложной из всех задач проектирования, то в этой области увеличение количества привлеченных людей действительно позволяет добиться лучших результатов. Это невозможно отрицать! Коллектив, пусть даже небольшой, намного лучше по сравнению с отдельным человеком справляется с задачей выявления скрытых потребностей и с изучением существующей системы, подлежащей замене. Как правило, с привлечением различных людей удается рассмотреть гораздо больше разных вопросов, а сами эти вопросы становятся разнообразнее. С увеличением количества затрагиваемых тем удается лучше понять особенности разрабатываемого проекта. В коллективе должна царить атмосфера сотрудничества, гарантирующая, что каждый член коллектива получит полную возможность найти ответы на все интересующие его вопросы, касающиеся будущего продукта.

Постановка целей. При любой организации процесса проектирования разработчики начинают свою работу с общения с людьми, заинтересованными в успехе создаваемого проекта. При этом основные рассматриваемые темы касаются целей и ограничений, в рамках которых разрабатывается проект. В ходе этого взаимодействия одной из самых сложных задач является обнаружение таких подразумеваемых целей и ограничений, о существовании которых потенциальные пользователи часто просто забывают. Фактически по итогам проводимых собеседований и на основе анализа того, что и как было сказано и что осталось невысказанным, проектировщик впервые оценивает функцию полезности создаваемого проекта.

Одна из наиболее важных задач в этой фазе состоит в наблюдении за тем, как пользователь выполняет свою работу сегодня, в каких обстоятельствах и с помощью каких инструментальных средств. Часто бывает полезным проведение видеосъемки в ходе этих наблюдений и многократный просмотр полученных материалов обследования.

Успешному проведению этой фазы весьма способствует привлечение достаточно большого количества сотрудников. Чем больше людей,

- тем больше вопросов они задают,
- замечают различные нюансы, пусть даже о них ничего не было сказано,
- вырабатывают независимые и даже, возможно, противоречивые мнения по поводу того, что было ими услышано,
- наблюдают за разными аспектами организации труда в текущих условиях,
- продуктивно обсуждают видеоматериалы обследования.

Принципиально различающиеся подходы к концептуальным исследованиям

На ранних этапах процесса проектирования, причем чем раньше, тем лучше, разработчики приступают к поиску решений (ведь пока еще никому не удается поставить их перед фактом, что все решения по поводу будущего проекта уже приняты), учитывая то, что в конечном итоге в предлагаемых ими решениях должны быть учтены даже невысказанные пользователями пожелания или ограничения.

Мозговая атака. После проведения обследования наступает время для мозговой атаки. Вначале каждый представитель коллектива проектировщиков подготавливает в схематическом виде свои предложения. Затем все члены коллектива приступают к рассмотрению идей друг друга, не обращая внимания на то, что некоторые предложения могут даже казаться на первый взгляд нелепыми. На этой стадии необходимо руководствоваться такими стандартными правилами: “Постараться выработать как можно больше предложений”, “Не высказывать критические замечания”, “Поощрять появление любых, пусть даже внешне кажущихся нелепыми предложений”, “По возможности объединять и уточнять предложения” и “Делать графические наброски по тем предложениям, где это имеет

смысл"[19]. Чем больше людей, тем больше идей. Если люди побуждают друг друга высказывать свои предложения, то количество выдвинутых идей существенно увеличивается.

При этом не нужно стремиться к тому, чтобы все предложения сразу "попадали в цель". В статье [83] приведен отчет об управляемом эксперименте промышленных масштабов, проведенном в компании Sandia Labs:

> *Все члены групп мозгового штурма в среднем выдвигают одинаковое количество идей по поводу электронных схем, независимо от продолжительности сеанса мозгового штурма. Но если проводится оценка качества сформулированных предложений по трем измерениям (новизна, осуществимость и эффективность), то показатели отдельно взятых представителей группы намного превосходят показатели всей группы в целом (с вероятностью $p<0,05$).*

Конкуренция как альтернатива сотрудничеству. В качестве альтернативного подхода при проведении фазы концептуального исследования можно попытаться задать стимулы к проявлению творческих способностей отдельных проектировщиков путем проведения конкурсов на лучшие решения по проекту. Такие конкурсы становятся особенно удачными, если цели и ограничения известны, однозначно сформулированы и доведены до всех участников, а ненужные ограничения тщательно исключены.

В архитектуре такой способ определения наилучшего проекта был принят в течение многих столетий. Например, Брунеллески впервые прославился, выиграв конкурс на проектирование купола собора Санта-Мария дель Фьоре во Флоренции в 1419 году (рис. 6.4). Предложенная им концепция оказалась новаторской, ее осуществимость была доказана с помощью масштабной модели, а сам архитектурный замысел лег в основу новых конструкций, применявшихся и в более позднее время, например, при строительстве собора Святого Павла и американского Капитолия.

В архитектуре и в гражданском строительстве, особенно если речь идет о крупных проектах, заказчик часто выступает в единственном числе, а количество проектировщиков, желающих получить работу, весьма велико. Поэтому самым естественным решением становится проведение конкурса между исполнителями.

С другой стороны, в среде разработки продуктов, в которой занят разработчик аппаратного или программного обеспечения, ситуация принципиально иная. Обычно принято поручать выполнение задания по созданию конкретного продукта целому коллективу проектировщиков. Разумеется, в отношении отдельных проектных решений представители группы проектировщиков неизбежно выдвигают противоречащие друг другу предложения, поэтому нередко приходится становиться свидетелем горячих дебатов по этому поводу. Но руководители разработки крайне редко прибегают к тому, чтобы перед разными коллективами была поставлена одинаковая цель, которая должна быть достигнута в обстановке конкуренции.

Рис. 6.4. Купол Брунеллески, Санта-Мария дэль Фьюре. Неизвестный художник. Акварель "Вид на Флоренцию из садов Боболи". XIX век. Музео ди Фирензе комьера, Флоренция, Италия. Галерея "Скала/Арт Рисорс", г. Нью-Йорк

Тем не менее иногда крупные корпорации на определенных этапах разработки продуктов устраивают конкурсы в целях выбора наиболее подходящего варианта среди различных проектов. В ходе разработки архитектурного проекта System/360 мы работали над созданием архитектуры стека в течение шести месяцев. Затем наступил первый цикл оценки продуктивности затрат. Полученные результаты показали, что принятый нами подход является приемлемым для компьютеров среднего и высокого класса, но в разрабатываемом семействе компьютеров с семью моделями обнаруживает плохие показатели применительно к продуктам низкого класса.

Поэтому мы устроили конкурс проектов. Группа архитекторов по собственному усмотрению разбилась примерно на тринадцать небольших коллективов (от одного до трех участников), и каждый коллектив подготовил свои наброски по архитектуре, руководствуясь заданным набором правил и сроков. Я судил этот конкурс и пришел к выводу, что два из тринадцати проектов оказались лучшими. Оба эти проекта на удивление были похожими, и больше всего удивляло то, что предложившие их коллективы довольно холодно относились друг к другу и даже не общались.

Мы взяли лучшее из этих проектов и заложили образцы для дальнейшей работы. (Во всем процессе проектирования наиболее острые, глубокие и продолжительные дебаты вызвало основное различие между этими двумя приложениями, заключающееся в том, что в первом случае применялся 6-битовый байт, а во втором — 8-битовый.)

Я полагаю, что это соревнование между проектировщиками, идея проведения которого была первоначально предложена Джином Амдалом (Gene Amdahl), оказалось весьма стимулирующим и плодотворным. После того как стоимостная оценка предыдущего варианта проекта стала столь разочаровывающей, предложение померяться силами заставило проектировщиков с новыми силами взяться за работу. Каждому участнику разработки пришлось глубоко погрузиться во все аспекты проектирования, что позволило значительно укрепить моральный дух и обернулось ценными достижениями на более поздних этапах проектирования. Кроме того, удалось успешно согласовать многие другие проектные решения. А полученный в результате проект оказался очень качественным[20].

Спонтанно возникающие конкурсы между проектами — борьба за то, какой из продуктов останется на рынке. Нередко происходит так: коллектив проектировщиков, допустим, Б, настолько успешно разрабатывает свой проект, что перекрывает часть целей вывода на рынок своего продукта, которыми руководствуется коллектив проектировщиков А. В этом случае возникает соревнование между проектами особого рода — борьба за окончательный выход продукта на рынок.

Я сам был свидетелем многих соревнований между продуктами. Они происходят по стандартному сценарию в пяти действиях.

1. Представители двух коллективов, которые могут даже не знать подробности о работе друг друга, встречаются, сравнивают свои будущие продукты, определяют целевые рынки и приходят к выводу, что в действительности выпуск одного продукта не помешает успешному выводу на рынок другого. Обоим коллективам приходится напрягать все свои силы, чтобы вовремя решить поставленные задачи.
2. Между тем приходит все более четкое понимание того, как будет встречен разрабатываемый продукт, то ли под влиянием рыночных прогнозов, то ли в результате критики со стороны скептически настроенного руководства.
3. Каждый из коллективов вносит изменения в свой будущий продукт, чтобы воплотить в нем свойства всех других продуктов, завоевавших долю на рынке, а не просто недостающие свойства.
4. Каждый коллектив начинает вербовать сторонников среди заказчиков, специалистов по маркетингу и представителей компаний, занимающихся прогнозами в области перспектив реализации предлагаемых продуктов.
5. Цели коллективов начинают конфликтовать, и эти конфликты неизбежно переходят на уровень руководителей, принимающих решения.

С этого времени сценарии начинают развиваться по разным направлениям: побеждает коллектив А; побеждает коллектив Б; обоим коллективам предоставляется возможность продолжить работу; от проектирования отстраняются оба коллектива, поскольку у них не хватает сил на решение всех задач, выдвинутых в ходе конкурентной борьбы.

Описанный выше ход развития событий может и обычно должен быть предотвращен руководителями, трезво оценивающими реальную обстановку. Но иногда наилучшее решение может состоять в том, чтобы дать возможность пройти до конца оба пути, к которым ведут два весьма разных подхода к проектированию (и даже способствовать этому).

Обзор проекта

Той фазой проектирования, в которой совместная работа становится наиболее важной и даже необходимой, является обзор проекта. В обзоре должны участвовать специалисты по многим направлениям: другие проектировщики, пользователи и (или) представляющие их лица, конструкторы, покупатели, изготовители, специалисты по сопровождению, эксперты в области надежности и безопасности, а также экологи.

Специалист по каждому направлению должен выполнять обзор документации проекта отдельно от других, с учетом того, что тщательный обзор требует времени, размышлений, а также, возможно, изучения справочных материалов, архивов и других проектов[21]. Благодаря этому каждый участник обзора вырабатывает собственное, уникальное мнение; в каждом случае выдвигаются разные вопросы и обнаруживаются различные недостатки. Но результаты обзора должны быть обязательно сведены воедино с участием всех представителей группы.

Обязательный общий обзор, подготавливаемый междисциплинарной группой. Обзор проекта, подготовленный с участием многих специалистов, становится весьма авторитетным, но особый вес придает ему то, что проект рассматривается с точки зрения разных научных и технических направлений. Коллектив, участвующий в обзоре, должен быть намного больше по сравнению с коллективом проектировщиков. К рассмотрению проекта необходимо привлечь всех заинтересованных лиц: тех, кому будет поручен выпуск продукта по будущему проекту; тех, кто будет его сопровождать; представителей пользователей; лиц, которые займутся продажей продукта, и т.д. Рассмотрим в качестве примера обзор проекта новой подводной лодки. Определенный недостаток обнаруживает чиновник, отвечающий за поставки; высказанная им обеспокоенность наталкивает специалиста по борьбе с неисправностями на мысль о наличии еще некоторых недостатков. Эксперт по производственной оснастке замечает, что при строительстве подводной лодки придется столкнуться с некоторыми трудностями; предложенное им решение позволяет развеять обеспокоенность, которую до сих пор выражал эксперт по акустике.

Проектировщики из отдела электрооборудования подводной лодки компании General Dynamics рассказали мне об обзоре, в котором руководитель верфи обнаружил наличие конструкции с полуцилиндрическим резервуаром и сразу же предложил заменить его цельным цилиндром, уменьшить высоту резервуара наполовину и накрыть его плоской пластиной. Таким образом, вместо приблизительно двадцати деталей, первоначально предложенных конструктором, удалось обойтись гораздо меньшим количеством. После этого руководитель верфи сказал: "Мы, строители подводных лодок, очень хорошо умеем обращаться с цилиндрическими конструкциями".

Приведу еще один подобный пример. Проектировщик из компании Brown & Root, г. Ледерхед, Англия, рассказал мне об обзоре проекта для глубоководной платформы бурения нефтяных скважин. Руководитель службы эксплуатации указал на определенный блок и отметил: "Его лучше сделать из качественной конструкционной стали".

"Почему?"

"Вы сможете окрасить его для защиты от коррозии на заводе перед монтажом, но после того как этот блок встанет на место, его больше никогда не удастся окрасить повторно".

Конструкторы перепроектировали всю периферию платформы, чтобы можно было и в дальнейшем получить доступ к этому важному блоку.

Использование графических представлений. При решении сложной задачи обзора проекта необходимо использовать вспомогательные средства, наиболее важным из которых является общая модель продукта — чертеж, полномасштабный макет из другого материала в натуральную величину или, как в случае подводной лодки, компьютерная модель, или же прототип какого-то механического узла, возможно, в виде архитектурной диаграммы на компьютере.

Для проведения междисциплинарного обзора проекта часто требуется применение более разнообразных графических представлений проекта по сравнению с тем, которые используют сами проектировщики. Дело в том, что далеко не каждый участник обзора сможет представить себе внешний вид конечного продукта, ознакомившись лишь с техническими или архитектурными чертежами. Мне довелось посетить многие предприятия, и я пришел к выводу, что компьютерные технологии виртуальной визуализации находят наиболее плодотворное применение именно при проведении подобных обзоров проектов[22].

Кроме того, для успешного проведения обзора проекта очень важно иметь модель продукта, с которой может ознакомиться каждый, а также предоставить возможность всем участникам обзора беспрепятственно обмениваться своими мнениями; наличие инструментальных средств, обеспечивающих совместную работу, становится непременным условием, если в обзоре проекта участвует группа, все представители которой не могут физически присутствовать в одном помещении. Здесь вступают в свои права средства дистанционной организации совместной работы.

Отсутствие возможности обеспечения совместной работы непосредственно в самом проектировании

Не оправданная практикой концепция сотрудничества при проектировании. В литературе, посвященной организации совместной работы на основе средств компьютеризации, часто можно встретить вымышленную версию сценария разработки проекта в обстановке сотрудничества. На это можно было бы закрыть глаза, но предлагаемая ошибочная концепция ложится в основу еще более углубленных академических исследований, посвященных созданию все менее и менее полезных технологических инструментальных средств обеспечения совместной работы.

В этом вымышленном сценарии коллектив проектировщиков работает над виртуальной или реальной моделью проектируемого объекта, будь то здание, механический узел, подводная лодка, блок-схема программы или совместно подготавливаемый текст. Члены коллектива один за другим предлагают свои изменения, при этом обычно изменения вносятся непосредственно в саму модель. Другие предлагают поправки к этим изменениям, происходит обсуждение, и шаг за шагом проект принимает окончательный вид.

Организация совместной работы над проектом, происходящая в действительности. Но это воображаемое представление в большей степени описывает то, как происходит обзор проекта, а не отражает фактический ход совместной разработки проекта.

Во всех многочисленных коллективах проектировщиков, за работой которых я наблюдал, в любое время за каждое направление проектирования отвечал лишь один человек. Такие люди работают в основном самостоятельно, подготавливая для других предложения о проектировании по их направлениям. Затем руководители направлений проводят совещания со своими сотрудниками, фактически осуществляя в сокращенной форме сеанс обзора проектирования. После этого руководители направлений обычно снова возвращаются к самостоятельной работе и подробно анализируют, каковыми должны быть выводы из решений и указаний, которые обсуждались совместно с другими.

Если участники совещания выдвигают альтернативные предложения, не принятые руководителем направления, то инициаторы таких предложений часто отказываются от них и разрабатывают другие варианты. Наконец, снова проводится совещание в целях выбора приемлемых предложений, принятия на их основе общих решений или даже смены путей проектирования.

Решение задач управления проектированием. Описанная выше вымышленная концепция организации проектирования, кроме всего прочего, предусматривает возможность лишь уточнения проектов, но не выдвижения каких-либо новаторских идей. Мало того, в этой вымышленной концепции нет места для модели внесения изменений в совместно разрабатываемый проект. Организация работы в составе коллектива приводит к сокращению графика за счет распределения

отдельных операций по исполнителям, но из того, что часть операций выполняется параллельно, вытекает необходимость обеспечения их синхронизации; такая задача полностью исключается, если проектирование выполняется силами одного человека. Допустим, при проектировании океанского танкера проектировщик А отвечает за прокладку воздуховодов, а проектировщик Б несет ответственность за прокладку паропроводов. Еще до того как каждый из них воплотит в деталях свой проект, должен быть применен некоторый механизм управления проектом, гарантирующий то, что проектируемым воздуховодам и паропроводам будут отведены разные области общего пространства и что такие требования по их прокладке будут соблюдаться и в дальнейшем. На случай возникновения конфликтов должна быть предусмотрена определенная процедура улаживания потенциальных разногласий. Необходимо ввести в действие некоторые средства управления версиями, которые позволили бы руководителям направлений проектирования проводить согласования версий проделанных ранее работ по проектированию с учетом одинаковых временны*х* меток.

Я был свидетелем одной попытки реализации вымышленной концепции организации проектирования, когда представитель заказчика рассматривал модель проекта атомной подводной лодки и предложил сместить одну переборку, чтобы создать большие удобства в работе для ремонтников оборудования. (Даже сама задача осуществления этого предложения в интерфейсе виртуальной реальности, применяемом к системе автоматизированного проектирования, является технически сложной. Дело в том, что создаваемые в этих системах модели действительной обстановки по своему характеру являются статическими, и это учтено в применяемых методах визуализации в реальном времени.)

Но в рассматриваемом случае не стоит даже заниматься решением задачи, поставленной представителем заказчика! Допустим, представитель заказчика хочет сместить переборку, чтобы узнать, как после этого будет выглядеть пространство отсека, и такая возможность предоставляется в имитационной версии модели. Но само подобное предложение о смещении переборки не может войти в общую версию проекта до тех пор, пока с привлечением определенного лица или с помощью некоторой программы не будут проконтролированы результаты смещения, обнаруживаемые в отсеке, находящемся по другую сторону от переборки, конструктивные последствия, изменения акустических свойств, изменения в прокладке труб и кабелей. Достаточно представить себе, какой ужас охватил бы ответственных конструкторов, узнавших о том, что представитель заказчика отдал распоряжении о смещении переборки, не имея не малейшего представления о том, какие проектные согласования были проведены ранее при выборе ее положения и какие ограничения были учтены. Если бы была введена в действие формальная система управления изменениями, то представитель заказчика не смог бы опробовать свое предложение даже на виртуальной модели.

Описанный выше воображаемый принцип организации совместного проектирования полностью исключает возможность достижения концептуальной

целостности. Проектировщик А шлифует проект в одной части; проектировщик Б вносит изменения в другую часть; проектировщик В проводит правки еще в одной части. Все эти проектировщики работают по своему усмотрению; проект, безусловно, создается совместно; но этот проект не может быть качественным. В действительности такая организация работ по проектированию настолько хорошо изучена, что для нее уже придумано пренебрежительное обозначение — *проектирование комитетом*. Если инструментальные средства организации совместного проектирования задуманы так, что поощряют проектирование комитетом, то их применение приносит больше вреда, чем пользы.

Особая важность соблюдения принципа индивидуальной ответственности при концептуальном проектировании

После завершения стадии исследований и выбора основной темы наступает время, когда основное внимание должно быть сосредоточено на достижении концептуальной целостности. Главный проектировщик передает ответственность за разработку частей проекта коллективу проектировщиков, оставляя за собой решение основных задач[23].

Тем не менее указанный принцип достижения концептуального проектирования может завести в тупик. В таком случае должна быть выбрана другая базовая схема и продолжено совместное исследование в целях выбора более продуктивной базовой схемы.

Отличительные особенности коллективов из двух человек

В приведенном выше обсуждении организации совместной работы в области проектирования речь шла о коллективах, состоящих более чем из двух человек. А в данном разделе будут рассматриваться коллективы, в состав которых входят два человека, обладающие заметными отличительными особенностями. Даже на концептуальной стадии проектирования, т.е. в условиях, когда существует наибольшая опасность нарушения концептуальной целостности, проектировщики, работающие вдвоем в полном согласии, способны создать не менее целостный проект, чем проектировщики, работающие индивидуально. В литературе, посвященной организации работы по программированию, тот же принцип подтверждается и в отношении стадии детализированного проектирования. Исходные показатели производительности двух совместно работающих проектировщиков, как правило, ниже, чем у двух проектировщиков, работающих отдельно, но относительное количество ошибок значительно меньше[24]. Считается, что примерно 40% трудозатрат во многих проектах связано с исправлениями, поэтому суммарная производительность проектировщиков, работающих в паре, становится выше, а сами продукты характеризуются большей надежностью.

И действительно, многие работы гораздо проще выполнять вдвоем. Плотнику нужен помощник, который держит вторую ручку пилы. Электрик не может обойтись без помощи, протягивая провода через кабельные каналы. Дети развиваются лучше, если за ними ухаживают двое помогающих друг другу родителей. Недаром в отношении семьи сказано: "Не хорошо быть человеку одному"[25], но нечто подобное часто можно высказать и в отношении единолично действующих проектировщиков.

Что же касается проектирования, то, по-видимому, при совместном участии в проекте двух человек работа происходит иначе по сравнению с тем, когда в ней участвуют то ли многочисленные коллективы, то ли индивидуальные проектировщики. Два человека обменяются идеями быстро и неформально, для чего не требуется брать слово для выступления или выяснять, кто из партнеров главный. Любой из этих двух сотрудников может высказывать свои замечания в любой момент. В процессе проектирования фактически происходят мгновенно сменяющиеся сеансы выдвижения предложений, обзора и критического анализа, высказывания встречных предложений, синтеза и принятия решения. Разработка идеи, как правило, происходит в виде единого потока обсуждения, в ходе которого не приходится задумываться над тем, что у кого-то из участников дискуссии могут быть иные мысли по сравнению с другими, как обычно происходит в дискуссиях с участием нескольких человек. Следы от двух вместе взятых карандашей можно провести по одному и тому же листу бумаги, не пересекая их и не сталкивая сами карандаши.

К тому же, по аналогии с тем, как одно лезвие ножниц точит другое, каждый участник обсуждения вдвоем побуждает другого более активно высказывать свои мысли, а человек, работающий отдельно от других, вынужден контролировать сам себя. По-видимому, даже сама необходимость высказать другому свои мысли, чтобы продвинуться в решении общей задачи, способствует более быстрому осознанию собственных ошибок и лучшей оценке более успешных предложений, выдвинутых партнером.

В классической статье Торренса [205], опубликованной в 1970 году, показано, что обсуждение проблемы вдвоем приводит к увеличению в два раза количества сформулированных оригинальных идей, идеи отличаются вдвое большей новизной, удовлетворенность работой возрастает, а сами участники работы стремятся взять на себя все более сложные задачи[26].

Совещания с участием работающих вдвоем проектировщиков все же приходится чередовать с индивидуальной работой для уточнения деталей, документирования творческих находок и подготовки предложений к следующему сеансу совместной работы.

Особенности организации труда программистов

Специалисты в области программирования вложили значительные усилия в создание инструментальных средств организации совместной работы на основе

компьютеризации для собственной дисциплины и других дисциплин. Но вызывает разочарование то, что лишь немногие из выдвинутых ими идей и созданных инструментальных средств вошли в категорию применяемых повседневно. (В качестве наиболее важных инструментальных средств, которые оказались успешными, можно назвать системы управления разработкой кода и средства отслеживания изменений в программе Word.) Причина такого положения дел, по-видимому, состоит в том, что многие специалисты в области программного обеспечения, взявшие на себя задачу создания инструментальных средств проектирования, не смогли учесть некоторые важные особенности коллективного проектирования, осуществляемого в действительности.

- Реальные проекты всегда намного сложнее, чем можно было бы себе представить[27]. Это особенно справедливо, поскольку самое обучение в области проектирования начинается с освоения примеров из учебников, которые в силу своего назначения чрезмерно упрощены. Реальное проектирование осуществляется с учетом необходимости достижения более сложных целей, в ходе него приходится учитывать более сложные ограничения, а конечный продукт оценивается по более сложным критериям качества. Фактически по мере развития процесса проектирования приходится соприкасаться со все более и более мелкими деталями, количество которых порой кажется бесконечным.
- В реальном коллективном проектировании всегда должен применяться процесс управления изменениями в проекте, чтобы левая рука не разрушала то, что сделано правой.
- Насколько углубленным не было бы сотрудничество, оно не может быть настолько глубоким, чтобы полностью устранить потребность в общении для бегства от "безысходности работы и одиночества размышлений".

Среди тем для диссертаций, подготавливаемых аспирантами, часто упоминается разработка инструментальных средств организации совместной работы в области проектирования, но фактически это не оправдано, поскольку такая задача может быть успешно решена лишь теми людьми, которые сами имеют достаточный практический опыт в области проектирования. Кроме того, для публикации в компьютерных журналах не следует неограниченно принимать такие статьи, которые не основаны на реальном опыте и (или) не показывают процесс проектирования реальных приложений.

Примечания и ссылки

1. Эту замечательную фразу процитировал Бернард Барух (Bernard Baruch), которую, по его словам, он услышал от своего адвоката.
2. См. [89].

3. Опытный руководитель одной из проектных организаций заблаговременно поручает отдельному проектировщику или проектировщикам, работающим вдвоем, приступить к исследованию в области некоторой перспективной технологии, реализация которой пока не представляется возможной.
4. См. [40], глава 2.
5. Бригадир верфи, принадлежащей компании Electric Boat, г. Гротон, шт. Коннектикут (по материалам личного общения).
6. Наиболее полным обнаруженным мною научным исследованием, в котором сравниваются способы организации труда с привлечением работающих в паре и отдельных проектировщиков, является [73].

   В Технологическом университете г. Делфта разработаны протоколы регистрации процессов решения одинаковых задач отдельным проектировщиком и коллективом, состоящим из трех человек. В ходе экспериментов производилась видеозапись, в коллективе обсуждения проводились устно, а отдельно действующий проектировщик должен был излагать свои мысли вслух. Видеопротоколы, подготовленные в университете Делфта, рассматриваются в двадцати главах книги, в каждой из которых используется собственный аналитический метод. В большинстве методов протоколы одного или обоих типов рассматриваются с точки зрения предварительно определенных авторами категорий активности. Во многих главах либо сравниваются действия и производительность двух указанных способов организации труда, либо анализируется социальное поведение в коллективе. Наиболее конкретные выводы сделаны в работе [105]: "Подробный анализ позволяет прийти к заключению, что способы доведения своей работы до завершения, применяемые отдельным проектировщиком и коллективом, почти не различаются".

   В работе [87] приведен обзор этой книги в разделе "Исследования по проектированию" (с. 475-476):

   В целом эти исследования открывают широкие перспективы и позволяют читателю понять предпосылки успеха и особенности процессов проектирования. Очевидно также, что полученные видеозаписи раскрывают богатый спектр характеристик поведения людей в процессе проектирования... Тем не менее стали полностью видимыми ограничения современных методов анализа протоколов. Каждое исследование, отдельно взятое, позволяет изучить только малую часть общего процесса проектирования. Понимание всего процесса достигается лишь в результате совокупного рассмотрения многочисленных исследований.

   В этой книге полностью представлено современное состояние исследований процессов проектирования с применением протоколов, достигнутое после тридцати лет работы в этом направлении, а также в более общей форме показано, как эти исследования связаны с различными теориями проектирования.

7. См. [40], глава 4, с. 42 и далее.
8. См. [25], раздел 1.4; [40], главы 4–7, 19.
9. См. [21], глава 6; [152].
10. См. [25], глава 14.
11. См. [158].
12. См. [108]; [27].
13. См. [218]; [196].
14. В свое время Р. Джозеф Митчелл (R. Joseph Mitchell), разработчик самолета Spitfire, предупредил одного из пилотов, участвующих в испытаниях (с нашей точки зрения — пользователя!), о том, как следует общаться с инженерами: "Если кто-либо из них скажет вам о самолете, будто он настолько сложен, что вы никогда не сможете разобраться в его конструкции, вспомните мои слова — все это чушь".
15. Эон Вудс (Eoin Woods) из компании Artechra сказал:

    Я не отношусь к совместному проектированию столь пессимистично, как вы. Я работал в коллективах, где мы энергично обсуждали ход разработки проектов, а затем совместно согласовывали решение (хотя иногда окончательное решение принимал "благосклонный диктатор"). Проект оставался целостным, поскольку в нем всегда доминировали одна или две мощные концепции проектирования, на основе которых происходило принятие всех прочих решений; проектирование и согласование детализированных решений не происходило по принципу работы комитета (по материалам личного общения [2009]).

16. Брэд Паркинсон (Brad Parkinson), который теперь работает в Стэнфордском университете, а в свое время был одним из двух разработчиков системы GPS, ответственным за создание системной архитектуры и заключение контрактов, указал, что стоящие перед ним задачи существенно усложнялись из-за наличия многочисленных подрядчиков, отвечающих за подготовку нескольких частей системы (по материалам личного общения [2007]).
17. См. [120].
18. Мэри Шоу (Mary Shaw) из университета Карнеги-Меллона задала вопрос: "Какие выводы на этом основании можно сделать о современных средах программирования и применяемых в них API-интерфейсах?"
19. См. [161].
20. Конкурсы между проектами организационной структуры — совсем другие; в этих проектах прежде всего приходится решать политические задачи. В борьбе участвуют различные противоборствующие силы, которые обычно даже не стремятся к общей цели создания организации, наилучшим образом выполняющей свои функции. Задача обеспечения оптимальной

работы организации рассматривается, как второстепенная по отношению к задаче наделения полномочиями тех или иных лиц.

21. Маргарет Тэтчер: "Нужны такие документы, которые позволяли бы все продумать заранее и проконсультироваться с коллегами (а не просто слайды с графиками)" (по материалам личного общения с сэром Джоном Фэрклоу). На деловых предприятиях в США слишком широко применяются обзоры в виде презентаций PowerPoint. Эти презентации часто имеют вид непонятных картинок с маркированными списками, которые позволяют каждому участнику просмотра интерпретировать полученные сведения так, как ему угодно; кроме того, с их помощью можно легко исключить не укладывающиеся в благостную картину, но важные детали.

Лу Джерстнер (Lou Gerstner), работающий по контракту исполнительный руководитель в компании IBM, первым бросил вызов всей этой корпоративной культуре, опубликовав статью [99]: "Ник открыл свой второй слайд к тому времени, как я подошел к столу и, проявляя максимальную вежливость к его коллективу, выключил проектор... Это вызвало чрезвычайно мощный волновой эффект... Проявились все эмоции вплоть до испуга. Это можно было бы сравнить с тем, как если бы президент Соединенных Штатов запретил использование английского языка на совещаниях в Белом доме."

22. См. [45].

23. Харлан Миллс (Harlan Mills) выдвинул концепцию поддержки коллектива проектировщиков во главе с руководителем, "хирургической бригады", которая была уточнена в книге [40], глава 3.

24. См. [224]; [61].

25. Бытие 2:18.

26. См. [205].

27. См., например, внушительные диссертации на звание доктора философии, [109] (Кэмбридж) и [182] (Гарвард), в которых подробно описано все, что касается реального проектирования.

Andrei State 2005

Andrei State 2005

7

Дистанционное сотрудничество

Новые взаимозависимости, обусловленные применением электроники, воссоздают мир в образе глобальной деревни.

MARSHALL MCLUHAN [1967],
"THE MEDIUM IS THE MESSAGE"

◂ *Представление Генри Фукса (Henry Fuchs) об офисе будущего. Рисунок Андрея Стейта (Andrei State), Университет шт. Северная Каролина*

Причины применения дистанционного сотрудничества

Начиная с этой главы, мы можем приступить к изучению принципов дистанционного сотрудничества. Основной вопрос состоит в том, почему в наши дни коллективы проектировщиков используют технологии связи для обеспечения взаимодействия людей, находящихся далеко друг от друга.

Специализация

В настоящее время одним из важных стимулов к развитию сотрудничества является крайне узкая специализация в научных и технических областях. Но специалиста в любой конкретной области вряд ли найдешь в каждом небольшом населенном пункте и даже в каждом городе. Не стало деревенских кузнецов, способных выполнять все необходимые работы по металлу, но появились ученые эксперты по материалам ограниченного применения, допустим, по сплавам титана; место городского пожарника занял сотрудник МЧС, который может быть направлен в любой уголок страны, например, для тушения пожара на нефтяной скважине стоимостью в миллионы долларов[1].

Работа на дому

Люди, как правило, привыкают к месту своего жительства и предпочитают его не менять. Для многих проживание неразрывно связано с их окружением в виде семьи, клана, культуры. Другие выбирают место жительства по тому принципу, находится ли оно в сельской или городской местности. Есть и такие люди, которые руководствуются предпочтениями, в основе которых лежит выбор климата или ландшафта: морского побережья или гористых территорий. Причем, чем более высокой квалификацией обладает человек, тем больше он имеет шансов выбрать то место проживания, которое подходит ему лучше всего. Применение технологий дистанционного сотрудничества способствует увеличению количества экспертов, проживающих там, где им нравится больше всего, но работающих в другом месте. Из числа моих собственных бывших студентов один поселился в Исландии, а другой в Бразилии, но оба работают в Силиконовой Долине.

Круглосуточная работа

Если члены коллектива проживают в разных часовых поясах, то могут обеспечить круглосуточную работу, даже если каждый из них выходит только в дневную смену.

Денежные затраты

Разные страны значительно отличаются друг от друга по прожиточному минимуму и уровню жизни, поэтому часто бывает возможным привлечение высококлассного специалиста на основе аутсорсинга за счет гораздо более низких затрат. При этом, разумеется, расширение количества лиц, привлеченных по принципу телеработы (с применением средств передачи данных), приводит к стремительному выравниванию уровня жизни между странами с различной степенью развития экономики, что можно рассматривать как внешнюю экономическую помощь, причем наиболее продуктивную.

Политика

Масштабы современных проектов, осуществляемых крупными международными предприятиями с правительственной поддержкой, таковы, что требуют разделения труда между несколькими странами, что связано с созданием территориально разрозненных подразделений. Рассмотрим в качестве примера проект создания самолета Airbus 380 как яркого проявления современных технических достижений (рис. 7.1). Не только производство, но и разработка этого самолета были разделены между Францией, Германией, Великобританией и Испанией.

Джеффри Джапп (Jeffrey Jupp), который в свое время занимал должность технического директора подразделения Airbus UK в Англии, рассказывал мне, как проектировались крылья самолета Airbus в Бристоле, с учетом того, что они должны состыковываться с фюзеляжем, разрабатываемым в Тулузе, и успешно поднимать его в воздух.

- Использовались все существующие возможности передачи данных.
- Несколько инженеров из Бристоля были откомандированы в Тулузу в качестве представителей британского подразделения.
- Каждый день самолет компании совершал полет из Бристоля в Тулузу и обратно для обеспечения оперативного обмена материалами между подразделениями.

Мой опыт показывает, что даже при дистанционном сотрудничестве обязательно должны применяться аналогичные виды обеспечения взаимодействия. К сожалению, при разработке самолета Airbus 380 не удалось также избежать связанных с дистанционным сотрудничеством потенциальных опасностей, которые еще больше усугублялись тем, что в проектировании участвовали представители разных стран. Во французском и британском коллективах проектировщиков использовался выпуск 5 программного обеспечения CATIA CAD, а в германском и испанском — выпуск 4. И вот, отчасти из-за различий между этими выпусками, жгуты кабелей, спроектированные одним коллективом, имели больший диаметр,

чем допускали кабелепроводы, разработанные другим коллективом. По этой и другим причинам произошли задержки начальных поставок, составляющие приблизительно 22 месяца, которые оказались очень неприятными[2].

Рис. 7.1. Airbus 380, iStockphoto

Распределение работ по территориально разрозненным подразделениям. Распределенная разработка семейства компьютеров IBM System/360, 1961–1965 гг.

Первые семь компьютеров семейства IBM System/360 одновременно разрабатывались в четырех местах, расположенных в трех странах: Пукипси и Эндикотт, шт. Нью-Йорк, США; Херсли, Великобритания; Беблинген, Германия. В этом семействе компьютеров впервые был реализован принцип совместимости снизу вверх и сверху вниз на уровне двоичного кода, а также впервые в компьютерной индустрии произошел переход от 6-битовых к 8-битовым байтам. Я был руководителем этого проекта. В главе 24 приведен практический пример проектирования архитектуры для компьютеров System/360. (Модель 20 не была совместимой сверху вниз, что было вызвано ошибочным решением системного архитектора Уильяма Райта (William Wright) и моим недосмотром.)

Требовалось одновременно разработать больше сорока новых 8-битовых устройств ввода-вывода, для чего приходилось привлекать узких специалистов, имеющих опыт работы в соответствующей области, задействуя в этом все большее количество подразделений различных компаний: Ла Год, Франция; Лидингё, Швеция; Уитхоорн, Нидерланды; Сан-Хосе, шт. Калифорния; Боулдер, шт. Колорадо;

Лексингтон, шт. Кентукки; и Эндикотт, шт. Нью-Йорк. Координации этих усилий в значительной степени способствовало применение принципиально нового подхода: скрупулезное определение стандартного логического, электрического и механического интерфейса, позволяющего подключить любое устройство ввода-вывода к любому компьютеру[3]. Но даже несмотря на это, управление распределенной разработкой всех указанных устройств представляло собой нелегкую задачу. Что же касается разработки программного обеспечения, то распределение работ приобрело еще бо́льшую широту.

Управление проектированием компьютеров, программного обеспечения и устройств ввода-вывода осуществлялось на основе тех же методов, которые применялись в компании BAE Systems, о чем речь пойдет ниже в данной книге. Применяемые нами средства передачи данных были намного более примитивны: я заключил договор на круглосуточную аренду первой трансатлантической телефонной линии IBM. Мы отказались от того, чтобы прямые и обратные рейсы совершал самолет нашей компании, но покупали много билетов на рейсы авиалиний. В группе по разработке архитектуры под руководством Амдала в г. Пукипси постоянно находился представитель британского подразделения, а наши представители из Пукипси работали в группах по реализации проектов процессоров в Великобритании и Германии.

Связь между подразделениями из разных стран поддерживалась не только благодаря тому, что совершались тысячи телефонных звонков и передавались многочисленные документы, но и с помощью многих личных встреч. Неразрешенные конфликты и сложности улаживались на ежегодных двухнедельных встречах с участием всех групп, причем каждый раз удавалось устранить до двухсот рассогласований.

Во всех разработках, осуществляемых по принципу дистанционного сотрудничества, обычно применялись одни и те же интеллектуальные ресурсы.

- Специалисты по различным технологиям, работающие в своих подразделениях.
- Сотрудники, работающие без выезда на другие площадки.
- Руководители, способные вести политику по организации функционирования многочисленных подразделений и распределению работы.

Этот подход к осуществлению разработки оказался очень успешным[4]. Однако не следует впадать в заблуждение, считая распределенную разработку общего продукта легким делом! Дело в том, что само распределение обязанностей требует выполнения большой дополнительной работы! Оказалось, что неформальные каналы связи в коллективе, все члены которого находятся в одном месте, имеют огромную значимость, которую мы недооценивали, пока не испытали на себе отсутствие таких связей. Пространственные барьеры оказались реальными![5] Не менее ощутимыми были временны́е барьеры, обусловленные различием часовых

поясов, причем зачастую они были важнее пространственных барьеров! Кроме того, весьма значимы языковые барьеры, которые также должны быть приняты во внимание![6]

Ввод принципов дистанционного сотрудничества в действие

В дальнейшем роль проектирования в режиме дистанционного сотрудничества будет только увеличиваться. Продолжается стремительное развитие технологий передачи данных. В связи с этим перед проектировщиками и руководителями групп проектирования встает задача оптимальной организации работы в режиме дистанционного сотрудничества.

Крайняя важность личных встреч!

Чтобы понять, насколько важны личные встречи, подумайте о том, как вы сами разговаривайте по телефону. Не правда ли, что ощущается меньший комфорт и даже обнаруживается недостаточная продуктивность общения, если разговор ведется с незнакомцем, а не с хорошо знакомым человеком.

Обычно люди готовы даже совершить небольшое путешествие, чтобы избавиться от необходимости использовать такие средства связи, как оборудование для проведения видеоконференций, телефон, электронную и обычную почту, для решения следующих задач.

- Назначить встречу за завтраком.
- Найти скидку на приобретаемую услугу.
- Заключить сложную коммерческую сделку.
- Организовать семейный отдых.
- Уволить своего помощника с административной должности.

Безусловно, при решении некоторых подобных задач предпочтительным является использование электронной почты или телефона, а не личная встреча (для чего требуется согласовывать наиболее подходящее время), тогда как для достижения других целей многие не откажутся преодолеть некоторое расстояние.

Известные мне примеры дистанционного сотрудничества, связанные с достижением наибольшего успеха, базировались на долгой истории личного знакомства, но даже и в этих случаях требовалось проведение дополнительных личных встреч в ходе продолжающегося дистанционного сотрудничества. Если же участники совместной работы, осуществляемой в разрозненных подразделениях, мало знакомы друг с другом, то следует хоть иногда преодолевать разделяющие их пространственные барьеры, поскольку это оправдано с точки зрения затрат времени и денег.

Когда я работал в компании IBM в должности руководителя, то наиболее продуктивно потратил те деньги этой компании, которые пришлось заплатить за автобус для перевозки административного персонала и секретарей, работающих по проекту S/360, на 60 миль от Пукипси до Уайт Плейнс, шт. Нью-Йорка. Эти люди провели в командировке полдня, пообедав и побеседовав со своими коллегами в штаб-квартирах подразделений, после чего до тех пор безликие голоса постоянных собеседников обрели плоть и кровь. Этот "шаг навстречу людям" оказался гораздо более эффективным, чем пустые призывы к укреплению сотрудничества.

Мне довелось слышать о том, что на начальном этапе проектирования самолета 777 компания Boeing отправляла для совместной работы представителей групп распределенного проектирования в Эверетт, шт. Вашингтон, сразу на несколько недель.

Иногда не отдавая себе в этом отчета, высоко оценивают значение личных встреч и сами люди. Именно поэтому, несмотря на наличие столь мощной технологии проведения видеоконференций, самолеты по-прежнему заполнены людьми, отправляющимися в командировку для решения деловых вопросов.

Качественные интерфейсы

Одной из самых сложных задач является определение качественных интерфейсов между компонентами, проектируемыми отдельно расположенными подразделениями. При этом работа не заканчивается после формального определения интерфейсов, поскольку оказывается необходимой непрерывная интерпретация семантики определений в режиме вопросов и ответов. По ходу дела приходится вносить изменения, управлять ими и доводить сведения об этом до всех заинтересованных лиц.

Еще одной важной частью создания архитектуры системы становится не просто определение интерфейсов, но и организация применения руководителями заранее установленного механизма, предназначенного для устранения разногласий во мнениях или вкусах. Безусловно, какой бы ни была организация работы, решения всегда должен принимать уполномоченный на это руководитель.

Труд руководителей оплачивается дорого, но если они по праву занимают свое место, то получаемая от них отдача чрезвычайно велика! От качества интерфейсов во многом зависит то, насколько велико будет количество ошибок, допущенных при проектировании. В литературе встречаются такие оценки, что стоимость исправления ошибок и переделки, пусть даже это касается лишь небольшой части проекта, может приближаться к половине затрат на проектирование. Но хуже всего то, что ошибки, обусловленные применением неточно определенных или произвольно трактуемых интерфейсов, обычно обнаруживаются слишком поздно, во время интеграции на уровне системы. Наиболее затруднительные в поиске, наиболее дорогостоящие в исправлении, эти ошибки способны сорвать график разработки всей системы.

Кроме того, качественные интерфейсы позволяют добиться подлинного удовлетворения от работы. Проектирование является увлекательным занятием, а улаживание разногласий с коллегами — обычно нет. Те, кто занимаются проектированием, чувствуют, как продвигается работа, а те, кому приходится улаживать разногласия в части трактовки интерфейса, жалеют о том, что время проходит даром. Если интерфейсы имеют высокое качество, то каждый из проектировщиков, участвующий в работе, ощущает радость от успеха порученного ему дела и получает привилегию утвердить своей подписью завершение определенного этапа работы. Кроме того, применение качественных интерфейсов способствует упрощению процедуры передачи ответственности за выполняемую работу по мере того, как малые компоненты объединяются друг с другом в ходе создания все более значимых и крупных подсистем.

Технологии дистанционного сотрудничества

Проходит одно десятилетие за другим, но прогнозы ученых мужей, специализирующихся в области технологий, согласно которым по мере развития средств передачи данных отпадет необходимость в пространственных перемещениях проектировщиков, все не сбываются. До сих пор нет признаков того, что это когда-либо случится[7]. Почему так происходит? И оправдаются ли эти прогнозы? Лично я разделяю такое мнение, что в результате создания все более удобных и приближенных к жизни технологий связи действительно удастся постепенно снизить необходимость в проведении личных встреч[8]. Но человеческое общение характеризуется наличием бесконечного количества нюансов, поэтому необходимость пребывания время от времени в одном помещении сотрудников, занимающихся совместным проектированием, никогда не исчезнет.

Возможность успешного применения всего спектра средств организации связи

Документ. Наиболее продуктивным средством дистанционного сотрудничества является документ, совместно используемый на основе технологий передачи данных по сети или электронной почтой. Документ содержит текст и рисунки, передающие буквальный смысл, поэтому его применение может стать основой проведения исследования, обеспечивает критический анализ и стимулирует взаимодействие.

Мы с Джерри Блау (Gerry Blaauw), создавая нашу совместную работу объемом в 1200 страниц ([25]), находились по разные стороны Атлантического океана и пришли к выводу, что наиболее эффективное сотрудничество осуществляется путем отправки друг другу черновиков по почте. Тем не менее это эффективное дистанционное сотрудничество строилось на базе девятилетней совместной работы лицом к лицу. В результате мы достигли глубокого понимания стиля проектирования друг друга, научились обходить потенциальные конфликты и

нашли оптимальные способы сотрудничества. Кроме того, наши представления о том, какой должна быть архитектура вычислительной системы, стали во многом одинаковыми. Но даже при наличии такого фундамента нам приходилось дополнять свой многократный обмен черновиками ежеквартальными продолжительными переговорами по телефону и трехдневными личными встречами один раз в полгода.

Особенно результативными всегда были встречи с глазу на глаз. Такие совещания по самой своей сути предназначались для согласования сложных вопросов, решение которых не удавалось найти другим способом. Мы пришли к выводу, что некоторые безуспешные попытки согласования текста были вызваны отсутствием общего представления о рассматриваемом содержимом. Обнаружив такую ситуацию, мы обычно проводили ее обсуждение в течение примерно получаса. В ходе этого нам приходилось раскрывать все более и более подробные сведения об архитектуре вычислительной системы.

Современным эквивалентом черновика, размеченного красным карандашом, является документ Word, корректируемый в режиме отслеживания изменений. В работе над таким документом может участвовать много редакторов, при этом каждый может вносить изменения так, что их можно легко отличить от исправлений, сделанных всеми другими редакторами. Средства отслеживания изменений программы Word хорошо продуманы. Но я все же считаю, что задача создания и изучения твердой копии документа, предназначенной для редактирования, гораздо проще, в значительной степени из-за того, что людям удобнее работать с листом бумаги, а не с его изображением на экране. Применяемые нами электронные технологии все еще не позволяют обеспечить такое же удобство. (Возможной причиной может являться также то, что я все еще живу в каменном веке.)

Телефон. Как средство дистанционного взаимодействия, на втором месте после документа стоит телефон, который иногда позволяет гораздо быстрее добиться согласования, чем электронная почта. В частности, многие пользователи электронной почты ощутили на себе последствия неправильного толкования их непродуманных писем, в условиях, когда нельзя дать устные пояснения и моментально устранить недоразумения. Мгновенная передача сообщений — плохая замена телефонной связи.

Совместная корректировка документа с обсуждением по телефону. Если корректировка документа в режиме совместного использования сопровождается обсуждением по телефону, то создается сочетание средств дистанционного взаимодействия, гораздо более продуктивное по сравнению с этими отдельно взятыми средствами. Характерной особенностью этого сочетания является дополнительное взаимодействие в реальном времени, благодаря которому удается обойтись без многословных письменных пояснений и заблаговременно устранить потенциальные недоразумения. Кроме того, что менее очевидно, совместно разрабатываемый документ позволяет внести бо́льшую конкретизацию и детализацию в телефонный разговор. При обсуждении документа по телефону приходится

дословно согласовывать одну формулировку за другой, в процессе чего выходят на поверхность многие проблемы, которые в противном случае остались бы незамеченными.

Это сочетание средств дистанционного взаимодействия является очень мощным. Сотрудник нашей лаборатории UNC Кертис Келлер (Kurtis Keller), штатный инженер-механик, занимался проектированием нового дисплея, закрепляемого на шлеме, совместно с Сэмом Дрэйком (Sam Drake) из Университета шт. Юта (UU). В нашем распоряжении была весьма эффективная, работающая в реальном времени широкополосная видеосистема конференц-связи между UU и UNC. Наша станция для проведения видеоконференций находилась в холле, на расстоянии 50 метров от офиса Кертиса, т.е. до нее нетрудно было дойти. Тем не менее мы обнаружили, что на всем протяжении процесса проектирования Кертис не сделал ни малейшей попытки воспользоваться системой конференц-связи. Он постоянно работал в своем офисе и общался с Дрейком по телефону, а когда нужно было обменяться графическими изображениями, оба эти участника проекта рисовали их на экранах своих рабочих станций.

Проведение видеоконференций

Непосредственно после широкого распространения средств проведения видеоконференций началась шумиха по поводу того, что эти средства "изменят всю картину" в области дистанционного сотрудничества, но фактически уровень их практического применения возрастал не так быстро, как предполагалось первоначально.

Чем же обусловлено замедленное внедрение? На первых порах пропускная способность каналов связи, применявшихся для проведения видеоконференций, была низкой, поэтому частота смены кадров оказывалась недостаточной и весь сеанс взаимодействия не выглядел как естественный. В наши дни достигнуты приемлемые скорости передачи видеоданных. Но возросло ли благодаря техническим усовершенствованиям удобство в работе?

- **Поле обзора.** Видеосвязь хорошо подходит для проведения сеансов взаимодействия двух собеседников, но если одна половина комитета проводит сеанс связи с другой, то невозможно видеть каждого и следить за выражениями лиц.
- **Повышение удобства совместного изучения документов и презентаций.** Желательно, чтобы можно было видеть одновременно и докладчика, и демонстрируемый им документ или слайд, а не пассивно воспринимать то, как видеокамера попеременно показывает то одно, то другое. В ходе участия в видеоконференции может возникнуть желание, например, чтобы содержимое какой-то таблицы было раскрыто более подробно. Любому из участников может потребоваться сделать личные заметки или

внести замечание, которое могли бы увидеть все. В действительности при проведении видеоконференции нельзя обойтись без виртуальной доски, предоставляемой обеим сторонам в совместное использование.

- **Повышение разрешающей способности.** Разрешающая способность средств проведения видеоконференций все еще недостаточно высока, чтобы можно было полностью показать для общего обзора страницу текста с размерами 210x297 мм или добиться того, чтобы лица участников были хорошо различимыми.
- **Улучшение способов отображения трехмерного пространства.** При проведении видеоконференций изображение обычно кажется плоским. Безусловно, из-за этого редко возникают недоразумения, касающиеся раскрытия темы самой видеоконференции, зато участники никак не могут забыть о том, что не беседуют со своими коллегами как при личной встрече.

Предпосылки получения наибольшей отдачи при проведении видеоконференций. В определенных обстоятельствах проведение видеоконференции оказывается более продуктивным по сравнению с телефонной связью, несмотря на все еще существующие технические ограничения. Но до сих пор ничто не может окончательно заменить личное непосредственное общение. Ниже указаны определенные ситуации, в которых приходится внимательно следить за мимикой лица и читать язык тела.

- Проведение отбора среди пришедших со стороны претендентов на получение работы для определения достойных финалистов.
- Решение задач, жизненно важных для одного или нескольких участников взаимодействия.
- Преодоление крайней неуверенности со стороны одного из участников взаимодействия.
- Устранение организационных или языковых барьеров.

Проведение видеоконференций на основе высокотехнологичных средств. Для достижения максимального реализма в работе систем конференц-связи был проведен значительный объем исследований. Мой коллега Генри Фукс (Henry Fuchs) усовершенствовал организацию видеоконференций с помощью визуальных подсказок, обозначающих глубину, и наглядно продемонстрировал, что в результате этого усовершенствования эффект присутствия стал гораздо более ярко выраженным. Проводилось слежение за положением головы каждого участника, что позволило ввести в действие мощный кинетический эффект глубины, заключающийся в том, что после изменения положения головы реконструированные объекты на экране сдвигались с учетом их расстояния от камер. Кроме того, с помощью многочисленных камер создавалось трехмерное изображение, которое демонстрировалось в стереоформате через два проектора с поляризованными фильтрами[9].

Применение технологий дистанционного сотрудничества на основе доставки по запросу или принудительной доставки. Развитие аппаратных средств и программного обеспечения для дистанционного сотрудничества стало результатом проведения большого объема академических исследований. Благодаря этому было создано значительное количество инструментальных средств и систем, часть которых распространяется на коммерческой основе, кроме того, проведен ряд научных конференций[10] и организован выпуск авторитетных журналов[11], посвященных этой теме (а также теме сотрудничества между людьми, не разделенными пространственными и временными барьерами).

Однако изучение этих инструментальных средств и систем вынуждает прийти к заключению, что большинство из них было создано на основе реализации технической идеи, а не анализа особенностей организации сотрудничества или потребностей в обеспечении сотрудничества. И действительно, проведенный в Интернете поиск по теме организации дистанционного сотрудничества показал, что 49 результатов из первых 50 относились к инструментальным средствам или обучению, но не к сотрудничеству в проектировании. А изучение литературы показало, что из 20 рассмотренных книг 19 были посвящены описанию инструментальных средств, а не их применению для решения задач.

Такая непродуктивная направленность усилий меня глубоко беспокоит. Она свидетельствует о расточительном использовании драгоценных ресурсов — сил и времени научных работников, а также о том, что обучение наших самых способных студентов происходит неправильно. Эффективная подготовка инструментария всегда начинается с пользователя и задачи. Мой опыт показывает, что создаваемые инструментальные средства становятся наиболее успешными, если они рассчитаны на реального пользователя, перед которым стоят реальные задачи. Недостаточно просто подготавливать прототипы, поскольку они изобилуют ошибками, к тому же при отсутствии связи с реальностью критические замечания становятся поверхностными и примитивными. Я много писал на эту тему в других своих работах, но положение так и не изменилось[12].

Примечания и ссылки

1. В работе [142] приведено сообщение о разработке концепции "коллективного интеллекта", которая предусматривает объединение усилий коллективов специалистов с помощью Интернета для создания крупных технических проектов.

   Но анализ последних разработок и результатов новейших исследований показывает, что применение моделей, открытых к инновациям, оказывается успешным только при условии их тщательного проектирования для решения определенной задачи, а также при мобилизации соответствующих стимулов для привлечения наиболее продуктивно работающих сотрудников. "Не следует считать, что вам достаточно лишь применить в своей работе сложившийся

подход, и дело пойдет на лад, — отметил Томас В. Мэлоун (Thomas W. Malone), директор Центра коллективного интеллекта Массачусетсского технологического института. — Это неправда. Здесь нет волшебства".

2. См. [58] — превосходный отчет, опубликованный в газете. См. также `http://en.wikipedia.org/wiki/Airbus_380`; в книге взята за основу версия этой страницы от 9 сентября 2008 года.
3. Для проведения этой работы потребовалась отдельная небольшая группа архитекторов.
4. В работе [226] приведено очень компетентное, полное и беспристрастное обсуждение проекта и встретившихся в нем трудностей по прошествии двух лет от даты объявления о его начале. В отношении совместного проектирования он отметил следующее: "Работа группы инженеров из разных стран оказалась весьма эффективной, что позволило компании IBM с полным правом утверждать, что компьютеры серии 360, вероятно, являются первым продуктом, действительно созданным в результате международного сотрудничества в области проектирования" (с. 142).

   Питер Фагг (Peter Fagg), технический руководитель проекта System/360, проделал феноменальную работу по управлению разработкой множества новых устройств ввода-вывода с участием многочисленных подразделений из разных стран, не будучи непосредственным руководителем ни одной из групп разработчиков.
5. В работах [113] и [202] описаны недостатки распределенной работы. В работе [117] приведен ряд отчетов по различным аспектам распределенной работы.
6. См. [100].
7. В статье [96] приведен отчет об интересном исследовании, в котором дано сравнение результатов сотрудничества в проектировании в условиях совместного и дистанционного расположения.

   В этой статье описано, как осуществлялась и какие выводы позволила сделать научно-исследовательская работа, в которой проводилось сравнение процессов и результатов выполнения рисунков парами студентов-проектировщиков, работающих рядом друг с другом, с другими парами, члены которых общались друг с другом дистанционно, с применением компьютеризированных средств... Как оказалось, участники лабораторных исследований совместной работы, проводимой при личном контакте, широко пользовались совместно разрабатываемыми рисунками и эскизами, тогда как при совместной работе в дистанционном режиме с применением компьютеров это происходило значительно реже. Для иллюстрации обнаруженного феномена использовалось понятие "действий, связанных с созданием графических рисунков".

С другой стороны, в работе [194] не только отмечено отсутствие различий, но и указано, что ученым удалось выявить преимущества и недостатки каждого режима работы:

> *Разработки по теме общих научных рабочих пространств развиваются настолько стремительно, что опережают критический анализ состояния дел в этой области. Можно ли считать, что возможности, открывающиеся в результате применения общих рабочих пространств, перевешивают их недостатки? Для оценки системы общего научного рабочего пространства мы провели управляемый эксперимент с неизменными критериями, в котором сравнивались результаты и сам процесс научной работы, выполняемой 20 парами участников (не получившие степени студенты старших курсов с научными специальностями), работающих рядом друг с другом и в отдалении друг от друга.*
>
> *Мы собрали научные результаты (получившие оценки преподавателей лабораторные отчеты) для изучения качества научной работы, данные последующего заполнения вопросника для измерения приспособляемости системы и данные последующего проведения интервью для определения впечатлений участников от выполнения научных работ в тех и других условиях. Мы выдвинули гипотезу, что участники исследования, как правило, показывают меньшую продуктивность, сообщают о большем количестве затруднений, а также менее склонны к принятию системы организации работы при дистанционном сотрудничестве.*
>
> *Вопреки ожиданиям, количественные данные показывают отсутствие статистически значимых различий в отношении продуктивности и приспособляемости. Причину получения этого нулевого результата помогли объяснить следующие качественные данные: участники сообщали о преимуществах и недостатках работы в тех и других условиях, но разработали обходные методы, позволяющие справляться с очевидными недостатками дистанционного сотрудничества. Безусловно, при рассмотрении в целом анализ данных привел к получению нулевых результатов, но общее изучение проблемы позволило нам сделать вывод о наличии положительного потенциала для разработки и принятия на вооружение систем организации общего научного рабочего пространства.*

8. На сайте `Economist.com` (см. [87]) приведены размышления анонимного автора о том, что "текущий циклический спад может сопровождаться структурным уменьшением количества деловых командировок, обусловленным успехами в информационной технологии". Таким образом, современный кризис может способствовать распространению средств проведения видеоконференций.

9. См. [173]; [207]; http://www.cs.unc.edu/Research/stc/inthenews/pdf/washingtonpost_2000_1128.pdf; в книге взята за основу версия этой страницы от 28 августа 2009 года.

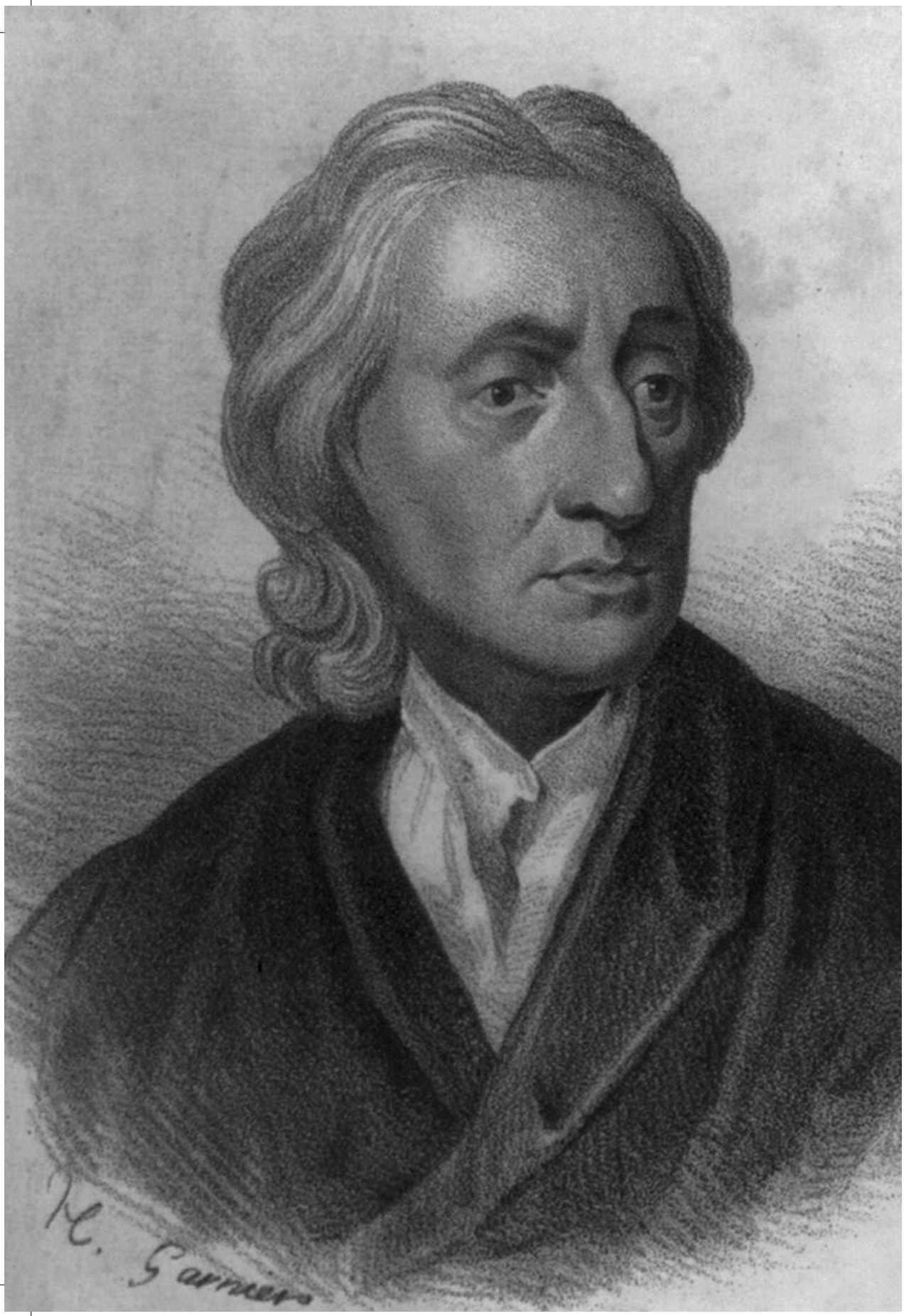
   В качестве основы для дистанционного сотрудничества исследовались также виртуальные миры, такие как Second Life. См. например, http://blog.irvingwb.com/blog/2008/12/serious-virtual-worlds-applications.html.

10. См. http://www.cscw2008.org/.
11. Computer Supported Cooperative Work (CSCW): *The Journal of Collaborative Computing*. ISSN: 0925-9724 (печатная версия); ISSN: 1573-7551 (электронная версия).
12. См. [41]; [44].

III

Перспективы создания теории проектирования

TheDesignofDesign.indb 137 30.10.2012 15:58:28



TheDesignofDesign.indb 138 30.10.2012 15:58:28

8

Проявления рационализма и эмпиризма в проектировании

Человеку свойственно ошибаться, и большинство людей во многих обстоятельствах, движимые чувством или заинтересованностью, подвержены искушению совершить ошибку.

См. [141]

...две операции нашего мышления, интуиция и дедукция, лишь в отношении которых мы можем утверждать, что на них следует полагаться в приобретении знаний.

См. [80]

◂ *Джон Локк (1632–1704), британский философ-эмпирик. Wikimedia Commons (commons.wikimedia.org)*

Сравнение рационализма и эмпиризма как систем взглядов

Могу ли я лишь на основании достаточно полного обдумывания правильно спроектировать сложный объект? Этот вопрос, будучи распространенным на область проектирования, позволяет провести "водораздел" между двумя давно сложившимися философскими системами, рационализмом и эмпиризмом. Рационалисты полагают, что ответ на этот вопрос положителен, а эмпирики считают его отрицательным[1].

Указанный "водораздел" гораздо шире, чем кажется на первый взгляд. По сути, мы сталкиваемся с фундаментальной философской проблемой — описанием природы человека как создателя.

Рационалист полагает, что человек изначально является здравомыслящим (и стремящимся творить добро), подверженным ошибкам и способным совершенствоваться под влиянием образования и воспитания. После правильного обучения, профессионального становления под действием опыта и достаточно тщательного обдумывания проектировщик становится способным создать безупречный проект. Поэтому цель методологии проектирования состоит в изучении того, как размышлять над проектом в направлении достижения его безупречности.

Эмпирик полагает, что человек изначально порочен и подвержен всем искушениям и ошибкам. Все, за что бы он ни брался, будет иметь недостатки. Поэтому цель методологии проектирования состоит в определении того, как выявить недостатки с помощью эксперимента, чтобы постепенно их исправить, возвращаясь к проекту снова и снова.

Примеры проявления тех и других взглядов весьма многочисленны. Аристотель полагал, что совершать научные открытия можно исключительно по дедукции, путем рассуждений, поэтому пришел к выводу, что тяжелые предметы должны при падении достигать земли быстрее, чем легкие. Галилей считал, что необходим эксперимент, и оказался настолько дерзким, что бросил вызов издревле несокрушимому авторитету Аристотеля.

Рене Декарт (1596–1650), по-видимому, наиболее прямо изложил рационалистические взгляды. Джон Локк (1632–1704) четко проявил себя как эмпирик.

До сегоднящнего дня французская наука продолжает преуспевать в создании красивых логических структур, начиная от проведенного Фурье анализа потока теплоты и заканчивая термодинамикой Карно и математическими построениями группы Бурбаки. Тем временем британская наука шагает от одного достижения к другому, придерживаясь эмпирической традиции; достаточно назвать работы Уатта, Фарадея, Хевисайда, Брегтса и др.

Проектирование программного обеспечения

Можно ли считать компьютерную программу математическим объектом, который можно представить в абстрактной форме и обосновать его правильность с помощью доказательства? Эту идею яростно отстаивают рационалисты во главе с Эдсгером Дейкстрой (Edsger Dijkstra)[2]. Они считают, что вся суть программирования сводится к правильно организованным тщательным размышлениям. Иными словами, можно и должно проектировать программное обеспечение как отвечающее критериям правильности, а затем доказывать правильность проекта. И этого достаточно[3].

В наши дни общепризнанно, что программа — чисто математический объект и в принципе может быть спроектирована в идеальной форме с помощью правильно организованных размышлений. Возникающие при этом трудности обусловлены несовершенством подготовки проектировщиков, а не информационной среды проектирования. Эмпирики полагают, что люди неизбежно совершают ошибки в определении целей, в разработке архитектуры программного обеспечения, в реализации в виде объектов (алгоритмов и структур данных), а также в реализации непосредственно в коде. Эта твердая уверенность в подверженности ошибкам предписывает применение такой методологии проектирования, которая предусматривает подготовку проекта, заблаговременное создание прототипов, предварительное тестирование с участием пользователей, итерационную последовательную реализацию, проверку на обширном наборе тестовых примеров и регрессионное тестирование после внесения каждого изменения.

Сам я — убежденный эмпирик

Я стал таковым прежде всего на основании собственного опыта. Только дважды в жизни мне удалось написать программу, которая работала правильно с самого начала и выполняла именно то, что от нее требовалось. Один из этих случаев относился к моей самой первой достаточно значительной программе. Когда я учился в аспирантуре Гарвардского университета в 1953-1954 гг., нам поручили разработку программ по индивидуальным проектам. Нам предоставили компьютер Harvard Mark IV, который был достаточно удобным в работе, но каждая группа из двух аспирантов получала только по два отрезка машинного времени продолжительностью в один час, в течение которых нужно было отладить и запустить на выполнение программы, предусмотренные проектом за семестр. Я работал в одной группе с Уильямом В. Райтом (William V. Wright), проявившим необычайные способности, и мы вдвоем придирчиво и не жалея сил проверяли вручную нашу скромную программу, состоящую из полутора тысяч строк. Выполнение этой программы прошло успешно с первого раза.

Кое-кто может утверждать, что этот эксперимент служит практическим доказательством правильности рационального подхода к проектированию. Но мы не доказывали правильность нашей программы, а проверяли ее вручную, моделируя ход выполнения. Кроме того, я сомневаюсь в том, может ли обычный работник достичь равного нам уровня мотивации и в своей повседневно работе проводить столь же тщательную проверку, как мы с Биллом. Да, в принципе это возможно. Но когда дело касается реальных людей и современного программного обеспечения реальных масштабов, на это не следует рассчитывать.

Какую роль играет опыт в проектировании правильных программ? На практике были попытки использования формальных методов доказательства для обоснования того, что ядро той или иной безопасной операционной системы спроектировано и реализовано правильно[4]. Безусловно, это — способ применения указанных методов в полном соответствии с их назначением, но прежде всего должна быть достигнута высокая степень уверенности в том, что формального доказательства будет достаточно.

Но нет никакого сомнения, что такая уверенность не может быть стопроцентной. История математики показывает, что многие доказательства, некогда признанные верными, в дальнейшем оказались ошибочными[5]. Формальное доказательство не гарантирует отсутствие ошибок. Но указанный метод имеет свое преимущество в том, что при проведении формального доказательства рассуждения проводятся иначе по сравнению с таковыми при разработке программы, поэтому весьма существенно повышаются шансы, что не останутся ошибки, не обнаруженные в том и в другом подходе.

По-видимому, ядро операционной системы относится именно к тем компонентам, которые требуют применения метода доказательства правильности. Если ядро является безопасным и правильно спроектированным, то появляется возможность свести к минимуму ущерб, причиненный ошибками, лазейками или кодом, введенным в любом месте программы с преступными намерениями. При доказательстве правильности программы приходится проводить значительный объем работы, сравнимый с тем объемом, который затрачивается на создание программы. Но доказательство не может подтвердить правильность исходной формулировки назначения программы[6].

Харлан Миллс (Harlan Mills) и его коллеги из компании IBM разработали вариант доказательства правильности проекта, который мне кажется вполне приемлемым. Это — метод "стерильных условий", руководствуясь которым Миллс и его коллектив подвергают интенсивному групповому исследованию каждый аспект проекта. Группа проектировщиков на своих совещаниях заслушивает доклад конкретного проектировщика, который объясняет, почему он считает представленный им проект правильным, после чего остальные участники совещания оспаривают его аргументы и лежащие в их основе неявные предположения[7].

Формальное доказательство правильности обычно неосуществимо, а отказ от всех усилий в направлении систематической проверки (как правило, в

экстремальных условиях) опасен. Мне кажется, что предложенное Миллсом систематическое, но неформальное групповое исследование логических аргументов позволяет достичь разумного и практичного баланса.

Примеры применения подходов на основе рационализма, эмпиризма и проверки правильности в других областях проектирования

Насколько я могу судить, нет ни одной области проектирования, кроме разработки программного обеспечения, в которой проектировщики хотя бы пытались доказать правильность проекта с применением строгих формальных методов. Возможно, причина этого состоит в том, что программное обеспечение, как и математические теоремы, является исключительно продуктом размышлений, поэтому можно вести речь о строгом доказательстве его правильности. В большинстве других областей проектирования создание проекта влечет за собой физическую реализацию, и никто не станет доказывать теоремы о материалах и их недостатках или о том, как распорядиться пространством, отведенным для размещения компонентов.

При проектировании организационной структуры, как и при разработке проекта программного обеспечения, не приходится сталкиваться с физическим воплощением. Но я не знаю ни об одной попытке доказать правильность или даже работоспособность намеченной организационной структуры. Безусловно, авторы работы "The Federalist Papers" (Записки федералиста) попытались доказать осуществимость на практике статей Конституции Соединенных Штатов с помощью тщательно продуманных логических аргументов. Непревзойденная мудрость этих людей до сих пор восхищает новые и новые поколения, но разразившаяся вслед за выпуском указанного труда Гражданская война в США (исключительно серьезная системная катастрофа) показала неполноту всех этих доказательств.

В областях проектирования, отличных от разработки программного обеспечения, применение методов доказательства правильности проектов остается невозможным, но в этих областях широко используется проверка проектов с применением бесчисленного количества методов анализа и моделирования.

На современном производстве механические детали подвергаются испытаниям под нагрузкой, вибрационным и акустическим исследованиям. Обзорные экскурсии, моделируемые в реальном времени или снятые на видеокамеру, позволяют архитекторам и клиентам апробировать предполагаемые сценарии эксплуатации спроектированных зданий. Испытания под нагрузочными напряжениями применяются для проверки способности сооружений противостоять снегопадам и ураганам. Динамические испытания под нагрузкой служат средством тестирования устойчивости к землетрясениям.

Компьютерное оборудование подвергается исчерпывающей проверке на многих уровнях: аппаратных средств, логического проекта и эксплуатации программ.

Всесторонние испытания проходят даже операционные системы компьютеров, которые еще только подготавливаются к массовому выпуску; они осуществляются (с огромными затратами времени) на модели компьютера, развертываемой на существующем оборудовании.

В связи с проведением таких обширных эмпирических исследований неизбежно возникает необходимость неоднократного повторения в процессе проектирования одних и тех же работ. При этом проводимые проверки все больше усложняются, а это означает, что применяемые критерии оценки полноты выполнения требований и соблюдения ограничений становятся все более точными. Но вместе с тем задача оценки того, насколько проект соответствует заданным целям, в значительной степени упрощается и решается более легко. Тем не менее, какими бы совершенными ни были подобные процедуры анализа и и моделирования, они не позволяют определить, правильно ли выбраны цели проекта и действительно ли предполагаемые условия эксплуатации проектируемого продукта будут таковыми на самом деле.

Могу ли я лишь на основании достаточно полного обдумывания правильно спроектировать сложный объект? Разумеется, нет. Таким образом, остаются необходимыми практические испытания и доработки. Из сказанного выше не следует, что с момента начала проведения проверки проекта работа над самим проектом должна прекращаться. В главах части III приведены некоторые соображения по этому поводу.

Примечания и ссылки

1. Рационализм и эмпиризм относятся к категории способов приобретения новых знаний, т.е. входят в сферу научного направления, известного под названием *эпистемологии*. Безусловно, взгляды тех, кто придерживается принципов рационализма и эмпиризма, внешне кажутся противоположными, но на деле невозможно добиться успеха, не прибегая в определенной степени к применению и тех и других способов. В качестве примера можно отметить, что Декарт в основном отстаивал эмпирический подход к научной деятельности, а Локк считал, что в основе развития математики должен лежать рационализм.

   За представленную в этой главе попытку распространить понятия эпистемологии на проектирование несу ответственность я сам, но мне кажется, что здесь я действительно зашел очень далеко и вполне могу высказать ошибочные взгляды.

2. См. [82].
3. См. [81], с. 174–186.
4. См. [130] и [131], где приведен хороший краткий обзор и описаны весьма впечатляющие современные достижения. Авторы утверждают, что в их

работе впервые была проведена полная функциональная проверка нетривиальной реализации ядра.

5. Например, было доказано, что для умножения матриц требуется n^3 операций скалярного умножения. Отмечен также недостаток, который заключался в предположении, что эти операции должны выполняться на векторах. См. [198].

6. Описан получивший широкую известность и поучительный случай аварии самолета компании Lufthansa в рейсе 2904, который сошел со взлетно-посадочной полосы в Варшаве из-за сбоя при подключении управляемых компьютером систем торможения. Применявшийся при этом программный код соответствовал спецификации, которая оказалась неправильной в сложившихся непредвиденных обстоятельствах. По материалам `http://en.wikipedia.org/wiki/Lufthansa_Flight_2904`; в книге взята за основу версия этой страницы от 16 июля 2009 года.

   Для обеспечения того, чтобы система реверса тяги и интерцепторы активизировались только в ситуации приземления, должны были соблюдаться все следующие условия для ввода в действие систем торможения с помощью указанного программного обеспечения:

   - к каждой главной распорке механизма приземления должна была прилагаться нагрузка не менее 12 тонн;
   - колеса самолета должны были двигаться со скоростью больше 72 узлов (133,43 км/час);
   - рычаги управления реверсом тяги должны были находиться в положении реверса тяги.

   Авария в Варшаве произошла из-за того, что не было выполнено ни одно из первых двух условий, и в результате не активизировалась самая эффективная система торможения. Первый пункт оказался невыполненным, поскольку самолет приземлялся под углом (чтобы противодействовать возможному сдвигу под влиянием ветра). Из-за этого не была достигнута нагрузка в 12 тонн, требуемая для срабатывания датчика, на обоих механизмах приземления. Второе условие также не было выполнено из-за эффекта гидропланирования на влажной взлетно-посадочной полосе.

7. См. [154].

   В справочние Wikipedia (см. `http://en.wikipedia.org/wiki/Cleanroom_Software_Engineering`; в книге взята за основу версия этой страницы от 30 октября 2008 года) приведен качественный общий обзор всего подхода, основанного на создании стерильных условий.

   Основными принципами процесса разработки на основе создания стерильных условий являются следующие.

Разработка программного обеспечения на базе формальных методов

В разработке на основе создания стерильных условий используется метод коробчатой конструкции для формулировки спецификации и проектирования программного продукта. Проверка того, правильно ли реализуется спецификация в проекте, осуществляется путем проведения обзора с участием всего коллектива проектировщиков.

Последовательная реализация в условиях применения статистического контроля качества

В разработке на основе создания стерильных условий используется итерационный подход, предусматривающий последовательное создание продукта, в ходе которого происходит постепенное приращение объема реализованных функциональных средств. Качество каждого приращения проверяется по заранее установленным стандартам, что позволяет определить, достаточно ли удовлетворительно проходит процесс разработки. Обнаружение любого отклонения от стандартов качества влечет за собой прекращение проверки на текущем этапе разработки и возврат к фазе проектирования.

Статистически обоснованное тестирование

Тестирование программного обеспечения в процессе, основанном на создании стерильных условий, выполняется в виде статистического эксперимента. Исходя из формальной спецификации выбирается и тестируется представительное подмножество программных траекторий ввода-вывода. Затем производится статистический анализ полученной выборки для выработки оценки надежности программного обеспечения и определения доверительного уровня этой оценки.

9

Модели пользователей — лучше неправильные, чем неопределенные

...Истина скорее может родиться из ошибки,
чем из путаницы.

См. [15]

◂ *Группа архитекторов. Андерсон Росс (Anderson Ross), Getty Images Inc. — Stockbyte*

Явно заданные модели пользователя и использования

Опытные проектировщики часто начинают работу с точного формулирования того, что им известно о пользователе, цели использования будущего продукта пользователем и режимах использования. А еще более умудренные опытом проектировщики также явно перечисляют то, что они не знают, но принимают в качестве гипотез, касающихся пользователей.

Если перед этими проектировщиками стоит задача разработки множества различных приложений или охвата многих разных категорий пользователей, то они подготавливают описания отдельно для каждого случая, а затем явно присваивают веса каждому определенному набору требований, чтобы составить модель использования[1].

Чем более детализированными и конкретизированными являются предположения, выдвинутые проектировщиками, тем более богатую почву они создают для обдумывания всех подробностей разработки проекта на ранних этапах. Кроме того, собранный таким образом материал так или иначе потребуется позже, по мере дальнейшего развития проекта. Если же такая оценка проекта проводится на предварительном этапе, это позволяет в будущем избежать многих ошибок[2].

Это — вполне оправданный подход, но применяется ли он в действительности?

Можно ли ожидать того, что люди на самом деле будут выполнять всю эту дополнительную работу до начала проектирования? Разумеется, ответом на данный вопрос является то, что этим занимаются лишь немногие из проектировщиков. Лично я считаю, что мы должны явно определять модели пользователей и использования в намного большем количестве проектов, чем принято сейчас. Ввод этого правила в действие позволит улучшить практику проектирования.

В настоящее время назрела потребность в создании явных моделей приложений и пользователей, что следует непосредственно из специфических особенностей современного проектирования: проведение самого проектирования с участием крупных коллективов, а также создание проектов сложных, а не простых инструментальных средств.

Коллективное проектирование

Фактически в ходе своей работы все проектировщики сознательно или подсознательно руководствуются моделями пользователей и использования. А в связи с переходом к коллективному проектированию выдвигается принципиально новое требование, согласно которому придерживаться одной и той же модели пользователя и одной и той же модели использования должен весь коллектив. Поэтому должны быть подготовлены явно заданные модели и сформулированы предположения.

Но это делается редко, поскольку обычно все члены коллектива, не обсуждая данную тему, полагают, что они и без этого придерживаются одинаковых взглядов. В конце концов, ведь каждый слышал, какие задачи поставил руководитель предприятия и какие требования выдвинул перед коллективом. Кроме того, каждый прочел документ, в котором определены цели. А все участники разработки и без того имеют достаточную подготовку.

Но в действительности дела обстоят не так уж просто. Фактически каждый из проектировщиков имеет разный опыт использования систем, аналогичных проектируемой, а на основании этого опыта формируются разные представления о типичном пользователе. Кроме того, каждый из проектировщиков на протяжении своей трудовой деятельности участвовал в создании разных приложений, а из этого следует, что у каждого из них сложилось собственное представление о том, каким должно быть вновь создаваемое приложение. Если в коллективе не будет выработан общепринятый набор явно заданных предположений, то каждый проектировщик станет руководствоваться неявно принятыми им предположениями. В ходе проектирования приходится главным образом самостоятельно принимать такие микроскопические решения, которые даже не заслуживают обсуждения, но каждый из участников проектирования будет принимать их по-разному, что приведет к потере концептуальной целостности.

Предполагаемые модели использования будут разными, а это неизбежно приведет к возникновению несогласованностей в проекте, создаваемом по принципу коллективного проектирования. Например, в нескольких частях операционной системы OS/360 (которая теперь именуется z/OS) отражены два полностью несовместимых подхода к отладке: в одном упор сделан на использовании пакетов, а в другом предполагается эксплуатация в режиме разделения времени с использованием терминалов. Решение о воплощении этих двух режимов использования не было принято сознательно; то, что это произошло, просто стало следствием участия в проектировании двух подгрупп, которые придерживались разных моделей использования[3]. Результатом случившегося стало неоправданное увеличение общего объема кода и появление несовместимостей.

Сложные проекты. По мере возрастания сложности инструментальных средств возрастает потребность в применении явно заданных моделей использования. Даже если речь идет о подготовке к выпуску лопат, необходимо явно определить, предназначены ли они для переброски угля, глины, зерна, снега или определенной смеси; будет ли ими пользоваться ребенок, женщина или мужчина; должны ли они быть предназначены для одноразового применения, или ими будет постоянно пользоваться чернорабочий. А насколько более велика потребность в создании явных моделей использования для грузовиков, электронных таблиц или зданий производственного назначения!

Кроме того, чем сложнее проект, тем меньше вероятность того, что все проектировщики окажутся специалистами в данной проблемной области, которые могли бы мысленно встать на место пользователей. Поэтому реализация неявно принятых моделей приводит к еще более опасным последствиям.

Организация проектирования в условиях отсутствия необходимых фактических данных

Зачастую сразу после того, как проектировщик приступает к созданию явных моделей использования, ему приходится сталкиваться с серьезными затруднениями: жестокая реальность показывает ему, насколько много еще осталось того, что он не знает. В ходе решения самой этой задачи он становится вынужден задавать вопросы, которые, возможно, не встали бы перед ним до перехода к гораздо более позднему этапу проектирования. В этом состоит неоспоримое преимущество подхода на основе создания явных моделей использования.

Предположим, ведется проектирование программного продукта для разработки маршрутов и расписаний движения школьных автобусов. Тщательные полевые исследования на базе двух или трех "представительных" школьных систем позволили выявить целый ряд фактов: временные ограничения, количество автобусов, количество водителей, места проживания учащихся. Тем не менее анализ этих фактов при подготовке к созданию модели использования для общей программы создания маршрутов и расписаний порождает лишь еще большее количество вопросов.

До какой степени системы, применявшиеся в качестве образцов, действительно являются представительными? В какие диапазоны укладываются все рассматриваемые параметры после того, как речь заходит обо всем множестве предполагаемых пользователей? Каковы распределения значений параметров?

Чем измеряются интенсивности изменений от одного периода составления расписаний до другого? Каковыми будут указанные диапазоны по истечении, допустим, пяти лет? Десяти лет?

Чем более однозначными становятся поставленные вопросы, тем более расплывчатыми становятся ответы. Что должен сделать проектировщик, поставив перед собой задачу создания явных моделей использования?

Руководствоваться предположениями!

Я вполне убежден в том, что после выхода за рамки вопросов, на которые можно получить ответ с применением достаточно хорошо продуманного анкетирования, проектировщик должен перейти к выдвижению предположений или, иначе говоря, к определению всеобъемлющего набора атрибутов и значений, характеризующихся некоторыми гипотетическими плотностями распределения, чтобы разработать полные, явно заданные и разделяемые всеми заинтересованными лицами модели пользователя и использования.

Четко сформулированное предположение всегда бывает лучше, чем невысказанное допущение.

Эта процедура может показаться примитивной, но из нее вытекает много преимуществ, которые описаны ниже.

Чтобы можно было правильно выдвинуть предположения в отношении значений и частотных распределений, проектировщик вынужден очень тщательно оценивать характеристики предполагаемого множества пользователей.

После формулирования значений и частотных распределений в письменном виде появляется почва для их обсуждения. *Проще критиковать что-то конкретное, чем создавать самому*, поэтому увеличивается вклад в работу, сделанный всем коллективом. Обсуждая представленные материалы, все участники знакомятся с предлагаемой постановкой, а все различия во взглядах на подготавливаемые модели пользователей, разделяемых отдельными проектировщиками, выплывают на поверхность. Кроме того, как правило, становятся видимыми другие нераспознанные предположения[4].

Явное перечисление значений и частотных распределений помогает каждому понять, какие решения зависят от тех или иных свойств множества пользователей.

А что еще более важно, это приводит к выдвижению следующих важных вопросов. Какие предположения действительно являются значимыми? Насколько велика эта значимость? Ценным становится сам по себе даже такой поверхностный анализ влияния принятых предположений. После того как обнаруживается, насколько важные решения зависят от тех или иных конкретных предположений, становится понятным, что не нужно экономить на разработке лучшей оценки.

Однако даже при этом многие предположения останутся спорными и не поддающимися проверке. Поэтому на главного архитектора возлагаются обязанности по принятию конкретного набора предположений и доведению их в качестве руководящих до коллектива проектировщиков.

Неправильная модель лучше, чем неопределенная!

Ознакомившись с вышеизложенным, читатель может возразить: “Исходя из чего я могу знать или даже формулировать предположения в отношении столь многих фактов, касающихся пользователя и использования?” Ответ состоит в следующем: “Соответствующие предположения так или иначе будут вами фактически сделаны”. Это означает, что при принятии каждого проектного решения будут учитываться сознательно или подсознательно предположения проектировщика, касающиеся пользователей и использования. В действительности это часто означает следующее: проектировщик, не сформулировавший свои предположения явно, ставит себя на место пользователя, создавая проект в таком виде, какой он сам пожелал бы иметь, будучи пользователем. Но он не пользователь.

Поэтому неправильные явные предположения намного лучше, чем неопределенные. Возможно, что неправильные будут подвергнуты сомнению, а неопределенные таковыми и останутся.

Примечания и ссылки

1. Модель использования представляет собой взвешенную совокупность результатов анализа примеров использования. В работе [178] весьма подробно рассмотрено понятие примеров использования.
2. Исчерпывающие данные по этой теме можно найти в работе [60].
3. См. [40], с. 56–57.
4. В проводимом мной расширенном курсе по архитектуре вычислительной системы к студентам предъявлялось требование, чтобы они учитывали этот этап в своих учебных проектах. Если они выполняли указанную задачу достаточно добросовестно, результатом становилось весьма существенное повышение качества проектов.

TheDesignofDesign.indb 155 30.10.2012 15:58:29

USA
USA

10

Лимитируемые ресурсы — дюймы, унции, биты, доллары

Чтобы обеспечить концептуальную целостность проекта, особенно если он разрабатывается по принципу коллективного проектирования, необходимо явно перечислить ограниченные ресурсы, открыто их отслеживать и жестко контролировать.

◂ *Ракета Аполлон. IStockphoto*

Общее определение понятия лимитируемого ресурса

В любом проекте имеется по крайней мере один ограниченный ресурс, который должен нормироваться или лимитироваться. Иногда также приходится оптимизировать использование одновременно двух или более ресурсов, но обычно только один из таких ресурсов является доминирующим, а остальные в основном имеют вид пожеланий или ограничений. Экономисты называют такие ресурсы *лимитируемыми*, а я предпочитаю при их определении подчеркивать одну из важных обязанностей проектировщика — продуманное распределение ресурсов[1].

Безусловно, проектировщики часто рассматривают в качестве оптимизируемого ресурса стоимость проекта или соотношение стоимости к производительности, но в действительности в своей работе редко действуют так, как если бы это на самом деле были ресурсы. Из этого следует, что для обеспечения концептуальной целостности проекта, особенно если он разрабатывается по принципу коллективного проектирования, необходимо явно перечислить ограниченные ресурсы, открыто их отслеживать и жестко контролировать.

Ресурсы, которые сложно оценивать в деньгах

При разработке проектов часто приходится сталкиваться с важными ограниченными ресурсами, которые сложно оценивать в деньгах. Ниже приведены некоторые примеры.

- Ширина участка морского побережья, принадлежащего владельцу строящегося дома у моря.
- Вес полезной нагрузки в космическом корабле или в рюкзаке.
- Пропускная способность памяти в любой архитектуре вычислительной системы, разработанной в соответствии с принципами, выдвинутыми Фон Нейманом.
- Допуск синхронизации в системе GPS в наносекундах.
- Календарные дни, оставшиеся до момента осуществления проекта перехвата астероида.
- Пространство памяти резидентного ядра в проекте OS/360.
- Распределение времени в программе конференции в часах.
- Количество страниц в предложении на получение гранта или в журнальной статье.
- Потребляемая мощность спутника связи (и запас электроэнергии).
- Теплота рассеивания высокопроизводительной микросхемы.
- Количество пресной воды на сельхозугодьях западных штатов США.

- Количество часов обучения студента в расписании подготовки к экзаменам.
- Задачи, выдвинутые в уставе политической партии.
- Продолжительность кино- или видеофильма, измеряемая в секундах или количеством кадров.
- Продолжительность времени в часах, отведенного в сутки на модернизацию и техническое обслуживание путей лондонской подземки.
- Количество битов формата в архитектуре вычислительной системы.
- Часы или минуты, расписанные в плане вооруженного нападения.

Разновидности денежных затрат и других ресурсов, которые могут оцениваться в деньгах

Иногда техническое задание к проекту содержит явное указание, что одним из лимитированных ресурсов являются денежные затраты, но и в этом случае необходимо учитывать конкретные направления расходования денежных средств. Например, для персональных компьютеров, выпускаемых сотнями миллионов, доминирующей является стоимость производства. Что же касается суперкомпьютеров, которые выпускаются в количестве не больше нескольких десятков, то в их проектах доминирует стоимость разработки.

Но даже в этом случае, чтобы упростить расчеты, проектировщики очень часто измеряют объем лимитируемых ресурсов не в денежном выражении, а в виде каких-либо вспомогательных единиц. В частности, в строительном проектировании архитекторы часто разрабатывают техническое задание и даже эскизные проекты, используя в качестве нормированного ресурса площадь здания в квадратных метрах. Архитекторы компьютерных систем, как правило, оценивают площадь поверхности создаваемой микросхемы с учетом количества битов регистра, а также наличия применяемых уровней кеша.

Применение вспомогательных единиц измерения позволяет достичь нескольких преимуществ: обычно такие единицы проще в использовании; разработка проекта на их основе может осуществляться задолго до того, как будет определено денежное выражение вспомогательной единицы измерения; количественные значения, выраженные во вспомогательных единицах измерения, остаются сопоставимыми при переходе от одного проекта к другому. Неизменное применение одних и тех же вспомогательных единиц измерения позволяет воспользоваться опытом разработки предыдущих проектов, даже если денежное выражение вспомогательной единицы измерения будет изменяться или даже колебаться. К примеру, остаются неизменными нормы площади на каждого студента в аудитории.

Но применение вспомогательных единиц измерения может также завести в тупик, если ими будут по-прежнему руководствоваться по привычке даже после того, как они перестанут соответствовать действительности. В частности, многие

проектировщики микросхем по-прежнему руководствуются данными о площади поверхности микросхемы, несмотря на то, что во многих случаях важным критическим ресурсом становится длина внутренней проводной разводки или количество выводов.

Изменение значимости лимитируемого ресурса

Иногда внедрение новых технологий приводит к тому, что вместо одного ресурса важным становится другой. Не учитывая этого, можно по неосторожности попасть впросак. По мере повышения плотности элементов микросхем вместо площади поверхности микросхемы лимитирующим параметром стало количество контактов ввода-вывода, поэтому данный параметр перешел в разряд нормируемых. Но в последнее время во многих проектах микросхем приходится нормировать не количество выводов, а рассеяние энергии. Сеймур Крэй (Seymour Cray) однажды произнес знаменитую фразу: "Охлаждение — это ключ к проектированию суперкомпьютера"[2]. А Джин Амдал (Gene Amdahl) примерно в то же время сказал мне, что в его проекте фактором, лимитирующим быстродействие, является внешняя емкость микросхемы[3].

Некоторые специалисты считают, что в качестве показателя оценки сложности программного обеспечения, следовательно, его вероятного размера, должно применяться значение количества объектно-ориентированных классов, а не функциональных точек. Однако опытные консультанты Сюзанна и Джеймс Робертсон (Suzanne и James Robertson) указывают, что, по их мнению, функциональные точки все еще остаются самым надежным показателем оценки[4]. Эоин Вудс (Eoin Woods), опытный разработчик, высказался на эту тему так:

> *Многие стремятся определить значения двух показателей — количество предоставляемых функций и количество модулей, которые должны быть созданы для предоставления этих функций. Показатель количества функциональных точек остается удобным, поскольку вполне может служить для оценки предоставляемых функций, тогда как объем программного обеспечения, определяемый количеством строк кода, количеством классов и так далее, в значительной степени зависит от стиля программирования. Вполне возможно сокращение количества функциональных точек и уменьшение объема программного обеспечения наряду с расширением поддерживаемых функций [комментарий рецензента].*

Может оказаться так, что непосредственно в ходе проектирования будет определена необходимость выбрать в качестве исходного другой лимитируемый ресурс, просто потому, что этого требует дополнительно полученная информация. Операционная система OS/360 (в версии от 1965 года) проектировалась с учетом эксплуатации на компьютерах с объемами памяти от 32 Кбайт (это не ошибка — Кбайт, а не Мбайт) до 512 Кбайт и больше, т.е. с такими объемами, которые отличаются больше чем в 16 раз. Безусловно, определенный объем памяти

приходилось оставлять для прикладных программ, поэтому налагались очень жесткие ограничения на размер резидентного ядра операционной системы: при наименьшем объеме памяти он должен был составлять не больше 12 Кбайт. Тогда мы были уверены, что ограничительным лимитируемым ресурсом является объем памяти. Мы были неправы.

OS/360 была одной из первых операционных систем, в которой для хранения кода операционной системы можно было применять жесткий диск. В операционных системах предыдущих поколений для этой цели служили магнитные ленты или перфокарты. С того времени как системы с произвольным доступом стали резидентными, замечательным образом увеличилось количество функций, которые можно было реализовать с использованием ограниченного объема оперативной памяти. В силу вступило правило: чтобы вызвать на выполнение функцию, просто считайте фрагмент с ее кодом с диска. И действительно, каждый представитель коллектива системных проектировщиков начал во все большей степени руководствоваться этим правилом, учитывая то, что фрагменты кода должны были быть небольшими, не выходящими за пределы буферной области с объемом 1 Кбайт в наименьшей конфигурации.

Один из самых опытных наших разработчиков, Роберт Ратрофф (Robert Ruthrauff), на ранних этапах разработки проекта OS/360 приступил к созданию программы моделирования производительности, и ему, к счастью, удалось достаточно быстро ввести ее в действие. Первые результаты оказались ужасающими! На нашей второй по быстродействию компьютерной модели (Model 65) применяемая система программирования могла компилировать только пять операторов Fortran в минуту! В тот же день было принято такое решение: лимитируемым ресурсом проекта должны стать показатели доступа к диску, а не объем памяти в байтах[5].

Что из этого следует?

Если коллектив проектировщиков должен строить свою работу с учетом наличия лимитируемых ресурсов, то как должна быть организована эта работа?

Явное определение состава лимитируемых ресурсов

Вполне естественно, что руководитель проекта должен прежде всего определить полный состав целей и ограничений. За этим должно следовать явное определение лимитируемых ресурсов. Заслуживает внимания то, что в настоящей главе, как правило, определяются и обсуждаются лимитируемые ресурсы проекта, а не процесса проектирования. Например, распределение специалистов по ключевым направлениям всегда является важным фактором успеха проекта разработки, но качество самого проекта не должно зависеть от этого.

Многое может определяться тем, как составлен план проектных работ, но от этого чаще всего зависит ход самого проекта, а не полученный результат.

Например, при подготовке конкурентоспособного предложения по организации работ в условиях жестких временны*х* рамок часто выдвигается лозунг: "Мы сделаем наилучший проект, создание которого возможно за выделенное нам время". С другой стороны, если разрабатывается проект по предотвращению падения астероида на Землю, то лимитируемым ресурсом становится количество дней до этой катастрофы. Аналогичным образом, план проведения проектных работ может оказаться лимитируемым ресурсом, если задачей является выход на рынок первыми с полностью новым продуктом в условиях конкуренции с другими компаниями.

Широкий контроль

Весь коллектив должен постоянно иметь доступ к сведениям о том, каковыми являются текущие показатели измерения критических ресурсов. Например, при разработке компьютерной техники представители каждой группы и подгруппы проектировщиков должны знать, предположим, за какие пределы не должна выходить мощность, потребляемая разрабатываемым ими компонентом, или какое количество операций доступа к диску в процессе диалоговой обработки запросов является допустимым.

Централизованное управление

Каким бы ни был критический ресурс, руководитель коллектива всегда должен хранить в своем "запаснике" небольшой резерв, которым можно было бы распорядиться в крайнем случае, подобно военачальнику, который скрывает от противника несколько батальонов, чтобы бросить их в бой в самый разгар сражения[6].

Крайне важно, чтобы распределение и перераспределение ресурсов сосредоточивалось только в одних руках. Джерри Блау (Gerry Blaauw) блестяще подтвердил правильность этого требования, определяя количество битов в слове состояния программы при разработке архитектуры System/360. Этот показатель, наряду с пропускной способностью памяти и количеством битов в форматах команд, вошел в число лимитируемых ресурсов, распределяемых системными архитекторами. А поскольку Джерри обладал глубоким пониманием особенностей системы, проявлял предусмотрительную скупость и изобретательность при выборе альтернатив в пределах существующей архитектуры, то принятые им решения привели к созданию весьма эффективной архитектуры[7].

Кен Айверсон (Ken Iverson), который был объявлен победителем конкурса Turing Award за создание языка APL, прежде всего стремился к достижению концептуальной целостности, поэтому принял в качестве лимитируемого ресурса число применяемых различимых языковых концепций. К нему поступали бесчисленные предложения от коллектива разработчиков и от сообщества

пользователей, касающиеся внедрения новых конструкций, которые воплощали бы дополнительные концепции. Однако, благодаря тому что Кен обладал полным пониманием назначения создаваемого языка, проявлял предусмотрительную скупость и изобретательность при реализации решений в рамках существующих языков, он добился создания чрезвычайно изящного языка.

А в наши дни образцом полного владения предметной областью, неукоснительного стремления к сохранению единообразия проекта и такой же предусмотрительной скупости может служить Марисса Мейер (Marissa Meyer), которая, как дракон, стоит на страже, охраняя сложившийся облик сайта Google[8].

Примечания и ссылки

1. В книге [192] исследования в области применения лимитируемых ресурсов также рассматриваются как решающий фактор процесса проектирования (с. 143-144).
2. См. [158].
3. По материалам личного общения [примерно в 1972 году].
4. По материалам личного общения [2008 год].
5. В изящном небольшом компиляторе языка Fortran компании Digitek (1965 г.) использовалось очень компактное, специализированное представление для самого кода компилятора, поэтому внешнее ЗУ не требовалось. Время, затрачиваемое на декодирование этого представления в режиме интерпретации, десятикратно окупалось благодаря тому, что была полностью исключена необходимость использования лимитируемого ресурса — операций доступа к диску (см. [36], глава 6).
6. Сравнение проектировщика с военачальником, планирующим сражение, заслуживает самого серьезного внимания, ведь именно тот из них, кто способен быстрее других пересмотреть свои замыслы перед лицом новых фактов и ограничений, и добьется победы.
7. См. [25], раздел 12.4.
8. См. [120].

11

Полезные свойства ограничений

Форма освобождает.

КРЫЛАТОЕ ВЫРАЖЕНИЕ В СРЕДЕ ХУДОЖНИКОВ

But I need four walls around me to hold my life, to keep me from going astray.

JAMES TAYLOR, ПЕСНЯ "BARTENDER'S BLUES"

Продукт универсального назначения сложнее спроектировать, чем специализированный.

◀ *Скульптура Давида, которую Микеланджело высек из блока мрамора, считавшегося непригодным. iStockphoto*

Ограничения

Ограничения могут порождать дополнительные трудности, но могут также проявлять полезные свойства. Ограничения уменьшают область поиска для проектировщика. Благодаря этому проектировщики получают возможность сосредоточиться на более узком направлении и ускорить разработку. Многие ученики старших классов средней школы не любят писать сочинения на свободную тему и предпочитают получать более конкретные задания. Это означает, что после снятия всех ограничений задача подготовки сочинения становится сложнее, а отнюдь не проще.

Об этом знал Бах. Его биограф Уолф писал: "Бах с самого начала своей творческой деятельности стремился выйти за сложившиеся рамки, однако предпочитал работать над выполнением четко сформулированного задания и преодолевать реальные заключающиеся в нем сложности"[1].

Ограничения не только сокращают область поиска, но и бросают вызов проектировщику, часто стимулируя тем самым создание принципиально нового продукта. Достаточно вспомнить историю создания Микеланджело скульптуры Давида. Согласно легенде, блок мрамора, над которым стал работать Микеланджело, был оставлен Антонио Росселлини за 25 лет до этого как непригодный, поскольку в нем была трещина. Микеланджело принял вызов и в результате разработал концепцию создания скульптуры Давида, беспрецедентную в истории искусства. Невозможно оставаться равнодушным, изучая то, как этот гений справился с дефектами блока и сумел выработать принципиально новый художественный подход.

Еще одним ярким примером может служить восстановление лондонских церквей Кристофером Реном. Рен, получив задание перестроить 50 англиканских церквей, уничтоженных Великим пожаром в 1666 году, столкнулся с весьма серьезными ограничениями. Работая над восстановлением каждой церкви, он был вынужден исходить из того, на какой площадке она должна находиться и в какой среде, а иногда даже использовать существовавший ранее фундамент. Кроме того, он должен был учитывать, что алтари англиканских церквей ориентируются на восток, поскольку Христос обещал вернуться как "звезда светлая и утренняя"[2]. Нашим современникам очень интересно посетить сегодня все 27 церквей Рена, которые сохранились по прошествии столь длительного периода времени и даже пережили бомбардировки времен Второй мировой войны. Особенно занимательно видеть, как расположена каждая площадка и какие проблемы стояли перед архитектором, оценить ограничения при выборе расположения каждой церкви и узнать, какие решения Рен принимал в различных случаях.

Проектировщикам шоссе на участке Блю Ридж Парквей в горах штата Северная Каролина приходилось до минимума сокращать площадь опор, касающихся земли, чтобы до предела уменьшить отрицательное воздействие на экологию. В результате было создано весьма изящное сооружение[3].

Целесообразность ограничений

В заданиях на проектирование часто встречаются искусственно введенные ограничения, положительным свойством которых является то, что проектировщик вправе их ослаблять. В идеальном случая ограничения должны стимулировать проникновение в неизведанные уголки пространства проектирования, способствуя тем самым реализации творческого потенциала. Но набор ограничений может также загнать разработчика в такой тупик, что невозможно будет найти даже единственное решение по проектированию.

Поэтому необходимо тщательно проводить различия между следующими типами ограничений.

- Действительные ограничения.
- Устаревшие ограничения, которые когда-то были действительными.
- Ограничения, неправильно воспринятые как действительные.
- Намеренно введенные искусственные ограничения.

Неактуальные ограничения. Проектировщик, имеющий большой стаж работы, может оказаться в положении льва, привыкшего мерять шагами свою клетку, и продолжать по привычке подчиняться ограничениям, ставшим устаревшими в результате развития технологии. Мысли, высказанные в главе 9 книги [40], теперь воспринимаются как забавные. В этой главе рассматриваются методы сокращения объема программного обеспечения в целях размещения в ограниченном объеме памяти. Эти рекомендации были крайне необходимыми в 1965 году, к 1975 году стали менее важными, но многим программистам все еще приходилось работать над сокращением объема программного обеспечения. (Безусловно, характеристики объема памяти все еще остаются значимыми для встроенных вычислительных устройств, особенно тех, которые в настоящее время становятся сердцем чрезвычайно развитых систем, называемых нами по привычке “сотовыми телефонами”.)

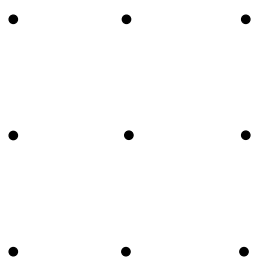


Рис. 11.1. Задача, в которой приходится сталкиваться с неправильно воспринятым ограничением: провести непрерывную линию, состоящую не более чем из четырех отрезков, через все девять точек.

Неправильно воспринятые ограничения. Эта проблема является более тонкой. Классический пример приведен на рис. 11.1. (Решение дано на рис. 11.5[4].) Еще одним примером могут служить ограничения, о которых шла речь в главе 3.

Для перемножения двух матриц 2 × 2 достаточно семи операций умножения вместо восьми, но, чтобы это понять, необходимо отбросить неправильно воспринятое ограничение, согласно которому должны использоваться векторные операции.

Еще одним примером того, сколько неприятностей может принести неправильно воспринятое ограничение, может служить проектирование компьютерной системы IBM 9020 для Федерального управления авиации (Federal Aviation Administration — FAA). По заданию FAA работали системные архитекторы MITRE Corporation, которые поставили перед собой задачу создания супернадежной системы. Они определили конфигурацию, которая показана на рис. 11.2.

Внешне она выглядит весьма неплохо. Соответствующий запрос на предложение был направлен специалистам компании IBM, которые пришли к выводу, что новая технология System/360 на полуинтегральных схемах очень хорошо показала себя согласно требованиям к надежности устройств.

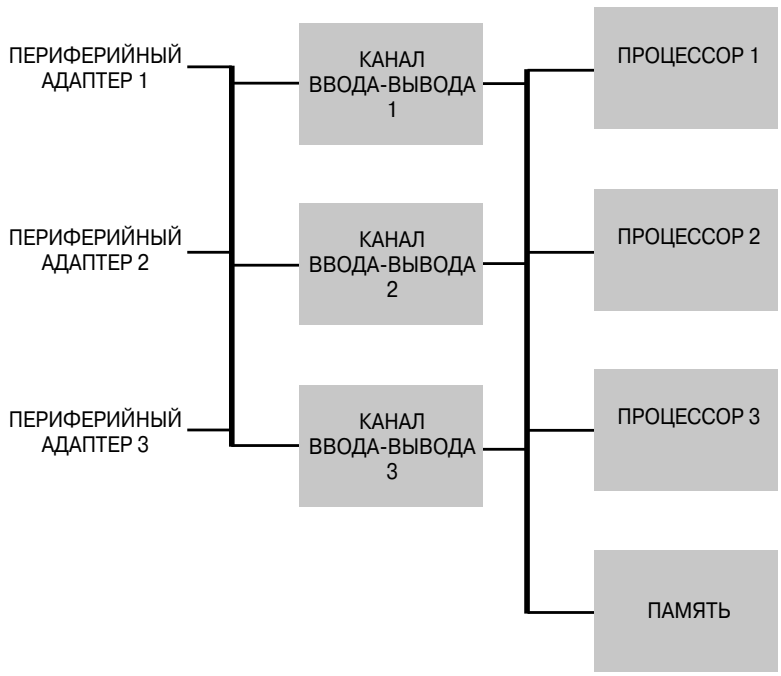


Рис. 11.2. Конфигурация системы, характеризующаяся использованием процессора с тройной модульной избыточностью и средств ввода-вывода, которая была предложена для реализации задания FAA в 1965 году

Оказалось, что компьютер среднего класса S/360 модели 50 более чем соответствует требованиям и по скорости, и по надежности, предъявляемым к процессорам. А требованиям к контроллеру ввода-вывода идеально соответствовала система ввода-вывода той же модели 50, реализованной с такими же устройствами памяти и информационными каналами, как процессор, но с другим микропрограммным обеспечением.

Таким образом, специалисты компании IBM спроектировали систему, показанную на рис. 11.3.

Тщательный анализ показал, что конфигурация, приведенная на рис. 11.3, вполне соответствует всем требованиям к производительности и надежности системы. Но, к сожалению, этот проект был отклонен.

Он не имел топологии, указанной в конфигурации. Системные архитекторы компании MITRE ошибочно выдвинули собственный вариант топологии и настаивали на его принятии как существенного ограничения, хотя в действительности то, что они ставили во главу угла, относилось к категории функциональности и надежности, но не топологии. В связи с этим корпорацией IBM было выдвинуто предложение, разработана и реализована конфигурация, которая показана на рис. 11.4. Но беспристрастный анализ надежности показал, что фактически эта конфигурация является значительно менее надежной по сравнению с той, что приведена на рис. 11.3, поскольку в ней применяется в два раза больше компонентов и еще больше разъемов, из-за которых могут возникать отказы. Но этот последний вариант полностью соответствовал заданным ограничениям!

Рис. 11.3. Система FAA в том варианте, который был первоначально предложен для реализации на основе компьютеров System/360 модели 50

Однако в последнем случае магическими оказались ценовые показатели. Не учитывая полностью реальные затраты, правительство заплатило за систему на рис. 11.4 столько, сколько должно было бы заплатить за систему на рис. 11.3. Корпорации IBM этот контракт действительно был на руку! Безусловно, неодинаковыми были бы и расходы на электроэнергию, кондиционирование и техническое обслуживание, связанные с эксплуатацией системы на всем протяжении срока ее службы.

Из этого следует извлечь урок: формулируя задание на проектирование некоторого продукта, необходимо указывать, какие свойства он должен иметь, а не то, каким образом должны быть реализованы эти свойства.

Если в качестве ограничений заданы способы реализации, то заранее отсекаются лучшие решения. Ради достижения высокого качества продукта и соблюдения интересов пользователя проектировщик должен вести борьбу против ложных ограничений!

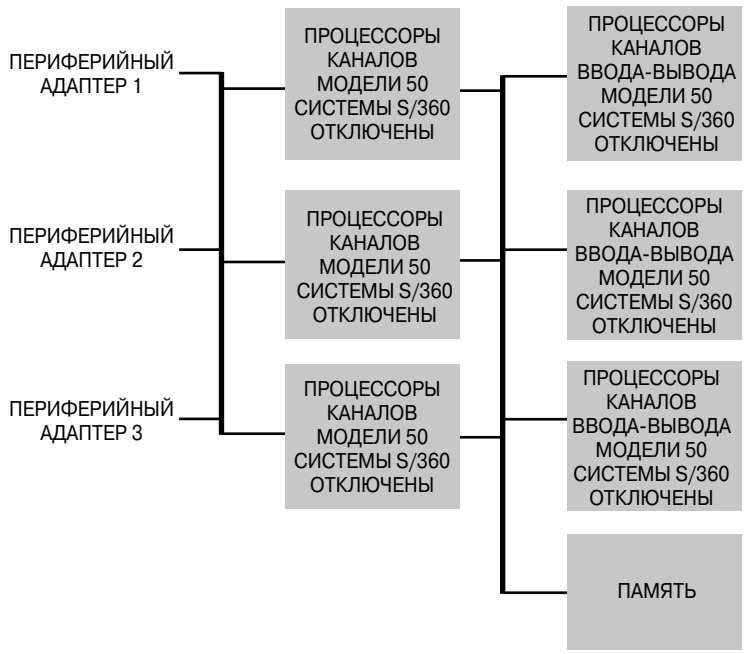


Рис. 11.4. Система FAA на основе IBM 9020 в том виде, в каком была выполнена ее поставка

Парадокс проектирования — более высокая сложность проектирования продуктов универсального назначения по сравнению со специализированными

Как уже было сказано в этой книге, самой сложной задачей при проектировании является точное определение того, что должно быть спроектировано. А в начале этой главы было указано, что ограничения обладают полезными свойствами и способны сделать задачу проектирования более простой, а не более сложной. На основании этого можно сделать вывод, что выбор для проектирования продуктов более специализированного назначения также способствует упрощению задачи проектирования.

На первый взгляд это утверждение кажется неправильным. В качестве примера сравним два задания: "Спроектировать дом площадью 1000 квадратных футов" и "Спроектировать дом 1000 квадратных футов для семьи с двумя детьми противоположного пола, расположенный в городе Чапел-Хилл, штат Северная Каролина, с выходом на север". Можно сразу же подумать, что первое задание проще.

В определенном смысле, безусловно, без дополнительных требований легче заниматься проектированием, но труднее противостоять критике. Если нет ограничений, то нет и критериев, позволяющих отстаивать превосходство проекта. Вообще говоря, проще сделать посредственный универсальный проект, чем посредственный проект специального назначения.

Тем не менее в последнем случае задача разработчика становится проще, если проект должен иметь не среднее, а превосходное качество. Процесс проектирования всегда начинается с того, что разработчик выявляет и конкретизирует цели и ограничения. Прежде всего необходимо сузить пространство проектирования. Чем более четко обозначена намеченная цель, тем больший путь к ее достижению уже проделан.

Переход от создания проектов общего назначения к специализированным проектам. Важно понять, как перейти от проекта общего назначения к специализированному. Рассмотрим пример разработки архитектуры вычислительной системы. Технология производства универсальных компьютеров достигла значительного развития. Подготовлены к производству и выведены на рынок сотни вариантов архитектур таких компьютеров. Всем известно, какие функции возлагаются на компьютеры общего назначения. Хороший проектировщик может подготовить техническое задание на новую модель универсального компьютера за несколько дней, поскольку не составляет особого труда принять определенный ряд архитектурных решений, перечисленных ниже.

- Форматы команд.
- Адресация и управление памятью.

- Типы данных и их представления.
- Набор операций.
- Порядок следования команд.
- Средства управления.
- Система ввода-вывода.

Проектирование архитектуры специализированной ЭВМ. С другой стороны, при проектировании специализированной ЭВМ приходится с самого начала выполнять значительный объем дополнительной работы. Необходимо тщательно изучить приложение, для эксплуатации которого будет применяться ЭВМ. В чем состоят особенности этого приложения? Каково распределение операций по частоте их выполнения? Каковы взвешенные оценки требований к характеристикам, предъявляемых заказчиком: производительность, стоимость, надежность, вес? В действительности должна быть разработана полностью определенная модель эксплуатации приложения на ЭВМ.

Проектирование архитектуры общего назначения, позволяющей достичь превосходства перед конкурентами. С другой стороны, если при разработке архитектуры общего назначения предъявляется требование создать конкурентноспособный продукт, то должна быть подготовлена модель работы пользователя в будущей системе, а эта задача — далеко не простая. При этом фактически необходимо изучить весь набор приложений, намеченных для эксплуатации, и для каждого из приложений определить конкретные требования и условия работы. В компьютере для научных вычислений основную нагрузку создают операции матричной алгебры и функции для решения дифференциальных уравнений в частных производных; при инженерных расчетах приходится прилагать основные усилия в части статистической обработки данных и вычисления формул; при выполнении запросов к базе данных важной становится оптимизация операций доступа к диску. Должно быть изучено каждое приложение.

После этого необходимо определить наиболее важные взвешенные характеристики по следующим аспектам.

- По всему набору приложений.
- По всему множеству намеченных компьютерных реализаций.
- По тому периоду времени, возможно, измеряемому десятилетиями, в течение которого должна применяться создаваемая архитектура[5].

Кроме того, на протяжении всего процесса проектирования достигнутые результаты должны быть проконтролированы на предмет того, соответствуют ли они прогнозируемым требованиям каждой категории пользователей. После завершения проектирования и разработки прототипа необходимо выполнить проверку прототипа с участием пользователей, относящихся к каждой категории.

Именно на это я всегда нацеливаю своих студентов в проводимом мною расширенном курсе архитектуры вычислительной системы, рассказывая, как разрабатываются проекты архитектур специального назначения. И я не принимаю от студентов курсовые работы, наполненные вымыслами и пустыми словами; результаты анализа характеристик приложений и пользователей должны быть точными. Тем не менее мои студенты часто замечательно справляются с подобным заданием, притом что трудно представить себе, как за отведенное им время можно спроектировать без заданных ограничений превосходную универсальную архитектуру. Очевидно, что задача проектировщика становится намного проще, если реальный проект разработки продукта общего назначения подкрепляется всесторонним анализом.

Проектирование программного обеспечения. Выше шла речь о том, какие парадоксальные зависимости связывают разработку универсальной и специализированной архитектур. Аналогичные закономерности наблюдаются и в области программного обеспечения. Достаточно представить себе, насколько взвешенными должны быть характеристики выразительной мощи, общности и экономичности, которыми должен обладать универсальный язык программирования, и становится ясно, что задача проектирования языка программирования специального назначения по сравнению с этим гораздо проще. В проекте специального назначения можно достаточно легко определить минимально необходимый набор функций.

Строительное проектирование. Парадоксальные зависимости между универсальным и специализированным проектированием наблюдаются также в области строительства. Гораздо легче превосходно спроектировать спальню, чем создать заслуживающий высокой оценки проект гостиной, поскольку гостиная предназначена для выполнения намного большего количества функций, требует рассмотрения многочисленных сценариев пребывания в ней людей, а также допускает применение гораздо большего количества вариантов меблировки[6].

Аналогичным образом легче спроектировать специализированную лабораторию, чем подготовить проект общего вестибюля в здании факультета информатики.

Сеть взаимосвязей

Следует всегда помнить, что ограничения могут принести проектировщику большую пользу, поэтому, если задача первоначально кажется не укладывающейся ни в какие рамки, проектировщик должен посвятить больше времени размышлениям о том, что от него действительно требуется и каковы модели пользователя и использования, а это может помочь ему найти некоторые сужающие ограничения, приемлемые и для проектировщика, и для пользователя.

Примечания и ссылки

1. См. [228], с. 387. Если Бах не видел достаточно жестких ограничений, которые налагало бы задание или квалификация имеющихся исполнителей, то сам иногда принимал довольно искусственные ограничения, которые могли бы заставить его проявить весь свой творческий потенциал. Примером может служить неоднократное повторение знаменитого мотива Баха (последовательности нот "си-бемоль, ля, до, си"). Но я не рекомендую принимать для стимулирования творческого потенциала искусственные ограничения в области разработки технического или программного обеспечения.
2. Откровения 22:16.
3. См. `http://www.blueridgeparkway.org/linncove.htm`; в книге взята за основу версия этой страницы от 18 июля 2009 года.
4. Решение задачи приведено ниже.

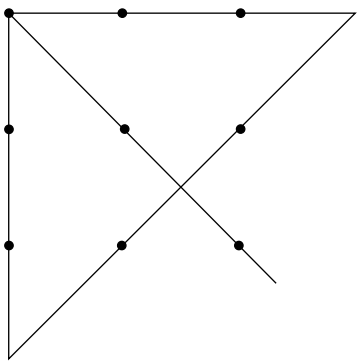


Рис. 11.5. Решение задачи с девятью точками

5. Первоначально предполагалось, что компьютерная архитектура System/360 рассчитана на срок эксплуатации 25 лет, при том условии, что будет происходить смена поколений средств реализации (см. [39]). Но мы тогда не могли предположить, что тенденции, определяющие более продолжительный срок применения языков программирования, архитектур вычислительных систем и архитектур операционных систем, окажутся намного более сильными. В компьютерах z/90 корпорации IBM по прошествии 45 лет продолжается использование архитектуры System/360, и не заметно признаков ее устаревания. Еще одним надежным свидетелем устойчивости тенденций к сохранению жизнеспособности является язык Fortran (созданный в 1956 году).
6. Для дома Кэнвуда в местечке Хемпстед неподалеку от Лондона архитектор Роберт Адам (1728–1792 гг.) спроектировал каждую деталь обстановки, вплоть до дверных ручек (путеводитель по дому Кэнвуда, см. [234]).

TheDesignofDesign.indb 175 30.10.2012 15:58:30

TheDesignofDesign.indb 176 30.10.2012 15:58:30

12

Эстетика и стиль в техническом проектировании

Firmitas, utilitas, venusitas (надежность, полезность, привлекательность)

См. [216]

Стиль — это одежда мысли, и удачно выраженная мысль, как хорошо одетый человек, приобретает большое преимущество.

ЛОРД ЧЕСТЕРФИЛД
(См. [57], LETTER CCXL, 361)

◂ *Монтичелло: Адаптация Джефферсона адаптации Палладием римского стиля. iStockphoto*

Эстетика в техническом проектировании

Древнеримскому архитектуру Витрувию принадлежит знаменитое утверждение, согласно которому качественная строительная архитектура позволяет достичь трех целей: "надежности, полезности и привлекательности"[1]. В этом изречении привлекательность приравнивается к надежности и полезности, хотя чисто с утилитарной точки зрения вполне достаточно было бы добиться только надежности и полезности.

Но правоту Витрувия подтверждает весь опыт человечества. Мы всегда выражаем желание, чтобы окружающие нас жилые и общественные здания удовлетворяли духовную потребность в красоте, и почти всегда готовы приложить к этому дополнительные усилия.

Пещерный человек разрисовывал стены своей пещеры; коренные американцы украшали вигвамы, в которых они жили, шкурами бизонов. Доисторические племена, обитавшие на Альбионе, украшали свою глиняную посуду, ставя на ней отметки ногтями. В течение столетий строительные конструкции и жилые помещения украшались лепными фигурами, резьбой по дереву, витражами, панелями и разноцветными драпировками. Стандарты красоты были во многом разными, изменяясь в зависимости от национальных особенностей и конкретной эпохи, но внешняя красота всегда была одной из целей при проектировании зданий.

Какую же роль играют красота и эстетика в техническом проектировании? Автомобили, самолеты и суда имеют физические формы, поэтому могут стать предметами, привлекательными для глаз. Но на этом вопрос не исчерпывается. Можно стать свидетелем того, как специалисты характеризуют разные языки программирования как "изящные" или "уродливые". Если речь идет о "превосходном" компьютере, это отнюдь не означает, что дается характеристика визуального оформления его как производственного изделия; скорее данное мнение относится к определенному свойству его логической структуры.

Возвращаясь взглядом к некоторым используемым нами вещам, мы снова и снова испытываем удовольствие. Но бывает и так, что предметы нашего пользования с эквивалентным назначением и столь же полезные не обладают такой привлекательностью, и каждый раз мы испытываем раздражение. Привлекательность может проявляться в сфере визуального, слухового, вкусового восприятия или может быть связана с осязательными ощущениями. Привлекательность может даже выражаться чисто в интеллектуальной сфере, как в примерах с красивым доказательством или удачно выраженной мыслью. В повседневном языке эта истина находит много разных выражений — иногда приходится слышать, как программу называют "изящной", и слушатели понимают, что под этим подразумевается[2].

В чем выражается красота логических рассуждений?

Краткость

Чтобы рассуждение было "изящным", оно прежде всего должно быть кратким. В одном из определений изящного изложения идей в математике сказано, что "оно выражает многое с помощью немногих элементов".

Это, безусловно, относится и к доказательствам. Но, к сожалению, слишком многие авторы учебников по математике полагают, что данное правило распространяется также на объяснения. Несомненно, не может считаться удовлетворительным такое объяснение, в котором высказано предельно мало на данную тему; для достижения полной ясности необходимо развертывать каждую мысль и иллюстрировать ее примерами.

Проектировщики языков программирования всегда рассматривают краткость как один из руководящих принципов. Разумеется, и при проектировании аппаратных средств необходимо придавать бо́льшую значимость достижению краткости, применению меньшего количества элементов[3].

Например, ядро языка Lisp является весьма небольшим, но при этом предусмотрены все необходимые условия для достижения его расширяемости и компонуемости, что способствовало созданию очень мощного языка. С другой стороны, язык Visual Basic изначально является значительно более сложным, но задача его расширения становится намного затруднительнее.

Другие соображения, касающиеся краткости. Тем не менее краткость не может служить исчерпывающим критерием. Например, что касается аппаратных средств, то добавление к проекту компьютера с виду "ненужного" компонента, такого как индексные регистры, позволяет существенно повысить его производительность и улучшить соотношение "производительность/стоимость". Такое расширение базового языка Lisp, как Common Lisp, позволяет программистам проще справляться со своими задачами.

В свое время Ван дер Пол (Van der Poel) спроектировал компьютер, в котором была реализована только одна машинная операция[4]. В этом компьютере любая инструкция выполнялась с применением этой единственной операции. Ван дер Пол сумел продемонстрировать, что предложенной им операции вполне достаточно; его компьютер был способен выполнять все те же действия, что и любой другой. Тем не менее задача разработки программ для его компьютера была очень сложной. Как ни странно, но для некоторых программистов применение его компьютера оказалось привлекательным, хотя этот процесс можно было сравнить с получением удовольствия от решения кроссворда — намеренно усложненной конструкции, не имеющей какой-то реальной пользы.

На практике для эффективного использования машины Ван дер Пола требовалось освоение целой библиотеки стандартных конструкций, которые было бы

сложно придумать самому. Со временем Ван дер Пол представил в виде готового кода некоторые подпрограммы и макрокоманды, в результате чего был создан своего рода язык высокого уровня, которым был способен овладеть гораздо более широкий круг пользователей.

Аналогичным образом развивался APL — весьма изящный и мощный язык программирования. Язык APL способен воплотить в своих расширениях целый ряд различных стилей программирования, начиная с ясного и простого и кончая замысловатым и сверхлаконичным. Безусловно, операторы языка APL имеют весьма несложные определения, но реализации часто применяемых функций основаны на таких синтаксических конструкциях, которые не сразу приходят в голову. Примеры приведены в книге [25], раздел А.6. Например, в программе А-41, "Приведение к четному", число делится на два, дробная часть отбрасывается, после чего производится умножение на два.

В некоторых кругах даже стало модно соревноваться в том, сколько функций удастся упаковать в одну строку APL, пусть даже хитроумную и непонятную на первый взгляд. "Однострочники" рекламируются и публикуются с ликованием и гордостью. Но и это занятие опять-таки можно скорее сравнить с соревнованиями по решению кроссвордов, а не считать его одним из направлений усовершенствования проектирования. Назначение языков программирования состоит в том, чтобы упрощать написание (и гораздо более часто — чтение) программ, а не становиться источником проблем[5].

Структурная ясность

Критерий краткости не может стать приемлемым еще по одной причине. Кроме всего прочего, приходится стремиться к достижению определенной простоты в языке программирования или в архитектуре вычислительной системы. Каждый пользователь должен находить прямой путь между тем, какие задачи он хочет решить на компьютере, и тем, как в действительности он будет эти задачи решать.

Естественные человеческие языки, которые также можно рассматривать как предназначенные для удовлетворения многих реальных потребностей, отнюдь не обладают краткостью. Клод Шеннон в своей книге [189] показал, что избыточность английского языка составляет приблизительно 50%[6]. Но даже при такой избыточности своего естественного языка люди не могут обойтись без метафорических высказываний, так же как и в языках программирования.

Структура. Для воплощения критерия изящества в техническом проектировании требуется, чтобы основная структурная концепция проекта стала явно очевидной; если же замысел проекта не может быть непосредственно выявлен путем логических рассуждений, то должен быть просто описан.

Метафора. Созданию изящного и постижимого проекта способствует применение знакомых и простых метафор, особенно в части разработки пользовательских

интерфейсов к проектируемому объекту. Одним из замечательных примеров является воплощение понятия "рабочего стола" в операционной системе Macintosh. Еще одним примером может служить понятие "электронной таблицы", впервые реализованное в программе VisiCalc, которая теперь заменена программным обеспечением Excel и Lotus.

Непротиворечивость

Мы с Джерри Блау много времени уделили изучению следующего вопроса: что представляет собой качественная архитектура вычислительной системы? Со временем стало ясно, что рассмотрение этой темы становится проще, если вопрос формулируется в таком виде: что такое приемлемый технический проект? В следующем подразделе кратко представлено содержимое раздела 1.4 нашей совместной с Джерри книги [25][7].

Что представляет собой качественная архитектура вычислительной системы?

Архитектура, которая не обеспечивает поддержку необходимых функций, является неприемлемой. Но она все равно может быть недостаточно совершенной, даже если в ней предусмотрены все требуемые функции. Еще один неправильный подход может состоять в применении настолько сложной архитектуры, что ее становится трудно изучать, а также запоминать, какие функции в ней реализованы и каким правилам они подчиняются. Конструкции, которые являются удобными в использовании, часто называют *совершенными*.

Мы с Джерри Блау считаем, что в основе всех принципов качества лежит гармоничность. Качественная архитектура является гармоничной в том смысле, что даже частичное ознакомление с системой, в которой воплощена эта архитектура, позволяет предположить, как устроена остальная часть системы[8].

Например, зная лишь о том, что в наборе операций компьютера предусмотрена операция извлечения квадратного корня, можно почти полностью представить себе, как определена эта операция. Для нее, как и для других арифметических операций с плавающей запятой, должны быть предусмотрены одинаковые форматы данных и команд. Точность, диапазон, округление и значность чисел должны быть сравнимыми с параметрами результатов других операций. Даже при попытке извлечения квадратного корня из отрицательного числа необходимо предусмотреть активизацию исключительных ситуаций и их обработку по аналогии с ситуациями деления на нуль.

Тем не менее может оказаться, что в действительности трудно судить о том, является ли архитектурное решение на самом деле непротиворечивым. Что касается архитектуры вычислительной системы, то в качестве некоторых критериев могут служить краткость описания, простота формирования кода и применимость для многих реализаций.

Производные принципы. Из непротиворечивости вытекают три основных принципа проектирования: ортогональность, уместность и общность.

Ортогональность: отсутствие ограничительных связей между независимыми компонентами. Изменение в одной ортогональной функции не должно иметь какого-либо видимого эффекта, обнаруживаемого в любой другой функции из того же набора функций. Например, в число функций будильника могут входить подсветка циферблата и звонок. Принцип ортогональности был бы нарушен, если бы звонок срабатывал только при условии включения подсветки. В нашей совместно написанной с Джерри книге мы приводим много примеров нарушения принципа ортогональности в архитектурах вычислительных систем.

Уместность: отказ от реализации ненужных функций. Должны быть реализованы только необходимые функции, каждая из которых удовлетворяет какому-то из основных требований. Например, назначению автомобиля соответствуют такие функции, как рулевое управление, регулирование скорости, освещение и очистка ветрового стекла.

Противоположностью уместности является избыточность. В качестве примера можно привести переключение передач в автомобиле. Вождение автомобиля возможно и без переключения передач; этот дополнительный компонент, предоставленный в распоряжение водителя, стал следствием общепринятой реализации.

Примером уместности в аппаратных средствах компьютеров является уникальная интерпретация нуля, которая характеризуется представлением в поразрядном коде с дополнением до двух. В отличие от этого и представление в прямом коде со знаком, и представление в поразрядном коде с дополнением до единицы предусматривает отображение нуля со знаком, а это вносит различие, являющееся избыточным. Чем больше правил приходится формулировать для описания того, как должно обрабатываться нулевое значение со знаком в арифметических операциях и операциях проверки на нуль, тем чаще возникают побочные эффекты при использовании этих операций.

Критерий краткости является одним из следствий применения критерия уместности. Еще одним является прозрачность, выражаемая как свойство реализации функции, согласно которому в этой реализации отсутствуют какие-либо видимые побочные эффекты. Например, если в компьютере применяется канал передачи данных с конвейерной организацией, это должно оставаться невидимым для программиста.

Общность: отсутствие ограничений на то, в каких областях применяется функция. Общность — это возможность использовать одну и ту же функцию для многих назначений. В общности подхода к разработке выражается терпимость проектировщика, характерная для его профессии, и его уверенность в том, что находчивость будущих пользователей его продукта превзойдет все мыслимые рамки, а потребности могут измениться, выходя за пределы любых возможностей к предсказанию. Проектировщик должен избегать соблазна ограничивать функцию своими собственными представлениями о ее назначении. Если вы не уверены в чем-то до конца, предоставьте другим свободу выбора.

В процессоре Intel 8080A предусмотрена операция, обозначенная как Restart. В действительности эта операция была предназначена для перезапуска процессора после прерывания. Но она была спроектирована со столь значительной общностью, что чаще всего стала применяться для возврата из подпрограмм.

Основными направлениями достижения общности являются открытость (резервирование пространства для будущих разработок), полнота наборов функций, декомпозиция функций на ортогональные компоненты и компонуемость.

Дополнительные преимущества единообразия

Применяя продукт, одной из характеристик которого является единообразие, пользователь приобретает уверенность в себе и получает возможность самообразования, поскольку сам продукт укрепляет и подтверждает его ожидания. Кроме того, благодаря единообразию разрешается конфликт между удобством в использовании и легкостью в изучении. Достижение легкости в изучении требует применения простой архитектуры, что можно видеть на примере арифметики с фиксированной запятой. А для обеспечения удобства в использовании требуется сложная архитектура, как показывает применение арифметики с плавающей запятой. Проектировщик, принимая решение о том, что операции с фиксированной запятой должны стать подмножеством операций с плавающей запятой, тем самым в значительной степени облегчает понимание пользователем всей созданной архитектуры.

Стиль в техническом проектировании

Услышав любой фрагмент классического музыкального произведения, хорошо подготовленный слушатель обычно может догадаться, в каком столетии написана эта музыка, и часто даже угадать имя композитора, несмотря на то, что он впервые прослушивает это произведение. Разглядывая незнакомую картину, мы часто можем утверждать: “Похоже на Рембранта” или “Это — голландский Золотой век”. Представители британских вспомогательных служб WREN во время Второй мировой войны научились узнавать характерные признаки “морзянки”, передаваемой журналистами стран антигитлеровской коалиции, поэтому могли следить за тем, как движутся отдельные воинские подразделения, за которыми были закреплены эти журналисты. Характерными чертами обладают также мосты, автомобили, самолеты и компьютерные архитектуры. Например, определенные модели компьютеров “выглядят так”, как будто их спроектировал Сеймур Крэй (Seymour Cray).

Одним из того, что способствует достижению привлекательности, является так называемый *стиль* (попытку определения этого понятия мы предпримем ниже). Созданию привлекательного образа способствуют следующие два компонента стиля: единообразие применяемого стиля и сами изначальные возможности стиля. Ни в архитектуре, ни в музыке, ни в кулинарии, ни в вычислительной технике

невозможно выработать стиль, который был бы привлекательным для всех без исключения; но отсутствие единообразного стиля не нравится никому[9].

Выше в данной книге я уже отстаивал такую точку зрения, что концептуальная целостность — самый важный атрибут проектирования. Безусловно, наиболее важной составляющей целостности проекта является его общая структура — сам скелет проекта. Но не менее важно придерживаться единообразного стиля в деталях, так сказать, во внешнем облике проекта.

Выбранный стиль должен неуклонно соблюдаться в каждом компоненте, даже несмотря на то, что от "оформления" компонентов, на первый взгляд, не зависит концептуальная целостность продукта. Продукт, выдержанный по стилю, не только является привлекательным, но и способствует улучшению общего восприятия проекта. В свою очередь, хорошо воспринятый продукт легче изучать, проще использовать, удобнее выводить из эксплуатации после завершения срока службы, легче обслуживать, удобнее расширять его функции.

Соблюдение правильно выбранного стиля важно при проектировании во всех областях и во всех жанрах.

Что такое стиль?

В чем же именно состоит характерный способ организации работы, который позволяет легко понять, что к данному продукту приложил руку мастер? Этот вопрос сложнее, чем кажется на первый взгляд.

Определения. В оксфордском словаре английского языка (Oxford English Dictionary) понятие стиля, в том смысле, который нас интересует, определено следующим образом.

> *14. Свойства литературного произведения, которые относятся к форме или выражению, а не к сути излагаемой мысли или темы.*
>
> *21. Конкретное выражение или форма искусного построения, выполнения или выработки; способ, с применением которого создано произведение искусства, рассматриваемый как характерный для отдельного художника, его эпохи и места деятельности.*

Пересмотренный полный словарь Вебстера (Webster's Revised Unabridged Dictionary) в издании 1913 года содержит следующее определение.

> *4. Способ представления, особенно в музыке или в другом виде искусства; характерный или специфический способ разработки идеи или достижения результата.*

В работе [3] стиль определен таким образом.

> *Стиль как выражение личных и профессиональных решений проектировщика — это средство, позволяющее уменьшить количество неопределенностей, с которыми приходится сталкиваться в процессе проектирования.*

Характерные особенности проработки деталей. Можно заметить, что различные работы одного и того же представителя мира искусства отличаются, допустим, по сюжетам картин, по жанрам или темам сочиняемой музыки, но остаются подобными благодаря наличию узнаваемого стиля. Точно так же, например, церковь Oak Park, возведенная архитектором Фрэнком Ллойдом Райтом (Frank Lloyd Wright), имеет определенные различия в композиции и расположении по сравнению с церквами других архитекторов, но больше напоминает жилые дома, построенные Райтом, по общему облику, применению архитектурных деталей и украшений, а также по палитре красок. Каким бы ни был стиль, он в большей степени затрагивает детали проекта, чем касается главного назначения или побудительной потребности в создании продукта[10].

Гипотеза: минимизация умственных усилий. При любом проектировании и при любой творческой работе приходится принимать сотни решений, касающихся мельчайших деталей, которые можно назвать *решениями на микроуровне*, или *микрорешениями*. Создается впечатление, что привычки и традиции — это механизм, с помощью которого люди экономят умственные усилия, снимая с себя часть бремени принятия решений в повседневной жизни. А если соблюдение традиций — это действительно врожденный человеческий признак, то он, безусловно, переносится и на нашу творческую деятельность. Если не обнаруживаются существенные причины поступать иначе, мы каждый раз в аналогичных обстоятельствах принимаем одни и те же микрорешения одним и тем же образом. Неизменное принятие микрорешений из одного и того же набора становится характерным для нашей работы и придает ей такую особенность и различимость, благодаря которой результаты работы становятся узнаваемыми на фоне результатов, достигнутых другими.

Единообразие всех микрорешений. Можно вполне рассчитывать на то, что принятые микрорешения должны быть единообразными не только во времени, но и в пространстве подобных решений. Что же касается взаимосвязанных микрорешений, то на их принятие, как правило, влияют одинаковые факторы, а человек, который прежде уже сталкивался с такими факторами, вполне естественно, будет оценивать их так же, как раньше.

Например, для Фрэнка Ллойда Райта (Frank Lloyd Wright) характерно использование прямых, а не кривых линий во всем выполненном им художественном оформлении, а также в архитектуре возведенных им зданий. Сеймур Крэй (Seymour Cray) неизменно стремился в каждом новом проекте достичь наивысшей производительности, но мало заботился о совместимости с ранее созданными им компьютерами.

Четкость стиля. По такому признаку, насколько проектировщику действительно удается достичь единообразия во всем диапазоне принятых им макро- и микрорешений, мы можем судить о том, насколько четким является его стиль, подразумевая под этим, что для описания стиля не нужно много слов. Из этого следует, что четкий стиль является более легко узнаваемым.

Четкостью стиля может характеризоваться даже такая причудливая архитектура, как барокко. Архитектура сооружений Райта является не только четкой, но и лаконичной, однако это — разные свойства.

Если не удается достичь единообразия проектных решений, то стиль конечного продукта становится плохо узнаваемым или невыразительным. Так или иначе, единообразие приводит к четкости, а четкость порождает привлекательность[11].

Я всегда пользуюсь таким рабочим определением:

Стиль — это совокупность различных повторяющихся микрорешений, каждое из которых принимается одинаковым образом, когда в нем возникает необходимость, даже притом, что контекст принятия решения каждый раз может быть иным.

Кроме того, принятие взаимосвязанных микрорешений должно происходить одинаковым образом.

Стиль, бесспорно, имеет право на существование. Прослушав несколько музыкальных аккордов, мы можем узнать, принадлежит ли это произведение Баху, Моцарту или Шуберту. На выставке знаменитых работ может быть собрано много картин Рембранта, а также других картин, которые когда-то были приписаны Рембранту, а теперь считаются созданными в его манере. Глубокое изучение стиля этого художника со временем позволило экспертам провести различие между теми и другими, но удивительно также то, что на это оказались способными даже непрофессионалы[12].

Специалисты, достаточно хорошо владеющие знаниями в области компьютерных наук, могут легко отличить компьютер Сеймура Крэя от компьютера, который разработал Джерри Блау или Гордон Белл[13]. Знаменитые "Заметки федералиста" были изданы без подписей авторов, но со временем удалось идентифицировать всех авторов по особенностям свойственного им прозаического стиля[14]. Программисты могут узнавать код, написанный друг другом. К.С. Льюис (C. S. Lewis) утверждал, что в чудесах, совершенных Иисусом, достоверно выражаются признаки творческого стиля Всевышнего[15,16].

Свойства стилей

Независимо от информационной среды проектирования стили имеют некоторые общие свойства.

Высокая трудоемкость работы, связанной с определением стиля. Во-первых, для выработки четкого стиля необходимо провести значительный объем работы по определению требований к стилю. Например, учебник The Chicago Manual of [English prose] Style, посвященный описанию стиля английской прозы, имеет объем 984 страницы. В словаре [93] Фоулера определение надлежащего использования артикля "the" занимает приблизительно 2800 слов. Прежде чем научиться сразу же узнавать стиль Баха, необходимо многое узнать о творчестве этого композитора.

Иерархическая структура описания стиля. Во-вторых, любое описание, характеризующее стиль (явно или неявно), по существу должно быть иерархическим. Рассмотрим характерные особенности английской прозы.

- Диалект и произношение.
- Личностная окраска, жанр, непосредственность описания, живость цвета, теплота тона.
- Равновесие и ритм — просодия.
- Словарный запас — например, относительное количество местоимений.
- Пунктуация.
- Композиционное расположение — шрифт, интервал и т.д.

В частности, при создании композиционного оформления англоязычного оригинала данной книги, как показано на посвященной ей веб-странице[17], учитывался общий стиль книги и стиль, характерный для отдельных глав, а также принимались решения по поводу оформления заголовков и абзацев, подбора шрифтов и других элементов.

Направления развития стилей. В-третьих, со временем происходит изменение и развитие стиля, что проявляется даже в стиле создателя стиля, т.е. самого человека. Знатоки способны отличить поздние работы Тёрнера от его ранних картин[18]. Великие стили, такие как готическая архитектура, часто изменялись настолько значительно, что в ней, допустим, можно различить готику раннюю, декоративную и перпендикулярную. Что же касается разновидностей стиля, то они развиваются еще быстрее, будь то поп-музыка, подростковый жаргон или английские парки XVII столетия.

Необходимость документального описания стиля для достижения его единообразия

Как уже было сказано, стиль проектирования отражает совокупность всех принятых микрорешений. А четкий стиль показывает, что множество микрорешений было единообразным. Четкий стиль может оказаться не самым лучшим, но невыразительный стиль ни в коем случае не может стать удовлетворительным[19].

Поэтому проектировщик, который ставит перед собой амбициозные задачи, должен всегда стремиться к достижению единообразия стиля. А еще большее внимание должен уделять соблюдению этого требования коллектив проектировщиков. Дело в том, что если, допустим, единственный автор, не отрываясь от работы, пишет статью объемом в десять страниц, то вполне можно рассчитывать на соблюдение в ней четкого стиля. Если же автор пишет книгу, то чаще всего обнаруживает необходимость документировать некоторые свои решения

по поводу стиля, принимаемые на микроуровне в процессе работы, чтобы добиться единообразия. Более того, даже для статьи с десятью страницами может потребоваться проверка стиля (или привлечение к работе опытного редактора окончательной версии), если к написанию ее разделов будут привлечены разные авторы.

В условиях разработки большинства проектов многие стилистические решения согласуются с самого начала. Научная статья подготавливается к печати в соответствии со справочником по стилю, принятым в редакции журнала; книга проверяется издателем. Проектировщики автомобилей руководствуются колоссальным набором стандартов SAE, а также каталогом пружин, гаек, болтов и прочих деталей, рекомендованных к применению в автомобильной промышленности. Проектировщики операционных систем руководствуются библиотекой стандартных подпрограмм.

Тем не менее в каждом конкретном проекте появляются многочисленные поводы для принятия микрорешений в связи с непредвиденными обстоятельствами. Занимаясь совместным написанием нашей книги [25], мы с Джерри Блау отражали принятые решения, касающиеся стиля, в отдельном документе, который в конечном итоге превратился в руководство по стилю объемом в 19 страниц[20]. Разумеется, этот документ служил дополнением к руководству Chicago Manual of Style и собственному документу по стилю издательства Addison-Wesley. Но какими бы всесторонними ни были основные документы, мы не находили в них ответы на многие вопросы, поэтому были вынуждены составить для себя отдельное руководство. Например, нам пришлось принять решение по поводу того, как следует обозначать определенный компьютер. Должны ли мы указывать изготовителя и номер модели? Повторять ли все, что служит обозначением конкретной модели, при каждом упоминании или давать развернутый текст только в первый раз? Или, может быть, приводить полное название единожды в каждой главе?

Разумеется, коллектив проектировщиков должен придавать большое значение надлежащему документированию проекта, идет ли речь о технических чертежах, строительных кальках или руководствах пользователя. Проектировщики должны также хотя бы отчасти объяснять свои намерения и подчеркивать, почему было принято то или иное решение, чтобы те, кому в дальнейшем придется реализовывать готовый проект, не оставили будущее здание без краеугольного камня, на котором все держится. Продукт, который не был тщательно документирован, не может считаться законченным. Кроме того, коллектив проектировщиков должен заботиться о сохранении концептуальной целостности будущего изделия в ходе его эксплуатации, поэтому отражать в документации бесчисленное множество микрорешений, которые составляют видимый извне проект и направляют его разработку.

Способы достижения хорошего стиля

Рецепт достижения хорошего стиля несложен: простые методы и упорная работа.

Целенаправленное изучение стиля других проектировщиков. Стремление на практике осуществлять попытки работать в непривычном для себя стиле. Это вынуждает заострять внимание на деталях и размышлять над тем, что раньше казалось вполне понятным. Такой подход к работе может привести к созданию превосходных произведений; достаточно лишь вспомнить "Старинные песни и танцы" Респиги, "Классические манускрипты" Фрица Крэйслера или его "Прелюдию и аллегро" (в стиле Пуньяни)[21].

Изложение своих суждений, которые лежат в основе принятых решений. Излагайте на бумаге свое мнение по поводу того, какой стиль вам нравится и почему, какие аспекты определенного стиля являются наиболее привлекательными и по какой причине.

В основе всего должна лежать работа, работа и еще раз работа.

Неоднократный тщательный пересмотр. Поиск стилистических несогласованностей.

Привлечение к работе только тех проектировщиков, которые отвечают всем требованиям. В создании ответственных проектов должны участвовать только те разработчики, которые имеют сложившийся стиль и хороший вкус, о чем свидетельствуют ранее созданные ими работы.

Примечания и ссылки

1. См. [216].
2. В книге [98] приведены убедительные доказательства в пользу того, что проектировщик должен руководствоваться не только теоретическими результатами, но и чувством прекрасного.
3. В своей книге [229] Стив Возняк с удовлетворением отметил, что в созданном им компьютере Apple I было применено на одну треть меньше компонентов по сравнению с другими компьютерами такого же класса.
4. См. [211]. Упрощенное описание компьютера Ван дер Пола приведено в книге [25], раздел 13.1. См. также [212].
5. Мы привели сложную однострочную конструкцию APL в книге [25]; см. листинг 9.8 на с. 511.
6. См. [189].
7. См. [25].
8. См. [22]; [23].

9. В книге [139] приведены убедительные доказательства в пользу того, что превосходство и уровень стиля ("высоколобый" и "узколобый" подходы) независимы.

10. Важным архитектурным примером является книга [7].

11. К примерам четких стилей, которые стали предметом моего особого восхищения, относятся некоторые образцы, иногда внешне кажущиеся надуманными и подражательными.

- Расположенный под открытым небом музей испанской архитектуры "El Poble Espanyol" в Барселоне.
- Площадка EPCOT в парке Walt Disney World с расположенными на ней национальными павильонами.
- Студенческий городок Университета Дьюка, построенный в готическом стиле.
- Десять павильонов Джефферсона на лужайке в Университете штата Вирджиния, каждый из которых иллюстрирует определенный стиль.
- Написанные еще в XVIII столетии прозаические произведения Сидни Смита.
- Упомянутые выше произведения Респиги и Крэйслера.

12. Одно из этих произведений считалось имитацией и после проведенного исследования было возвращено в Чикагский художественный институт, в место постоянного хранения, как подлинное.

13. Дополнительные примеры приведены в главах 14 и 15, а также в разделе 12.4 книги [25].

14. См. [156].

15. См. [140], глава 15.

16. В книге [56] приведен серьезный обзор определения понятия стили и попытка применения к нему количественных критериев.

17. См. `http://www.cs.unc.edu/~brooks/DesignofDesign`.

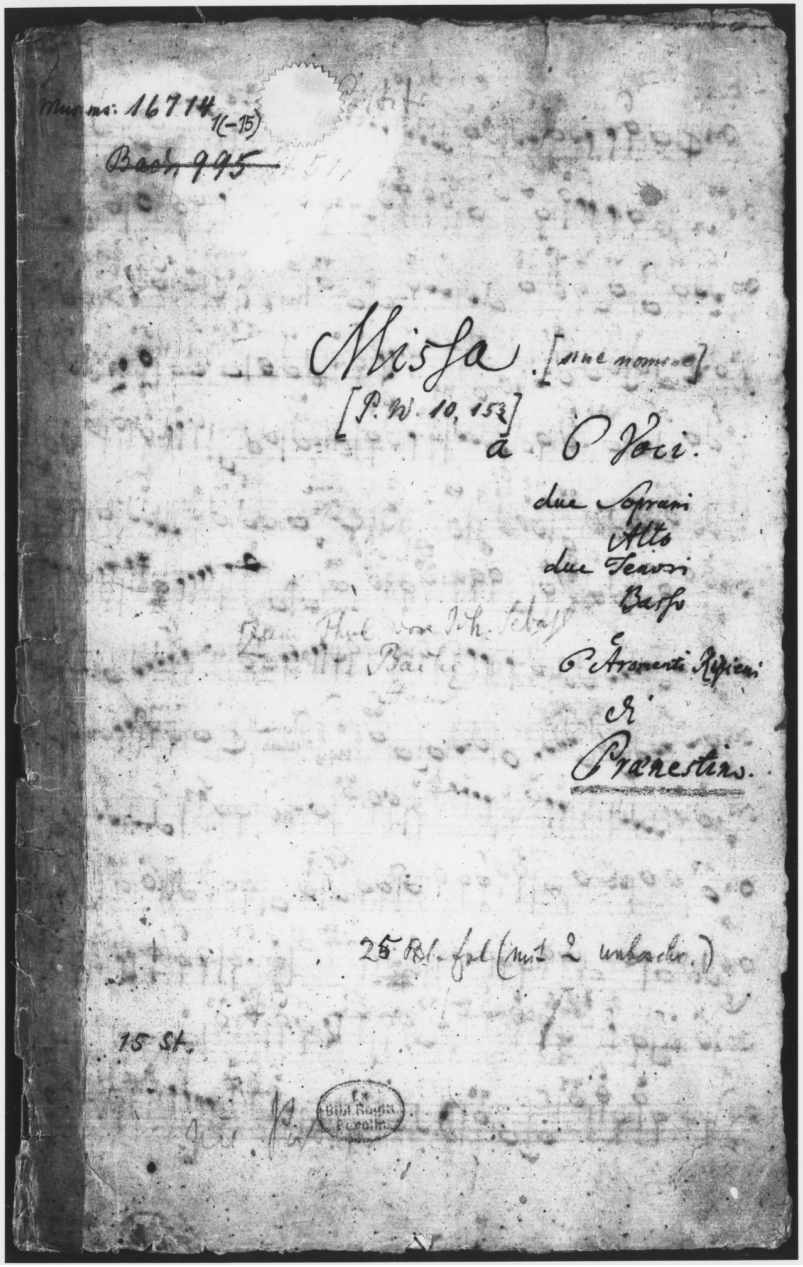
18. Стили большинства художников развиваются. Выставка "Моне в Нормандии", которая проводилась в 2006-2007 гг. в Художественном музее штата Северная Каролина, показала потрясающее развитие стиля этого художника за годы его творчества.

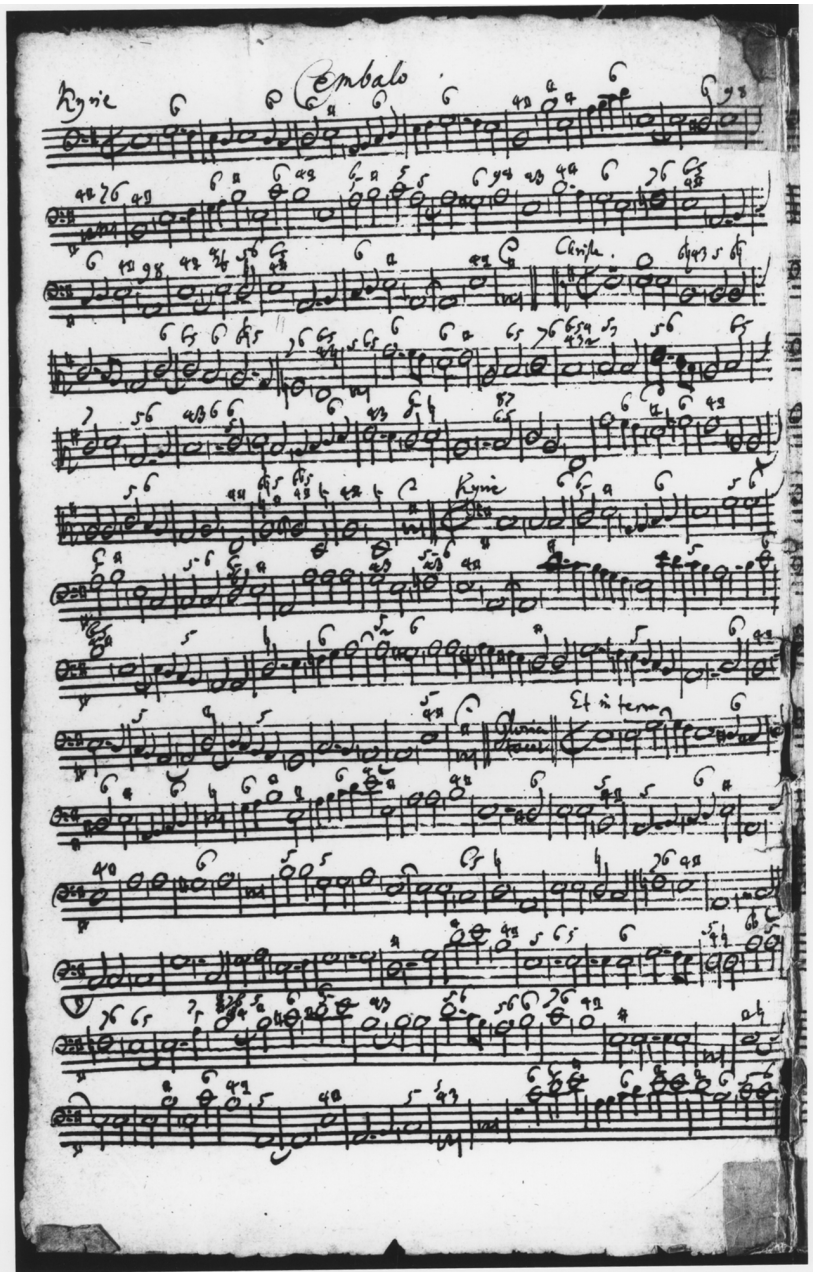
19. Ошеломляющая барселонская церковь "Святое семейство" архитектора Гауди кому-то может нравиться или не нравиться, но концептуальная целостность ее стиля не вызывает сомнений.

20. Это руководство приведено на веб-странице к этой книге, `http://www.cs.unc.edu/~brooks/DesignofDesign`.

21. Крэйслер впервые опубликовал "Классические манускрипты" в 1905 году, утверждая, что нашел их в одном старинном женском монастыре на юге Франции. Дело в том, что его скромность подвергалась большим испытаниям в связи со слишком частым появлением собственного имени в качестве композитора на программах проводимых им концертов. В 1935 году Крэйслер признал, что сам сочинил "Классические манускрипты", после чего дальнейшие публикации этих произведений осуществлялись как музыкальных пьес "в стиле классики".

TheDesignofDesign.indb 192





30.10.2012 15:58:31

13

Применение образцов в проектировании

...Плодотворный поиск в огромном пространстве вариантов возможен только при наличии определенного представления о том, что должно быть найдено. Без определения предварительных условий того или иного рода нельзя понять, где искать, или узнать, найдено ли искомое. Это также, по-видимому, способствует созданию для всех архитекторов одинаковых исходных предпосылок, поскольку все они вправе опираться в поиске на принятые ранее решения и сложившиеся представления о стиле...

См. [116]

◂ ***Страницы из копии мессы Палестрины, собственноручно написанные Бахом. Государственная библиотека в Берлине***

Принципиальная новизна лишь немногих проектов

Большая привлекательность новаторских разработок. Мало кому выпадает удача заниматься проектированием в качественно новой области. Можно только представить себе, насколько интересно было заниматься созданием первого искусственного спутника Земли, первого мобильного телефона, первого интерфейса WIMP ("window, icon, menu, pointing device" — оконный интерактивный интерфейс), первого аэровокзала, первого суперкомпьютера!

Общий удел. Но, как правило, даже во многом новые проекты опираются на более ранние разработки, предназначенные для аналогичных целей и созданные на основе аналогичных технологий. Возможно даже, что предыдущее изделие было разработано самим проектировщиком; а если дело обстоит иначе, то он, несомненно, видел, изучал и, может быть, использовал подобные изделия.

Это означает, что мы обязаны определить истинную роль образцов, или прототипов, в проектировании. Как проектировщик должен их изучать и использовать? Следует ли в каждой области проектирования создавать постоянно пополняемое и доступное хранилище образцов? Как это сделать? Кто этим должен заниматься?

Функции образцов

Образцы могут служить надежными моделями для новых проектов; неявно определять перечень контрольных вопросов, на которые необходимо найти ответ при решении задач проектирования; предупреждать о потенциальных ошибках; а также становиться пусковой площадкой при проектировании в принципиально новых областях.

Вот почему прославленные творцы вкладывали большие усилия в изучение работ своих предшественников. Палладио (1508–1580 гг.) не только изучал по книгам здания, возведенные Витрувием (22 г. до н.э.), но и отправился в Рим, чтобы измерить и описать сохранившиеся памятники, выявляя наиболее удачные концепции и пропорции, выработанные в процессе развития древнеримской архитектуры. Эта утомительная и неблагодарная работа позволила Палладио не только разработать собственные оригинальные проекты, но и написать книги по проектированию, в которых был заложен один из наиболее устойчивых архитектурных стилей.

Джефферсон тщательно изучил не только книги Палладио, но и здания в Париже, построенные на основе предложенных им концепций[1].

Иоганн Себастьян Бах взял на работе шестимесячный неоплачиваемый отпуск и отправился в путешествие длиной в 250 миль для изучения произведений композитора Букстехуде и знакомства с его идеями. (В конечном итоге Бах потерял работу, поскольку не вернулся вовремя из отпуска.) Произошло так, что Бах получил большую известность как композитор, чем Букстехуде, но его неоспоримое

превосходство стало результатом постижения и применения методов предшественников, а отнюдь не их игнорирования[2].

Я уверен, что для достижения большего успеха технические проектировщики должны поступать аналогичным образом, но в действительности этого не удается сделать из-за ускоренного темпа современного проектирования.

Разумеется, отдельные проектировщики предпринимают усилия по изучению накопленного опыта, но эта работа должна проводиться на уровне технических дисциплин проектирования в направлении создания доступных хранилищ образцов и проведения их критического анализа с привлечением знающих специалистов, поскольку без этого больше невозможно создавать действительно качественные проекты.

Характерные особенности проектирования аппаратного и программного обеспечения

Каково состояние работ по проектированию на основе образцов в области создания аппаратного и программного обеспечения? Я считаю, что по справедливой оценке в этой области состояние дел намного хуже по сравнению с дисциплинами проектирования, которые сложились раньше. Тем не менее уже достигается понимание того, что нужно больше вкладывать в разработку основ самого проектирования и выделять ресурсы на организацию проектирования с применением образцов. Однако следует отметить, что в планах работ по разработке это требование еще не находит достаточного отражения, а степень проникновения подхода на основе образцов в практику проектирования все еще является неудовлетворительной.

Изучение используемых образцов

В сложившихся ранее дисциплинах проектирования, таких как архитектурное проектирование, можно отчетливо наблюдать, насколько отличаются друг от друга начинающие проектировщики и опытные профессионалы с точки зрения использования образцов.

Новички используют образцы, с которыми им довелось соприкоснуться на собственном опыте. С другой стороны, подготовленный профессионал, как правило, желает приобщиться к значительно большему объему накопленных знаний, изучая целые библиотеки образцов, представляющих различные эпохи, разные стили и всевозможные теоретические школы. А те, кому улыбнулась удача, находят более подготовленных спутников в путешествиях по этим сокровищницам знаний, притом что их наставники раскрывают наиболее важные особенности предшествующих работ и объясняют различия между ними.

Что же касается проектирования аппаратного и программного обеспечения, то часто можно наблюдать, как в качестве образцов используются лишь собственные,

достигнутые ранее результаты; в нашем деле даже самые выдающиеся профессионалы не изучают доступные образцы.

Компьютеры. В архитектурах вычислительных систем обнаруживается глубокое влияние компьютеров, с которыми разработчик архитектуры ранее имел значительный опыт работы. Поэтому первые модели миникомпьютеров DEC весьма напоминают компьютеры Whirlwind, разработанные в Массачусетсском технологическом институте, мэйнфреймы System/360 корпорации IBM испытали сильное влияние моделей IBM 704 и 1401, а исходные варианты микрокомпьютеров, вполне очевидно, были выпущены с учетом результатов, достигнутых при создании миникомпьютеров DEC PDP-11.

Необходимо также учитывать опыт, накопленный самой корпорацией, поскольку компьютеры, в отличие от строительных объектов, разрабатываются системными архитекторами, работающими в тех же компаниях, которые затем реализуют готовые проекты. Таким образом, разработчики в большей степени знакомы с компьютерами, созданными в своих компаниях, чем с изделиями конкурентов, поскольку чаще всего проходят начальное обучение на их основе, а затем занимают свое место в сложившемся коллективе со своими традициями. Например, стиль разработок, принятый в корпорации Intel, отражается в конструкциях всех создаваемых в ней микропроцессоров.

В этой области широко распространен подход к проектированию, в основе которого лежит дальнейшее развитие ранее созданных моделей. Успешное создание определенных моделей компьютеров влечет за собой появление целых семейств одного поколения и следующих поколений, которые обычно проектируются по принципу расширения функций предшествующих моделей.

Программное обеспечение, выпускаемое в рамках массового производства. В программных продуктах, таких как Microsoft Word, также явно обнаруживается воплощение того же принципа, который соблюдается при проектировании компьютеров, т.е. создание следующих поколений на основе прогрессивной модификации функций и реализаций. Эти процессы развития программного обеспечения были тщательно исследованы и описаны в книге [137][3].

Заказное прикладное программное обеспечение и операционные системы. В историческом плане бо́льшая часть заказного прикладного программного обеспечения и операционных систем становится воплощением ранее полученного опыта их проектировщиков, но не отражает успехи, достигнутые во всей этой дисциплине.

Тем не менее в последнее время стало шире практиковаться изучение документации по подобным системам и ознакомление с имеющимися образцами, что способствует плодотворному обмену идеями в этой области. Проявлением этой тенденции является выход из печати таких книг, как [95], в которой глубоко рассматриваются структуры данных и подходы к разработке архитектуры на уровне компонентов, и [53], где речь идет о применении крупномасштабных образцов, воплощающих в себе конструкцию всей системы. Как оказалось, весьма велика

потребность в исчерпывающих описаниях целых систем, которые позволяли бы ознакомиться со всеми реализованными в них концепциями. Но, вообще говоря, такие описания подготовлены лишь для немногих операционных систем.

Изучение проектных требований к образцам

Нетривиальной является даже сама задача организации обучения проектировщиков на базе образцов, принятых на вооружение в соответствующих областях проектирования. Например, для изучения архитектуры системы принято знакомиться со справочным руководством по этой системе. А для изучения реализации часто можно воспользоваться эксплуатационной документацией. Но для получения общего представления о системе приходится изучать многочисленные технические статьи и книги о подобных изделиях, чтобы понять, на основе каких требований она была создана.

Однако в документации главным образом находят отражение сведения о том, как реализована та или иная функция, и почти ничего не говорится о самом назначении этой функции. К тому же многие проекты так и остаются до конца не описанными и не раскрытыми в документации, подготовленной их разработчиками; отчасти это обусловлено тем, что создатели этих проектов, как всегда, лишены возможности на что-либо отвлекаться, в связи с большим объемом работы над подготовкой следующего проекта.

Как ни странно, более подробные описания публикуются в те периоды, когда происходит становление новой технологии или последующие кардинальные ее пересмотры, поскольку при этом конфликтуют многие разные подходы и ведутся горячие споры. Но и эти статьи, как и доклады о победах в военных действиях, всегда служат констатацией факта, к тому же чаще всего освещают проблему лишь под определенным углом зрения; иначе говоря, в этих репортажах об успехах в большей степени отражено доминирование определенной точки зрения, чем это фактически наблюдалось в процессе проектирования. Сами же проектировщики хорошо знают, насколько часто в процессе разработки встречаются сложные препятствия, тупики, бесперспективные отклонения от принятого направления, а также происходит пересмотр целей. Несмотря на это, описания, в которых беспристрастно представлен весь ход работы, со всеми преодолеваемыми затруднениями, встречаются крайне редко, хотя и являются для нас наиболее поучительными.

Поэтому наиболее плодотворным источником идей, допустим, в области архитектуры компьютерных процессоров, становится исследование имеющихся образцов. Фактически развитие этой технологии происходит в современную эпоху, поэтому в отношении ее остается еще много тем для обсуждения и описания. Но в целом развитие данной области началось с испытания самых разнообразных подходов к проектированию и свелось к “стандартной архитектуре”. Мы с Блау подробно рассматриваем эту эволюцию в главе 9 нашей книги [25]. Основные

этапы революционных изменений в развитии аппаратных средств были отмечены появлением виртуальной памяти, мини-компьютеров, микрокомпьютеров, архитектуры RISC и т.д. Каждое новое достижение становилось стимулом к возобновлению дискуссий, что, в свою очередь, способствовало лучшему пониманию задач, стоящих перед этой областью.

Компьютеры первого поколения

Наиболее важной из когда-либо опубликованных работ по вычислительной технике является статья [52].

Она является исключительно важной, поэтому ее обязан прочесть каждый специалист в области теории вычислительных машин и систем. В указанной работе последовательно развиваются концепции хранимой в памяти программы и арифметического модуля с тремя регистрами, а также изложено много других интересных идей. Каждый вопрос подвергнут всестороннему рассмотрению; рассуждения авторов весьма убедительны.

Морис Уилкс (Maurice Wilkes) поделился своими впечатлениями по поводу одного из первых черновых вариантов этой статьи:

> *"Я просидел до поздней ночи, читая данное сообщение. ...Я сразу понял, насколько реальны изложенные в нем предложения, и с тех пор ни разу не усомнился в том, что разработка компьютеров будет происходить именно в этом направлении"* [4].

Далее Уилкс сообщает, что в этой статье представлены идеи, которые родились в дискуссиях с участием Преспера Экерта (Presper Eckert), Джона Мочли (John Mauchly) и Джона фон Неймана (John von Neumann), которые проводились в Университете штата Пенсильвания. Он сожалеет о том, что авторство этих чрезвычайно плодотворных идей обычно приписывают одному фон Нейману и что сейчас приходится прилагать определенные усилия для устранения этого недоразумения.

После появления статьи [52] многочисленные группы разработчиков из многих организаций приступили к созданию компьютеров с хранимыми в памяти программами, используя логические устройства на вакуумных лампах. Первые успехи были достигнуты в Манчестере, где был создан работающий, но слишком маломощный для того, чтобы быть полезным, компьютер Baby, и в Кембридже, где впервые появился практически применимый компьютер с программами, хранимыми в памяти, EDSAC. История этих проектов нашла превосходное отражение в литературе: [223]; [222].

К числу наиболее важных первых прототипов суперкомпьютеров относятся Stretch корпорации IBM и CDC 6600 корпорации Control Data. В книге [50] приведен наиболее значительный объем сведений об этих проектах. Но в этой книге самой интересной является глава 17, содержащая описание принципиально иного типа компьютера, который оснащен сопроцессором, предназначенным для

выполнения криптоаналитических операций, однако почти отсутствует описание области применения этого компьютера и не приведены соображения, на основании которых были заданы его характеристики.

В своем развитии модель CDC 6600 быстро достигла уровня модели Stretch как самого быстродействующего компьютера в мире и стала доминирующей в области машинных вычислений для проведения научных исследований. Она явилась предшественником семейства суперкомпьютеров Cray. Большой объем сведений по истории этого проекта приведен в книге [203].

Компьютеры третьего поколения

Возможности архитектуры вычислительной системы второго поколения были быстро исчерпаны; как оказалось, в них не предусмотрено применение достаточно большого количества битов в регистрах адресации, поэтому отсутствует возможность поддерживать большие объемы памяти, которые позволили бы им стать экономически оправданными и необходимыми. Наступил этап, на котором пришлось вынужденно прервать поступательное развитие архитектур многих линеек продуктов, хотя при этом возникли значительные трудности. К счастью, с появлением интегральных схем стало возможным значительное снижение стоимости реализации, а языки высокого уровня обеспечили перекомпиляцию созданного ранее программного обеспечения, поэтому переход на новую архитектуру прошел относительно безболезненно. С развитием новых компьютерных архитектур возникла необходимость в проведении дополнительных теоретических исследований.

В частности, некоторые теоретические обоснования архитектурных решений, принятых при создании компьютеров моделей System/360, приведены в книге [25], хотя сама эта книга не является сугубо теоретической. Например, в ней описано много случаев, когда обоснованием принятых решений служил исключительно наш с Блау личный опыт. Краткие сведения по истории развития проекта System/360 приведены в книгах [10] и [24].

Виртуальная память

Автоматический перенос страниц с блоками команд и данных между низкоскоростным, но крупным хранилищем данных и меньшей, но высокоскоростной памятью был впервые реализован в проекте Atlas, разработанном в Манчестерском университете. Разработчики операционных систем с разделением времени из Университета штата Мичиган и Массачусетсского технологического института вскоре предложили обобщить эту концепцию в виде полноценной виртуальной памяти с огромным пространством имен. Компьютеры такого типа были созданы корпорациями General Electric и IBM. Итак, снова произошла революция и снова стали необходимыми теоретические обоснования, которые, в частности, приведены в следующих книгах: [200], [79], [11].

Революционные изменения, связанные с появлением мини-компьютеров

После создания логических схем на транзисторах и диодах стоимость реализации проектов компьютеров существенно снизилась. Были разработаны мини-компьютеры, такие как PDP-8 корпорации DEC, с выходом которых мир изменился, поскольку вычислительная техника стала доступной даже для небольших отделов, тогда как раньше лишь отдельные учреждения могли позволить себе приобретать ЭВМ и содержать обслуживающий персонал. К компьютерной технике стали приобщаться широкие массы людей, и это оказалось не менее важным, чем исключительно технические достижения, измеряемые показателями производительности и стоимости. Мини-компьютеры изготавливались тысячами и применялись совместно, а не заменяли так называемые универсальные ЭВМ.

Изготовителей универсальных ЭВМ вполне удовлетворяли те бизнес-модели, в рамках которых они работали, поэтому, оставаясь загруженными заказами, ни о чем не заботящимися и вполне удовлетворенными жизнью, все эти компании проспали наступление эпохи мини-компьютеров. А в это время появилось много новых компьютерных фирм. Наиболее успешной из них оказалась Digital Equipment Corporation. Технические характеристики и история развития мини-компьютеров DEC приведены в книге [19].

Принципиальные изменения, связанные с появлением микрокомпьютеров и архитектуры RISC

Создание интегральных схем повлекло за собой становление следующего этапа развития и распространения вычислительной техники. Произошло резкое снижение стоимости компьютеров, в связи с чем возможность приобретать и обслуживать собственные компьютера получили не только небольшие отделы, но и частные лица. Микрокомпьютеры стали изготавливаться в миллионах экземпляров.

Настало время задуматься о своем будущем и изготовителям мини-компьютеров, которые когда-то также были весьма довольны успехом своего дела, оставаясь загруженными заказами, ни о чем не заботящимися и вполне удовлетворенными жизнью. Теперь уже они не заметили, как наступила революция, связанная с появлением микрокомпьютеров. Корпорация Hewlett-Packard сумела выжить, а корпорации DEC этого не удалось. Некоторые же изготовители универсальных ЭВМ, в частности IBM, сумели вернуться в игру и стать крупными поставщиками персональных микрокомпьютеров.

В связи в этими революционными изменениями также потребовалось проведение большого объема теоретических исследований, которые отчасти представлены в следующих книгах: [118], [166], [171][5].

Таким образом, развитие компьютерных наук стало все в большей степени складываться по примеру других научных дисциплин, на основе документирования накопленных фактов, публикации исторических обзоров, внимательного

рассмотрения причин и следствий революционных изменений, а также подробного описания тех образцов и артефактов, которые иллюстрируют поворотные моменты в развитии.

Возможности усовершенствования принципов проектирования на основе образцов в проектных дисциплинах

Прежде всего необходимо найти ответ на вопрос о том, что если проектировщики действительно должны выходить за рамки личного и корпоративного опыта для овладения идеями и методами, накопленными в рамках всей проектной дисциплины, то какую отдачу от этого они должны получить?

Коллекции образцов

Как показывает предыдущий раздел, в области проектирования архитектуры вычислительных систем можно найти большое количество хорошо описанных образцов. Очевидно, что следующим шагом в использовании этого интеллектуального богатства может стать сбор и публикация систематизированных коллекций образцов. Гордон Белл (Gordon Bell) и Аллен Ньюэлл (Allen Newell) впервые представили достаточный объем сведений для того, чтобы проектировщики могли воспользоваться их наработками, в своей превосходной книге [18]. Свой вклад внесли также Хеннесси (Hennessy) и Паттерсон (Patterson), опубликовавшие книгу [112], в которой большой интерес представляет приложение E. Кроме того, и мы с Блау дополнили общедоступную коллекцию образцов проектирования в главах 9–16 нашей книги [25].

Применение коллекции образцов

Следующим шагом после публикации материалов проектов становится внимательная, беспристрастная критика представленных в них образцов проектирования. Что касается проектирования вычислительной техники, то подобную критику можно встретить и в книгах с материалами проектов, и в журнальных обзорах, в которых рассматриваются описания конкретных компьютеров.

За критическим рассмотрением следуют анализ, сравнение одного образца с другим и оценка значимости образцов с учетом назначения каждого из них. Тем не менее аналитики часто склоняются к тому, чтобы подвергать критике выбор целей при создании продукта, а не оценивать эффективность проектирования в ходе достижения этих целей. К сожалению, такой анализ не может оказать существенную помощь будущим проектировщикам.

Очередным шагом должен стать сравнительный анализ. Он позволяет выявить, какие характеристики проекта являются перспективными и какие тупиковыми.

Обнаруживается, что некоторые подходы к решению задач проектирования приносят свои плоды, а другие влекут за собой бесполезные затраты сил. Поэтому добросовестный анализ должен приводить к созданию все лучших и лучших версий проектных рекомендаций, представляющих собой обобщение всего накопленного опыта. В большинстве технических дисциплин подобные рекомендации оформляются в виде руководств по проектированию и в конечном итоге — в виде стандартов.

Особенности проектирования программного обеспечения

Область проектирования вычислительной техники развивается очень быстро, и в ней нагляднее всего обнаруживаются перечисленные выше этапы: сбор образцов, критический и сравнительный анализ, а также обобщение полученных результатов в виде рекомендаций. В этом отношении проектирование программного обеспечения еще намного отстает.

Возможно, причиной является исключительно новизна этого направления. Разработка программного обеспечения как научная дисциплина была признана в 1968 году[6]. С другой стороны, разработка вычислительной техники на научной основе началась с 1937 года[7]. До сих пор в нашем распоряжении имеются лишь описания отдельных образцов операционных систем[8], несколько сборников описаний языков программирования и связанных с этим теоретических исследований[9], а также определенный объем других материалов по этой теме.

Составление описания архитектуры операционной системы представляет собой намного более трудную задачу по сравнению с описанием самого компьютера. Намного сложнее даже отдельные функции, а само количество функций значительно больше в первом случае, чем во втором. Кроме того, гораздо сложнее поддается описанию семантика операций, допустим, таких как Link (связь), по сравнению, например, с операцией Divide (деление). По моему мнению, задача проектирования программного обеспечения на два порядка величины сложнее, чем проектирование вычислительной техники. Именно по этой причине на примере даже наиболее выдающихся образцов программного обеспечения явно наблюдается запаздывание в сборе, критическом анализе, сравнительном анализе и синтезе. Поэтому меня радует, что Гради Буч (Grady Booch) взял на себя обязательства окончательно собрать для печати свою книгу *Handbook of Software Architectures*, которая в настоящее время выложена на веб-сайт[10].

Кто этим должен заниматься? Систематизация образцов для исследования — это задача ученых, а не проектировщиков. Ученые и проектировщики существенно отличаются друг от друга в том, какие задачи они ставят и как организуют свою работу. Проектировщики часто не успевают завершить один проект, как начинается следующий, поэтому у них не остается времени на размышления по поводу сделанного, не говоря уже о том, чтобы заниматься теоретической деятельностью в своей области. Ученых привлекают только зрелые проектные

дисциплины (бывает также, что становятся готовыми размышлять над своей работой проектировщики, накопившие достаточный опыт).

Создание стимулов к развитию теоретических исследований. К сожалению, в наши дни сообщество проектировщиков программного обеспечения недостаточно ценит и поощряет работу систематизаторов в своей области. В частности, вряд ли можно прожить на то, что приносит деятельность по сбору и обобщению опыта создания программного обеспечения. Во многих организациях считают, что задача изучения образцов программ должна быть скорее возложена на отдел истории науки и техники (которого часто и не существует), чем на какой-либо производственный отдел.

Преимущества образцов — возможность применить готовый материал, оценить оригинальность и проверить значимость достигнутого успеха

На этом мы должны остановиться более подробно. Выше в данной главе при рассмотрении проблем применения образцов в проектировании почти не были затронуты некоторые вопросы, весьма значимые для каждого проектировщика:

- Прежде всего, многие считают, что копирование ранее созданного проекта, или прецедента, представляет собой лишь применение готового материала ради экономии собственных сил. Считается, что честный профессионал не может этим заниматься, не испытывая муки совести.
- Ведь люди становятся проектировщиками потому, что любят создавать что-то новое. А какую радость может доставлять ограничение собственного полета фантазии из-за того, что ты замкнулся в железной клетке стиля, выбранного кем-то другим до тебя?
- Мир высоко ценит оригинальность и новизну, поэтому вознаграждает эти качества, поскольку их носители приобретают уважение со стороны других, репутацию, а иногда даже известность и благосостояние.
- Человек может внести свой особый вклад в достижения человечества, лишь проявляя собственный уникальный дар предвидения. Применение готовых образцов равносильно игнорированию или подавлению своих оригинальных идей, и это, наверное, не лучший способ реализации своих возможностей[11].

Несколько иной взгляд на вещи

Чтобы предотвратить неправильную трактовку излагаемых здесь идей, я обязан подчеркнуть следующее: я не утверждаю, что большинство проблем проектирования можно решить путем адаптации готовых образцов, а также не предлагаю заниматься их рабским копированием.

Я являюсь сторонником изложенных ниже представлений.

- Проектировщик должен быть хорошо знаком с образцами творчества в своей области, а также изучать их сильные и слабые стороны. Стремление к новизне не может служить оправданием для незнания ранее достигнутого.
- В инженерном деле и, может даже, в искусстве безоглядное стремление к достижению новизны (такое, что даже не предвидится получение от него "пользы" в каком-то практическом смысле) — это глупая идея и эгоистичное оправдание свой заносчивости, поскольку риск непреднамеренных отрицательных последствий является неизбежным.
- Проектировщики, глубоко познавшие стиль своих предшественников, имеют в своем распоряжении целую россыпь сокровищ, из которой они могут выбирать все необходимое для собственного новаторства.

Использование готовых материалов

Разумеется, проектировщик, стремящийся решить свои задачи с минимальными усилиями и без особого напряжения, может сократить для себя объем работы до предела, выбрав готовый образец и лишь изменив его по месту. Но, вообще говоря, даже те, кто просто копирует готовые решения, не прибегают к устаревшим или малопригодным образцам, а выбирают лишь наиболее современные и "модные". Именно поэтому, например, в США имеется огромное количество строений с посредственной архитектурой и невзрачных домов, напоминающих провинциальное поместье, выполненное в стиле "прерий", который основал Фрэнк Ллойд Райт (Frank Lloyd Wright).

Но в любой области проектирования необходимо подавлять в себе стремление двигаться по самому легкому пути, а также проявлять весь свой энтузиазм и усердие для изучения полной совокупности образцов, доступных в этой области.

Оригинальность и значимость достигнутого успеха

В наши дни принято подчеркивать большое значение новизны и оригинальности новых проектов, но, по моему мнению, это неправильно. Остаются справедливыми слова Витрувия, приведенные в эпиграфе к настоящей главе, которые можно перефразировать так: независимо от сферы проектирования, желательным является такой проект, который отвечает функциональным потребностям, обеспечивает надежность и устойчивость против отрицательных факторов, а также приносит удовлетворение пользователю. В качестве образцов изделий, созданных по таким проектам, можно назвать мебель Шейкера, столовые приборы Ревера, острогубцы Пека и Стоу[12].

Обладают ли эти изделия оригинальностью? Несомненно, и это достойно восхищения. Эти проекты относятся к прежним временам, но есть и недавно

разработанные проекты, настолько искрящиеся новизной, что нельзя не порадоваться изяществу решений: карманный набор фальцовочных инструментов фирмы Лезерман, игрушка "шагающая пружина", вантовый мост.

Но мы обязаны подчеркнуть, что предметом восхищения в этих случаях является чрезвычайное изящество новых решений старых задач, а не сама по себе оригинальность. В этом можно убедиться самому, поняв, что пользование подобным предметом, будь то инструмент или игрушка, каждый раз доставляет удовольствие. И эти приятные ощущения не ослабевают. С другой стороны, новинка в чистом виде способна лишь раздразнить, но не удовлетворить наше стремление к обладанию идеальной вещью. Чудо, случившееся семь дней назад, уже не вызывает прежних ощущений. По мере того как новинка становится привычной, она перестает восхищать. Любитель новинок вынужден постоянно находиться в поиске. Он не может стать наконец-то удовлетворенным[13].

Оригинальность как цель или побочный продукт. Те, кто стремятся к оригинальности, скорее склонны найти что-то новое, но не понимают, что гораздо важнее добиться того, чтобы чувство восхищения новинкой оставалось непреходящим. С другой стороны, те, кто желают создавать исключительно работоспособные проекты, с большей степенью вероятности могут предложить действительно новые проектные решения, ценность которых не уменьшается со временем, причем оригинальность их проявляется почти что в качестве побочного продукта.

Значимость достигнутого успеха. Близко связанным со стремлением к оригинальности является желание повысить значимость достигнутого успеха, или сделать для себя имя. Этот издревле известный недостаток людей, который явился следствием грехопадения прародителей человечества, отрицательно отражается практически на всех проектах и приводит ко многим разрушительным последствиям.

В частности, он ярко проявился на примере строительства вавилонской башни. "И сказали они: построим себе город и башню, высотою до небес, и сделаем себе имя, прежде нежели рассеемся по лицу всей земли"[14].

Шелли в одной из своих поэм показал, как проявляется это желание и в старину, и в наши дни: "Мое имя Озимандиас, король королей. Смотрите на мои творения, великие, и потеряйте надежду!"

Стремление быть оригинальным привело к провалу многих творческих начинаний[15].

Примечания и ссылки

1. См. [122].
2. В книге [206] можно найти такие строки: "В действительности не было ни одного музыкального направления, начиная от творчества Палестрины и заканчивая работами других композиторов, с которыми Бах имел

возможность ознакомиться в свое время, образцы которых он тщательно не скопировал бы собственноручно".

3. См. [137]; [164].
4. См. [221].
5. См. [118], [166], [171].
6. См. [159].
7. См. [1].
8. Multics, UNIX, OS/360, Linux.
9. См. [183]; [219]; [20], том 2.
10. См. [31].
11. Собор Святого Павла, возведенный Реном, показывает, какие замечательные произведения могут создаваться как в рамках традиции, так и вне ее.

    Мне всегда приятно видеть, как содержатели парка Дисней-уорлд создают свои аттракционы, придерживаясь требуемых стилей и вместе с тем проявляя потрясающую оригинальность. Достаточно вспомнить игровые площадки "Замок Золушки", "Остров Тома Сойера", "Дом с привидениями", "Дом на дереве швейцарской семьи Робинсон" и "Главная улица в городе XIX века". В этом контексте даже карикатурные преувеличения и пародирование стилей осуществимы и могут оказаться восхитительными.
12. В книге [111] представлен превосходный краткий обзор работ Витрувия. В ней утверждается то, что Витрувий сформулировал метод проектирования, по существу воплощающий в себе подход на основе ветвящегося дерева, который ведет к выбору одного из 45 типов домов. Это — главная ссылка относительно использования образцов и упрощенных методов проектирования.
13. Я думаю, что именно здесь пролегает реальный водораздел между благочестивыми удовольствиями и сатанинскими игрищами. Ведь только подлинные наслаждения приносят удовлетворение, или сатисфакцию (от латинского слова satis — "достаточно"). Нормальный человек способен сказать, что ему достаточно пищи, достаточно сна, достаточно работы, достаточно игры, достаточно любовных утех, а извращенная личность всегда ищет новые деликатесы, стремится найти невиданные оттенки вкуса, допускает все бо́льшие отклонения от нормы.
14. Бытие 11.
15. Я работаю в таком здании. В своем архитектурном контексте многоэтажный дом "Ситтерсон-Холл" мог бы легко составить совершенный визуальный образ четырехугольника вместе с расположенной напротив гостиницей "Кэролайна Инн". Он имеет такую же высоту, оформлен в изящном

колониальном стиле и построен из такого же кирпича. Но стремление к ложно понятой оригинальности побудило Ситтерсона возвести над зданием непохожую на соседнюю, уродливую стальную крышу, а на третьем этаже предусмотреть мансардные окна, которые расположены слишком высоко для того, чтобы можно было сидя любоваться окружающим видом. В результате целостный визуальный образ четырехугольника так и не сложился.

TheDesignofDesign.indb 208 30.10.2012 15:58:32

14

В чем могут ошибаться опытные проектировщики

Заклинаю вас ранами Христа — не отбрасывайте заранее мысль, что вы можете ошибаться.

ОЛИВЕР КРОМВЕЛЬ, 1650 Г.

Но когда я обнаруживаю свою ошибку, я счастлив!

ФИОРЕЛЛО ЛА ГАРДИА

Наибольшую озабоченность всегда вызывают не ошибки опытных проектировщиков при проектировании предмета, а неправильный выбор предмета проектирования.

◀ *Крушение неправильно спроектированного аэродинамически моста Tacoma Narrows. AP Wide World Photos*

Ошибки

В любой области любитель допускает множество мелких ошибок, которые никогда не сделал бы профессионал. Обучение, производственная практика и опыт работы помогают профессионалу освоить качественные методы.

Но если профессионал попадает впросак, то эти ошибки оказываются грандиозными, — строятся мосты, которые обрушиваются во время их возведения; появляются дома без лестниц между этажами; выпускаются компьютеры, расходующие впустую почти всю оперативную память; создаются языки программирования, имеющие слишком богатый набор средств, чтобы их мог изучить простой смертный.

Генри Петроски (Henry Petroski) предположил, что после каждого революционного открытия в области материалов или методов проектировщики на первых порах фактически действуют следующим образом:

- вначале продвигаются мелкими шагами;
- овладевают новым подходом;
- начинают смело внедрять новинки, часто забывая про исходные позиции;
- переходят все границы в своей смелости и самоуверенности, которые могут многократно возрасти под действием честолюбия и из-за отсутствия конкурентов.

Генри процитировал результаты исследования, показывающие, что между крупными крушениями мостов неизменно проходит 30-летний период, а также предсказал, когда следует ожидать очередной катастрофы[1]. Крушение моста I-35W в Миннеаполисе доказало его правоту.

По-видимому, основной причиной грандиозных ошибок профессионалов является выход на передний план нового поколения проектировщиков, которые с самого начала усваивают новые методы. Не почувствовав муки рождения общепризнанной истины, путь к которой часто лежит через острые дискуссии с выдвижением предположений и их опровержением, вновь испеченные проектировщики недостаточно полно учитывают противоречивый контекст, в котором складывались современные знания.

Кроме того, новые поколения проектировщиков не всегда до конца понимают, какое место методы, взятые ими на вооружение, занимают во всем арсенале применимых методов. У меня часто возникает ощущение, что некоторые профессионалы, хорошо знакомые с отдельными деревьями, не видят леса и не задают себе вопрос: “Какой в целом смысл имеет то, чем я занимаюсь?” Знаменитое высказывание Томаса Джефферсона можно перефразировать так, что профессионал часто весьма озабочен правильным созданием предмета, поэтому не может остановиться и спросить себя: “Правильно ли выбран сам создаваемый предмет?”

При разработке семейства компьютеров System/360 наша замечательная группа квалифицированных компьютерных архитекторов отвергла мысль о создании средств автоматического управления памятью, забыв о том, что картину маслом можно исправлять только до тех пор, пока не высохла краска.

Но особенно опасен для профессионального проектировщика успех. Неудача побуждает к анализу, исследованию, повторному обдумыванию. Успех укрепляет веру и в выбранный метод проектирования, и в себя. Слишком доверять тому и другому может оказаться опасным.

Самый неудачный из когда-либо созданных компьютерных языков

Ярким примером ошибки экспертов является создание языка управления заданиями (Job Control Language — JCL) операционной системы OS/360 корпорации IBM, известного теперь под названием языка управления заданиями MVS для z/OS. По моему глубокому убеждению, это — самый плохой язык программирования, созданный когда-либо и кем-либо. Он был разработан под моим руководством, и я не снимаю с себя вину, несмотря на то, что в принятии решений по поводу его создания принимали участие все уровни администрирования.

Весьма поучительно вначале изучить недостатки JCL как языка программирования. После этого необходимо разобраться в том, как коллектив проектировщиков программного обеспечения, состоящий из истинных профессионалов, в который, например, входили разработчики оригинальной версии языка Fortran и ведущие теоретики в области компьютерных языков, сумели с самого начала пойти в столь ошибочном направлении.

Хотя эти ошибки были допущены 45 лет тому назад, язык JCL все еще продолжает использоваться и сохраняется почти полностью в первоначальной форме. Многие наши ошибки таковы, что нам так и не удается от них избавиться. А расплата за эти ошибки является нескончаемой.

Краткая характеристика языка JCL

OS/360 была с самого начала спроектирована как пакетная операционная система, несмотря на то, что даже на первых порах пользователи терминалов могли работать с ней в интерактивном режиме, передавая задания в очереди заданий, запуская их на выполнение, запрашивая сведения о состоянии и выводя полученные результаты. Язык управления заданиями — это язык сценариев, который задает параметры и значения, используемые в ходе вычисления пакетного задания, указывает, какие входные файлы должны быть смонтированы и где должен быть расположен выходной файл, а также выполняет множество прочих вспомогательный функций, связанных с управлением программой и файлами данных. Например, в сценарии JCL может быть указано, как осуществить компиляцию

исходной программы, провести связывание с библиотечными программами, выполнить задачу применительно к определенным наборам данных, а также напечатать, записать на диск и архивировать на ленте требуемую часть выходных результатов.

Но в действительности язык JCL является сложным в изучении и использовании. Поэтому сложилась такая практика, что набор команд JCL, который позволил успешно управлять процессом вычисления одному пользователю, в дальнейшем копировался вслепую другими пользователями. Только наиболее смелые программисты решались углубиться в сценарий JCL, чтобы изменить что-либо, кроме самых очевидных параметров. Даже в наши дни к хранящимся в архивах программам на языках Fortran и COBOL прилагаются сопутствующие им сценарии JCL, которые созданы так давно, что получили название "пыльные колоды".

Основные недостатки языка JCL

Самым большим упущением разработчиков JCL оказалось то, что он в действительности должен был создаваться как язык программирования, но не воспринимался как таковой с самого начала его проектирования.

Единственный язык планирования для всех языков программирования. Язык JCL является весьма несовершенным, начиная с самой его концепции.

В комплектацию OS/360 вошли компиляторы для широкого набора языков программирования, по меньшей мере шести, не считая языков Fortran и COBOL. А каждый пользователь был обязан знать по крайней мере два языка: JCL и выбранный им язык программирования. Но многие программисты ограничивались лишь собственно языком программирования, поэтому были вынуждены просто заимствовать готовые сценарии JCL и передавать друг другу "пыльные колоды".

Но в действительности пользователям требовались не только возможности поэтапного запуска заданий, как в сценариях JCL, но и средства планирования запуска по времени, подобные возможностям времени компиляции, предусмотренным в языке PL/I и макроассемблере S/360. Это позволило бы каждому программисту работать в рамках единственного языка, задавая одни действия как запланированные на время компиляции, другие действия назначать для выполнения на этапе планирования, а третьи — на этапе прогона программы.

Применение в языке JCL синтаксиса ассемблера S/360, а не синтаксиса языка высокого уровня. По ошибке приняв решение ввести в действие единственный язык планирования, проектировщики к тому же выбрали неправильный язык. Даже в 1966 году, спустя один год после запуска и ввода в эксплуатацию полной версии OS/360, задания на ассемблере составляли всего лишь 1% всех заданий. Произошла радикальная смена подходов к использованию языков программирования, но это осталось незамеченным создателями OS/360.

Не полное соответствие синтаксиса JCL синтаксису ассемблера S/360. В язык JCL было внесено настолько много изменений по сравнению с исходной версией, что изучение синтаксиса ассемблера S/360 не гарантировало понимания синтаксиса JCL.

Зависимость от структуры записей на перфокарте. В исходной версии Fortran была учтена необходимость эксплуатации этого языка на компьютере IBM 704, который имел длину машинного слова, равную 36 битам (по состоянию на 1956 год), поэтому допускалась использование в каждой строке 72 знаков и строк продолжения. Знаки, которые следовали за позицией 72 до конца перфокарты, игнорировались. (Столбцы перфокарты 73–80 первоначально использовались для последовательной нумерации карт в колоде с программой, чтобы можно было легко собрать рассыпавшиеся карты!)

В языке JCL был унаследован этот формат на основе перфокарт, причем это произошло именно в тот момент, когда в остальной части OS/360 за основу был взят терминальный доступ. (И в то время, и впоследствии в терминалах не предусматривалось даже указание позиции символа, поэтому пользователи не могли определить момент достижения ими позиции 73 при вводе.) Произошла радикальная смена подходов, причем даже на уровне самого системного продукта, но разработчики языка JCL этого не заметили.

Слишком малое количество глаголов. Создатели языка JCL гордились тем, что в этом языке предусмотрено всего лишь шесть глаголов: JOB, EXEC, DD и т.д. И он при этом справляется со своей работой. Но количество функций, которые должен выполнять этот язык, превышает шесть[2]. Руководствуясь ложно понятыми представлениями об изяществе и простоте, разработчики языка не выяснили до конца, насколько сложными являются реальные задачи, которые он должен решать, поэтому им пришлось во многих случаях вводить в действие дополнительные конструкции языка, подготовленные на скорую руку.

Функции глаголов, возложенные на параметры объявлений. В условиях недостатка глаголов приходилось так или иначе реализовывать необходимые функции глаголов. Поэтому в язык JCL была введена инструкция DD (Data Declaration — объявление данных), в которой был предусмотрен (слишком) большой набор параметров в виде ключевых слов. Многие из этих параметров, например DISP, по существу представляют собой командные глаголы, указывающие, какие действия должны быть выполнены с набором данных после завершения текущего шага работы.

Почти полное отсутствие возможностей ветвления. Центральным понятием в большинстве языков программирования является переход по условию. Но в языке JCL это понятие не играет столь важной роли: возможности ветвления реализованы с запозданием, имеют ограниченный охват, а само ветвление происходит с учетом параметров.

Отсутствие итерации. В языке JCL нет такого оператора, который непосредственно позволял бы организовать итерацию; для достижения этой цели

приходится строить неуклюжие конструкции на основе ветвления. Разработчики языка не сумели предположить, что в сценарии времени выполнения может потребоваться повтор действий.

Отсутствие четко определенных средств вызова подпрограмм. Разработчики не смогли также понять, что может возникнуть необходимость вызова подпрограмм в сценарии времени выполнения. Это — еще более непростительное упущение, поскольку во многих программах JCL широко используются открытые подпрограммы; под этим подразумеваются повторяющиеся последовательности команд, идентичные во всем, кроме нескольких параметров.

Причины столь значительных недоработок при создании JCL

Профессиональные программисты, занимавшиеся разработкой языка JCL, внесли слишком большую долю личного опыта в решение этой задачи. У них сложились чрезвычайно устойчивые представления о том, что такое программа, поэтому была утрачена возможность подойти к решению задачи по-новому, с более широких позиций. В этом случае следование образцу привело к катастрофическим последствиям.

Основные теоретики, руководившие созданием языка JCL для OS/360, приступили к реализации проекта OS/360 сразу после завершения разработки операционной системы для компьютера IBM 1410/7010 (1963 год), оказавшейся весьма успешной. Но с точки зрения функций OS 1410/7010 была, по-видимому, на два порядка величины более простой по сравнению с OS/360. Это была в полном смысле слова пакетная операционная система, предназначенная для классических приложений управления файлами, без дистанционной обработки. Функции планирования, связанные с установлением имен файлов и распределением устройств ввода-вывода, осуществлялись с применением нескольких простых управляющих перфокарт, которые размещались перед колодой каждого задания до того, как должен был произойти ввод этого задания с помощью устройства чтения перфокарт, а этот метод был унаследован еще от тех операционных систем, которые функционировали на основе магнитных лент.

Разработчики языка JCL для OS/360 рассматривали свою задачу как развитие приобретенного ими опыта работы с компьютером 1410/7010: проектирование, как они объясняли свою цель, "нескольких инструкций для планировщика, вводимых с помощью перфокарт". Эта ошибка оказалась фатальной. Как оказалось, в основе каждой части этой формулировки цели лежала неправильная концепция, а поскольку разработчики не смогли этого понять, проектирование языка JCL с самого начала пошло по неправильному пути.

В действительности небольшое количество требуемых инструкций управления было приемлемым только для операционной системы 1410/7010, что же касается

языка JCL для OS/360, то стремление к достижению цели с помощью минимального набора инструкций выразилось в стремлении к упрощению. Именно поэтому в языке JCL было предусмотрено слишком мало глаголов. Ошибкой было не только максимальное сокращение количества типов инструкций; таковой было и неявное предположение, что для управления каждым заданием достаточно будет применить лишь несколько карт того или иного типа. Но на практике в сценарии JCL обычно приходилось включать десятки инструкций.

Вторым источником недоразумений, не позволившим с самого начала выбрать правильные исходные концепции, была ставка на использование перфокарт. Перфокарты стремительно вытеснялись из обихода под влиянием расширения использования других носителей информации, но несмотря на это, на их основе проводилась разработка концепций всего нового языка программирования JCL.

Из того, что должны были применяться управляющие перфокарты, следовало, что каждый шаг задания должен был отдельно интерпретироваться и выполняться почти полностью независимо от других, а это в действительности имело место в применявшихся ранее операционных системах. Именно в этом состоят причины создания столь неполноценного языка программирования, в котором ограничены возможности ветвления, итерации и вызова подпрограмм.

По моему мнению, именно ориентация на использование перфокарт как отдельных носителей для текстов команд стала причиной того, что никто из нас не мог понять истинное назначение JCL как языка программирования, команды которого интерпретируются и выполняются в запланированное время. Нашим основным упущением как разработчиков было то, что, создавая автомобиль, мы мыслили как пешеходы.

Еще одно упущение состояло в том, что язык JCL в действительности никогда и не проектировался — он просто рос. Если бы мы признали его в качестве языка управления системой, то, возможно, смогли бы спроектировать как настоящий язык, используя знания и опыт наших проектировщиков других языков.

Но этого не произошло. Задуманный с самого начала как "макет для нескольких управляющих перфокарт", он создавался как побочный с точки зрения достижения основной цели — проектирования самого планировщика заданий. Тем не менее по мере появления в процессе проектирования OS/360 все новых и новых задач сопровождения файловой системы, управления сетью дистанционной обработки и т.д. приходилось возлагать на язык JCL все новые и новые функции управления планированием времени или расширять состав параметров. Но в этом языке было заложено мало гибкости, общности, а также отсутствовала развитая структура, поэтому предъявление к нему новых требований приводило лишь к появлению новых параметров в виде ключевых слов, чаще всего в инструкциях DD. Поэтому ключевые слова, которые должны были быть прилагательными в объявлениях, становились императивными глаголами, в которых приходилось учитывать всевозможные результаты действий.

Усвоение уроков

1. Изучайте неудачные примеры еще более тщательно, чем полностью успешные.
2. После каждого успеха посмотрите на себя критически. Успех укрепляет веру в использованный метод проектирования, в сам проект — и в самого себя. Все это может привести к развитию самонадеянности.
3. Размышляйте над проектируемым объектом с позиций, стоящих выше этого объекта, и старайтесь представить себе среду, в которой он будет использоваться. Не произошло *ли* за истекший период изменение подходов? Остались *ли* действительными ваши предположения, которыми вы руководствовались десяток *лет* тому назад? Является *ли* проектируемый предмет именно тем, что требуется?

Примечания и ссылки

1. См. [168]. Автор цитирует оригинальное исследование [191].
2. Еще несколько глаголов было добавлено в ходе развития JCL. Текущий стандарт JCL (по состоянию на ноябрь 2008 года) приведен по адресу `http://www.isc.ucsb.edu/tsg/jcl.html`. Оригинал стандарта см. [123].

TheDesignofDesign.indb 218 30.10.2012 15:58:32

15

Становление проектирования как самостоятельной научной дисциплины

Примерно в XVI веке в большинстве европейских языков появился термин "проектирование" или его эквивалент... Возникновение этого термина прежде всего указывало на то, что произошло отделение проектирования от выполнения.

МАЙКЛ КУЛИ, 1988 Г.

◂ *Первый полет братьев Райт, г. Нэгз Хэд, шт. Северная Каролина. Библиотека конгресса США*

Становление проектирования как направления деятельности, отличного от реализации и эксплуатации

Одной из наиболее ярких тенденций в развитии проектных дисциплин в XX веке явилось постепенное становление проектирования как профессии, отличной от производства и эксплуатации.

Достаточно сравнить, как складывалась работа изобретателей в XIX и XX веке. Эдисон изготавливал рабочие версии всех своих изобретений в лаборатории. Генри Форд сам собрал свой автомобиль. Уилбер и Орвилл Райт построили свой самолет собственными руками.

А может ли столетие спустя инженер по вычислительной технике изготовить самостоятельно нужные ему микросхемы, не говоря уже о том, может ли он выплавить необходимые для этого кремний и медь из песка и медной руды? Действительно ли проектировщик самолетов разбирается во всех нюансах сложных производственных процессов самолетостроения или в том, как работает сложное программное обеспечение, применяемое для динамической стабилизации воздушного судна? Разве сам архитектор занимается разработкой строительной конструкции здания и его укреплением с учетом требований сейсмостойкости?

Аналогичным образом проектировщик отделен от пользователя и во многих других областях производственной деятельности. Поэтому, что касается, например, архитектуры, то проектировщики больниц, административных зданий, предприятий по переработке ядерного топлива или лабораторий биофизики вряд ли могут опереться на свой личный опыт эксплуатации подобных сооружений в качестве пользователей и должны полагаться на результаты исследований предполагаемого поведения пользователей с привлечением представителей пользователей или, что еще хуже, людей, действующих от имени пользователей, которые сами удалены на шаг или два от реальных пользователей. Вряд ли можно найти такого проектировщика военных кораблей, который сам бы командовал кораблем, не говоря уже о его участии в морском бою.

В прежние времена проектировщики находились во многом в другой ситуации. Это положение усугубляется и в связи со стремительной сменой поколений технических средств. Проектированием современных автомобилей занимаются главные инженеры проектов, которые провели свои юные годы, ремонтируя и гоняя по дорогам автомобили, которые в наши дни давно считаются устаревшими. Многие современные ведущие специалисты по телекоммуникациям могли начать свое приобщение к этой отрасли с получения лицензии на радиосвязь и сборки приемопередатчика на одной радиолампе в средней школе. В Великобритании многие работающие инженеры-механики прошли подготовку по применявшимся в свое время "сэндвичевым" программам 1-3-1: год практического обучения в компании, три года учебы в университете за счет компании и еще один год практики до начала проектирования. Многие американские инженеры подготовлены

в рамках программ совмещения учебы с практикой, которые предусматривали чередование теоретических занятий с приобретением опыта на производстве.

Но, к счастью, проектировщики не всегда теряют связь с производственной деятельностью. Например, такая отрасль, как разработка программного обеспечения, еще настолько молода, что большинство работающих в ней системных архитекторов сами в свое время занимались программированием. Проектировщики таких товаров личного пользования, как планшеты iPod, телефоны iPhone и автомобили, прежде всего, сами являются пользователями этих товаров и воплощают в своих проектах собственные представления о путях дальнейшего усовершенствования проектов.

Проектировщики таких операционных систем, как UNIX и особенно Linux (с открытым исходным кодом), исходили из собственных потребностей, создали инструментальные средства для личного пользования и лишь затем поделились результатами своей работы со всеми заинтересованными лицами. По моему мнению, именно этим объясняется успешность многих инициативных проектов и приверженность этим проектам других пользователей.

Причины становления проектирования как самостоятельного направления

Первая причина является очевидной. В XX веке достигнуты ошеломляющие успехи во всех областях науки и техники, а развитие технологий производства достигло таких высот, что потребовалась все более узкая специализация инженерной деятельности, а подготовка специалистов стала занимать все больше и больше времени. Появились даже такие отдельные специальности с полной занятостью, как прогнозирование землетрясений и разработка композитных материалов.

Вторая причина является менее очевидной, но, возможно, столь же весомой. Проектируемые изделия стали настолько сложными, что требуется специализация даже в самом проектировании, продолжительность обучения проектировщиков увеличивается, а трудозатраты таковы, что поглощают всю энергию специалиста. В наши дни не осталось фактически ни одной простой технологии. Высокие требования предъявляются даже к процессу производства печенья, поскольку готовый продукт должен иметь не только хороший вкус, но и достаточно продолжительный срок хранения, а тесто должно быть отделено от начинки[1].

Отрицательные последствия отделения проектирования от других дисциплин

Что же произошло в конечном итоге? Какие последствия мы наблюдаем? *Частое непонимание своей задачи!* Архитекторы строят изящные здания, в которых трудно работать. Инженеры проектируют панели управления ядерными реакторами, которые с большим трудом осваивают операторы атомных электростанций.

Проектировщики, стремясь до предела упростить реализацию, заходят слишком далеко, поэтому функциональная пригодность или производительность готового изделия становится неприемлемой. Обмен информацией между пользователями и проектировщиками, а также между проектировщиками и производственниками становится крайне ограниченным. Человеку гораздо проще самому заниматься и постановкой, и решением задач, чем кого-то привлекать для того или другого. Поэтому все чаще наблюдаются примеры того, как отсутствие полного понимания между специалистами разных направлений приводит к разрушительным, дорогостоящим или непредвиденным последствиям.

Способы исправления существующего положения

Прежде всего необходимо признать следующее: проектировщики обязаны осознать, что уже в XX веке их профессия полностью отделилась от практики, поэтому предпринимать целенаправленные и сосредоточенные усилия для смягчения отрицательных последствий происшедшего.

Способ 1. Экспериментирование с помощью сценариев

Зачастую приносит пользу даже весьма небольшой объем экспериментов на основе сценариев практической эксплуатации. В тех условиях, когда отсутствует возможность провести эксперименты на реальном объекте, полезные результаты могут быть получены и с помощью качественной модели. Макеты в натуральную величину обеспечивают пробные прогоны сценариев работы на рабочем месте или за пультом управления. Это означает, что во многих случаях может принести пользу создание виртуальной среды.

После получения задания на проектирование консоли оператора для компьютера IBM Stretch я имел возможность пользоваться только протоколами наблюдения за работой операторов на компьютере, практически не имея данных об относительной частоте и значимости выполняемых ими функций.

К тому времени коллектив разработчиков компьютера Stretch получил двухнедельный летний отпуск, во время которого многие отправились на отдых. Но я вместо этого стал посещать вычислительный центр, в котором эксплуатировались компьютеры модели 709 для лабораторий Пафкипси (Poughkeepsie), и выполнял в течение двух недель функции стажера при операторе компьютера.

За этот период я узнал очень многое. Чаще всего я занимался монтированием лент, но стремился почувствовать "биение сердца" научного вычислительного центра и внимательно наблюдал за тем, как работали ведущие операторы[2].

Этот опыт "пользователя" привел меня к выводу, что проект первой консоли оператора должен быть основан на программном управлении (иными словами, что по существу должен быть создан жестко связанный терминал), поэтому необходимо отказаться от идеи создания консоли, непосредственно зависимой от структуры аппаратных средств и ориентированной на управление их работой.

Такой подход должен был обеспечить применение многочисленных консолей многими операторами и гибкое распределение задач между операторами, а также интерактивную отладку программ в реальном времени.

Но я должен признать, что спроектированная мною чрезмерно сложная консоль редко использовалась в каких-либо практически эксплуатируемых компьютерах Stretch по такому принципу, как было задумано. Интерактивная отладка в реальном времени вошла в повседневную работу лишь значительно позже, что было достигнуто отчасти благодаря внедрению разработанной Тэдом Коддом мультипрограммной операционной системы для компьютера Stretch, которая пришла на смену стандартного программного обеспечения Stretch[3].

Тем не менее полученный опыт пригодился после моего подключения к проекту создания операционной системы OS/360. Появились все возможности для широкого внедрения интерактивной отладки в реальном времени, а полученные ранее результаты эксплуатации показали, как создать более простую консоль и обеспечить полную программную поддержку.

Столь же значительный опыт был получен мною во время творческого отпуска, который проходил в течение семестра в биохимической лаборатории Дейва и Джейн Ричардсонов в г. Дьюк. Повседневно наблюдая за решаемыми в лаборатории задачами, я понял, что должны быть созданы графические инструментальные средства визуализации молекул, с помощью которых можно было бы изучать структуру и функции белка.

Филипп Крачтен (Philippe Kruchten) систематизировал такого рода опыт, выполняя функции главного архитектора системы управления воздушным движением Канады:

> *Все причастные к разработке программного обеспечения были направлены для практического обучения в службу управления воздушным движением, посещали занятия по теории управления воздушным движением, а затем просиживали целые дни рядом с контролерами в оперативном зональном центре управления, пытаясь понять, в чем суть их деятельности. Аналогичным образом, специалисты по теории управления воздушным движением были направлены на такие курсы, как объектно-ориентированное проектирование и программирование на языке Ada, а это позволило достичь такого положения, что специалисты в той и другой области освоили достаточно большой общий словарь, чтобы эффективно сотрудничать и совершенствовать навыки друг друга[4].*

Способ 2. Тесное взаимодействие с пользователями на основе инкрементной разработки и итерационной поставки

Предложенная Харланом Миллсом (Harlan Mills) система инкрементной разработки и итерационной поставки представляет собой наилучший способ организации и поддержания самого тесного взаимодействия с пользователями непосредственно от запуска проекта[5]. При этом проектировщики прежде всего

создают работающую версию с минимальным набором функций, передают ее пользователям для эксплуатации или, по крайней мере, проводят контрольный прогон. С участием наиболее типичных пользователей можно успешно испытывать даже такие товары, которые предназначены для широкого рынка.

В моей собственной практике случилось такое, что при создании и испытании интерактивных графических систем в качестве инструментальных средств для ученых наблюдалась весьма неожиданная реакция первых пользователей на предлагаемый нами прототип системы. Почти неизменно мои предположения в части того, как пользователи будут работать с новым инструментальным средством, оказывались неправильными.

Возглавляемый мной коллектив затратил почти десять лет на воплощение нашей мечты о создании виртуального изображения белка "величиной с целую комнату". Я считал, что химики смогут легче понять структуру сложной молекулы, если будут в этом огромном изображении видеть, где находятся связи типов C и N. Испытав много разочарований, мы наконец смогли определить требования к изображению с высоким разрешением, для реализации которых оказалось достаточно применить портативный дисплей, который крепится на шлеме. Химики получили возможность сами передвигаться по структуре белка и изучать интересующие их участки.

Наш первый пользователь посетил нас в рамках своей регулярной двухнедельной командировки; освоение инструментария шло нормально и достаточно быстро. На следующем занятии было то же самое.

А третье занятие принесло неожиданный результат: "Можно ли сделать так, чтобы я оставался на месте, а двигалась модель?" Итог десятилетней работы был перечеркнут одним предложением! Оказалось, что отдача от использования средств навигации по структуре молекулы не стоила затрачиваемых при этом физических усилий.

Аналогичным стал наш опыт создания системы планирования лучевой терапии. Задача радиолога состоит в том, чтобы правильно сориентировать многочисленные лучи, которые бы разрушали опухоль, но не затрагивали здоровые ткани и чувствительные органы, особенно такие как глаза. Мы подвешивали полупрозрачную виртуальную модель тела пациента в пространстве, что позволяло врачам двигаться вокруг нее и рассматривать с любой точки зрения. Но пользователи категорически потребовали, чтобы им была предоставлена возможность самим спокойно сидеть и поворачивать виртуальную модель пациента на любой угол.

Способ 3. Параллельная разработка

Проектировщики должны занимать более активную позицию и лично участвовать в ходе реализации на всех ее этапах. Иногда даже фрагментарный и нетипичный опыт реализации может неожиданным образом дополнить идеализированное или исправить ложное представление проектировщика о том, как

происходит воплощение проекта в жизнь. Поэтому я настоятельно рекомендую проектировщикам как можно чаще бывать на производстве.

Тем не менее всегда остается опасность, что скромный опыт реализации, отражающий только типичные ситуации, не позволит достичь достаточно высокого качества проектирования, если проектировщик может опереться лишь на то, что он лично усвоил, принимая участие в реализации, ведь такой опыт по самой своей природе остается сосредоточением фрагментарных наблюдений. По-видимому, наибольшей отдачи от усилий специалистов всех категорий можно достичь, применяя параллельную разработку в качестве основного подхода к проектированию. При этом подходе в процессе проектирования непосредственно участвуют те, кому в дальнейшем придется заниматься реализацией проекта; эти специалисты, главным образом конструкторы, определенно обладают гораздо более широким опытом практической реализации проектируемых объектов, поэтому способны компенсировать ограниченный набор представлений проектировщика в области реализации. (В области создания программного обеспечения аналогичный подход иногда называют просто *адаптивным методом разработки*.)

Более широкое привлечение конструкторов к участию в процессе проектирования позволяет лучше учитывать их запросы. С другой стороны, рядовые рабочие, даже имеющие превосходные навыки воплощения типичных технических чертежей в реальных изделиях, могут оказаться менее способными заранее представить себе готовую конструкцию, изучая чертежи общего вида и чертежи отдельных деталей, поэтому не могут в такой же степени помочь, когда требуется выявить ошибки или понять, какие сложности возникнут при реализации, в чем я убедился, наблюдая за работой верфи. Процесс параллельного проектирования можно усовершенствовать, обеспечив применение вместе с чертежами общего вида и чертежами отдельных деталей более развитые средства визуализации, вплоть до систем создания виртуальной среды.

Способ 4. Обучение проектировщиков

Назрела необходимость включить в учебные планы подготовки проектировщиков оба указанных выше метода, а также обеспечить прохождение практики в части изучения требований и пожеланий пользователей[6].

В своей классической статье, которая вышла в 1985 году и до сих пор не потеряла актуальности, Гоулд (Gould) и Льюис (Lewis) изложили три принципа проектирования, поставив на первое место изучение потребностей пользователей и стоящих перед пользователями задач "путем вступления с пользователями в непосредственный контакт с самого начала проектирования". Эти авторы, проводя свои исследования, обнаружили, что многие проектировщики, которые считали себя тесно работающими с пользователями, фактически лишь слушали или читали описания потребностей пользователей, исследовали профили пользователей, а затем, как они выражались, "проводили презентации, консультации или проверки проектов" с участием пользователей на последних этапах[7].

Иными словами, для обучения проектировщика не менее важным, чем все прочее, является приобретение опыта работы в механическом цехе, на строительной площадке или в группе разработчиков программного обеспечения.

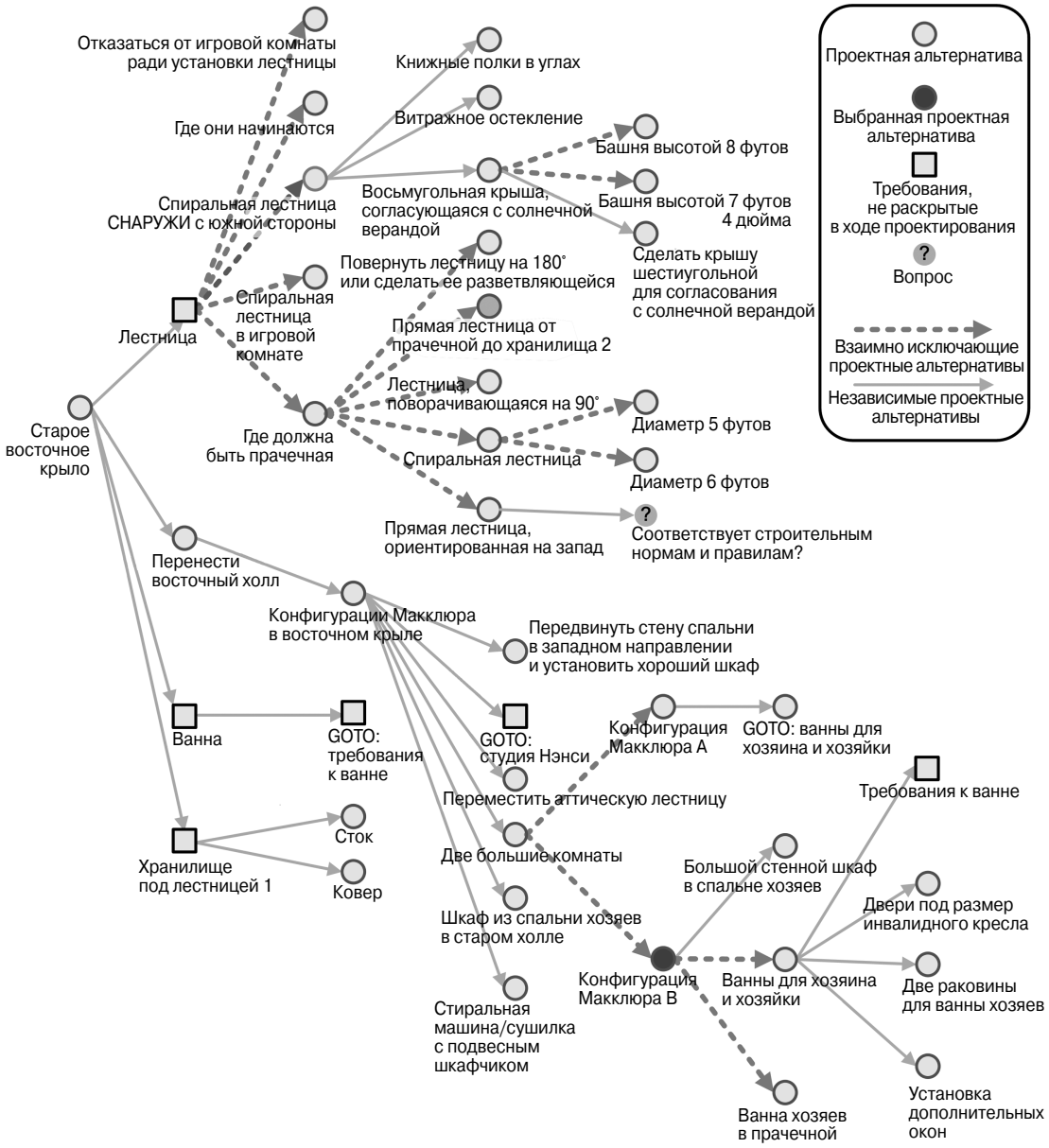
Будущие проектировщики обязательно должны приобретать опыт непосредственного общения с пользователями и практической реализации, поэтому в планы их обучения необходимо вводить соответствующие курсы и занятия, пусть даже для этого потребуется сократить количество часов теоретического обучения. Безусловно, следует также вести обучение использованию инструментальных средств на основе аналитических методов и формальных методов синтеза, а что касается наиболее усовершенствованных методов, то проектировщик может их осваивать самостоятельно по мере необходимости. Что же касается доведения правильных навыков проектирования до уровня инстинктов, то дела обстоят сложнее. В современных планах обучения проектированию должно быть учтено, что проектирование приобрело статус отдельной профессии, в связи с чем приходится предпринимать напряженные усилия по приобщению молодых проектировщиков к реальному миру реализации и эксплуатации.

Примечания и ссылки

1. См. [90].
2. А также вслушивался в звуки. Я разделяю ностальгию Гради Буча: "Мне не хватает звуков, которые сопровождали работу компьютеров старых поколений. Я мог определить, что делает моя программа, по звуку, исходящему от компьютера".
3. См. [64].
4. См. [132]. Автор далее сообщает: "Кое-кто вставлял палки в колеса, протестуя против кажущихся ненужными затрат времени, но в дальнейшем все были буквально поражены тем, насколько это помогло им выполнять свою работу".
5. См. [153].
6. Просмотрев примерно 22 предложения по организации лабораторного курса разработки программного обеспечения, я признал необходимым и возможным требовать привлечения сторонних пользователей, с которыми должны работать студенческие коллективы и требования которых они должны удовлетворять. Пользователи должны были выделять время для еженедельных встреч с коллективом, в ответ на что они могли бы получить удобные для использования прототипы программного обеспечения.

Я поддерживал проекты, которые были бы полезными в случае успеха, но не обязательно необходимыми. Студенческому коллективу нужно дать право на неудачу.

7. См. [106]. По словам автора: “Эти принципы таковы: заблаговременное и непрерывное сосредоточение на потребностях пользователей; эмпирическое измерение полезности и итерационное проектирование”.


Проектная альтернатива
Выбранная проектная альтернатива
Требования, не раскрытые в ходе проектирования
Вопрос
Взаимно исключающие проектные альтернативы
Независимые проектные альтернативы
Старое восточное крыло
Лестница
Отказаться от игровой комнаты ради установки лестницы
Где они начинаются
Спиральная лестница СНАРУЖИ с южной стороны
Спиральная лестница в игровой комнате
Книжные полки в углах
Витражное остекление
Восьмугольная крыша, согласующаяся с солнечной верандой
Башня высотой 8 футов
Башня высотой 7 футов 4 дюйма
Сделать крышу шестиугольной для согласования с солнечной верандой
Повернуть лестницу на 180° или сделать ее разветвляющейся
Прямая лестница от прачечной до хранилища 2
Где должна быть прачечная
Лестница, поворачивающаяся на 90°
Спиральная лестница
Прямая лестница, ориентированная на запад
Диаметр 5 футов
Диаметр 6 футов
Соответствует строительным нормам и правилам?
Перенести восточный холл
Конфигурации Макклюра в восточном крыле
Передвинуть стену спальни в западном направлении и установить хороший шкаф
Ванна
GOTO: требования к ванне
GOTO: студия Нэнси
Конфигурация Макклюра A
GOTO: ванны для хозяина и хозяйки
Переместить аттическую лестницу
Две большие комнаты
Шкаф из спальни хозяев в старом холле
Стиральная машина/сушилка с подвесным шкафчиком
Хранилище под лестницей 1
Сток
Ковер
Конфигурация Макклюра B
Большой стенной шкаф в спальне хозяев
Ванны для хозяина и хозяйки
Требования к ванне
Двери под размер инвалидного кресла
Две раковины для ванны хозяев
Установка дополнительных окон
Ванна хозяев в прачечной

16

Представление процессов и критериев проектирования

В сотрудничестве с Шарифом Раззаком (Sharif Razzaque)

Есть много способов совершить ошибку, выбрав неправильный подход к использованию цифровой ЭВМ, поэтому еще одна ошибка вряд ли изменит суть дела.

См. [222]

Выбирайте как можно тщательнее слова, описывая то, что вам непонятно.

◂ *Часть диаграммы из архива проекта дома Бруксов, в которой показан ход проектирования флигеля. Шариф Раззак*

Введение

Для проектировщиков один из путей приобретения наибольшего объема знаний, исходя из опыта каждого проекта, состоит в подробном документировании хода развития проекта; в этом документе должны быть отражены не только объекты проектирования, но и способы решения задач проектирования. Кроме того, в подобном документе должны быть описаны критерии проектирования, что становится неоценимой помощью для тех, кто занимается сопровождением готовой системы; это позволяет предотвратить появление многочисленных ошибок, вызванных неосведомленностью. Однако документирование процессов проектирования и критериев проектирования представляет собой намного более сложную задачу, чем кажется на первый взгляд[1].

Проблемой создания компьютерных инструментальных средств, способных оказать помощь в этом деле, занималось несколько исследовательских групп[2]. Мы с доктором Шарифом Раззаком также решили внести свой вклад в подготовку компьютерного представления хода разработки конкретного проекта. В свое время я со своей женой Нэнси занимались проектированием пристройки к нашему дому площадью 1700 кв. футов и весь ход этого процесса описывали обычным языком в журнале, объем которого в целом составил 235 страниц. (Разработка этого проекта кратко описана в главе 22. Значительная часть дерева проектирования, относящегося к этому проекту, представлена на веб-сайте, сопровождающем настоящую книгу.)

В данной главе описаны наши наблюдения, касающиеся того, каковым является характер реального процесса проектирования и как следует документировать процессы проектирования. Мы разделяем свою ответственность за изложенное в ней следующим образом.

- Мы отвечаем за работу, сделанную вместе.
- Раззак отвечает за ту работу, которую он сделал сам.
- Раззак отвечает за свои комментарии и выдвинутые им гипотезы.
- Наша ответственность становится общей, если мы оба выражаем с чем-то согласие.

Линеаризация сети знаний

Ванневар Буш (Vannevar Bush) в описании проекта предложенной им системы Memex указал, что для представления всех взаимосвязей между элементами знаний, как правило, требуется граф общего вида, а не плоский граф[3].

Но такой граф трудно представить и почти невозможно понять. Поэтому представители всех научных областей стремятся линеаризовать используемые ими представления знаний и дополняют это линейное представление одним или несколькими вспомогательными представлениями.

Обычно применяемый при этом процесс линеаризации пространственного графа состоит в следующем.

1. Удаление ребер в графе в целях преобразования этого графа в дерево. Этот процесс приводит к созданию иерархической структуры даже в той совокупности взаимосвязей, в которой таковая до сих пор не существовала, независимо от того, является ли такая структура желательной или нет.
2. Отображение дерева на линейный граф одним из нескольких известных способов, как правило, с применением алгоритма поиска в глубину.

Рассмотрим, например, научную книгу. Темы, затрагиваемые в книге, сплетены в сложный клубок взаимосвязей. Но структура самой книги не может быть иной, чем линейной: страница следует за страницей; строка следует за строкой; слово — за словом. Поэтому автор организовывает совокупность тем в виде дерева и показывает это дерево в оглавлении, где главы состоят из разделов, а разделы из подразделов. Номера страниц показывают, как дерево тем преобразуется в линейную форму.

Но оглавление не является единственным дополнением, позволяющим воссоздать структуру взаимосвязанных тем. В конце книги находится предметный указатель, в котором тематическое содержание книги представлено в виде словаря терминов, которые перечислены в алфавитном порядке. Номера страниц, где можно найти определения любого применяемого в книге термина, по существу выстраиваются в цепочку, пронизывающую всю книгу. В предметном указателе восстанавливаются многие связи, которые были исключены при преобразовании совокупности тем, представляющих содержимое книги, в дерево оглавления.

Осуществление подобного процесса применительно к достаточно большой совокупности книг приводит к организации библиотеки. В системе нумерации библиотеки Конгресса США (называемой также *десятичной системой Дьюи*) весь огромный объем хранимых книг рассматривается как взаимосвязанный и отображается в дерево. Это дерево по принципу обхода с поиском в глубину отображается в линейный граф, который, в свою очередь, показывает, как книги должны быть расставлены по полкам библиотеки. И это отображение дополняется предметными указателями, в каждом из которых восстанавливаются оборванные связи, а темы, затронутые в книгах, выстраиваются в цепочки, такие как указатель авторов, указатель названий, тематический указатель.

Особый интерес представляет тематический указатель, поскольку сама расстановка книг на полках уже учитывает распределение книг по основным темам. В тематическом указателе также учтено, что любая научная работа затрагивает не только свою основную тему, но и целый ряд дополнительных тем.

А в такой энциклопедии, как Википедия, отображение пространственной структуры совокупности знаний достигается благодаря применению развитого набора перекрестных ссылок, позволяющих мгновенно переходить к другим

источникам по интересующей теме. Эта возможность становится существенным дополнением к нашему интеллектуальному инструментарию.

Аналогичную многомерную структуру имеет любое пространство проектирования, поэтому в ходе разработки проекта приходится решать сложную задачу представления знаний, касающихся данного проекта. Итак, нелегко представить эффективно даже неизменные по сути сведения о проектируемом объекте, а что касается динамически развивающихся процессов проектирования, то задача их представления изначально является еще более сложной.

Следует отметить, что в главе 2 книги Саймона "Rational Model of design" принята гипотеза о возможности применения дерева проектных решений, которое показывает в каждом альтернативном узле вторичные проектные решения, основанные на принятии соответствующего выбора. В идеальном случае следовало бы приводить для каждого выбора критерии, на основании которых было принято данное решение. Но необходимо также учитывать, что между самими решениями могут быть установлены многочисленные и сложные взаимосвязи, поскольку наряду с обоснованиями отдельно взятых решений имеются такие предпосылки, исходя из которых могут быть приняты равнозначные или в большей степени отличные друг от друга решения.

Сбор данных, касающихся процесса проектирования

Мы поставили перед собой цель перевести в конкретную форму неявно сформированное дерево проектирования флигеля дома Бруксов, чтобы, во-первых, дополнить краткое текстовое описание практического примера процесса проектирования, приведенного в главе 22, и, во-вторых, показать, как процесс проектирования развивается во времени. Но еще более важно то, что мы стремились уточнить многие нюансы процесса проектирования дома Бруксов, в частности, перечисленные ниже.

- Не противоречат ли друг другу записи журнала с описанием хода проектирования и воспоминания Фреда Брукса?
- Происходило ли временами столкновение мнений и по какому поводу?
- Какие спорные моменты удалось разрешить и каким образом?
- Занимались ли Фред и Нэнси Бруксы систематическим исследованием дерева проектирования?
- Подтверждают ли результаты проведенного анализа те выводы, которые изложены в оставшейся части этой книги?

Как оказалось, результаты, полученные при попытке восстановить дерево проектирования, были более познавательными, чем изучение структуры самого дерева. В действительности изучение полученного дерева проектирования принесло

настолько мало информации, что даже вызвало разочарование. Мы признали данный этап работы экспериментом, который окончился неудачей.

Принятая нами методика изучения процесса проектирования дома

Мы начинали с поиска имеющегося в наличии программного обеспечения, которое позволяло бы строить изображения деревьев проектирования. В конечном итоге мы остановились на программе Compendium[4], представляющей собой инструментальное средство, позволяющее регистрировать и фиксировать шаги процесса проектирования в ходе его развертывания.

Я положил в основу дерева проектирования заметки, приведенные на первой странице нашего журнала, и продолжил постраничную обработку журнала, переводя сведения из заметок в узлы и связи дерева проектирования в соответствии с письменной схемой преобразования, которая была подготовлена мною предварительно. Я сразу же столкнулся с трудностями, которые вынудили меня усомниться в том, что ведение журнала осуществлялось должным образом. В связи с этим мне пришлось изменить схему преобразования и тем самым отклониться от подразумеваемых рекомендаций по эксплуатации программы Compendium.

В конечном итоге была создана описанная ниже методика работы. После каждой корректировки схемы преобразования я возвращался назад на одну страницу в журнале и перерабатывал дерево в программе Compendium, чтобы оно соответствовало новой схеме. После этого я продолжал обработку журнала, что неизбежно приводило к обнаружению конфликтующей записи журнала, которая не вписывалась в принятую схему преобразования. Это становилось причиной повторного исследования. Проектирование осуществлялось следующим образом.

- Применение (и развитие) схемы преобразования.
- Использование программы Compendium для восстановления дерева решений.
- Осуществление самого процесса проектирования.

Эта методика, в ходе осуществления которой происходило обнаружение недостатков в схеме преобразования, устранение этих недостатков и повторное выполнение предыдущих этапов, применялась снова и снова, изо дня в день, что привело к постепенному уменьшению необходимости в повторах. Нам удалось постепенно выработать намного более приемлемую схему, что позволило даже смириться с оставшимися ее недостатками и добиться существенного прогресса в формировании дерева проектирования на основе записей журнала.

Определение понятия дерева проектирования

Лишь много позже я понял, что проблемы поиска схемы преобразования для дерева проектирования были в основном обусловлены отсутствием точного определения понятия дерева проектирования. Я мысленно руководствовался определением, которое оказалось неформальным, неявным и расплывчатым. Осуществлявшийся мною поиск пригодной для использования схемы преобразования дерева оказался одновременно поиском определения понятия дерева проектирования, которое было бы строгим, всесторонним и достаточно точным, чтобы быть практически применимым.

Но первоначально применявшееся мною определение было неформальным и неявным, поэтому мне не удавалось изобразить на листе бумаги даже то, как примерно должно выглядеть дерево проектирования, не говоря уже о создании схемы проектирования, независимой от программного инструментального средства.

Первоначальное расплывчатое представление о дереве проектирования выглядело примерно так, как показано на рис. 2.1. Это представление напоминает дерево опций, которое мог бы построить заказчик, определяя конфигурацию ноутбука, который должен быть собран в соответствии с его требованиями. Каждый вопрос, который должен быть рассмотрен в ходе проектирования (т.е. вопрос, по поводу которого должно быть принято решение), изображается в виде узла. Равнозначные рассматриваемые вопросы проектирования, такие как, на примере ноутбука, характеристики визуального отображения информации и требования к системе передачи сообщений о событиях, являются ортогональными по отношению друг к другу, и проектировщик должен ответить на каждый из них. В книге [25] такие вопросы проектирования именуются *ветвями атрибутов*.

Каждый узел вопроса проектирования имеет по одному дочернему узлу для каждого из вариантов проектных решений. В примере с ноутбуком необходимо, в частности, выбрать один из возможных размеров дисплея. Предусмотренные при этом варианты выбора представляют собой взаимно исключающие альтернативные ветви. Проектировщик выбирает одну из ветвей для каждого независимого вопроса проектирования.

Большинство вариантов выбора является таковым, что за ними следуют очередные вопросы проектирования (например, если в разрабатываемом устройстве решено использовать светящуюся шкалу, то приходится выбирать способ подсветки). Эти узлы вопросов проектирования являются дочерними по отношению к предшествующим им узлам решений. Таким образом, подобное дерево решений включает и независимые, и взаимно исключающие варианты выбора проектных решений, а конечный продукт представлен не как выбор единственного узла, а скорее как набор из многих альтернативных узлов проектных решений, по одному на каждый листовой узел, выбранный в ответ на каждый независимый вопрос проектирования.

В дереве могут быть также представлены критерии выбора, для чего с каждым вариантом выбора должны быть связаны узлы, представляющие аргументы за и против выбора этого узла. Кроме того, каждый узел вопроса проектирования должен иметь связанный с ним узел с описанием того, какой выбор сделан и почему.

Таковым было первое определение понятия дерева решений с критериями, которое, по-видимому, наиболее полно соответствовало типам узлов, применяемым в программе Compendium. Я выбрал описанное ниже отображение. Каждый вопрос проектирования был представлен в виде пиктограммы вопроса. Каждая альтернатива проектирования обозначалась пиктограммой "Идея". С каждой пиктограммой "Идея" были связаны дочерние пиктограммы "За" и "Против". Выбранная альтернатива обозначалась пиктограммой "Согласие" с примечанием, описывающим, в чем состоят причины согласия с принятым решением.

Наконец, было решено, что вопросы проектирования, даже независимые, должны быть упорядочены в виде иерархической структуры, с учетом того, к какой части проектируемого дома они относятся. Например, предполагалось, что данные обо всех проблемах проектирования гостиной будут сгруппированы под одним узлом "Гостиная", что позволит учесть структуру и принятые обозначения записей журнала.

Брукс с самого начала разделил задачу проектирования на три отдельные группы проблем (рис. 16.1).

Новые данные, полученные в процессе проектирования

Проектирование должно служить не только выполнению известных, но и раскрытию до сих пор неизвестных требований

Постраничный анализ журнала вскоре показал, что даже после определения программы архитектурного проектирования требования продолжали изменяться. Например, приглашенный архитектор Вес Макклюр (Wes McClure) предложил добавить два павильона, с северной и с южной стороны[5]. Семейство Бруксов в конечном итоге отклонило эту идею, поскольку, во-первых, после этого в доме не было бы единственного центрального места, больше всего подходящего для встреч в составе всей семьи, и, во-вторых, строительство южного павильона потребовало бы спиливания большого всеми любимого черного дуба. Но интересно то, что ни наличие общего центра сбора, ни сохранение дуба не было заведомо сформулировано как требование к тому времени, когда Макклюр предложил свои павильоны. Анализируя предложенное проектное решение, связанное со строительством павильонов, Бруксы столкнулись с необходимостью рассмотреть эти требования.

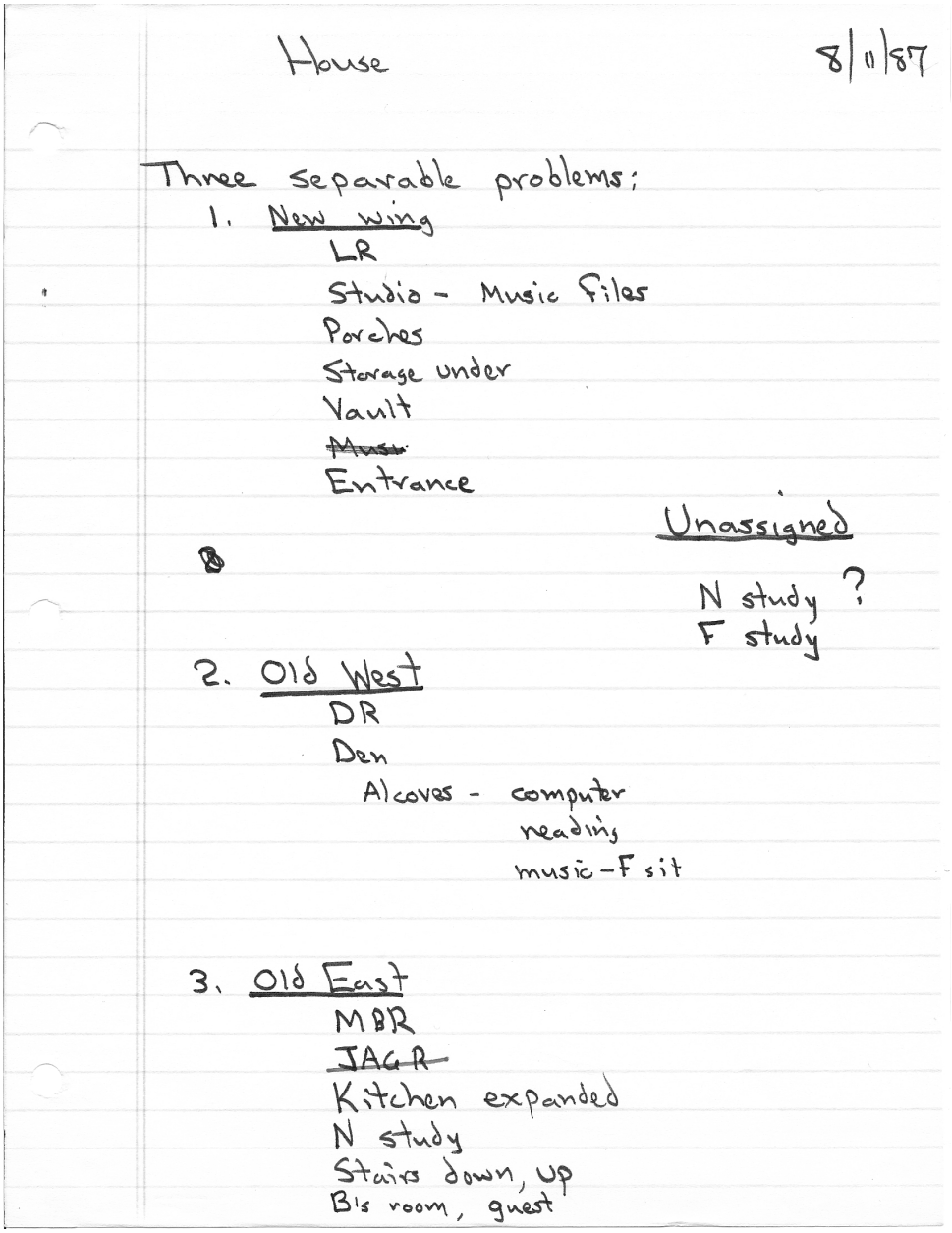


Рис. 16.1. Разделение задачи проектирования флигеля дома на отдельные проблемы

По мере рассмотрения журнала аналогичный ход событий повторялся снова и снова. Иными словами, цель проектирования состоит не только в удовлетворении известных требований, но и в выявлении неизвестных. Полученный нами опыт

подтверждает теорию Шёна, о которой шла речь в главе 3. Хорошо продуманный процесс проектирования должен способствовать проявлению указанной тенденции, а не ее подавлению.

Проектирование — это не просто выбор среди альтернатив, но и осознание их существования

После того как перед проектировщиком возникает очередная проблема, касающаяся проектирования, ему приходится определять все возможные альтернативные варианты решения, в том числе те, с которыми до сих пор ему не доводилось сталкиваться. Некоторые варианты решения являются очевидными или известными по предыдущему опыту (возможно даже, заимствованными из образцов). Но есть и такие решения, которыми проектировщик не пользовался ранее и поэтому нуждается в их освоении. Например, журнал показывает, с каким трудом семья Бруксов проводит выбор между двумя возможными конфигурациями музыкального зала, причем после выполнения большого объема аналитической работы и нескольких попыток компенсировать недостатки одного и другого варианта им так и не удается выбрать наиболее подходящий. Наконец, Бруксы обнаруживают третью конфигурацию, которая им сразу же кажется приемлемой. Примечания в журнале: "Конфигурация С-А действительно ведет к цели!"

Аналогичные ситуации снова и снова находят свое отражение в журнале. Как уже было сказано в главе 3, покупка Бруксами участка земли у соседа не рассматривалась с самого начала как очевидное решение задачи размещения музыкального зала. Как оказалось, проектирование во многом представляет собой процесс осознания того, что существуют не учтенные ранее варианты решения задач проектирования. И в этом полученный нами опыт подтверждает теорию Шёна.

Изменение дерева проектирования в ходе развития проекта и способы представления этого изменения

Три указанные отдельные проблемы проектирования складываются в структуру, которая весьма напоминает дерево выбора конфигурации ноутбука, за исключением того, что узлы самого низкого уровня в этой структуре не являются листовыми, а скорее представляют собой проблемы, требующие разработки вспомогательных проектов. Но даже само дерево показывает, что выполненный ранее объем проектирования может успешно использоваться в дальнейшей работе. Например, каковы причины размещения входа в новом флигеле? Успешным решением оказалось и то, что входная дверь дома была размещена в новом флигеле, а не в западной части, как было задумано ранее. Дело в том, что некоторые проектные решения были приняты без явного развертывания дерева проблем проектирования.

Поэтому при просмотре журнала было обнаружено, что некоторые комнаты в процессе проектирования были размещены в одном, а не в другом крыле здания по неизвестным причинам. После этого Брукс понял, что он рассматривает свой дом одновременно и как задуманный первоначально (до разработки проекта строительства), и как спроектированный в конечном итоге. Например, в последнее время Брукс считает, что студия Фреда должна быть спроектирована как детская в новом флигеле, потому что так было задумано вначале. А когда Брукс заполнял первую страницу журнала, он решил расположить студию Фреда на первом этаже, но не определил ее точного местонахождения. Но в период от начала проектирования до его завершения прошло много этапов.

На каждом этапе были приняты важные проектные решения, которые влекли за собой существенное изменение организационной древовидной структуры проекта. При каждом изменении в проекте одни и те же комнаты и части дома часто становились сгруппированными под иными узлами высокого уровня. Например, в исходном проекте дома и в проекте от 1987 года было предусмотрено размещение спальни хозяев в восточном флигеле, тогда как согласно окончательному проекту эта спальня должна находиться в новом флигеле. Какая древовидная структура должна быть принята в конечном итоге? Как показать развитие процесса проектирования во времени?

Нами был принят такой подход, который предусматривает привязку всех узлов дерева проектирования к организационной структуре помещений в построенном доме, что позволяет заложить в основу дерева проектирования стабильную структуру. Окончательный проект дома был разделен на две фазы, первую, которая, в свою очередь, предусматривала разделение на новый флигель и существующую западную часть дома, и вторую, в которой рассматривалась кухня и игровая комната.

Некоторые альтернативные варианты проектирования являются настолько важными, что для них должны быть отведены узлы, расположенные достаточно высоко в дереве проектирования. Например, альтернатива "Провести зеркальную перестановку проектируемых комнат во всем здании" должна быть показана на более высоком уровне, чем фаза I и фаза II (или новый флигель, существующая западная часть и существующая восточная часть). Но данный конкретный узел должен быть исключен из окончательного варианта дерева проектирования как вызывающий ненужную путаницу, ведь он больше не будет рассматриваться в процессе проектирования. Если же такой узел высокого уровня будет сохранен, то в дальнейшем станет постоянно привлекать внимание и мешать в работе.

Мы рассмотрели возможность сгруппировать данные обо всех таких непродолжительных исследованиях в узле верхнего уровня, назвав его "Поиски на ранних этапах", поскольку обнаружили, что во многих проектах в начале разработки происходит рассмотрение нескольких радикально отличающихся друг от друга вариантов. Но отказ от альтернативных вариантов решения задач проектирования

происходит и в каждой стадии разработки проекта. По существу, не возникает значительного различия в тех случаях, когда отказ от какого-то варианта происходит на самой ранней стадии или на более поздней, за исключением того, что отказ на более ранней стадии затрагивает бо́льшую часть дерева; по мере развития и стабилизации процесса проектирования далеко идущих изменений становится все меньше и меньше.

Но со временем изменяется и сама структура дерева решений. Для документирования таких изменений требуются новые динамические инструментальные средства, которые еще не существуют. Такие средства должны позволять отслеживать не только то, как растет дерево в результате развертывания ветвей из листовых узлов, но и то, как происходит вырезка узлов и связанных с ними поддеревьев в одной ветви и вставка в другую ветвь.

Сравнение дерева решений с деревом проектирования

Как показано на рис. 2.1, полностью законченный проект (т.е. готовый продукт) представлен не единственным узлом в дереве решений, а скорее множеством листовых узлов. Изучение такого дерева обычно не позволяет понять, является ли отраженное в нем состояние проектирования соответствующим наилучшему возможному на данный момент законченному проекту.

Для представления пространства всех возможных проектов требуется совсем другое дерево, в котором каждый узел задает поддерево проектных продуктов, как и все связанные с этим узлом решения, дополнительно уточняющие проект. Мы называем такую древовидную структуру *деревом проектирования*. Каждый лист в этом дереве представляет полностью отдельный проект.

Дерево проектирования комбинаторно сложнее, чем соответствующее дерево решений. Например, дерево решений, узлы которого представляют ответы на n независимых бинарных вопросов проектирования, соответствует дереву проектирования с $2n$ узлами. Но для любого достаточно сложного проекта это дерево становится настолько большим, что проблема его конструирования кажется непреодолимой для обычного человека и возникают сомнения в том, может ли работа над этой проблемой привести к появлению каких-либо плодотворных идей. Дерево проектирования — это слишком громоздкая конструкция, чтобы им можно было действительно пользоваться при проектировании.

Но сама концепция дерева проектирования позволяет лучше понять, как должно осуществляться проектирование, и извлечь из этого выводы. В частности, возникает аналогия с методологией динамической разработки программного обеспечения: каждая сборка, произведенная в конце рабочего дня, соответствует одному узлу в дереве проектирования, а полученный при этом программный продукт представляет наилучший готовый результат проектирования на данный момент.

Сравнение модульных и тесно интегрированных проектов

Оформляя записи в журнале проектирования дома Бруксов в виде дерева решений, мы обнаружили, что выбор среди альтернативных вариантов решения часто зависит от того, какой выбор был сделан ранее среди других вариантов. Например, музыкальный зал мог быть размещен в северной или западной части нового флигеля (это — решение высокого уровня), но от выбранного варианта его размещения зависят дальнейшие возможности расположения студий, гостиной и кухни!

Если бы все эти варианты были показаны в дереве решений, то оно стало бы весьма громоздким, поскольку некоторые альтернативные решения становились бы зависимыми сразу от нескольких других вариантов. Например, вместо решения "Выбор места для музыкального зала" с набором простых альтернатив пришлось бы столкнуться со следующими комбинациями альтернатив:

- музыкальный зал — в западной части; гостиная — в северной части;
- музыкальный зал — в северной части; кухня — в южной части;
- музыкальный зал — в северной части; кухня — в северной части.

И даже эти комбинации вариантов весьма зависят от других альтернативных вариантов проектирования.

В своей книге [5] Александер указал на недостатки такой тесной зависимости (иными словами, отсутствие модульности), связанные с тем, что усложняется задача исправления проекта. Таким образом, проектировщик, выбирая более или менее модульную организацию проекта, может по существу не только взвешивать достоинства альтернативных проектов, но также преднамеренно и оправданно жертвовать качеством будущего проекта ради упрощения внесения изменений в будущем. Именно по этой причине Парнас указал в своей чрезвычайно важной статье [164] следующее: "Программное обеспечение проектируется с учетом возможности простого расширения или сужения функций"[6]. Кроме того, в любом случае может потребоваться выбрать, что важнее — качество проекта или скорость и легкость самого процесса проектирования.

Модульные проекты можно гораздо проще представить в виде деревьев решений. И действительно, именно эта особенность модульных проектов считается само собой разумеющейся.

С другой стороны, разбивка на слишком мелкие модули становится причиной многих других недостатков. Оптимальными являются такие проекты, в которых рассматриваются компоненты, позволяющие достичь сразу нескольких целей. Рассмотрим однокорпусной автомобиль: его корпус не только служит для придания автомобилю привлекательного внешнего вида и размещения пассажиров, но и выполняет роль несущей конструкции. Автомобиль, спроектированный как

однокорпусной, легче и прочнее по сравнению с тем, в основе которого лежит многозвенная конструкция. Но на основе проекта пикапа с многозвенной конструкцией можно проще создать проект внедорожника, чем при использовании проекта однокорпусного автомобиля.

Программа Compendium и альтернативные инструментальные средства

Мы исследовали несколько пакетов программ, надеясь обнаружить тот, который в большей степени подходит для реконструирования и анализа дерева проектных решений в нашем конкретном примере. Полученные результаты описаны ниже.

Программа Task Architect

Программа Task Architect[7] действительно может оказаться полезной для проектирования, но при ее использовании оправдались не все наши ожидания. Task Architect — это инструментальное средство упрощения анализа задач, т.е. структурированного исследования способов выполнения конкретной работы. К примерам использования этой программы относятся не только задачи, выполняемые вручную, такие как установка фар на конвейере сборки автомобилей, но и мыслительные задачи, такие как принятие решения о переходе из одной стадии приземления в другую.

Таким образом, при проектировании дома Бруксов программа Task Architect могла бы использоваться для обретения лучшего понимания практических сценариев эксплуатации семейством Бруксов своего дома (приготовление пищи, проведение общих мероприятий, обучение музыке и т.д.). Программа Task Architect не создавалась как средство реконструкции дерева решений и не может быть легко приспособлена для выполнения этой работы (но мы, безусловно, попытались это сделать!).

Инструментальные средства управления проектированием

Как и программаTask Architect, такие средства управления проектированием, как Microsoft Project, OmniPlan или SmartDraw, могут оказаться полезными в процессе проектирования системы, но не кажутся подходящими для представления деревьев решений.

Дело в том, что в основе моделей инструментальных средств управления проектированием лежат такие методы анализа критического пути, как PERT (Program Evaluation and Review Technique — метод оценки и обзора программ). Но создается впечатление, что методология PERT позволяет поддерживать только модель водопада. Методология PERT не предусматривает использование задач и (или) узлов, выбираемых по условию, поэтому при ее использовании неявно предполагается, что основные проектные решения уже приняты[8].

Система IBIS и ее потомки

Система IBIS (Issue-Based Information System — система обработки информации на основе задач) была разработана в 1980-х годах для коллективного принятия решений и документирования обоснований, которые лежат в основе этих решений[9]. Как и Compendium, эта система предназначена для использования в процессе принятия решений в целях повышения продуктивности совещаний по вопросам проектирования и оказания помощи при выявлении слабых логических аргументов.

Для каждого рассмотренного нами инструментального средства мы подготовили схему соответствия функциональных возможностей инструмента и наших требований, чтобы понять, насколько оно нам подходит. Прежде всего мы кратко ознакомились с программой Compendium, а затем — с программой Task Architect. А после того как настало время ознакомиться с системой IBIS, мы обнаружили, что в ней естественным образом осуществляется поддержка многих полей и типов узлов, которые мы сочли необходимыми.

Система IBIS была реализована на основе программы с командной строкой. А Конклин (Conklin) подготовил графическую версию IBIS, или gIBIS[10]. Программа gIBIS должна была стать более подходящей для наших потребностей, чем Compendium, но мы не смогли найти версию, которая работала бы на наших компьютерах. Программа Compendium фактически является потомком gIBIS. Поэтому вернемся к программе Compendium.

Программа Compendium

Программа Compendium имеет много преимуществ. Она является чрезвычайно гибкой, а со временем это ее качество становится еще более явно выраженным. Ее разработкой занимается энергичный коллектив программистов, которые чрезвычайно быстро реагируют на просьбы и крики о помощи. База данных отслеживания запросов и выявленных ошибок для программы Compendium является интерактивной и общедоступной. В обсуждении связанных с ней вопросов на сетевом форуме активно участвует большое сообщество пользователей. Таким образом, возможности программы Compendium постоянно наращиваются и охватывают все новые и новые области применения.

Тем не менее, оглядываясь в прошлое, мы не можем рекомендовать это инструментальное средство в целях применения непосредственно для проектирования либо для документирования процесса разработки и обоснований принятых проектных решений.

Что касается самого проектирования, то мы хотим выразить следующее опасение: если проектировщики в процессе разработки используют инструментальное средство создания структурированных аннотаций или программ, то лишают себя возможности выражать расплывчатые, зачаточные идеи, что приводит к снижению качества концептуального проектирования. Это во многом напоминает

использование инструментальных средств автоматизированного проектирования, которые требуют слишком большой точности, чтобы с их помощью можно было быстро исследовать творческие идеи, тогда как эскизы позволяют проектировщику представлять еще не сложившиеся окончательно замыслы. Сам Конклин заметил, что программа gIBIS оказалась слишком структурированной и громоздкой с точки зрения поддержки творческих аспектов процесса проектирования[11].

Что касается поставленной нами задачи реконструкции дерева решений, то мы не считаем программу Compendium наиболее подходящим программным инструментальным средством. В конечном итоге мы остановились на такой схеме преобразования, которая оказалась весьма отличной от той, для которой предназначена программа Compendium. Вместо непосредственного использования возможностей программы Compendium для получения помощи при реконструкции дерева проектирования мы были вынуждены находить оригинальные способы переориентации функций программы Compendium для использования по другому назначению (например, мы приспособили узел ссылок Compendium для описания требований).

Кроме того, создаваемые нами деревья (несмотря на значительные усилия по сокращению их размеров) оказались намного больше по сравнению с тем, что легко поддерживает пользовательский интерфейс Compendium, притом что рассматриваемая нами задача проектирования была небольшой по сравнению с более существенными программными проектами. В таких больших деревьях трудно находить узлы, а еще большие сложности возникают при печати деревьев или их экспорте в графическом виде.

В окончательно принятой нами схеме преобразования используется следующее:

- структуры программы Compendium, приспособленные к нашим целям;
- разработанные нами структуры, которые в дополнение к структурам Compendium позволяют создавать единообразные деревья проектирования.

В конечном итоге мы пришли к выводу, что для представления деревьев проектирования лучше всего подходят универсальные средства построения схем, которые поддерживают такие функции, как автоматическая компоновка деревьев, автоматическое перенаправление стрелок связей и обеспечивающие поиск ассоциированные примечания. Для этих целей могут подойти программы Microsoft Visio и SmartDraw.

DRed как одно из перспективных инструментальных средств[12]

Согласно имеющимся у нас сведениям, наибольший успех в области применения компьютерных средств для создания документации с описанием требований к проекту был достигнут в результате широкого применения программы DRed в компании Rolls-Royce. Программа DRed была разработана Робом Брейсуэллом

(Rob Bracewell) в Центре технического проектирования Кембриджского университета по субсидии, полученной от компаний Rolls-Royce, BAE Systems и от Британского совета инженерных и физических научных исследований (UK Engineering and Physical Sciences Research Council)[13].

Назначение программы DRed состоит в документировании обоснований проектирования в связи с принятием проектных решений. Ее концептуальная структура весьма напоминает структуру программы gIBIS. А с точки зрения эксплуатации DRed в значительной степени подобна программе Compendium. Тем не менее, поскольку программа DRed в основном предназначена для сбора данных, касающихся обоснования проектных решений, ее развитие сосредоточивается на этой функции, а не на подготовке вспомогательных материалов для совещаний по проектированию[14].

Широкому распространению программы DRed в компании Rolls-Royce весьма способствовало то, что в этой компании и раньше большое внимание уделялось сбору данных для обоснования проектов. (Другие спонсоры проекта, в частности компания BAE Systems, не придавали такого значения выполнению указанной функции, поэтому программа DRed не нашла там широкого распространения.) В должностные обязанности технических проектировщиков компании Rolls-Royce всегда входила подготовка текстовых отчетов, содержащих сведения о том, на основании чего было принято то или иное проектное решение. Поэтому для групп проектировщиков стало большим подспорьем решение руководства о том, что вместо требуемых ранее текстовых отчетов могут предоставляться документы DRed. Задача оформления необходимой документации с помощью программы DRed оказалась намного проще.

Один из разработчиков этой программы, Марко Аурисиккио (Marco Aurisicchio), вспоминает, насколько быстро происходило ее внедрение в компании Rolls-Royce. В Кембриджском университете Аурисиккио выполняет ключевую роль по взаимодействию с компанией Rolls-Royce, поэтому никто лучше него не знает, как программа DRed используется в этой компании. Марко также преподавал на многочисленных курсах по использованию программы DRed, проводившихся в компании Rolls-Royce. А в компании Rolls-Royce ключевую роль в области эксплуатации программы DRed играет штатный сотрудник этой компании Майкл Мосс (Michael Moss), к которому инженеры Rolls-Royce непосредственно обращаются для получения поддержки. Кроме этого, Майкл отбирает и располагает по приоритетам отзывы, касающиеся программы DRed, которые затем передаются в группу Брейсуэлла в Кембриджском университете. Нам кажется, что успех внедрения программы DRed в компании Rolls-Royce во многом стал следствием поддержки личных связей между пользователями этой программы и ее разработчиками.

В разных группах проектировщиков программа DRed используется немного по-разному, но часто бывает так, что новое фактическое наполнение для этой программы создается при работе на виртуальной доске во время совещаний проектировщиков. Затем один из участников совещания получает задание составить

формальную документацию DRed исходя из зафиксированных эскизов, появлявшихся на виртуальной доске. Документация DRed создается также в ходе индивидуальной работы над проектами. Она используется и для концептуального, и для детализированного проектирования. Кроме самих проектировщиков, документацию DRed используют руководители проектных групп и инженеры-технологи, непосредственно занятые на производстве, причем для этого им не приходится прибегать к посторонней помощи.

Примерно 30% всего инженерного состава компании Rolls-Royce общей численностью около 600 человек в различных отделах и лабораториях этой компании во всем мире прошли обучение по использованию DRed, по крайней мере, на кратких курсах. Инженеры, вновь принятые на работу в компанию Rolls-Royce, проходят производственную практику в сочетании с шестинедельными курсами проектирования, в группах по четыре человека. По окончании обучения группа получает задание в виде типичного проекта, например, им предлагают для решения реальную проблему, которую для себя наметили проектировщики компании, но еще не имеют возможности ею заняться; как правило, это не столь ответственный проект, поэтому, если стажеры терпят неудачу, это не сказывается на работе компании[15].

Наибольшее дерево DRed, составленное разработчиками этой программы из Кембриджского университета, состоит из 190 листов схем, со средним количеством узлов на листе, равным 15. Если количество схем достаточно велико, то создается также обзорная схема, на которой в виде узла показана каждая детализированная схема.

Разумеется, все это способствует развитию культуры проектирования в компании Rolls-Royce. Успешное завершение каждого проекта приводит к увеличению количества схем DRed и других документов, которыми могут воспользоваться специалисты этой компании. Схемы DRed являются очень удобными для использования в качестве иллюстративного материала на презентациях, а при анализе результатов проектирования позволяют проводить изучение проекта по многим аспектам. Но сами документы DRed не могут разрабатываться по принципу формального управления версиями. Специалисты компании Rolls-Royce предприняли попытку организовать формальное управление версиями и убедились в чрезвычайной сложности этой задачи, до такой степени, что "если бы пришлось работать над документами DRed на основе управления версиями, то пришлось бы отказаться от программы DRed".

Программа DRed нашла также широкое применение в такой области, о которой не предполагали разработчики, — в качестве инструментального средства организации работы по диагностированию неисправностей и документированию результатов группами сопровождения производственного оборудования, отвечающими на запросы пользователей со всего мира. "Вот данные о том, когда, где и как произошел отказ оборудования, и вот все показания приборов, в которых это зафиксировано. Как теперь узнать, какими причинами вызван этот отказ?"

К сожалению, программа DRed не доступна для всеобщего распространения. Этой интеллектуальной собственностью владеют компании Rolls-Royce и BAE Systems, которые в настоящее время желают оставить ее исключительно в своем распоряжении.

Примечания и ссылки

1. В работах [144] и [209] приведены доводы в пользу проведения сбора данных для обоснования. Книга [155] представляет собой довольно полный сборник опубликованных статей по обоснованию проектирования. Потрясающим примером полного сбора данных обоснования может служить книга Мэдисона [145]. Текст этой книги можно также получить в Интернете по адресу `http://www.constitution.org/dfc/dfc_0000.htm`.
2. Интерес представляют также работы [160]; [65]; [134]; [135]; [32]; [51].
3. Работа [54] — это превосходная статья, в которой выдвинута идея создания системы Memex, с обобщенными ссылками и персонализированными следами ссылок, проходящими через граф знаний, что во многом напоминает современную систему веб. Предложенная в статье технология была примитивной, но сама ее концепция оказалась провидческой и весьма перспективной.
4. См. работу [190]; `http://compendium.open.ac.uk/institute/`, в книге взята за основу версия этой страницы от 25 июля 2009 года.
5. См. планы в главе 22.
6. В книге [164] в качестве основной структуры используется дерево проектирования.
7. См. `http://www.taskarchitect.com/index.htm`, в книге взята за основу версия этой страницы от 25 июля 2009 года.
8. Этот метод частично основан на технологии PERT (Project Evaluation and Review Technique) и может помочь руководителю провести выбор среди альтернативных вариантов проектирования, основываясь на том, к каким последствиям приводит переход по той или иной ветви графика PERT.
9. См. [160].
10. См. [65].
11. См. [65], с. 324-325.
12. Этот раздел основан на совместном интервью Брейсуэлла и Аурисиккио, которое было проведено Бруксом 19 июня 2008 года. Для Брукса была проведена полная демонстрация системы, которая показала все ее возможности.
13. См. [32].
14. См. [13].
15. В курсе проектирования и разработки программного обеспечения Университета Северной Каролины в Чапел-Хилл (UNC-Chapel Hill) применяются точно такие же критерии выбора проектов, выдвинутые реальными пользователями.

IV

Система проектирования домов, идеальная с точки зрения специалиста по информатике

TheDesignofDesign.indb 247 30.10.2012 15:58:34

TheDesignofDesign.indb 248 30.10.2012 15:58:34

17

Система проектирования домов, идеальная с точки зрения специалиста по информатике, — от замысла до компьютерного представления

Пирамиды, соборы и ракеты созданы не благодаря открытиям в области геометрии, теории структур или термодинамики, а благодаря тому, что вначале были изображены (буквально представлены) в умах тех, кто задумал эти объекты.

См. [92]

◂ *Рабочее место пользователя графической системы UNC GRIP для молекулярных исследований. Джеймс Липскомб (James Lipscomb)*

Постановка задачи

Предположим, что достаточно представить себе идеальную компьютерную систему для архитектурного проектирования домов и других сооружений, и она появится. Как должна выглядеть эта система?

Мысленное представление

Разумеется, профессиональный архитектор лучше, чем кто-либо, может представить себе, как в целом должна выглядеть система проектирования зданий. Но я взял на себя смелость предложить свою версию интерфейса "проектировщик–компьютер" для идеальной системы проектирования домов, поскольку, во-первых, сам разработал небольшой любительский проект дома, во-вторых, тематика строительной архитектуры является более конкретной и доступной для широкой аудитории, чем программная архитектура, и, в-третьих, я имею более чем полувековой опыт работы по созданию человеко-машинных интерфейсов.

В предлагаемом интерфейсе нашел значительное отражение опыт разработки графической системы UNC GRIP для молекулярных исследований по заказу Северокаролинского университета в Чапел-Хилле, в котором я участвовал со своими студентами. Эволюция этой интерактивной графической системы реального времени проходила в течении многих лет, причем мы работали над инструментарием для сложнейших задач с выдающимися химиками — специалистами по белкам[1].

В настоящей главе рассматривается только процесс функционального проектирования домов и не затрагиваются процессы конструктивной или системной разработки, хотя, по моему мнению, та же система была бы применимой и в конструктивной разработке при создании механических и электрических систем и даже мебели.

Это — не единственная система проектирования, которую можно считать идеальной. Один из разделов в книге Ульмана [210] посвящен описанию такой системы для механического проектирования[2]. Тем не менее здесь рассматривается во многом другая проблематика. Ульман поставил задачу создания программы, способной обеспечить управление знаниями в таких больших масштабах, насколько это возможно, что, безусловно, представляет собой важную задачу. Меня же в этой области главным образом привлекает обеспечение взаимодействия между сознанием проектировщика и автоматической системой, поскольку я считаю, что важнее всего добиться успешной реализации замыслов проектировщика.

Последовательное приближение

Общепризнано, что успех проектирования достигается с применением нисходящих методов. Вначале подготавливается текстовый документ с заданием, в котором в первую очередь перечислены ключевые идеи, а затем указано, что из них

следует. Например, разработка программы начинается с размышлений об алгоритме и структурах данных. Разработка плана дома начинается с определения того, какие функциональные пространства требуются для реализации сценариев эксплуатации дома, после чего рассматривается задача обеспечения связи между этими пространствами. Но на самых ранних этапах проектирования здания должно быть уделено достаточное внимание решению эстетических задач путем правильного подбора архитектурных форм.

Великие проектировщики, даже те, кто проявляет наибольшую свободу мысли, редко начинают свою работу с нуля; как правило, проектировщики опираются на богатое наследство, полученное от их предшественников[3]. Они воспринимают одни идеи из одного образца, другие — из другого, дополняют их собственными идеями, стремясь воплотить всю совокупность замыслов в проекте, обладающем концептуальной целостностью и характеризующемся ярко выраженным собственным стилем.

Успешные проекты часто начинаются с того, что берется весь комплекс намеченных замыслов и чистый лист бумаги. Первый эскиз начинается с крупных блоков, после чего происходит последовательное уточнение, корректировка размеров и добавление все большего количества деталей.

Тёрнер Уиттед (Turner Whitted) в 1986 году предложил другой, возможно, лучший способ создания модели физических объектов (первоначально — в контексте компьютерной графики), который состоит в том, чтобы вначале подготавливалась полностью детализированная модель, которая не соответствует разрабатываемому изделию, а лишь напоминает его. После этого происходит корректировка одного атрибута за другим в целях постепенного продвижения к тому изделию, которое пока что существует в виде мысленного представления, или к действительным свойствам реального объекта.

Уиттед назвал этот метод последовательным приближением[4]. Если встать на реальную почву, то можно заметить, что последовательное приближение — это именно та программа, которой руководствовались представители естественных наук за прошедшие несколько столетий, поскольку естествознание основано на разработке моделей, все более точно отражающих существующие природные явления[5].

А с тех пор, как люди стали заниматься проектированием искусственных объектов, сам процесс проектирования стал представлять собой постепенную материализацию идеала объекта, присутствующего в сознании. Метод последовательного приближения, взятый на вооружение, оказывает существенную помощь. Он предусматривает появление на каждом шаге прототипа, доступного для изучения. Прототип с самого начала является действительным; иными словами, в его структуре нет несогласованностей. Прототип всегда полностью детализирован, поэтому его восприятие, допустим, визуальное и слуховое, не вводит в заблуждение.

Таким образом, можно представить себе примерно такую последовательность проектирования дома.

- “Я хочу построить дом с тремя спальнями в том стиле, который принят в штате Джорджия”.
- “Он должен быть обращен фасадом к северу”.
- “Зеркально отобразите его по вертикали”.
- “Установите ширину гостиной, равную 14 футам”.
- “Уменьшите продольный размер кухни на один фут”.
- “Выполните внешнюю отделку из белой штукатурки вместо кирпича”.
- “Покройте кровлю желобчатой черепицей вместо плоских кровельных плиток”.

Я пришел к выводу, что подход, предложенный Уиттедом, является вполне приемлемым, и выдвинул его в качестве основы для своей идеальной системы. Этот подход может изменить весь характер современного проектирования. Он способствует созданию новых инструментальных средств, стимулирующих продуктивное мышление.

Библиотека моделей

Поэтому разработка идеальной системы была начата с создания богатой библиотеки прекрасных образцов полностью детализированных проектов. Если проектировщик начинает свою работу с образцов, которые сами характеризуются непротиворечивостью стиля, то появляется гарантия, что такая непротиворечивость станет одной из черт будущего проекта. По мере дальнейшего использования системы происходит рост и библиотеки моделей. Чтобы было проще наполнять библиотеку моделей, следует применять методы машинного зрения для обследования трехмерных объектов и создания структурированных моделей на основе полученных данных.

В приведенной выше воображаемой последовательности проектирования дома согласно устным указаниям подразумевалось, что проектировщик действует, пользуясь хорошо знакомым содержимым библиотеки образцов; в частности, был назван конкретный образец стиля.

Но каким бы опытным ни был проектировщик, после достижения библиотекой определенного объема он не сможет в ней ориентироваться столь же легко. Как и в любой другой проблемной области, может оказаться, что опытный проектировщик глубоко изучил один аспект проектирования домов, обладает достаточной компетентностью в других аспектах, а что касается прочего, должен в случае необходимости дополнительно изучать рассматриваемую тему. Что же касается новичка в этой области, то ему придется разбираться во всем, с чем он сталкивается. Это означает, что для библиотеки необходимо предусмотреть инструменты ее просмотра по иерархическому и последовательному принципам.

Самый продуктивный способ структурирования библиотеки состоит в использовании методов решения таксономических задач, но следует учитывать, что уже проделан большой объем работы в части подготовки терминологии и концептуальной структуры для подобных библиотек[6].

Потенциальные недостатки подхода, основанного на последовательном приближении

Как уже было сказано, в наши дни метод последовательного приближения представляет собой наиболее продуктивный и удобный в использовании подход к созданию систем проектирования, но ему изначально присущи некоторые потенциальные недостатки.

В частности, некоторые специалисты отмечают, что наличие широкого выбора образцов станет неявно ограничивать творческий потенциал проектировщика. Было бы ли возможным появление творений Брунеллески, Ле Корбюзье, Гери и Гауди в условиях изобилия предъявляемых им образцов?

Я утверждаю, что эти архитекторы так или иначе создали бы свои великолепные произведения. Ни один из них не был дилетантом. Все они обучались по работам своих предшественников. Как и Бах, они вводили новации, опираясь на мастерство, а не наталкивались на них случайно, по незнанию. Такие гениальные самоучки, как "бабушка Мозес" (Анна Мэри Мозес), в наши дни встречаются редко.

Возможно, более существенным потенциальным недостатком является то, что каждый образец, как правило, лишь частично соответствует проектным требованиям к разрабатываемому объекту, причем ни один инструмент, каким бы великолепным он ни был, не способен соединить разрозненные части образцов, на базе которых мог бы быть создан новый проект.

Наиболее весомые потенциальные недостатки метода последовательного приближения заключаются в том, что в нем используется библиотека. Некачественные модели, слишком малое количество моделей, слишком узкое разнообразие моделей — это именно те ограничения, связанные с использованием библиотеки, которые не позволяют в полной мере реализовать потенциал метода последовательного приближения. Особенно значительное влияние эти недостатки оказывают в начальный период развития системы.

Применение способов визуального представления для ввода данных в компьютер

Проблему, заключающуюся в том, что результаты умственного труда должны быть перенесены в компьютерную модель, приходится решать независимо от того, выполняется ли проектирование на основе образца или полностью с нуля.

Воздушный замок, родившийся в воображении, проектировщик должен воплотить в реальный объект, используя голос, обе руки, голову и, очевидно, ноги.

Бакстон (Buxton) вместе с коллегами из компании Alias Systems и со своими студентами из Университета Торонто впервые предложил интерфейсы ввода данных двумя руками[7]. Ведущая рука выполняет точные манипуляции, а другая рука создает контекст для ее действий (в дальнейшем эти руки для простоты будем называть правой и левой). При совместном движении обеих рук происходит задание приблизительных размеров: "Это должно быть примерно таким".

Ритмика глагола–существительного

В языках проектирования, как и вообще в командных языках, каждое высказывание состоит из глагола (определяющего действие) и существительного (определяющего объект). Существительное может иметь дополнительно выбираемые "уточнители" — прилагательные, фразы или предложения. С глаголом может быть связана наречная фраза (рис. 17.1). В лингвистике наблюдается любопытный феномен, заключающийся в том, что прилагательные, или атрибуты, часто бывают назначены существительному-объекту с помощью глагола, как в следующих примерах: "Выполните эту дверь как имеющую ширину 32 дюйма"; "Покрасьте западную стену в зеленый цвет".

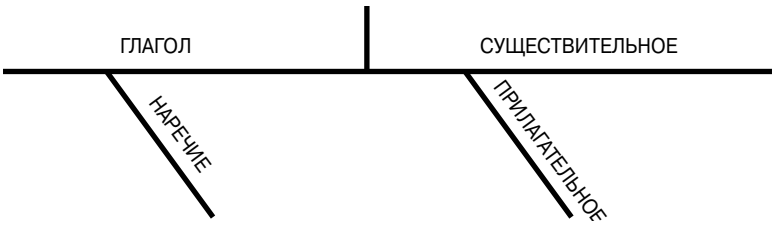


Рис. 17.1. Структура повелительного предложения

В пространственных языках проектирования обычно удобнее всего задавать существительное по методу непосредственного указания, для чего используются одна или обе руки. Что же касается глаголов, то их удобнее всего задавать голосом; в наиболее распространенном интерфейсе типа WIMP (Windows–Icons–Menus–Pointing — окна, пиктограммы, меню и позиционирующее устройство) используется неестественный способ ввода с помощью меню. Безусловно, иногда меню может оказаться чрезвычайно удобным, поскольку позволяет задавать варианты выбора. Но пользователь, выполняющий снова и снова знакомые задачи, в этом не нуждается.

Работая в рамках интерфейса WIMP, пользователи действуют в регулярном ритме: указывают или вводят с клавиатуры спецификацию существительного, после чего переводят указатель на команду меню (глагол). Таким образом, при редактировании документа пользователь выбирает блок текста, переводит курсор на команду "Вырезать", возможно, выбирает еще один фрагмент текста, задает команду "Скопировать", выбирает положение между символами и указывает команду "Вставить".

Задание глаголов

В действительности, действуя одной рукой в интерфейсе ввода, невозможно задать стабильный ритм, поэтому работа является утомительной и малопроизводительной. Перемещение указателя из позиции существительного в меню глаголов приводит к потере правильной позиции. Следующее существительное обычно располагается близко от предыдущего, поэтому указатель приходится возвращать примерно в ту позицию, где он находился перед этим. При этом приходится компенсировать потерю ориентации, для предотвращения чего были разработаны специальные методы задания глаголов с повышенной частотой. Например, двойной щелчок рассматривается как глагол "Открыть", а ввод с клавиатуры приводит к активизации глагола "Вставить".

Наиболее распространенные глаголы имеют клавиатурные эквиваленты, которые могут быть введены левой рукой. Этот метод представляет собой блестящее изобретение.

В распоряжении новичка всегда имеется доступный стандартный метод — выбор глагола из меню. А эксперт может воспользоваться методами, позволяющими повысить скорость ввода, предусматривающими возможность работы обеими руками и не требующими перевода указателя из позиции существительного. Но что лучше всего, начинающий пользователь может постепенно осваивать один глагол за другим исходя из того, насколько часто ему приходится применять конкретные глаголы.

Голосовые команды. Однако наиболее естественным способом задания команд является голосовой ввод. Поэтому в нашей идеальной системе должна применяться программа распознавания речи с ограниченным словарем, устойчивая к различиям в характеристиках голосов дикторов, имеющая богатый словарь синонимов и позволяющая пользователю самостоятельно накапливать и корректировать словарь. При этом остается возможным задание команд с помощью меню, а также ввод команд с клавиатуры. Таким образом, если пользователь потерял голос, он может воспользоваться другим способом ввода.

Общепринятые глаголы. В современных системах архитектурного автоматизированного проектирования применяется широкий набор общепринятых глаголов, который был сформирован с учетом многолетнего опыта работы с этими системами. Возможно, этот набор даже слишком велик, но если пользователю предоставляется возможность составить персональное меню команд и в дальнейшем осуществлять в нем выбор с помощью более простой процедуры, то система остается удобной в использовании независимо от количества предусмотренных в ней команд. В качестве примеров наиболее распространенных команд можно указать следующие.

- Повернуть.
- Создать копию.

- Сгруппировать.
- Выполнить привязку.
- Выровнять.
- Отодвинуть.
- Масштабировать.
- Выбрать объект из библиотеки.
- Задать имя.

Вероятно, наиболее продуктивной является команда "Выбрать объект из библиотеки", независимо от того, применяется ли подход к проектированию по методу последовательного приближения, или нет. Мы применяем этот подход, поэтому в нашей системе команда выбора из библиотеки используется чаще и является более важной.

В библиотеке все объекты представлены в одинаковом масштабе, поэтому преимущественно применяется команда "Выбрать и масштабировать", согласно которой после выбора масштабирование задается автоматически в соответствии с текущим рабочим масштабом.

Определение существительных

При определении объектов и областей в континууме пространства и времени люди используют чрезвычайно разнообразные способы!

Определение по имени. В обыденной речи определение объектов обычно осуществляется с применением имени, заданного явно или неявно подразумеваемого под местоимением. Желательно, чтобы такая же возможность была предусмотрена в идеальной системе. Причем, даже если считать, что распознавание речи происходит со 100%-ной точностью, остается задача правильной трактовки голосовой команды. К чему относится атрибут "левый"? Что означает "передний"? Какой из объектов рассматривается как "красный"? Какой является "наибольшим"? Что значит "это"? Даже если удается однозначно указать требуемый объект, часто возникают затруднения при определении области действия команды. Чтобы наделить систему интеллектом и способностью ориентироваться в мире хотя бы на уровне трехлетнего ребенка, необходимо проводить синтаксический и семантический анализ, научиться сохранять и использовать контекст.

Кроме того, приходится учитывать, что один и тот же объект может упоминаться под разными именами, а одно и то же имя может относиться к разным объектам. Насколько колоссальной может оказаться такая работа, показала попытка создания объединенной базы данных со стандартными определениями для сухопутных войск, военно-морских сил и военно-воздушных сил США. Нелегко было даже добиться того, чтобы во всех этих родах войск на вопрос "Который

час?" отвечали одинаково. В военно-воздушных силах использовалось среднеевропейское время, в сухопутных войсках — местное, а каждый военный корабль жил по местному времени района базирования авианосца его боевой группы.

Поэтому наша идеальная система должна давать возможность отдельным пользователям и группам пользователей создавать индивидуализированные словари синонимов, которые дополняют и переопределяют системный словарь. Для создателей идеальной системы попытка рационализации всей номенклатуры библиотеки моделей была бы равносильна погружению в глубокое болото, поэтому сами пользователи должны взять в свои руки и развивать необходимые инструменты.

Применение указателей на двухмерном представлении. Ошеломляющий успех интерфейса WIMP показал, насколько мощным является способ применения указателей для выбора определенных и видимых объектов. Этот способ действительно получил очень широкое распространение.

Но его нельзя считать исчерпывающим. Создавая свою систему GRIP, мы исходили из того, что наши клиенты-химики будут выбирать из нескольких сотен аминокислотных остатков в белке, указывая на них. Однако первый же клиент попросил, чтобы ему дали возможность использовать клавиатуру для указания остатков по их трехзначному номеру. Этот химик работал с данным конкретным белком в течение многих лет и знал числовые обозначения белковых остатков буквально назубок! Для непосредственного указания требуется корректировать положение фокуса ввода на экране до тех пор, пока необходимый остаток не станет четко различимым в сложном трехмерном представлении. Кроме того, становится причиной значительного утомления работа протянутой рукой, которая держит световое перо.

Применение эскизов в двухмерном представлении. Архитекторы проектируют трехмерные сущности. Но в своей работе они используют главным образом двухмерные представления — точные чертежи и приблизительные наброски. Это правило остается в силе даже применительно к самым причудливым двухмерным проекциям, позволяющим в основном представить себе, как будет выглядеть трехмерный объект. Создается впечатление, что рисование эскизов оказывает существенную помощь в процессе мышления[8].

Я не считаю, что когда-либо такие двухмерные изображения уступят свое первенство, пусть даже появятся самые развитые и удобные инструментальные средства трехмерного моделирования. Изображение на сетчатке глаза является двухмерным. Двухмерны плоские поверхности, по которым так легко скользит рука. Поэтому наша идеальная система должна предоставлять средства работы с двухмерными изображениями, позволяющие указывать объекты и рисовать эскизы, допустим, на основе сенсорной поверхности со световым пером, которая позволяет определить не только положение, но и силу нажатия пером.

При работе с определенными двухмерными изображениями, такими как карты или чертежи, часто возникает необходимость проводить разбиение на отдельные области. Если с изображением связаны иерархические структуры, например,

такие, которые охватывают территориальные образования (страны, штаты) или архитектурные объекты (этажи, комнаты), то задача разбиения изображения становится намного проще.

Еще одна задача состоит в том, что во время работы с изображением необходимо иметь возможность точно указывать, где находится тот или иной объект и какие он имеет размеры; при этом желательно, чтобы операции позиционирования выполнялись двумя руками. Предположим, на чертеже необходимо провести линию с точно заданной длиной. Правая рука держит перо, направленное в требуемую точку, а левая рука вводит на клавиатуре значение, определяющее длину. Такой вариант — удобнее всего.

Позиционирование и рисование эскизов в трехмерном пространстве. Все сказанное выше применительно к двухмерным изображениям относится и к трехмерным. Задача указания позиции изначально требует больших физических усилий, поскольку для подведения указателя к точке требуется движение рукой. Движения, осуществляемые в кинематической цепочке "палец–запястье–локоть", имеют примерно такую же относительную точность, как и движения "кисть–локоть–плечо", поэтому желательно так организовать работу, чтобы оба локтя основную часть времени находились на опоре[9].

Сложнее задавать произвольные пространственные области и углы их поворота, чем определять объекты и их позиционирование. В разговоре мы обычно показываем форму произвольных областей, одновременно двигая двумя руками: "Облако выглядело примерно так". Наша идеальная система должна работать именно по такому принципу. В части определения объектов и операций для работы в трехмерном пространстве было сделано очень многое[10]. В частности, значительный объем усилий был направлен на создание средств, допускающих задание спецификации от руки; по большей части существуют лишь более неуклюжие замены для этого.

Определение текста

Большинство блоков текста должно представлять собой короткие строки, размещаемые на проектных чертежах, которые, главным образом, задают названия и размеры. Наилучшим инструментальным средством для решения задачи ввода подобного текста, как и глаголов, являются программы распознавания речи.

Спецификации должны состоять из блоков текста — стандартных абзацев, выбранных из базы данных и параметризованных. Что касается задания больших объемов описательного текста, то мы не предусматриваем ввод его под диктовку, поскольку ввести самому и отредактировать абзацы можно намного быстрее по сравнению с диктовкой и последующей правкой. Но, идет ли речь о вводе коротких или длинных фрагментов текста, нельзя обойтись без алфавитно-цифровой клавиатуры.

В ходе разработки нашей системы UNC GRIP мы пришли к выводу, что лучше всего разместить клавиатуру под рабочим столом. Практика показала, что клавиатуру приходится выдвигать для работы на ней довольно редко; вполне можно предположить, что такая же ситуация будет наблюдаться и в идеальной системе.

Определение наречий

"Произвести движение".

"В каком направлении? Как далеко? До какой точки?"

"Выполнить поворот".

"В каком направлении? На какой угол?"

"Создать дубликаты".

"Сколько? В каком направлении расположить? Какие задать промежутки?"

"Выбрать дверь и подогнать ее по масштабу".

"Какую ширину она должна будет иметь?"

Приведенный выше диалог, состоящий из команд и запросов, включает не только глаголы и существительные; с большинством глаголов связаны обстоятельства, обычно заданные в виде предложных фраз[11].

Большинство таких наречий требует задания количественных данных. Кроме того, данные должны быть заданы точно; редко удается правильно указать позицию на рабочем столе вручную. Требуемую точность во многом можно обеспечить, применяя косвенные указания с помощью вспомогательных глаголов: "Выполнить привязку к сетке", "Выровнять по..." и т.д.

Многие из этих уточнений могут быть заданы с помощью небольших меню; к этому относятся палитры цветов, списки материалов, расписания с указанием времени завершения. В таком случае организация работы по принципу выбора из меню становится более простой и экономически оправданной. Выполнение операций проектирования домов можно сравнить с выполнением операций доступа к оперативной памяти в том, что в них проявляется ярко выраженная тенденция к локализации: после выполнения каждой операции (в первом случае — после принятия любого конкретного проектного решения) могут рассматриваться многочисленные независимые варианты выбора, но чаще всего выбор осуществляется из небольшого подмножества. Важно также иметь возможность применения настраиваемых меню.

Остальные требования к количественной точности проще всего могут быть выполнены с помощью цифровой клавиатуры. Наш опыт показывает, что в ходе проектирования клавиатура используется достаточно часто. В отличие от алфавитно-цифровой клавиатуры, не возникает необходимость убирать цифровую клавиатуру подальше.

Определение точки зрения и способа представления

Какое бы творческое наполнение ни содержало архитектурное проектирование, результатом его становятся планы и разрезы. Однако при проверке работы желательно ознакомиться с трехмерным представлением всего проекта, а лучше всего — совершить виртуальное путешествие через трехмерную модель. Определение текущего представления трехмерного проекта дома — это важный частный случай применения спецификации "существительное + прилагательное".

Одни параметры просмотра приходится изменять непрерывно, а другие — менее часто. Это не означает, что параметры подразделяются на *динамические* и *статические*. Динамическими являются все параметры просмотра, и часто даже возникает необходимость обеспечить плавную и постепенную корректировку редко изменяющихся параметров.

Виртуальные путешествия внутри объекта

При моделировании прогулки внутри дома приходится добиваться того, чтобы изображение сменялось последовательно в связи с изменением координат x и y, а также курса наблюдателя. При этом происходит перемещение точки просмотра на двухмерном чертеже, представляющем один этаж здания, и изменение угловой ориентации виртуального наблюдателя на той же плоскости.

Рост наблюдателя, с высоты которого он осматривает здание, считается постоянным. Разумеется, рост не изменяется после перехода с одного этажа на другой. Тем не менее иногда приходится устанавливать другое значение роста наблюдателя, если этого пожелает один из участников группы проектирования или если возникнет необходимость оценить внутреннюю перспективу здания с разных точек зрения.

Потребность в том, чтобы перспектива оценивалась при разных наклонах головы наблюдателя влево и вправо, практически не возникает. Люди выполняют такие движения головой достаточно редко. Гораздо чаще требуется обеспечить виртуальный осмотр внутренней части объекта при разных наклонах головы (вниз и вверх), но и эти движения гораздо реже по сравнению со сменой координат x и y, а также изменением курса.

Устройство EyeBall. Мы пришли к выводу, что идеальным средством виртуального просмотра внутренних пространств в архитектуре является специальное устройство ввода-вывода. Речь идет об устройстве, получившем название EyeBall, которое состоит из следящего датчика с шестью степенями свободы, размещенного в манипуляторе в форме бильярдного шара с основанием, срезанным на 0,5 дюйма. Манипулятор оборудован двумя кнопками, удобными для нажатия указательным и средним пальцами (рис. 17.2). При скольжении манипулятора по рабочей поверхности на плане происходит перемещение V-образного курсора.

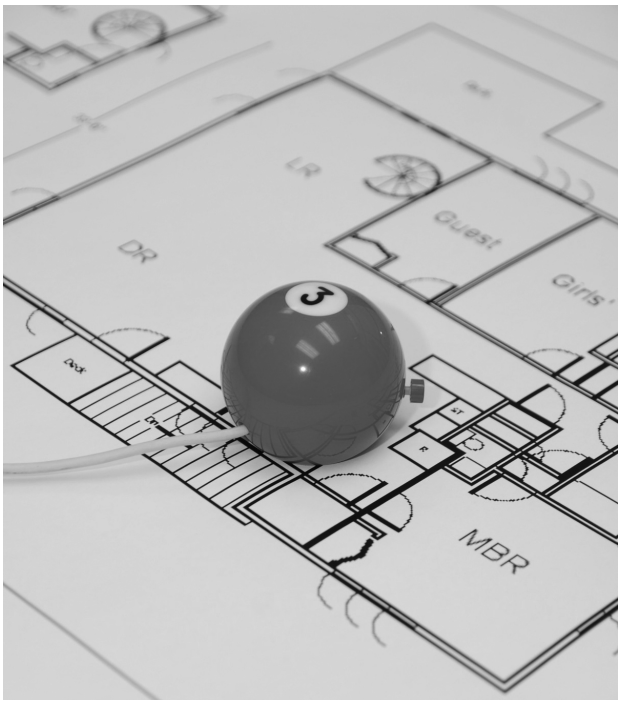


Рис. 17.2. Устройство визуального указателя UNC–Chapel Hill EyeBall. Келли Гаскилл (Kelli Gaskill), Университет Северной Каролины

Этот манипулятор обеспечивает наиболее удобное непрерывное определение всех шести параметров просмотра: точки зрения (3 параметра), направление просмотра (2 параметра) и наклон головы во время просмотра (1 параметр), причем в первую очередь происходит определение координат x и y, а также курсового угла. Кнопка для указательного пальца аналогична одной из кнопок обычной мыши, а вторая кнопка служит фиксатором. Для изменения точки зрения так, чтобы виртуальный наблюдатель поднялся на более высокий этаж, необходимо приподнять устройство EyeBall, нажать фиксирующую кнопку, поставить устройство на стол и отпустить фиксатор. Высота наблюдателя, измеряемая расстоянием от пола, остается неизменной. Привлекательным свойством этой операции перехода между этажами является то, что точка зрения плавно перемещается в направлении оси z, как если бы происходил подъем в лифте со стеклянными стенами; резкая смена изображения не происходит, поэтому в ходе виртуального путешествия внутри здания практически не возникает потеря ориентации.

Манипулятор EyeBall имеет также другие режимы использования, кроме формирования внутренних представлений с помощью плана. В частности, с его применением можно развернуть на рабочей станции трехмерную модель с контекстным

индексом, в которой с помощью светящейся точки показано положение виртуального наблюдателя. Работая с этой моделью, можно просматривать контекстный индекс и указывать новое положение виртуального наблюдателя и направление осмотра[12].

В том и в другом режиме для определения роста наблюдателя, иными словами, высоты его глаз над полом, служит специальный движок, хотя необходимость в этом возникает очень редко. Расположение двух кнопок манипулятора по умолчанию выбрано в расчете на то, что 5% пользователей устройства будут составлять женщины, а 95% — мужчины.

Осмотр снаружи

Визуальный указатель "Toothpick". Как будет описано в следующей главе, при использовании любого рабочего места проектировщика часто возникает необходимость осмотра проектируемого объекта с разных точек зрения; к этому относится осмотр самого проектируемого изделия, а также различные контекстные представления.

Что касается дома, то одним из контекстных представлений может быть вид снаружи, а другими — вид всей комнаты или внутренней стены. Решая вопрос о том, как задать произвольное внешнее представление объекта, мы пришли к выводу, что наиболее удобным устройством является позиционный джойстик с двумя степенями свободы, показанный на рис. 17.3.

Манипулирование этим устройством обычно производится левой рукой для указания направления осмотра объекта. Положением покоя считается то положение, в котором устройство Toothpick указывает на пользователя. Чтобы получить возможность выбора позиций осмотра в пространстве 4 пи стерадиан, мы удвоили угол смещения устройства Toothpick от его положения покоя. Таким образом, после полного сдвига устройства влево или вправо позиция осмотра перемещается вокруг всего объекта, что позволяет осматривать его и спереди, и сзади. К моему удивлению, пользователи устройства нашли, что такое удвоение действительно позволяет создать простой и естественный интерфейс. Указанные движения дополняют нажатия часто используемых основных кнопок, определяющих четыре параметра возвышения и позволяющих сформировать некоторые представления с обзором трех четвертей пространства (из угла куба со сторонами, выровненными вдоль осей, который охватывает рассматриваемую область); предусмотрен также более редко используемый движок, который определяет, с какого расстояния производится просмотр.

Восприятие глубины. Принципы эстетики диктуют, что в архитектурном проектировании должно учитываться распределение масс, т.е. относительное расположение заполненных и пустых объемов. Для контроля над распределением масс также необходимо обеспечить создание трехмерного представления. При этом наиболее информативным становится достижение кинетического эффекта

глубины, при котором происходит относительное движение глаз наблюдателя и самой сцены. Используя такой кинетический эффект, можно гораздо лучше понять особенности проектируемого объекта, чем при использовании его трехмерного представления (бинокулярного зрения), а если в распоряжении проектировщика находятся и тот и другой метод моделирования, то результаты становятся еще более весомыми.

Рис. 17.3. Устройство позиционирования Toothpick; копия устройства UNC, созданного профессором Чарльзом Молнаром (Charles Molnar) в Вашингтонском Университете г. Сент-Луи

Но возникает вопрос о том, как задать требуемое движение? Очевидно, что достижение кинетического эффекта можно обеспечить путем свободного перемещения манипулятора EyeBall, но для этого требуются значительные усилия. Было бы желательно, чтобы не приходилось ни на что отвлекаться при изучении распределения масс. Работая над своей системой графического представления молекул GRIP в университете UNC-Chapel Hill, мы пришли к выводу, что вращение всей сцены вокруг вертикальной оси способствует существенному улучшению

восприятия при анализе распределения масс, поэтому предусмотрели режим вращения, в котором сцена периодически поворачивается в прямом и обратном направлениях, не требуя ввода дополнительных команд. Пользователю предоставляется возможность задавать амплитуду и периодичность разворотов в прямом и обратном направлениях, хотя обычно использовались значения по умолчанию.

Примечания и ссылки

1. Описание системы приведено в работах [33]; [41]; [37]. Замечательной особенностью этого пользовательского интерфейса было то, что он имел 21 степень свободы для изменения модели и управления дисплеем, а также предусматривал применение клавиатуры. Все эти возможности действительно использовались на практике. Ни одно устройство не имело больше трех степеней свободы; перегрузка функций устройств также не применялась. Пользователи инстинктивно стремились достичь известных местоположений, чтобы в них проводить изменение значений.
2. См. [210] в работе [74].
3. Здесь весьма уместным является упоминание работ Александера — [5]–[7].
4. Тёрнер Уиттед, по материалам личного общения (1986).
5. В компьютерной графике также использовалось постепенное улучшение освещения сцены, даже в те времена, когда компьютеры были гораздо менее быстродействующими. В первую 1/30 секунды происходит прорисовка изображения с простым резким освещением. В последующие интервалы времени выполняются итерационные вычисления для моделирования все более сложного рассеянного освещения. При этом возникает буквально потрясающий визуальный эффект, например, в изображении внутреннего пространства здания тени разбегаются по углам комнаты по мере смягчения освещения от более резкого до более рассеянного.
6. См., например, описание сорока стилей домов на сайте `http://www.houseplans.net/house-plans-with-photos/` (в книге взята за основу версия этой страницы от 30 июля 2009 года).
7. См. [55].
8. См. [103].
9. См. [12].
10. См., например, [66].
11. Можно представить себе, что эта система должна быть построена так, чтобы в ней можно было задавать подобные уточняющие вопросы после неполного указания существительного или глагола. Но даже в этом случае

необходимо иметь возможность отключить эту функцию; опытные проектировщики неявно подразумевают, что *должно* следовать за текущим этапом, поэтому применение данной функции отвлекало бы их от работы и было бы навязчивым.

12. См. [197].

TheDesignofDesign.indb 266 30.10.2012 15:58:35

18

Система проектирования домов, идеальная с точки зрения специалиста по информатике, — от компьютерного представления до замысла

Задумав строить,
Исследовать сперва мы станем почву,
Потом начертим план; когда ж готов
Рисунок дома, вычислить должны,
Во сколько обойдется нам постройка.
Но коль превысит смета наши средства,
Что сделаем? Начертим план жилища
Размеров меньших иль затею бросим.

УИЛЬЯМ ШЕКСПИР [1598], ГЕНРИХ IV, ЧАСТЬ 2

◂ *Общий вид АРМ архитектурного проектировщика. Рисунок Андрея Стейта (Andrei State)*

TheDesignofDesign.indb 267 30.10.2012 15:58:35

Двухстороннее взаимодействие

Компьютеризация умственного труда требует в первую очередь организации двухстороннего взаимодействия человека и машины. При этом следует учитывать, что человек получает основной объем информации из внешнего мира с помощью зрения. Однако это не единственный путь. Для получения представления о текущей ситуации, улавливания предупреждений, восприятия изменений в окружающей среде особую роль играет слух, а для передачи другим полученных сведений — речь. К еще более глубоким уровням сознания относятся осязание (как чувство) и обоняние. Обыденный язык богат метафорами, отражающими эту глубину. Люди "улавливают", что происходит в сложных когнитивных обстоятельствах, когда им приходится "брать бразды правления в свои руки", поскольку они "чуют неладное".

Визуальные индикаторы — одновременное применение большого количества окон

Проектировщики, работающие за компьютером, обычно используют только одно активное окно. Тем не менее специалисты в области компьютеризации уже давно поняли, что для проектировщиков требуется по меньшей мере два окна, а размеры экранов не должны быть столь малыми, как сейчас[1]. Необходимо найти ответ на вопрос о том, какие визуальные индикаторы желательно было бы иметь в идеальной системе проектирования зданий.

Изображение координатной сетки и представление на экране чертежей

Я считаю, что в реальном проектировании основным средством всегда будут оставаться двухмерные чертежи, поэтому прежде всего необходимо обеспечить представление на экране электронного устройства координатной сетки.

- **Угол.** В проектных организациях, в которых чертежи изготавливаются вручную, поневоле приходится просматривать готовое изображение и вносить в него изменения с применением одной и той же рабочей поверхности. После перехода к использованию средств компьютеризации появляются более удобные способы организации работы. Для манипуляции компьютеризированными графическими инструментами применяется отдельная поверхность, а это означает, что больше не приходится отводить руку от чертежа, чтобы посмотреть, как он выглядит. Исследования показали, что корреляция движения мыши или пера по одной поверхности с изображением этого движения на другой не вызывает затруднений при использовании графических инструментов, основанных на этом принципе.
- **Размеры рабочей поверхности.** В компьютеризированном строительном проектировании применяются графические планшеты разных размеров,

но чаще всего используются размеры 30×48 дюймов. Эти размеры выбраны исходя из того, на какую длину можно удобнее всего протянуть руку.

- **Разрешающая способность дисплея.** Разрешающая способность дисплея должна соответствовать зоркости глаз, которая составляет примерно одну минуту дуги. Этому требованию отвечает экран компьютера с разрешающей способностью 1920 пикселей, расположенный от глаз на расстоянии, превышающем в два раза его ширину. Такие дисплеи с плоским экраном теперь можно найти во многих офисах.
- **Расстояние просмотра изображения.** Как оказалось, проектировщики меньше всего утомляются, если просматриваемое ими изображение создается на экране, который находится от них на расстоянии от 6 до 8 футов (2–2,5 м).

На своих графических планшетах они почти всегда изображают двухмерные чертежи. Кроме чертежей, на экране могут быть показаны эскизы, проекции и т.д. А в наши дни широкое применение получили средства послойного изображения чертежей с управляемой прозрачностью, которые незаменимы в тех случаях, когда возникает необходимость сосредоточить внимание на каком-то одном аспекте, сохраняя при этом общий концептуальный контекст.

Двухмерное контекстное представление

В своем Северокаролинском университете мы разработали много систем компьютеризированного проектирования, предназначенных для создания строительных чертежей, молекулярных моделей, двухмерных диаграмм для публикации и т.д. Наблюдая за тем, как в действительности пользователи осуществляют с их помощью проектные работы, мы неизменно обнаруживали одну и ту же универсальную последовательность.

1. Изучение большого пространственного фрагмента для выявления контекста.
2. Увеличение масштаба.
3. Создание некоторой локальной части или манипулирование ею.
4. Уменьшение масштаба.
5. Повтор.

Очевидно, что это неестественное и непродуктивное поведение обусловлено тем, что по традиции в проектировании применяется единственная рабочая поверхность, на которой поочередно выполняется одна операция за другой. Для исправления такого положения необходимо предоставить проектировщику возможность выводить одновременно и контекстное, и детализированное представление, чтобы можно было переходить от одного представления к другому

движением глаз, а не руки. Мы обязаны прежде всего решить эту задачу. Кроме того, следует отказаться от идеи изображать контекстное окно в виде эскиза в одном из углов детализированного представления, поскольку контекст также должен быть показан достаточно подробно. В наши дни стоимость оборудования настолько снизилась, что для уважающей себя проектной организации попытка экономии на покупке дисплеев становится непростительной. Стремитесь к тому, чтобы оба представления, контекстное и детализированное, были постоянно показаны на больших экранах!

Обычно контекстное представление выполняется как пространственное или даже как общий план. Но для таких операций, как извлечение объекта из библиотеки, желательно было бы иметь еще одно представление с условным изображением библиотеки (в виде иерархического древовидного представления или содержимого определенного узла), а на другом представлении показывать то место на чертеже, где должен быть расположен объект.

Трехмерное представление

В конечном итоге двухмерные абстрактные чертежи должны превратиться в трехмерные здания, в которых будут жить и работать люди. В идеальной системе проектировщик должен постоянно видеть трехмерное изображение будущего дома в том виде, который полностью соответствует современному состоянию работы над проектом, причем во всех подробностях.

По-видимому, для решения этой задачи идеально подходит технология создания проекций в виртуальной среде. Применять такую среду, как CAVE, полностью обеспечивающую эффект присутствия, нет необходимости; вполне приемлемым был бы вариант, в котором формируется небольшое вертикальное изображение. Для этого подошло бы даже единственное стандартное окно с трехмерным представлением. Разумеется, для проектировщика необходимо предусмотреть возможность сидеть перед чертежной поверхностью, а не ходить вокруг нее. Ему обязательно должен быть предоставлен набор удобных графических инструментов. Что касается упомянутого дисплея с трехмерным изображением, то он должен быть вспомогательным инструментальным средством создания проекта, тогда как программное обеспечение CAVE предназначено для просмотра, а не для проектирования.

При этом должно быть решено несколько технических проблем. Прежде всего следует исключить необходимость постоянно носить стереоскопические очки для просмотра трехмерного изображения во время проектирования. На глаза и без этого бремени возлагается достаточно большая нагрузка. Поэтому должен быть предусмотрен переключатель режима просмотра, возможно, даже ножной. Еще один вариант состоит в том, что на трехмерном дисплее должно создаваться стереоскопическое изображение, не требующее дополнительных средств для просмотра.

Заметим, что здания, как правило, имеют плоские стены, поэтому для изучения их внешнего вида с помощью объемных проекций вполне подходят плоские экраны. Разумеется, возникает также необходимость в использовании визуального указателя, идеальным примером которого является EyeBall, для управления стандартными режимами показа изображения, что позволяло бы просматривать проекцию здания под разными углами зрения. Но в идеальной системе необходимо также предусмотреть режим формирования моментального снимка, который позволяет сформировать изображение дома на плоской поверхности.

Осмотр снаружи

Особенно полезными способами изучения результатов проектирования дома должны стать способы, предусматривающие создание изображения дома снаружи, под широким углом зрения. При этом, как и при осмотре интерьера, необходимо предусмотреть возможность изучать внешний вид дома, применяя элементы управления изображением с учетом изменения положения солнца в разные часы и времена года.

Следует предусмотреть средства управления позицией наблюдателя при осмотре интерьера и экстерьера здания, причем в идеальном случае эти средства должны быть распределены по двум разным наборам инструментов. Как правило, внешний осмотр должен начинаться с указания координат x и y на контекстном дисплее, после чего будет неизменно создаваться изображение, по умолчанию ориентированное по центру здания. Это позволяет легко обеспечить изучение экстерьера здания с любой точки зрения.

Обычно самый потрясающий эффект создается при моделировании того, как будет выглядеть здание ночью после включения внутреннего освещения. К этому приему создания картин часто прибегал художник Томас Кинкэйд (Thomas Kinkade), поэтому его произведения выглядят такими очаровательными, привлекательными и уютными.

Я пришел к неожиданному выводу, что большую пользу приносит применение еще одного варианта осмотра экстерьера — создание представления здания при ночном освещении с динамически удаляемой ближайшей стеной. Внешний осмотр здания в этом режиме позволяет получить полное представление об этом здании (рис. 18.1).

Представление рабочей книги

Еще в одном окне должна одновременно отображаться рабочая книга проектировщика. Перед каждым сеансом проектирования, выполняемого на рабочей станции, проектировщики имеют в своем распоряжении следующее.

- Все, что касается текущего состояния проекта.
- Планы действий.

Рис. 18.1. Вид жилого здания в разрезе. DeltaSphere, Inc.

Каждый сеанс проектирования, выполняемого на рабочей станции, завершается следующим.

- Обновление проектов в ходе работы и планы будущих действий.
- Журналы, в которых должны отражаться все действия проектировщиков; применение либо журналов, либо системы управления версиями необходимо для обеспечения автоматического отслеживания с возвратом в предыдущее состояния.
- Заметки, для которых в идеальном случае должна быть предусмотрена возможность ведения под диктовку, чтобы можно было не прерывать выполнение вручную операций проектирования; заметки должны касаться того, какие варианты были опробованы и почему, какие из них были отвергнуты и почему, а какие сохранены и почему.

Пояснения, изложенные в заметках, не могут быть зафиксированы в журналах действий, но являются чрезвычайно важными для обоснования сложного проекта. В частности, просмотр заметок помогает замечательным образом снова войти в курс дела после перерыва в работе. С их помощью проектировщик может быстро вспомнить, чем он руководствовался, рассматривая определенный альтернативный вариант проектирования. К тому же обоснования вариантов становятся просто бесценными для новых членов коллектива и для тех, кто подхватывает эстафету от проектировщика.

В идеальном случае план действий и итоговые заметки должны чередоваться в одном документе, будучи выделенными разным цветом или шрифтом.

В нашей идеальной системе в журнале действий предусмотрены двухстраничные вставки, которые показывают текущие значения стоимостных оценок, а также значения некоторых других регламентируемых показателей, таких как площадь помещений в квадратных футах.

Представление спецификации

При возведении здания приходится руководствоваться не только чертежами, но и текстовыми спецификациями. В идеальном случае накопление текстовых сопровождающих документов происходит одновременно с чертежами, а не откладывается на более поздний срок.

Это не так уж сложно, как кажется на первый взгляд, учитывая то, что в проектировании предусмотрено применение режима постепенного приближения. Спецификации должны создаваться в соответствии с жестко заданным форматом. Если каждая из исходных моделей в библиотеке имеет собственную спецификацию, то проектировщик может работать по принципу внесения в спецификацию изменений в ходе работы над моделью. Эта задача значительно упростилась после давно произошедшего создания обширной коллекции Sweets File, которая теперь называется Sweets Network[2]. В этой коллекции собраны данные по миллионам продуктов с изображениями, описаниями и спецификациями. Сам файл коллекции и его таксономию сопровождает компания McGraw-Hill Construction; содержимое предоставляют поставщики продуктов; архитекторы и подрядчики подписываются на услуги доступа к коллекции. Каждый, кто причастен к работе над этой коллекцией, извлекает для себя пользу.

Таким образом, для идеальной системы требуется четвертое двухмерное окно с постоянно уточняемыми текстовыми спецификациями. В идеальном случае изменения, внесенные с помощью этого дисплея, должны автоматически переноситься на дисплей с изображением. Многие описания продуктов в коллекции Sweets Network уже включают модели автоматизированного проектирования в стандартизированной форме, поэтому задача поддержания такой связи не является концептуально сложной. Распространение изменений с изображения обратно на спецификации — гораздо более сложная, исследовательская задача[3].

Звуковое сопровождение

Еще много лет тому назад я получил очень сильные впечатления при просмотре видеозаписи, с помощью которой группа архитекторов из Хельсинки демонстрировала подготовленную с применением компьютерной графики модель предложенного проекта реконструкции жилого дома. Визуальные средства были хороши, но особо ничем не выделялись. Но видеозапись сопровождалась звуками, зарегистрированными на детской площадке. И даже притом что людей не было видно, эти закулисные шумы поневоле заставили зрителей почувствовать себя участниками происходящего.

Задача подготовки самого звукового сопровождения является несложной; гораздо сложнее подготовить сценарий, на основе которого будет формироваться это сопровождение. В частности, что касается нашей задачи, то для сцен, демонстрируемых внутри и вне помещения, необходимо выбрать подходящие источники звука: работающий телевизор, шум проходящего транспорта, работа стиральной машины, игры и ссоры малышей. В таком случае прогулку по виртуальному дому можно будет сопровождать воспроизведением заранее подготовленной звукозаписи, что позволит одновременно и усилить восприятие, и подчеркнуть назначение рассматриваемого помещения.

Необходимая для этого технология акустического моделирования уже достаточно развита[4]. Если же какие-то звуки на данный момент не могут быть сформированы в режиме реального времени, то следует пойти по пути аппроксимации и стремиться каждый раз подбирать все более и более подходящее звуковое сопровождение. Существующие сейчас затруднения во многом обусловлены недостаточным быстродействием акустических систем, но этот недостаток успешно преодолевается.

При акустическом моделировании приходится также учитывать распространение звука через двери и окна. Следует ли предусмотреть моделирование с учетом того, какие двери и окна открыты или закрыты? Должно ли учитываться пространственное расположение этих проемов? Если разнообразие учитываемых факторов будет достаточно велико, то количество возможных рассматриваемых комбинаций может стать просто колоссальным. Соблюдение требований к акустике помещения, которые могут выражаться в виде достаточно простой спецификации, может лишь с большим трудом быть проверено с помощью моделирования. Одно из возможных решений состоит в использовании двухмерных диаграмм интенсивности звука, формируемых с учетом определенной высоты от пола, которые привязаны к сетке, покрывающей весь чертеж дома, и изменяющихся в интерактивном режиме в зависимости от открытия и закрытия дверных и оконных проемов. На этом графическом изображении может быть указано место восприятия звука. И в этом случае удобным средством определения местоположения и ориентации головы слушателя в модели может стать манипулятор EyeBall.

Восприятие с помощью осязания

По-видимому, ни одно из чувств не способно вызвать у человека столь глубокие ощущения, как осязание[5]. Однако при всем желании я не смог найти ни одного вероятного направления использования существующей технологии тактильной обратной связи в идеальной системе.

Обобщения

Идеальная система проектирования зданий обладает такими особенностями, что от нее можно легко перейти к обобщениям, распространяющимся на многие другие области. При этом не столь существенно то, что не всегда предусмотренные функции применимы в полной мере. Например, в идеальной системе построения программного обеспечения вряд ли понадобятся все трехмерные возможности. Но и в эту систему необходимо включить богатую стартовую библиотеку, дисплей проектирования, контекстный дисплей, дисплей рабочей книги и дисплей тестовых примеров, соединив все компоненты необходимыми перекрестными связями.

Осуществимость

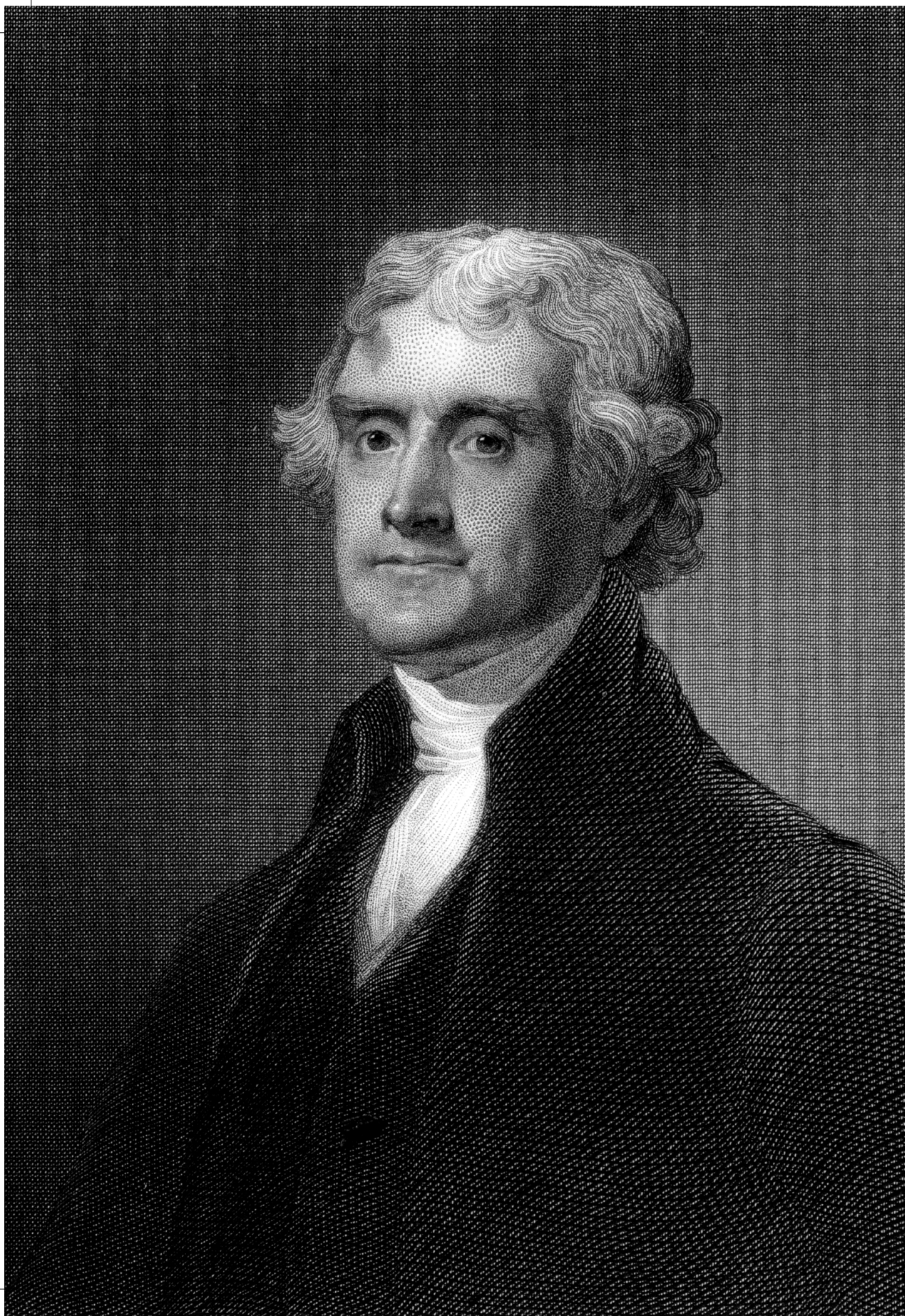
Можем ли мы с полной уверенностью утверждать, что теперь идеальная система может быть создана? Разумеется, да. Все предусмотренные в ней технологии вполне могут быть призваны на службу. Но возникает еще один вопрос: может ли создание идеальной системы стать экономически оправданным, если речь идет об относительно скромных проектах, таких как проектирование отдельных зданий? Я полагаю, что этого можно добиться, по крайней мере, в крупных фирмах, которые способны вложить соответствующие финансовые средства, а затем предоставлять систему в режиме разделения времени многочисленным проектировщикам.

Как можно подойти к созданию этой системы? Создавать ее путем постепенного наращивания! Любой проект, в котором была бы предпринята попытка создать такую систему без тщательной теоретической и практической подготовки, почти наверняка окончилась бы неудачей. Однако поэтапное создание системы под постоянным контролем профессиональных проектировщиков должно стать успешным. Труднее всего было бы собрать первую версию качественной библиотеки исходных моделей.

Можно представить себе, что структура системы, а также такие необходимые форматы, как стандартный формат ввода и формат описания, могут разрабатываться в рамках академической научно-исследовательской работы, после чего может начаться накопление моделей на принципах добровольного участия в рамках движения за разработку с открытым исходным кодом. Именно здесь могут быть успешно предложены стимулы к участию в создании открытого репозитария[6]. Мне кажется, что многим архитекторам понравилась бы мысль о включении их проекта жилого дома, здания или другого объекта в общедоступную библиотеку. Ведь это не только повысило бы престиж, но и стало бесплатной рекламой, равносильной публикации в журнале. А в дальнейшем, после накопления достаточно объемистой централизованной библиотеки, было бы не слишком трудно усовершенствовать отбор проектов с учетом использования. Проекты, которые никого не интересуют, стали бы постепенно изыматься из библиотеки.

Примечания и ссылки

1. См. [40], с. 194.
2. См. `http://products.construction.com/`.
3. В основе системы, обеспечивающей человеко-машинное взаимодействие, даже при решении такой относительно простой задачи, как редактирование текста, должны лежать некоторые весьма надежные механизмы. С ними можно ознакомиться на примере системы Lilac Кена Брукса (Ken Brooks) с двумя представлениями, которая описана с работах [47] и [48]. Проблема заключается в том, что среди двух представлений, обеспечивающих в этом случае взаимодействие человека с компьютером, одно представление содержит намного более подробную информацию по сравнению с другим, поэтому для проведения изменений в более простом представлении требуется одновременно не только изымать, но и вводить определенную информацию в более сложном представлении.
4. См. [201].
5. См. [151].
6. См. [174], глава 4.



19

Создание выдающихся проектов великими проектировщиками, а не участниками обычного процесса проектирования

В основе работ SEI (Software Engineering Institute) над воплощением идеи достижения зрелости процесса разработки ПО лежит основная предпосылка, заключающаяся в том, что качество программного продукта в значительной степени определяется качеством процессов разработки программного обеспечения и процессов сопровождения, используемых для создания этого ПО.

СМ. [167]

...Кое-кто может видеть в них сумасшедших, а мы называем их гениями, поскольку те, кто достаточно безумны, чтобы считать себя способными изменить мир, так и делают.

СТИВ ДЖОБС (STEVE JOBS),
РЕКЛАМА APPLE (1997)

◀ *Томас Джефферсон. iStockphoto*

Великие проекты и процессы разработки продуктов

Трудно найти такие цитаты, как в эпиграфах к этой главе, авторы которых проявляли бы большее несогласие. Кто из них прав?

При подготовке к написанию этой главы я провел небольшую субъективную классификацию известных компьютерных продуктов, разделив их на две группы по тому признаку, есть ли у этих программ горячие сторонники. Эта классификация постепенно стала более развернутой, как показано на рис. 19.1, с некоторыми новыми дополнениями. Полагаю, что это разделение в какой-то степени показывает, является ли та или иная разработка великим достижением в проектировании. (Эта оценка ни в коей степени не связана с коммерческим успехом, который определяется сложным сочетанием многочисленных факторов, а не только качеством проекта.)

Поразительной особенностью этой схемы оказалось то, что (насколько я могу определить) каждый из продуктов в правом столбце был создан в результате формального процесса разработки продукта, который характеризуется участием многих людей и прохождением многочисленных согласований. А каждый из продуктов в левом столбце был создан за пределами нормального процесса разработки готовых изделий.

Аналогичные примеры можно найти и в других областях: атомная бомба; ядерная субмарина; баллистическая ракета; самолет-невидимка; самолет Spitfire; пенициллин; стеклоочиститель, работающий в прерывистом режиме. Все перечисленные выше новаторские проекты были разработаны небольшими коллективами, без применения обычных процессов разработки продуктов, что стало следствием либо сложившихся обстоятельств, либо сознательно принятого решения.

Да	**Нет**
iPhone	Сотовый телефон
Apple II	ПК
Интерфейс Macintosh	Windows
UNIX	z/OS (MVS)
Pascal	Algol
Fortran	Cobol
Python	Appletalk

Рис. 19.1. Клубы активных пользователей программного обеспечения. (Согласно [40], рис. 16.1)

Преимущества и недостатки сложившихся процессов разработки продуктов

Указанная выше закономерность, разумеется, не всегда соблюдается[1], но, несмотря на это, заслуживает внимания и требует получения ответов на следующие важные вопросы.

- Почему такое большое количество выдающихся проектов было создано без применения сложившихся процессов разработки продуктов?
- Для чего нужны процессы разработки продуктов? Почему бы от них не отказаться?
- Можно ли вообще создать выдающийся проект в рамках обычного процесса разработки продуктов? Как это сделать?
- Как добиться того, чтобы процессы разработки продуктов способствовали созданию выдающихся проектов и облегчали эту задачу, вместо того, чтобы препятствовать этому?

В чем процессы разработки продуктов препятствуют созданию выдающихся проектов

Я считаю, что стандартные корпоративные процессы проектирования продуктов в действительности не позволяют создавать поистине новаторские и выдающиеся проекты. Причины этого нетрудно понять, рассматривая, как и каким образом развиваются корпоративные процессы проектирования. Процессы разработки продуктов существуют для того, чтобы внести порядок в естественный хаос, возникающий при разработке новых продуктов.

Любой процесс по самой своей природе консервативен, поскольку предназначен для создания аналогичных (пусть даже немного разных) предметов в рамках единой упорядоченной структуры. Именно по этой причине поистине ни на что не похожие, действительно новаторские продукты никак не укладываются в стандартные рамки. Достаточно лишь подумать, насколько далек по своему замыслу персональный компьютер от колоссальных корпоративных ЭВМ, занимавших целые здания в 1960-х годах, или даже от мини-компьютеров 1970-х годов, обслуживающих отделы.

Стандартный процесс разработки продуктов по самой своей природе нацелен на достижение предсказуемости: его применение должно привести к созданию в указанный срок и с регламентированными затратами примерно такого продукта, который определяется деловыми потребностями, поскольку обстоятельства не позволяют ждать, пока найдется великий проектировщик или будет потрачено значительное время на решение проблемы. Предсказуемые и выдающиеся проекты создаются в разных условиях.

По своей сути процесс разработки продуктов — это "продолжение прошедшей войны", поэтому в нем стимулируются тактические приемы, приносившие успех в прошлом, и отвергаются те, которые оказалось неудачными. А если должен быть создан такой продукт, который начинает "новую войну" (перекрывает принципиально новые потребности или поддерживает не существовавшие ранее режимы работы), то выбранный ранее подход к применению оправданных и отклонению неприемлемых тактических приемов может оказаться совершенно не подходящим. Рассмотрим в качестве примера коммуникатор iPhone. В действительности его почти нельзя сравнивать с простым мобильным телефоном, не говоря уже об устройстве ведения голосовых переговоров с помощью наземных линий связи, который был создан Александром Беллом и запатентован его компанией AT&T.

Что же касается процесса разработки однотипных продуктов, то он по самой своей сути заранее направлен лишь на достижение предсказуемых результатов, в ходе чего должны быть заблокированы "неудачные" идеи и исключены "недосмотры". Этот процесс направлен на то, чтобы не открывать дорогу продуктам, которые не будут пользоваться намеченным спросом, иметь стоимость больше ожидаемой, порождать надежды в отношении функций и графиков поставки, которые не могут быть оправданы. Есть еще один более тонкий нюанс: корпоративные процессы разработки продуктов нацелены также на предотвращение создания линий продуктов, вызывающих растерянность у покупателей, когда одна и та же компания выпускает товары, которые становятся безжалостными конкурентами по отношению друг к другу, и клиенты не знают, что покупать. Таким образом, неудача может явиться результатом многих причин, поэтому сложившиеся процессы разработки продуктов, как правило, предусматривают достижение согласования со стороны многих специалистов, каждый из которых является экспертом в части определения конкретной причины потенциальной неудачи.

В связи с необходимостью достижения такого согласия возникают многочисленные факторы, которые исключают на корню возможность создания выдающихся проектов. Во-первых, в обязанности каждого специалиста, отвечающего за свое направление разработки, входит прежде всего предотвращение ошибок, а не создание чего-то грандиозного. Поэтому каждый из них отдельно ищет причины, чтобы не дать хода новаторским идеям. И даже если не удается полностью предотвратить появление принципиально нового продукта, существующие механизмы часто сглаживают наиболее яркие его особенности в результате достижения компромиссов в той или иной части. Но ведь именно новаторские свойства позволяют выйти на передовые позиции!

Во-вторых, сложившиеся процессы разработки продуктов таковы, что требуют согласования не только на основе современных и актуальных идей, но и идей, оставшихся в прошлом, которые законсервировались в виде установленных правил. Разумеется, процессы разработки продуктов не остаются неизменными и развиваются, но вместе с ними растут и все кодексы правил, причем каждая ошибка

и весь приобретенный опыт воплощаются в новых правилах или в новых процессах утверждения, направленных на предотвращение повторения предыдущих ошибок. Постоянно рождаются все новые и новые правила, почти не встречая барьеров, а после их ввода в действие больше ничто не может их отменить, пока не возникает реальный кризис. Природа вещей такова, что бюрократия становится все более непреодолимой, процессы развития — все более замедленными, а организации — все менее поворотливыми, хотя внешне наблюдается их процветание и расширение[2].

Когда я занимался разработкой оборудования для семейства компьютеров System/360 компании IBM в начале 1960-х годов, наша модель 20 мэйнфрейма S/360 разрабатывалась в лаборатории IBM в г. Боблинген, ФРГ. Коллектив этой лаборатории состоял из очень талантливых людей, и во главе его стояли сильные лидеры. Это подразделение IBM существовало уже немало лет, и в нем было создано много коммерчески успешных продуктов для специализированных рынков, но его специалистам так и не удавалось достичь реального успеха во включении своих проектов в главную линейку продуктов IBM, иными словами, они не могли выйти на международный рынок этой компании. Для нашего проекта System/360, достижение успеха в котором было реальной проблемой для компании, такое положение дел вызывало тревогу, поэтому меня направили в Боблинген, чтобы я помог исправить ситуацию.

Вскоре была выявлена основная причина. Инженеры германского подразделения всегда добросовестно и тщательно соблюдали официальную корпоративную процедуру создания продуктов IBM, изложенную в качестве руководства в документе объемом свыше 100 страниц. Успешные руководители проектов в других лабораториях осмеливались отступать от процедурных правил в нужное время; секретом успеха был правильный выбор такого времени![3] После того как я это понял, в Германию был отправлен один из таких опытных "манипуляторов" процедурами для управления проектом. Он позволил всем талантливым специалистам из боблингенской лаборатории реализовать свой потенциал. Компьютер S/360 модели 20 оказался феноменально успешным.

Наконец, отметим, что процессы согласования истощают силы проектировщиков-новаторов, съедая материальные и временные ресурсы. Для достижения согласования постоянно приходится участвовать в совещаниях, количество которых зачастую выходит за разумные пределы. А совещания занимают время, зачастую очень много времени. Никто не думает о том, что великие проектировщики встречаются редко и далеко не везде; их время чрезвычайно дорого.

Чем оправдано существование сложившихся процессов разработки продуктов

Может показаться, что я по-донкихотски выступаю на стороне революционного ниспровержения всех корпоративных процессов проектирования в пользу

творческого хаоса. Но это не так. Без таких процессов нельзя обойтись по многим причинам. В определенный момент возникает необходимость получить одобрение со стороны руководства корпорации, чтобы двигаться дальше; когда-то следует привлечь к работе опытных людей, которые помогли бы обнаружить очевидные оплошности; а бывает и так, что должно быть выполнено согласование графика и бюджета разработки продукта. Весь секрет заключается в том, что эти процессы необходимо держать в рамках, чтобы они не мешали созданию выдающихся проектов и чтобы обсуждались в основном те проблемы, которые препятствуют созданию замечательного произведения, а не предпринимались попытки задушить его в колыбели.

Продукты, создаваемые на основе ранее выпущенных. Отметим также, что процесс проектирования продуктов всегда будет играть важную роль, поскольку большинство проектов не предназначено для осуществления каких-то кардинальных инноваций. На это есть серьезные основания. После того как в руки пользователей попадает действительно успешный и инновационный продукт, обнаруживаются по меньшей мере четыре разных эффекта, которые описаны ниже.

- Практическое применение показывает недостатки, которые должны быть устранены в последующих выпусках продукта.
- Пользователи находят для новшества неожиданные применения, что способствует расширению концептуального определения продукта, обычно сопровождаемому наращиванием его возможностей.
- Новшество демонстрирует свою полезность, в связи с чем возникает потребность в создании еще более развитого продукта, а пользователи проявляют готовность платить за это больше.
- Как противовес этим стимулам к изменению продукта, по мере роста его популярности возникает тенденция к сохранению преемственности; пользователи не хотят, чтобы следующая версия полюбившегося им продукта стала революционной, поскольку стремятся снова получить нечто знакомое и привычное.

Это означает, что при подготовке к выпуску последующих версий продукта приходится соблюдать намного более строгие ограничения, а возможности подлинного новаторства практически сходят на нет. Вместе с тем успех порождает многочисленные возможности и открывает новые направления для усовершенствования в последующих версиях продукта. Но ни одна компания не может охватить сразу все перспективные области. Поэтому при подготовке последующих версий продукта приходится тщательно выбирать среди всех возможных сценариев дальнейшего развития. Разработка этих версий должна находиться под постоянным контролем для обеспечения того, чтобы вновь выпускаемый продукт

по-прежнему отвечал своему назначению и был благожелательно воспринят пользователями, для которых он предназначен.

В наибольшей степени оправдывают себя процессы разработки продуктов, в которых циклически повторяются следующие фазы:

- определение требований к продукту;
- рыночный прогноз;
- оценка производственных затрат;
- прогнозирование оптимальной продажной цены.

Это позволяет эффективно решать обе указанные выше задачи: выбор направления развития продукта и контроль над его созданием.

Однако точный рыночный прогноз невозможен, если не накоплен определенный объем сведений о коммерческой реализации аналогичных продуктов. А для исчерпывающей оценки производственных затрат необходимо иметь опыт разработки и производства подобных продуктов.

Это позволяет должным образом направить процессы разработки продукта на создание последующих версий продукта. С другой стороны, для разработки действительно новаторского изделия необходимо выйти за рамки сложившегося процесса.

Повышение уровня практического осуществления проектирования. Проектировщики, даже хорошо освоившие процессы разработки и выпуска продуктов, не могут стать выдающимися проектировщиками, не имея на то способностей. Поэтому поистине новаторские продукты редко появляются там, где нет действительно талантливого проектировщика. Однако лишь глубокое освоение навыков проектирования позволяет разработчику проявить свой талант и подняться выше среднего уровня в этой области. Об этом не следует забывать.

Представители сообщества разработчиков программного обеспечения всегда уделяют большое внимание совершенствованию применяемых процессов проектирования. Жизнь показывает, что это очень важно, ведь, насколько я знаю, почти нет таких коллективов проектировщиков, которые при низком или среднем уровне организации работы смогли бы превзойти своих коллег, которым удалось лучше организовать свою деятельность.

В этом отношении невозможно переоценить важность работы Уотса Хэмфри (Watts Humphrey) и Института программной инженерии (Software Engineering Institute — SEI) по созданию и внедрению модели технологической зрелости (Capability Maturity Model — CMM)[4]. Модель CMM представляет собой формализованное средство измерения характеристик компонентов качественного процесса проектирования, которое, по общему мнению, оказалось весьма полезным. Было показано, что низкие результаты аудита по модели CMM в проектной компании могут служить свидетельством необходимости пересмотра применяемой

в этой организации методологии проектирования, а также заимствования опыта более успешно работающих проектных организаций. Наиболее перспективным направлением повышения общего уровня разработок в коллективе является усовершенствование применяемых процессов разработки. В таком отношении хорошо проявила себя модель CMM, и в этом нет ничего удивительного. Но, независимо от того, сколько усилий затрачивается на улучшение самих процессов, это не позволяет поднять верхнюю планку возможностей коллектива. Какой бы выдающейся ни была организация процессов проектирования, она, отдельно взятая, не позволяет создавать выдающиеся проекты; для этого нужны огромные усилия действительно талантливых людей. В СМИ промелькнула ссылка на высказывание одного из представителей компании Apple, который отметил: "Мы находимся на уровне I [модели технологической зрелости] и останемся на нем навсегда"[5]. Однако результаты, достигнутые Apple, говорят сами за себя.

Необходимость в использовании сложившегося процесса даже в новаторском проектировании. Меня часто спрашивали: "Нет сомнения, что при разработке проекта S/360, которая требовала координации работы международных лабораторий и учета бизнес-потребностей многих рынков, нельзя было обойтись без тщательно организованного процесса. Как же вам удавалось извлекать из него только пользу и избегать удушающих ограничений, связанных с неизбежными формальностями?"

Наш основной коллектив проектировщиков был достаточно надежно защищен от влияния обычно применявшихся в IBM процессов, в ходе осуществления которых часто возникали неприятные нестыковки, а также испытывал надежную поддержку со стороны влиятельных руководителей разных уровней. Нам предоставили исключительные возможности привлекать к работе любых талантливых людей из других групп разработчиков компании. Мы не испытывали нехватки денег.

Но когда настало время воплотить проект в серийно выпускаемые продукты, возникла необходимость пройти стандартные процедуры технологической подготовки. До всех исполнителей было доведено указание высших руководителей компании, согласно которому данный проект должен стать поистине революционным, поэтому потребуются определенные отступления от обычных процессов. Началась повседневная борьба, в ходе которой наш коллектив, коллектив проектировщиков, неизменно пытался внести больше новшеств в процедуры подготовки к производству, прогнозирования, оценки, ценоопределения и так далее, а коллектив исполнителей, ответственный за выпуск готовой продукции, с неизменным успехом парировал эти попытки. Тем не менее подготовкой к выпуску занимались специалисты высокой квалификации, и их производственные навыки стали основой для общего успеха.

Процедурные рамки сковывают творчество, но без определенной регламентации невозможно обойтись. Как выйти из этого положения?

Выдающиеся проекты создаются выдающимися проектировщиками. Найдите их!

Например, сам я не обладаю способностями ни в области спорта, ни в области музыки, но глубоко уверен в том, что, во-первых, достичь высокого уровня подготовки в качестве музыканта, бейсболиста, танцовщика или проектировщика может только талантливый человек, и, во-вторых, степень проявления таланта в каждой области у различных людей может оказаться весьма неодинаковой.

Даже в таком коллективе, где собраны специалисты примерно с одинаковым образованием и опытом, обнаруживается, что разные люди наделены талантами свыше далеко не одинаково. Мне самому приходилось сталкиваться и с некоторыми профессионально работающими людьми, и с некоторыми студентами, которые были чрезвычайно одаренными проектировщиками. Они оставили яркий след в моей памяти. История искусства изобилует примерами, подтверждающими эту мысль.

Кроме того, невозможно найти двух человек, которые были бы наделены одинаковыми талантами в тех или иных областях. (Я думаю, Бог распорядился так, что каждый из нас обладает в чем-то уникальными способностями, которые позволяют дать другим людям то, чего у них нет.) Поэтому прозорливый руководитель организует работу так, чтобы рамки ответственности были очерчены вокруг людей с учетом того, в чем состоят их сильные стороны и как они могут себя проявить, вместо разграничения ответственности между членами коллектива с помощью таких рамок, которые рассматривает как идеальные сам руководитель, исходя из абстрактных соображений.

Наши фундаментальные демократические права и обязанности состоят в том, чтобы обеспечить равенство всех людей перед законом. С этим связана намного более трудная цель — обеспечить для всех равные возможности. Но, преследуя эти цели, мы не должны закрывать глаза на то, что врожденные способности распределяются среди людей крайне неравномерно, и еще более непредсказуемо складываются факторы, способствующие воспитанию, шлифовке и проявлению природных талантов.

Поэтому должны быть предусмотрены такие структуры и процессы, которые способны предоставить свободу и право создавать выдающиеся проекты тем людям, которые показали себя на это способными[6].

Необходимость в поддержке великих проектировщиков со стороны требующих инноваций авторитетных руководителей

Прежде всего, старшие руководители организации должны сами проявлять горячее стремление к созданию инновационных продуктов на основе выдающихся проектов. Это стремление отчетливо проявлялось в компании Apple в первый период правления Стива Джобса, стало менее ярко выраженным при его преемниках и снова набрало силу, когда Стив вернулся к руководству компанией. Стремление к новаторству было также весьма заметно в компании IBM при Томасе Дж. Уотсоне-старшем (главный администратор в 1914–1956 гг.) и Томасе Дж. Уотсоне-младшем (главный администратор в 1956–1971 гг.). Можно также привести большое количество аналогичных примеров.

Способы организации процесса, способствующего созданию выдающихся проектов

Я на себе испытал те сложности, с которыми приходится сталкиваться, становясь участником обременительного организационного процесса, и приобрел определенный опыт обхода или нарушения некоторых его процедур, но не имею опыта в упрощении или реструктуризации подобных процессов. Этим обязано время от времени заниматься каждое предприятие. Я могу лишь пожелать, чтобы подобная работа была поручена первоклассному организатору с предоставлением ему широких полномочий по устранению ненужных процедур и лимитированию времени на выполнение оставшихся.

При проектировании нового организационного процесса подготовки к производству или реструктуризации существующего приходится прилагать значительные усилия по преодолению сил противодействия, в которых воплощаются естественные тенденции к отрицанию всего непредсказуемого. Необходимо добиться того, чтобы организационный процесс позволял, допускал и даже поощрял создание выдающихся проектов.

Для этого процесс подготовки к производству должен прежде всего позволять явно идентифицировать проблемы первостепенной значимости и сосредоточиваться на принятии решений по ним, и только по ним. В основе применяемой при этом организации работ, безусловно, должна лежать защита интеллектуальной собственности. Применяемые при этом средства обязаны надежно обеспечивать неприкосновенность "драгоценностей короны", т.е. самых важных секретов, но не менее важно, чтобы не затрачивались напрасно ресурсы на защиту от постороннего взгляда того, что не имеет особой значимости. При организации системы безопасности необходимо соблюдать сдержанность и умеренность, помня о том, что службы защиты часто не знают меры в своих усилиях. В главе 27, "Практический пример. Организация объединенного вычислительного центра", показано, насколько успешным оказался подход на основе первоочередного выявления наиболее значимых вопросов в конкретном случае.

Кроме того, в процесс подготовки к производству должны быть встроены простые и быстродействующие механизмы преодоления сбоев, для ввода которых в действие достаточно обращения любого координатора проекта и утверждения только одного руководителя, обладающего соответствующими полномочиями. Другими словами, те, от кого зависит судьба проекта, всегда должны быть готовы беспристрастно разобраться в ситуации и проявить здравый смысл, а рядовые исполнители должны знать об этом и без колебаний обращаться за помощью: "Нет таких правил, которые не могут быть нарушены ради интересов дела".

Обеспечение концептуальной целостности — проявление большего доверия главному проектировщику при создании проекта

Самым важным атрибутом выдающегося проекта является концептуальная целостность, а единственный способ ее достижения — делегирование всех полномочий по решению задач разработки лишь одному специалисту (если же в этой работе участвуют несколько человек, то они должны во всем достигать полного единогласия). Поэтому мудрый руководитель смело поручает выполнение каждого задания по проектированию одаренному главному проектировщику[7].

Из такого решения вытекает много следствий. Во-первых, сам руководитель не должен вмешиваться в процесс проектирования. Искушение взять на себя уточнение всех нюансов действительно велико, поскольку в таком коллективе руководитель не может не быть по своей подготовке проектировщиком. Однако возможно, что руководитель одарен как проектировщик не больше своего подчиненного (проектировать и руководить — две весьма несходные профессии), и к тому же ему неизбежно приходится уделять внимание решению других задач, которые непосредственно входят в круг его обязанностей.

Во-вторых, для всех должно быть совершенно незыблемым правилом, что главный проектировщик, пусть даже имея в своем подчинении лишь несколькими помощников, обладает всей полнотой власти над проектом и занимает по отношению ко всем исполнителям такое же положение, как и административный руководитель проекта.

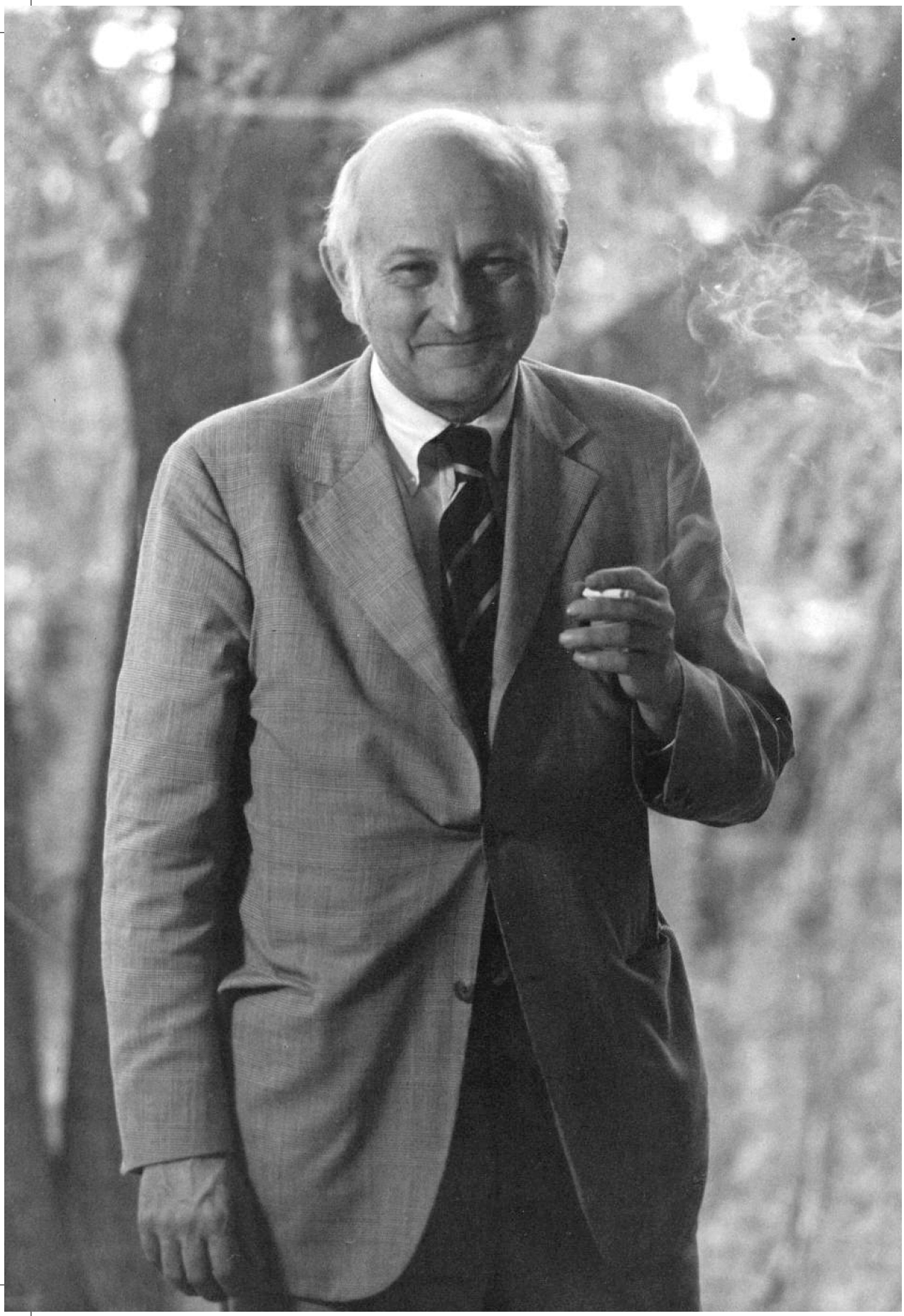
В-третьих, руководитель проекта должен защищать главного проектировщика от любых сторонних попыток вмешаться в ход проекта и планировать работу так, чтобы творческий коллектив не ощущал на себе жесткого прессинга, связанного с возможным изменением сроков.

В-четвертых, главному проектировщику необходимо предоставлять любые инструментальные средства и любую помощь, которые ему потребуются. Важнее всех заданий то, которое выполняют разработчики.

Примечания и ссылки

1. В докладе [2] констатируется факт, что это правило остается истинным для многих из наиболее успешных и новаторских проектов разработки оружия.
2. Этому универсальному феномену подвержено также программное обеспечение, что убедительно показано в классическом исследовании Лемана и Белэди роста энтропии в течение времени в OS/360 компании IBM (см. [136]). Более полный обзор приведен в важной работе [137].
3. См. статью "Template zombies", глава 86, из книги [77]. Я не считаю, что эта лаборатория IBM была поражена болезнью "превалирования формы над содержанием", которой страдают слепые приверженцы шаблонов. Ее руководители всего лишь добросовестно выполняли письменные указания, отправленные из далекой штаб-квартиры.
4. Такая стойкость была отмечена наградой National Medal of Technology в 2005 году.
5. Джеймс Робертсон (James Robertson) из компании Atlantic Systems Guild сообщил мне эти слова представителей Apple в 2008 году (по материалам личного общения).
6. В работе [70] описан замечательный практический пример применения методики Гордона Мюррея (Gordon Murray), проектировщика гоночных автомобилей, неизменно завоевывающих призовые места. В этой статье приведен отчет о деятельности коллектива, руководимого главным проектировщиком, в ходе которой удается обойтись без многих формальных процессов и обернуть к своей пользе ограничения.
7. Заместитель премьер-министра Великобритании обратился в Королевскую инженерную академию наук (Royal Academy of Engineering — RAE) с просьбой провести исследование и составить отчет по поводу того, как повысить безопасность железнодорожного транспорта. RAE поручила решение этой задачи комитету, состоящему только из одного человека: ее президента, сэра Дэвида Дэвиса (David Davies). Отчет был подготовлен в рекордное время и оказался кратким и четким, изобилующим конкретными рекомендациями (см. [75]).

TheDesignofDesign.indb 291 30.10.2012 15:58:36



TheDesignofDesign.indb 292 30.10.2012 15:58:36

20

Предпосылки появления выдающихся проектировщиков

Гений может развиться и проявить себя даже при отсутствии обучения, но это не означает, что такой редкий цветок не надо поливать и подрезать.

МАРГАРЕТ ФУЛЛЕР (MARGARET FULLER),
1820–1850 ГГ., ЗАПИСЬ В ДНЕВНИКЕ

Каждый человек, который поднимается выше среднего уровня, не может не получить два образования: первое — от своих преподавателей; второе, более личное и важное, — от себя.

СМ. [101]

◂ *Джон Кок (John Cocke) (1925–2002 гг.), компьютерный гений. Компания IBM, Научно-исследовательская лаборатория Уотсона*

В предыдущих главах было показано, что для создания выдающихся проектов нужны прежде всего выдающиеся проектировщики, и их не могут заменить даже наиболее хорошо организованные процессы проектирования. Безусловно, в наши дни техническое проектирование всегда требует привлечения больших коллективов, но остается в силе правило, что эти коллективы формируются вокруг выдающихся проектировщиков: Джон Рёблинг (подвесной бруклинский мост), Джордж Готалс (Панамский канал), Р. Дж. Митчелл (самолет Spitfire), Сеймур Крэй (CDC 6600, суперкомпьютеры Cray 1), Кен Томпсон и Деннис Ричи (UNIX).

Создатели высокотехнологичных проектов сталкиваются с особыми сложностями, обусловленными наличием изначального конфликта между принципом индивидуального проектирования, без которого невозможно достичь больших успехов в искусстве, литературе и технике, и принципом коллективного проектирования, без которого нельзя обойтись в связи с высокой сложностью применяемых артефактов и высоким темпом развития современных экономик.

- Возможности подготовки выдающихся проектировщиков.
- Необходимость прежде всего разработать процессы проектирования, которые поддерживают и поощряют развитие выдающихся проектировщиков, и вместе с тем не позволяют заключать их в узкие рамки и нивелировать их работу.
- Необходимость найти способы обеспечения поддержки выдающихся проектировщиков с привлечением к их работе коллективов.

Выбор правильных способов обучения проектированию

В наши дни формальное обучение проектировщиков организовано совершенно неправильно. В частности, Шён высказал такую мысль (эта цитата немного перефразирована)[1]: “Согласно принципам технической рациональности, обучение всем профессиям должно быть организовано так, как до сих пор ведется обучение инженеров: сначала преподавание соответствующих базовых и прикладных предметов, а затем формирование практических навыков”.

В этом Шён категорически не согласен с теми, кто поддерживает принципы технической рациональности. Он утверждает, что освоение всех профессиональных навыков должно осуществляться на практике под пристальным критическим наблюдением и что данное требование справедливо для медицины, юриспруденции, администрирования, архитектуры, искусства, музыки, работы в сфере социальных отношений и, разумеется, технических дисциплин. Это признано в медицинском образовании, поэтому с третьего года обучения студенты проводят все больше времени в клиниках, участвуют в консилиумах и принимают на себя

ответственность за конкретных пациентов. В архитектурном образовании эта истина никогда не подвергалась сомнению, поэтому за всю его историю занятия в проектных мастерских всегда рассматривались как самые важные.

С другой стороны, в США основная доля времени в процессе формального обучения технических проектировщиков затрачивается на прослушивание лекций в аудиториях и на проведение в лабораториях стандартных экспериментов, а не на выполнение проектов, подвергаемых критике. Это относится в еще большей степени к программным инженерам.

Я признаю справедливость утверждений Шёна. При таком образовании мы неправильно используем самый дорогой ресурс в процессе обучения: время студентов. Поэтому технические институты все чаще предусматривают проведение критически оцениваемых практических работ по проектированию в своих учебных расписаниях, несмотря на то, что это требует от учебного заведения дополнительных затрат.

Не подлежит сомнению то, что инженеры, независимо от того, проходят ли они подготовку в области аппаратных средств или программного обеспечения, должны обладать базовыми научными знаниями, лежащими в основе профессии, причем такие знания должны быть приобретены в первую очередь. В своем стремлении улучшить подготовку будущих специалистов наиболее прогрессивные образовательные учреждения отвергают и этот принцип, предлагая студентам разрабатывать критически оцениваемые проекты с первых лет обучения, одновременно с теоретической подготовкой. Но в учебных расписаниях по компьютерным наукам такое все же встречается довольно редко.

Следует также отметить, что учебные расписания многих известных технических учебных заведений часто основаны на так называемых "кооперативных" или "многослойных" программах, согласно которым студенты чередуют практические занятия на рабочих местах (и обучение в компании) с получением академического образования базового и заключительного уровней. Тем не менее почти нельзя найти такие учебные расписания занятий по компьютерным наукам, в которых применялся бы такой принцип.

Неудовлетворительное состояние современного академического формального образования главным образом обусловлено тем, что в его основе по-прежнему лежит обучение студентов на лекциях и семинарах, а не приобретении ими знаний в ходе создания критически оцениваемых проектов. Если будет решено действительно организовать обучение на основе усвоения стилей проектирования, то учащиеся будут разрабатывать архитектуру вычислительных систем с учетом всех ограничений, как это делал Крэй, создавать фуги в стиле Баха или строить здания в стиле Рена. Затем хорошо осведомленный и проницательный наставник сможет указать на стилистические несогласованности и подвергнуть критике, допустим, то, что превосходный во всех отношениях проект не соответствует фактически поставленным узким целям.

Если обучение проводится в области технических дисциплин или компьютерных наук, то свои критические замечания сможет высказать лишь такой наставник, который обладает не только определенной степенью уверенности в своей правоте, но и, конечно, смелостью. Но чаще всего мы не желаем выражать субъективную критику в произвольной форме и не умеем этого делать, по-видимому, потому, что предпочитаем целиком опираться на научно обоснованные решения. Тем не менее без этого нельзя обойтись при обучении проектированию.

Роль наставника, подвергающего критике предлагаемые проекты, могут также выполнять по отношению друг к другу сами студенты. Для проектировщика наиболее эффективный способ изучения сложившихся стилей проектирования состоит в том, чтобы обучать этим стилям других проектировщиков.

Привлечение к работе преимущественно с учетом задатков в области проектирования

Руководители, принимающие на работу новых проектировщиков, слишком часто оценивают их подсознательно по тем принципам, которых сами придерживаются в своей деятельности: "А может ли он успешно выполнять ту работу, которую делаю я?" Таким образом, выбор чаще всего делается в пользу яркой личности, лидера, человека, который эффективно проявляет себя на совещаниях. В результате можно не заметить талантливого интроверта, если он неразговорчив и особенно не похож на других. Но многие выдающиеся проектировщики принадлежат именно к этой категории людей! (Я не утверждаю, что если человек внешне не проявляет себя, то он потенциальный выдающийся проектировщик. Об этом невозможно узнать заранее.) Мы, руководители, иногда не можем разглядеть в человеке талант, допуская тем самым утрату перспектив для самого человека и потерю для общества.

Как обеспечить более успешный отбор перспективных сотрудников? Во-первых, необходимо постоянно напоминать самим себе, кого мы ищем. Во-вторых, следует обязательно изучать портфолио проектных работ кандидата, а не только выслушивать его устные отчеты о прошлом опыте. Например, в компании Microsoft существуют программы испытания кандидатов. К сожалению, такая практика еще не является общепринятой в компаниях, занимающихся разработкой программного обеспечения.

Целенаправленная подготовка будущих специалистов

Во многих ведущих промышленных и военных организациях предусмотрены сложные и тщательно продуманные процессы повышения квалификации специалистов, позволяющие подниматься по карьерной лестнице от рядовых

исполнителей до уровня руководителей и даже до уровня высшего управленческого состава. Причем на каждом этапе карьеры, например военной, предусмотрены разные программы обучения: для вновь испеченного лейтенанта, для майора и генерала. Это позволяет быстрее всего выявлять и стимулировать развитие перспективных и талантливых людей. Обязательным является контроль со стороны наставников. Тщательно запланированная ротация должностей позволяет приобретать разносторонний опыт. Наиболее перспективные сотрудники назначаются помощниками к ведущим профессионалам. Например, ассистентами судей Верховного Суда становятся самые многообещающие молодые юристы.

Господь помог на этом пути сформироваться многим лидерам. Моисей, благодаря стечению обстоятельств, часто присутствовал на судебных заседаниях, проводимых египетским фараоном, где и приобрел опыт руководства нацией. Давид, арфист короля Саула, наблюдал за тем, как осуществляется государственное правление и вырабатываются справедливые судебные решения. Апостол Павел был подготовлен к тому, чтобы ясно формулировать и разъяснять полностью новое понимание путей, открываемых Богом, получив превосходное образование в части Ветхого Завета в качестве писца у Гамалиила, величайшего раввина тех дней[2]. После встречи с Иисусом Христом он радикально переменил свои взгляды и был на некоторое время отправлен в пустыню, чтобы он мог заново продумать все свое мировоззрение[3].

Я не наблюдаю признаков того, чтобы руководители большинства технических организаций целеустремленно занимались формированием эффективного подхода к подготовке ведущих специалистов по техническим направлениям, не говоря уже о выдающихся проектировщиках, деятельность которых столь важна для преуспевания компании.

Как сделать принцип "разных лестниц" действующим и признанным

В отличие от менеджеров, которые сами знают, как продвигаться по службе, при подготовке проектировщиков необходимо прежде всего специально разработать подходящий путь карьерного роста, такой, что позволяет компенсировать затраченные усилия и сформировать социальный статус, отражающий истинную значимость работы проектировщика для предприятия с творческой направленностью. Для этого обычно применяется так называемый принцип "разных лестниц". Я привел описание этого принципа в другой своей работе; а здесь лишь повторю, что можно сравнительно легко назначить зарплаты, соответствующие конкретным ступенькам карьерной лестницы (в частности, имеются и рыночные механизмы, позволяющие решить эту задачу), но необходимо также заранее предусмотреть равные атрибуты престижа тем, кто продвигается по разным лестницам: одинаковые офисные помещения, сравнимую поддержку рядового персонала, такие же обратно пропорциональные степени расширения круга обязанностей прибавки к зарплате[4].

Почему подход на основе "разных лестниц" требует особого внимания? Возможно потому, что выполняющие руководящие функции менеджеры, которым также ничто человеческое не чуждо, по своему глубокому убеждению склонны считать собственные задачи более трудными и важными, чем проектирование, и скрепя сердце вынуждены оценивать степень проявления новаторства и творчества в этой "второстепенной" области.

Планирование формальных мероприятий по обучению

Подающие надежды проектировщики (как и подающие надежды менеджеры) нуждаются в том, чтобы неизменно продолжалось их систематическое обучение, чередуясь с реальной профессиональной практикой, которая проводится под критическим взглядом главного проектировщика.

В связи с чем требуется снова и снова возвращаться к систематическому обучению? Современный мир непрерывно изменяется. В высокотехнологичных дисциплинах отрицательные последствия прекращения дальнейшего повышения квалификации в своей области вскоре становятся очевидными и почти угрожающими. С тех пор, как я соприкоснулся с компьютерной отраслью в 1952 году, моя профессиональная интеллектуальная деятельность стала напоминать плавание по бурному морю. Не успеваешь справиться с одной волной, преодолев очередной рубеж в приобретении знаний, как на горизонте встает следующая! Этот мир постоянно меняется, поэтому заставляет всегда оставаться восприимчивым к новинкам и тем замечательным возможностям, которые они приносят с собой. Таким образом, первой причиной, которая обусловливает необходимость систематического образования, является непрерывная смена интеллектуального и практического инструментария.

Лично я пришел к выводу, что наиболее эффективным и экономичным методом переобучения являются краткие формальные курсы, и стараюсь посещать их в среднем один раз в год. Чем это обусловлено? Разве не возможно оставаться осведомленным о современных событиях, читая журналы и посещая конференции? Возможно! Но я — сторонник систематического образования и считаю, что хороший преподаватель, который тщательно подготавливает сбалансированный краткий обзор темы, способен повысить для меня эффективность обучения примерно в два с лишним раза, быстро преподнести взвешенный обзор и раскрыть перспективы, что потребовало бы от меня изучения не меньше нескольких десятков журналов. Как преподаватель, который берет на себя обязанности предоставить такую же эффективность освоения новых знаний своим клиентам, я стараюсь купить такие услуги для самого себя.

Второй причиной является необходимость постоянного углубления и расширения знаний. Этой цели можно наиболее эффективно достичь, изучая и удачные, и провальные проекты предшественников и современников. С данной точки зрения еще одним существенным преимуществом, предоставляемым систематическим

образованием, является отстраненность, поскольку профессиональный преподаватель всегда остается более объективным, изучая конкурирующие концепции проекта и стили. В любой проектной организации сложилась своя корпоративная культура, опирающаяся на собственные традиции и взгляды, что касается также спонсируемого компанией систематического образования. В этом и состоит наилучший стимул для подающих надежды проектировщиков и их наставников, чтобы искать источники систематического образования за пределами своей компании.

Одно из моих самых плодотворных решений в роли руководителя подразделения компании IBM не имело никакого отношения к разработке продукции. Оно состояло в отправке в середине карьеры многообещающего инженера в качестве полностью занятого служащего IBM в Мичиганский университет для получения степени доктора философии. Это решение, казавшееся в свое время тяжело загруженному руководителю работ по созданию архитектуры вычислительной системы довольно непродуманным кадровым мероприятием, принесло компании IBM такую отдачу, перед которой померкли самые смелые мечты. Защита Тэдом Коддом (Ted Codd) диссертации на степень доктора философии стала для него хорошей подготовкой к исследовательской карьере, а его исследования привели к открытию концепции реляционных баз данных и получению премии Тьюринга[5]. Реляционные базы данных стали основным приложением для самой прибыльной линейки компьютеров IBM больше чем на двадцать пять лет.

Планирование приобретения опыта работы во многих областях

Мы должны готовить подающих надежды проектировщиков так же, как в самых лучших организациях теперь готовят подающих надежды менеджеров. Ключевым словом является "план"; необходимо продумывать даже само продвижение карьеры молодого человека с учетом приобретения наиболее разнообразного опыта и достижения наибольшей степени участия в работе, ставить перед молодежью все более важные задачи и требовать все большей ответственности.

Часто оказывается, что наибольшую пользу приносит назначение молодых проектировщиков на должности именно в тех организациях, где находятся пользователи проектируемых ими объектов. Я сам сравнительно недолго работал в коммерческом вычислительном центре обработки данных, в котором создавалась программа учета заработной платы для сорока штатов США; в научном вычислительном центре, занимавшемся расчетами траекторий ракет; в криптоаналитическом вычислительном центре; в лаборатории по коммутации телефонных сигналов, которая решала задачу идентификации тех, кто набирал номера телефонов по четырем линиям коллективного пользования; а также в лаборатории технической физики, где производились измерения земных колебаний с точностью до тысячных долей дюйма. Этот опыт оказался буквально бесценным, поскольку помог мне понять на интуитивном уровне, в чем состоят требования пользователей

компьютеров. Как было описано в одной из предыдущих глав, я в течение двух недель занимал в вычислительном центре более низкую должность ученика оператора ЭВМ, заправляя в устройства магнитные ленты. Это также позволило мне в большей степени приобщиться к действительности и воспользоваться этим опытом при проектировании пульта оператора. Девью (Dewey) был прав, указывая, что мы учимся на деле; эффективная программа подготовки проектировщика должна предусматривать получение разнообразного опыта.

Планирование творческих отпусков, проводимых за пределами организации

Проектировщиков, карьера которых уже начала складываться, можно поощрять и стимулировать, предоставляя им творческие отпуска. Это может быть временная помощь клиенту, преподавание в учебном заведении или временное назначение на работу в государственном учреждении. Инвестиции, позволяющие предотвратить застой в работе людей с новаторским складом ума, всегда оправдываются.

Творческий характер руководства работами творческого направления

История Джона Кока (John Cocke) и Ральфа Гомори (Ralph Gomory). Джон Кок был, по-видимому, наиболее выдающимся, а также, безусловно, наиболее творческим человеком, которого я когда-либо знал. Вместе с ним мы подключились к разработке суперкомпьютера Stretch в компании IBM в июле 1956 года, будучи новоиспеченными докторами физических наук. Нам сразу же выделили одну секцию на двоих в общем зале, а затем перевели в офис, рассчитанный на двух человек. Это совместное размещение нас вполне устраивало, поскольку Джон любил работать в одиночестве по ночам, а я работал днем. Джон испытывал глубокую страсть в отношении всего, что касалось компьютеров. Я подозреваю, что он день за днем посвящал больше часов размышлениям о компьютерах, чем обо всем остальном, вместе взятом. Джон глубоко разбирался не только в компьютерной технике, но и во всех связанных с ней научных и технологических направлениях[6]. Кроме того, в нем сочетались редкие личные качества, он был одновременно очень вдумчивым и открытым. Как моряк в старинной песне, "он один мог справиться с тремя", когда требовалось объяснить другим свои последние идеи. Джона нельзя было не любить. Он был приветливым, в высочайшей степени благородным и всегда неравнодушным. Многие особенности личности Джона, его стиля и поведения ярко отражены в статье Харвуда Колски (Harwood Kolsky)[7].

Кок сумел привлечь к сотрудничеству выдающихся людей, которые помогали постигать и воплощать в жизнь его чрезвычайно многочисленные идеи. Тремя из этих идей, каждая из которых достойна премии Тьюринга, были следующие: идея конвейерной обработки команд, предложенная совместно с Колски (Kolsky)[8],

идея глобальной оптимизации компилятора, выдвинутая с участием Френа Аллена (Fran Allen) и Джека Шварца (Jack Schwartz)[9], и идея архитектуры вычислительной системы с сокращенным набором команд, предложенная совместно с Джорджем Радином (George Radin)[10].

Остается лишь гадать, как Джон Кок, столь выдающийся гений, не способный руководить даже небольшой группой, редко что-либо публиковавший и главным образом размышлявший и общавшийся с талантливыми людьми, сумел внести такой большой вклад в создание современной техники. Эта история посвящена двум гениям, и вторым является Ральф Гомори, директор IBM по исследовательской работе и старший вице-президент по науке и технике. Гомори, как и Кок, получил Национальную научную медаль (National Medal of Science) за личный вклад в науку.

Гомори создал саму организацию, ее атмосферу и стиль управления, направленный на то, чтобы каждый человек в подразделении IBM Research сумел вложить в общее дело что-то свое, раскрывая таланты, которыми он обладал. Когда речь шла о Коке, Ральф говорил: "Я не отношусь к Джону как-то иначе по сравнению с другими". Но посторонний человек, слыша его слова, не мог знать об одной особенности: Ральф действительно относился одинаково к каждому из своих выдающихся сотрудников, учитывал при этом их разные характеры и потребности. Он также говорил, тая гордость, что "Джон был самым высокооплачиваемым человеком в подразделении IBM Research, поскольку вносил самый большой вклад"[11].

Настоятельная необходимость во всем оберегать талантливых людей

Избавление талантов от всего постороннего

Как только станет ясно, что рядом с вами выдающийся проектировщик, необходимо сделать все, чтобы он занимался только проектированием. Для достижения высокой производительности проектирования требуется добиться бесперебойности в работе, не прерывающего умственного состояния высокого творческого накала и концентрации. Все мы, проектировщики, испытали это восхитительное чувство и жаждем его повторения. Перу ДеМарко (DeMarco) и Листера (Lister) принадлежит превосходное обсуждение того, в чем заключается смысл бесперебойной работы, какова ее значимость и как ее добиться организационными средствами[12].

В современных организациях возникает много препятствий и отвлечений, из-за которых нельзя добиться бесперебойной работы:

- встречи;
- звонки по телефону;

- электронная почта;
- правила и обязательные требования;
- состоящие в штате бюрократические и "сервисные" группы, которые разрабатывают правила так, чтобы им было проще выполнять свои собственные функции;
- клиенты;
- посетители и журналисты, причастные к разрабатываемой тематике.

Во многих творческих организациях приняты такие организационные требования, как соблюдение тишины по утрам, чтобы хоть в какой-то период работа проводилась без помех. Когда в IBM была предпринята попытка догнать компанию Apple, выпустившую свои первые персональные компьютеры, главный администратор Джон Опель (John Opel) организовал закрытую лабораторию в городе Бока-Ратон, чтобы разработать свой ответ от IBM. Другим служащим IBM, включая даже штатных сотрудников корпорации с четко определенными правами на это, просто не разрешалось входить в лабораторию.

Аналогичным образом я сам закрыл доступ в помещения, где разрабатывался проект System/360, для всех, кто не был связан с проектом, как для сотрудников IBM, так и для клиентов, с января по апрель 1964 года. У нас было слишком много работы, которую нельзя было прерывать[13].

Я проводил интервью с Джеффри Джаппом (Jeffrey Jupp), техническим директором компании Airbus UK, по вопросам проектирования и изготовления в Великобритании крыльев для самолетов Airbus, фюзеляжи которых проектировались и изготавливались во Франции. Наша беседа сложилась просто замечательно, и в конце я спросил, можно ли поговорить с главным разработчиком проекта. Ответом было твердое "Нет". Я понял мотивы Джеффри и отнесся к ним с уважением.

Защита проектировщиков от неумелых руководителей

Посредственный или несдержанный руководитель способен задушить творческий потенциал любого проектировщика. Недалекие руководители часто не могут понять, какие жемчужины появились в их коллекции. Иногда они не осознают, какую важность имеет проектирование для достижения общего успеха руководимого ими коллектива. Иногда они не понимают, что необходимо сделать для того, чтобы волшебство проектирования воплотилось в реальность.

Иногда руководитель затаивает злость или не может согласиться с тем, что подчиненный проектировщик фактически его превосходит. Иногда для руководителя становится оскорбительным, если выдающийся проектировщик получает больше, чем он сам. Результатом становится отказ в поддержке, не предоставление помощи, мелкая оскорбительная травля.

Таким образом, для руководителей высшего звена задача вполне очевидна: они должны активно работать над воспитанием руководителей первого уровня, стремясь в конечном итоге развить у них способность использовать не только чужие, но и собственные таланты и сознавать свою особую роль, обучая их навыкам поддержки коллектива и лидерства.

Избавление проектировщиков от выполнения руководящих функций

Я не раз был свидетелем того, как людей с потенциалом выдающихся проектировщиков отвлекали от задач проектирования и вынуждали принимать на себя функции руководителей. Им так и не удавалось реализовать свой потенциал. К сожалению, корпоративная культура наших организаций поощряет такое перенаправление усилий и даже вынуждает проектировщиков выполнять обязанности руководителей. Чтобы противостоять такому сложившемуся подходу, необходимо быть действительно целеустремленным и даже решительным человеком.

Сеймур Крэй (Seymour Cray), самый выдающийся проектировщик суперкомпьютеров, подает нам вдохновляющий пример. Крэй занимался проектированием компьютеров первого поколения на вакуумных лампах в тех благоприятных условиях, которые сложились в то время в городе Сент-Пол, штат Миннесота, сначала в компании Engineering Research Associates, а затем в корпорации Control Data Corporation (CDC). Чтобы спроектировать компьютер CDC 6600, он перевел весь свой коллектив, по его словам, "из тридцати пяти человек, включая швейцара" в отдаленное и уединенное место и освободил себя от всех прочих обязанностей в CDC[14].

Результаты создания модели CDC 6600 оказались чрезвычайно успешными, поэтому, снова войдя в состав руководства корпорации CDC, Крэй распорядился, с согласия CDC, об отъезде коллектива разработчиков на изолированную от внешнего мира рабочую площадку в прериях, принадлежащую вновь созданной корпорации Cray Computer Corporation, для работы над новой моделью. Он лично наблюдал за всеми аспектами создания компьютера Cray 1, начиная с электрических схем и систем охлаждения и кончая компилятором для языка Fortran.

Затем, поскольку корпорация Cray Computer достигла невиданных успехов и снова стала привлекать его к руководящей работе, Крэй отправил свой коллектив на новое место, в штат Колорадо, чтобы основать корпорацию Cray Research Corporation. Лишь автомобильная авария по вине пьяного водителя положила конец его выдающимся достижениям[15].

Как развить свои способности в качестве проектировщика

Если человек осознал свое призвание в качестве технического проектировщика, то ему необходимо определить пути развития своих навыков. Прежде всего,

может ли он рассчитывать на то, что найдет в этом помощь вне рамок собственной дисциплины? Я считаю, что ответ на этот вопрос должен быть положительным. Проектировщик должен начать с планирования программы роста для себя самого[16]. В решении этой задачи ему никто не поможет.

Развитие способности представлять проекты в виде эскизов

Проектировщики учатся проектированию, проектируя. Некоторые из эскизов должны быть полностью детализированы, поскольку не зря говорят, что "дьявол скрывается в деталях", и многие большие корабли погибли, наскочив на малозаметную подводную скалу. Превосходные примеры того, как лучше всего готовить эскизы, можно найти в записных книжках Леонардо да Винчи. Вполне может оказаться так, что подающий надежды молодой проектировщик программного обеспечения ведет дневник, в котором он фиксирует заинтересовавшие его идеи в изученных образцах, а также отражает итоги собственных построений.

Стремление к получению обоснованной критики собственных проектов

Дональд Шён (Donald Schoen) в своей превосходной книге [186] убедительно доказывает, что в действительности практическая работа в условиях внешнего критического контроля — единственный успешный метод обучения практической работе. Он перечисляет одно направление человеческой деятельности за другим (юриспруденция, медицина, архитектура, строительство, ремесло в средневековых гильдиях), которые в процессе свей эволюции подошли к этому методу обучения (скорее всего, независимо друг от друга)[17]. В основе применяемого в наши дни процесса подготовки диссертации доктора философии лежит точно такой же метод обучения практике исследовательской работы.

Изучение образцов и прецедентов

Начинающий проектировщик, занимаясь этим, идет по пути многих великих проектировщиков. Роберт Адам (Robert Adam) изучал творчество Кристофера Рена (Christopher Wren). Рен подробно знакомился с работами Палладио (Palladio). Палладио попросил своего отца, чтобы он поддержал его намерение отправиться в Рим и заняться рисованием развалин великих римских зданий. Римляне изучили и соединили воедино архитектурные стили этрусков и греков. Каждый выдающийся проектировщик овладел богатым наследием своих предшественников, а затем добавил новые, собственные концепции.

Чтобы изучение образцов принесло наибольшую пользу, необходимо прежде всего выработать к ним уважительный подход. Эти прецеденты, репутация которых оставалась незыблемой в течение столетий, несмотря на критику, носят печать глубокого превосходства. В более новых областях доступный обозрению

промежуток может измеряться всего лишь десятилетиями. Но как бы ни был обширен набор прецедентов, задача студента состоит в том, чтобы найти и освоить созданные ранее превосходные образцы творчества, пусть даже его склонности или новые обстоятельства, в которых он находится, побуждают его двигаться совсем в ином направлении.

Специалист в области компьютерной архитектуры должен ознакомиться со всем разнообразием серийно выпускаемых компьютеров. Не зря ведь кто-то в свое время счел эти модели достойными вложения реальных денег. (Намного более многочисленные варианты архитектур остались лишь на бумаге, в чертежах и публикациях, поэтому, разумеется, не прошли столь строгой проверки, как практически реализованные, и, следовательно, не заслуживают такого внимания.)

Вырабатывая для себя правильный подход к овладению опытом проектирования на основе изучения прецедентов, очень важно не подвергать сомнению компетентность своего предшественника. Иными словами, столкнувшись с чем-то неожиданным, мы должны в первую очередь искать ответ на вопрос:

"Что побудило этого безусловно талантливого проектировщика поступить именно так?"

Было бы ошибкой заведомо занимать такую позицию:

"Почему он допустил подобный просчет?"

Обычно истинный смысл принятого проектного решения можно найти, поняв, какие цели преследовал проектировщик и с какими ограничениями сталкивался; обнаружение этого обычно становится источником настоящего прозрения. Безусловно, в разногласиях между супругами чаще всего самым мудрым и обычно правдивым ответом на вопрос "Почему ты сделал...?" или "Почему ты не сделал...?" становится "Это просто недоразумение", но недоразумения чрезвычайно редко становятся причиной проектных решений, и об этом всегда следует помнить при их изучении.

При любой возможности стремитесь стать участником обсуждения проектировщиками своей текущей работы. При любой возможности старайтесь прочесть все, что проектировщики пишут о своей деятельности.

Мы с Джерри Блау приобрели очень многое после того, как провели собственные исследования архитектур вычислительных систем других разработчиков и преобразовали полученные результаты в общий формат. Мы выработали единую структуру изложения, приняли стандартный масштаб эскизов, согласовали общепринятые элементы текстового описания, взяли на вооружение общий формальный язык описания, а также договорились о том, что каждая система будет представлена в сопровождении краткого критического анализа основных характеристик и особенностей[18].

Проект самообразования — поэтажный план дома площадью 1 000 квадратных футов

В школе проектирования Университета штата Северная Каролина принято, что проектировщики с самого начала изучения курса проектирования приступают к выполнению одного полезного упражнения, независимо от того, в какой дисциплине они специализируются.

Задание. Спроектировать план дома площадью 1 000 квадратных футов для семьи с двумя родителями и двумя детьми: сыном в возрасте трех лет и дочерью в возрасте шести лет. Площадка для застройки находится в северной Вирджинии, в пригородной зоне, на расстоянии 50 футов от шоссе, ширина участка — 70 футов, на участке имеются деревья, фасад дома ориентирован на юг.

Журнал. Ведите датированный журнал с вопросами, касающимися проектирования, решениями и обоснованиями. Ниже приведены некоторые вопросы, которые подлежат рассмотрению.

- Подготовьте более детализированное архитектурное задание, руководствуясь своим воображением. Внесите его в свой журнал.
- Какие ограничения следуют из этого задания?
- Что рассматривается в качестве лимитируемого ресурса? Как вы справляетесь с распределением этого ресурса?
- Каким требованиям к проекту вы следовали явно или неявно?
- Если приходится сталкиваться с альтернативными вариантами развития проекта, как вы определяете наиболее приемлемый?
- Вы использовали какое-либо инструментальное средство автоматизированного проектирования? Если да, оцените его продуктивность в сравнении с разработкой проекта, проводимой вручную, на различных этапах решения задачи.
- Какие действия выполнялись вами на разных этапах? Проанализируйте свой журнал и нарисуйте схему принятой вами траектории проектирования.
- Оценка. Какими достижениями ознаменовался ваш проект? В чем заключались основные недостатки?

Примечания и ссылки

1. См. [186].
2. Деяния 22:3; статья "Гамалиил" в Википедии (`http://en.wikipedia.org/wiki/Gamaliel`), в книге взята за основу версия этой страницы от 25 апреля 2008 года.
3. К Галатам 1:17-18; Деяния 22:3. Однако, в отличие от них всех, Иисус не получил образование у знатоков Ветхого Завета и должен был опираться на

сохранившиеся в его памяти сведения из Священного Писания. Он обрел полностью новое понимание религиозных канонов, много лет размышляя над ними в мастерской плотника (От Луки 2:41-52).

4. См. [40], с. 118–120, 242.
5. Проведите поиск по имени Edgar Frank Codd, чтобы найти его работы.
6. Будучи больным и прикованным к креслу, Кок поделился со мной новыми знаниями и своими последними идеями, касающимися того, как их можно применить для разработки компьютеров.
7. `http://www.cs.clemson.edu/~mark/kolsky_cocke.html`, в книге взята за основу версия этой страницы от 26 ноября 2008 года.
8. См. [62]. В действительности эта работа посвящена конвейерной обработке команд, а не виртуальной памяти, в том виде, в каком это понятие трактуется сегодня.
9. См. [63]; [9].
10. Концепция RISC (Reduced Instruction Set Computer) часто трактуется неправильно. Основная ее идея — не сокращенный набор команд, а набор упрощенных, т.е. более примитивных, команд. В своей крайней форме эта концепция подразумевала отказ даже от команд для работы с частями операндов, такими как "Сдвиг на N битов" или "Умножение". Это позволяет свести до минимума размер цикла "аккумулятор–сумматор–аккумулятор", и в случае использования кеша команд и оптимизирующего транслятора работа компьютера существенно ускоряется. Я не знаю никого, кроме Джона, мастерски владевшего проблематикой проектирования компьютеров и оптимизации компиляторов, кто был бы способен выдвинуть эту идею. Джордж Радин внес важный вклад, но исходные статьи, [171] и [172], должны по праву носить имя Кока как ведущего автора, хотя я подозреваю, что он не написал в них ни одного слова.
11. По материалам личного общения с Ральфом Гомори (Ralph Gomory), ноябрь 2008 года.
12. См. [76].
13. Подборки писем от 4 февраля 1964 года главе отделения маркетинга, а также моему непосредственному начальнику и руководителю лаборатории приведены на веб-сайте к данной книге.
14. См. [158].
15. `http://americanhistory.si.edu/collections/comphist/cray.htm`; за основу взята версия этой страницы от 12 августа 2009 года.
16. Наилучшая рекомендация по планированию академической карьеры, которая мне когда-либо встречалась, приведена в книге [115], с. 21.

    Когда молодой германский ученый начинает свою карьеру, он обычно выбирает три или четыре крупных области, к которым чувствует реальный интерес,

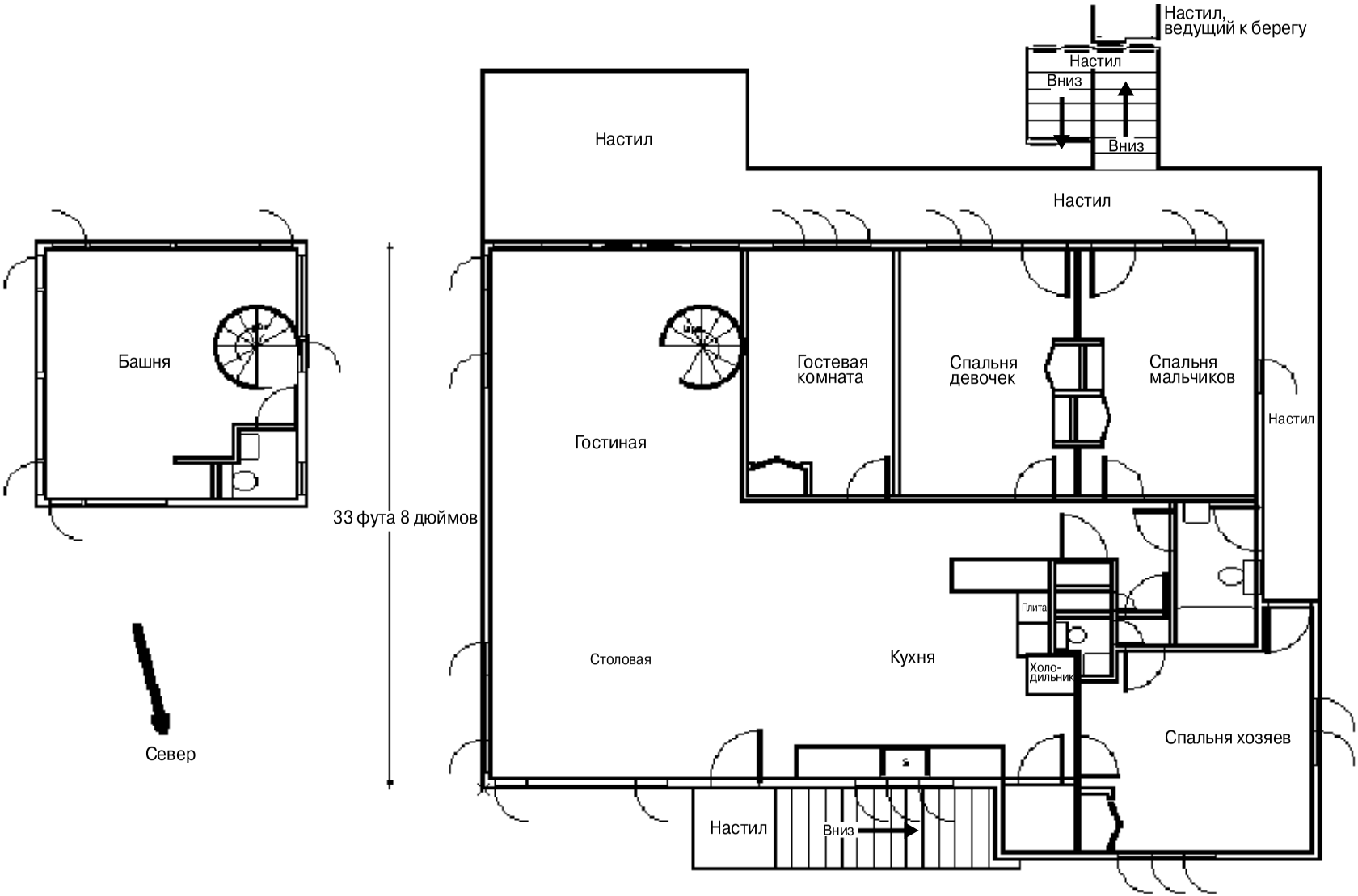
в которых предстоит сделать еще много работы, которые (что очень важно) тесно связаны друг с другом и которые (что важнее всего) он рассматривает как находящиеся в самом сосредоточении изучаемой им темы. Он стремится в максимально возможной степени сделать эти области предметом своих первых лекций и семинаров. Он подготавливает по этим направлениям серии лекций, затем дополняет и развивает каждую серию, пока они не вырастают в целую книгу. Если перспективный ученый достаточно энергичен и восприимчив, то вскоре становится автором трех или четырех книг, каждая из которых рекомендует и поясняет другие. Затем он продолжает ...стратегически увеличивать [каждую область] из года в год, пока не накопит действительно признанный всеми объем знаний по всей данной проблематике. ...Ученые, которые планируют свое обучение и ведение преподавательской деятельности таким образом, обычно находят... что они настолько расширили круг своих интересов и накопили такой объем знаний, которых хватило бы не на одну, а на три карьеры.

17. См. [186].

18. См. [25], главы 9–16. Этот стандартный формат описан в главе 9.

TheDesignofDesign.indb 309 30.10.2012 15:58:37

TheDesignofDesign.indb 310



30.10.2012 15:58:37

21

Изучение практического примера. Дом на побережье "View/360"

Самый красивый дом в мире
(тот, который вы построили сами).

ВИТОЛЬД РЫБЧИНСКИ, 1989 Г.

◂ *План первого этажа и башни дома на побережье "View/360"*

Основные факты и особенности

Назначение этого практического примера. Он показывает, что даже самый простой и понятный проект требует принятия очень большого количества решений и учета многочисленных соображений.

Решение в чистом виде. Разместить дом как можно ближе к океанской береговой линии, но так, чтобы он не выходил за пределы земельного участка, на который распространяется страховка. Оказалось, что при этом дом располагается ближе к океану примерно на 40 футов по сравнению с домами всех соседей, а это влечет за собой повышение риска того, что дом может быть смыт волнами.

Лимитируемый ресурс. Как оказалось, для этого проекта дома лимитируемым ресурсом является ширина океанского побережья, соответствующего земельному участку, от чего зависит обзор и воздействие ветра.

Прозорливое решение по установке винтовой лестницы. Эта деревянная конструкция, задуманная в целях экономии площади пола, оказалась местом, превосходно подходящим для размещения эстетически привлекательных пространственных форм и притягательным для тех, кто восхищается окружающей природой.

Изменения в ходе строительства. Изменения в проекте, внесенные в ходе его реализации, способствовали существенному улучшению внешней привлекательности дома, созданию его незабываемого вида и повышению удобства. Не были использованы все возможности, обнаруженные в ходе внесения изменений в проект, и оказалось, что это — ошибка.

Размещение свай. Я выполнял роль архитектора-любителя и привлек профессионала, но нам обоим не удалось тщательно продумать и найти правильное размещение свай под центрами сосредоточения веса дома, поэтому сваи были размещены исходя из принципа их равномерной нагрузки. Сваи ушли в песок на разную глубину, и дом подвергся усадке в нескольких местах из-за их необоснованного размещения.

Вводные сведения и условия выбора местоположения

Местоположение

Штат Северная Каролина, Касвелл-Бич, Касвелл-Бич Роуд, 321; координаты: 33°53.6' северной широты, 78°2.1' западной долготы. Строительная площадка находится на острове, протянувшемся с востока на запад, с одной центральной дорогой. Один ряд земельных участков пролегает между Атлантическим океаном и дорогой, а второй располагается между океаном и рекой Кейп-Фир с ее болотами. Дом находится на побережье океана и повернут на 15° к западу от южного направления.

Владельцы

Семья Фредерика и Нэнси Брукс.

Проектировщики

Фредерик и Нэнси Брукс — архитектура; Артур Когсвелл, член Американского института архитекторов, — строительное проектирование и разработка профиля крыши для башни.

Даты

1972 г. дом подведен под крышу и заселен 1997 г., строительство закончено.

Члены семьи, проживающие в этом месте, по состоянию на август 1972 года

Родители: Фредерик и Нэнси.
Дети: Кеннет — 14 лет, Роджер — 10 лет, Барбара — 7 лет.
Бабушка: Октавия, 71 год.
Близкий друг наших детей: Чандлер — 10 лет.

Цели

Основная цель. Наша основная цель заключалась в том, чтобы построить удобный, непохожий на другие дом для отдыха членов семьи и друзей, которые могли бы воспользоваться естественными преимуществами пребывания на океанском побережье. Дом не был предназначен для сдачи в аренду.

Другие цели

- Максимальное использование возможностей любоваться природой.
- Создание подходящего к данному месту, непретенциозного, успокаивающего интерьера.
- Предоставление всем присутствующим возможности дышать свежим морским воздухом и днем и ночью.
- Обеспечение 14 спальных мест и возможность одновременного питания 22 человек.
- Включение в проект комнаты для бабушки/для гостей, спальни для хозяев, спальни для мальчиков и спальни для девочек — всего четырех спален.
- Включение в проект достаточного количества душевых кабин и туалетов.
- Постройка дома, способного противостоять ураганам со скоростью ветра, по меньшей мере, 100 миль в час. Ураганы достигают угрожающей силы

примерно два раза в год и становятся разрушительными приблизительно один раз в десять лет.

- Предусмотреть в проекте кухню, предназначенную для работы одного человека, но так, чтобы можно было привлечь к работе на ней от 4 до 6 человек.
- Предусмотреть в проекте, чтобы шумные игры наших мальчиков и их друзей никому не мешали.
- Обеспечить максимальное снижение затрат на обслуживание.
- Организовать строительство дома по принципу "делаем все вместе", чтобы дать возможность каждому члену семьи внести свой посильный вклад и укрепить семейные узы.

Возможности

Строительный участок. Участок занимает 75 футов прибрежной океанской полосы. На юго-восток открывается прекрасный вид острова Болд-Хэд (Кейп-Фир), и можно наблюдать за движением судов, проходящих в Кейп-Фир и обратно. Как фронтон определена сторона здания, обращенная к океану, а не сторона, которая смотрит на дорогу. Почва — крупный прибрежный песок; растительность — низкий кустарник, морской овес и вьющийся смилакс.

Дюны. Дом вполне может быть расположен на расстоянии вплоть до 65 футов от верхней отметки воды, поскольку он защищен рядом дюн.

Виды. Поскольку остров — узкий, имеется не только обзор на 180° берега с передней стороны дома, но обзор на 135° реки Кейп-Фир и ее болот с задней стороны.

Ветер. Площадка, на которой располагается дом, имеет естественную ориентацию на 15° к западу от южного направления. Преобладающее направление ветра — от южного до юго-западного, а дует он в основном в теплую погоду.

Ограничения

Бюджет. В свое время у нас не было достаточного количества денег, чтобы сразу построить дом с четырьмя спальнями.

Время. Наша семья могла заниматься строительством из года в год только в летнее время.

Законодательные и юридические требования

- Дом должен был располагаться на 16-футовых сваях, причем часть каждой сваи длиной не меньше 8 футов должна была находиться в земле.
- Требование по отступу здания от соседних участков с каждой стороны составляло 10 футов.

- Дом должен был быть рассчитан на одну семью.
- Должны были соблюдаться законодательные требования по подводу электроэнергии и размещению резервуара для стоков.

Дюны, отделяющие дом от океана. Повреждения, причиненные дюнам, должны были быть сведены к минимуму, например, путем установки над ними для прохода дощатого настила.

Виды снабжения. Площадка снабжалась только электроэнергией и водой, а газ и канализация не были проведены.

Юридические требования. Приблизительно 65 футов со стороны участка, прилегающей к океану, было намыто волнами с тех пор, как участок был внесен в кадастр в 1938 году. Мы имеем акт отказа от права, а не акт с гарантией права собственности на эту землю.

Требования к экстерьеру. Юридические требования к внешнему виду дома не предъявлялись или были такими, которыми можно пренебречь.

Проектные решения

При строительстве дома было решено не задавать жесткие сроки.

- В первую очередь предусматривалось: постройка закрытого помещения, которое могло быть просушено и заселено, установка ванной с водопроводом и канализацией, монтаж резервуара для стоков, а также подведение к дому временного электроснабжения.
- Все первоначально доступные наличные деньги должны быть направлены на максимально возможное увеличение полезной площади дома и на установку окон.
- Все внутренние работы, включая отделку стен, установку дверей, устройства шкафов, прокладку электропроводки и, по большей части, прокладку водопровода и канализации, должны быть проведены членами семьи.

Должна использоваться вся длина участка, составляющая 75 футов. В этом месте длина большинства прибрежных участков составляет 50 футов. А дома на побережье океана, по большей части, являются узкими и длинными. Чтобы в максимальной степени воспользоваться допустимой шириной дома, составляющей 55 футов, мы развернули свой дом в сторону океана, поэтому он имеет большую ширину, чем глубину. Таким образом, нам пришлось разрабатывать собственный поэтажный план, а не брать стандартный образец.

Воспользоваться замечательными видами.

- Установить дом настолько близко к океану, насколько это возможно, но оставаться в пределах, определяемых актом с гарантией права собственности на землю.

- Перед помещением третьего этажа должна быть построена башня, застекленная со всех четырех сторон, которая позволит в полной мере насладиться открывающимися видами.
- **Результаты принятых решений.** Крыша не должна быть слишком крутой. Конек крыши не должна быть выше уровня подоконников в башне.
- Должны быть предусмотрены многочисленные и большие окна на всех этажах.
- **Результаты принятых решений.** Каркас здания необходимо укрепить с учетом возможного перекоса.

Необходимо предоставить возможность всем жильцам дома дышать свежим морским воздухом.

- В каждой спальне должен быть предусмотрен обзор прибрежной океанской полосы, пусть даже частичный.
- Должна быть обеспечена возможность открыть весь дом для свежего ветра; применение кондиционирования воздуха не предполагается.
- **Результаты принятых решений.** Необходимо подготовиться к тому, что в любом месте могут появляться влажность и следы солевых отложений.
- Должна быть предусмотрена пара раздвижных дверей шириной 6 футов перед гостиной.
- Перед домом должен быть установлен широкий навес для защиты от солнца.
- Необходимо предусмотреть использование створчатых окон, которые позволяют максимально увеличить площадь открытого проема и могут быть повернуты с учетом направления ветра.
- **Результаты принятых решений.** Окна должны устанавливаться так, чтобы их открытые проемы были ориентированы на юго-запад или северо-восток.

Все строительные конструкции должны соответствовать требованиям защиты от проникновения влаги. Дом должен быть устойчивым и по отношению к обычной влажности, вызванной поступлением морского воздуха, и к осадкам, вызванным непогодой. Должна использоваться деревянная обшивка для максимального уменьшения необходимости в отделочных работах. Необходимо отказаться от использования ковров; разрешена раскладка лишь отдельных циновок, причем чаще всего небольших.

Дом должен оптимальным образом соответствовать условиям проживания весной, летом и осенью. Необходимо предусмотреть отопление на тот случай, если дом придется посетить зимой. Вместо центрального отопления необходимо установить электрическое плинтусное отопление; это влечет за собой

значительное увеличение эксплуатационных расходов за единицу времени, но капитальные затраты существенно уменьшаются.

Локализация и максимальное снижение шума.

- Необходимо предусмотреть для каждой спальни, включая спальню для девочек, спальню для мальчиков и спальню для хозяев, отдельную наружную дверь, чтобы "ранние пташки" могли отправляться на берег, не тревожа спящих.
- Дом должен быть разделен на зону спален и зону общего пребывания; спальни, ванные комнаты и связывающий их холл необходимо отделить от того пространства, где все могут находиться без ограничений.

Шумные игры мальчиков не должны никому мешать. Спальня для мальчиков должна находиться в дальнем конце зоны спален.

Спроектировать уместный, непретенциозный, успокоительный интерьер. Для всех стен должна использоваться деревянная обшивка, не окрашенная и не покрытая обоями. Необходимо использовать темную деревянную обшивку и темное покрытие пола для гостиной и башни, где отблески от солнца бывают наиболее яркими; то же требование относится к таким общедоступным помещениям, как кухня и столовая, а также к спальне для мальчиков. Во всех случаях следует стремиться к тому, чтобы создавалось ощущение затемненного, спокойного, прохладного помещения. В других спальнях для поднятия настроения необходимо использовать светлую деревянную обшивку. Для холлов, куда не попадают солнечные лучи, должна быть предусмотрена белая обшивка.

Количество спальных мест должно достигать 14. Четыре человека могут ночевать в спальне для мальчиков, четверо — в спальне для девочек, двое — в спальне для хозяев, двое — в спальне для гостей, двое — в гостиной и двое — в башне. В гостиной должны быть установлены два дивана, которые могут использоваться как спальные места. Для башни необходимо предусмотреть две кровати, которые можно разворачивать, сворачивать и убирать. Должны быть предусмотрены по две откидные койки и две стационарные односпальные кровати и в спальне для девочек, и в спальне для мальчиков.

Количество мест за столами должно достигать 22. В центральной части дома необходимо установить три стола: два стола на 8 персон каждый и один на 6 персон.

Потолки не должны иметь покрытия. Для экономии и создания приятного визуального эффекта потолки не должны иметь покрытия, кроме как в ванных комнатах и в гостиной. Стропила состоят из сдвоенных балок 2×12 на 4-футовых опорах и рассчитаны на снеговую нагрузку толщиной до 1 фута. Крыша состоит из соединенных в шип балок 2×6 с изолирующей прокладкой в верхней части, а на нее положен настил из рубероида и белое кровельное теплоотражающее покрытие. В интерьерах в качестве отделки применяется в основном только покрытие лаком.

Результаты принятых решений. Почти во всех помещениях роль потолка выполняют балки перекрытия, поэтому установка скрытой электропроводки затрудняется.

Сокращение площади пола, занимаемой лестницей в башню. Башня расположена перед домом, поэтому там же должна находиться лестница, ведущая в нее, но при этом приходится соблюдать требование по максимальному сокращению общей площади, занимаемой домом. Гостиная — единственная общедоступная комната, которая расположена со стороны фронтона дома, поэтому лестница в башню должна начинаться с нее.

Если учесть требования, что окна гостиной должны полностью открывать вид на юг, а сама гостиная должна соединяться с другими общедоступными помещениями, находящимися в северной части здания, то лестница должна примыкать к восточной или западной стене. На восточной стене предусмотрены окна для обзора, поступления солнечных лучей и проникновения ветра, поэтому лестницу решено было установить со стороны западной стены.

Мы поняли, что драгоценное пространство будет лучше всего использоваться, если лестница в проекции будет занимать площадь в виде квадрата, а не прямоугольника, поскольку в последнем случае придется сузить всю гостиную. Таким образом, было решено построить спиральную лестницу. В предвидении того, что сталь будет подвергаться коррозии под воздействием отложений соли, было решено сделать деревянную лестницу. И тогда лестница, которая создавалась как необходимый конструктивный элемент, превратилась в своеобразный предмет искусства.

Проектирование карниза с учетом проникновения дневного света. Карниз должен иметь такую ширину, при которой в полдень в комнаты, расположенные со стороны фасада, с сентября по март попадает солнечный свет, а с марта по сентябрь обеспечивается защита от солнечных лучей.

Расчет оптимальной ширины фасада

С учетом возможностей и ограничений мы пришли к выводу, что ширина дома может достигать 55 футов, или 660 дюймов. Поскольку было решено использовать в полной мере возможность наслаждаться свежим ветром и любоваться видами океана, критическим ресурсом проектирования становится выбор ширины фасада.

Гостиная. Наибольшие требования по достижению этих целей (открытость для свежего ветра и видов на океан) мы предъявляли к гостиной. Поэтому было очевидно, что гостиная должна занять значительную часть фасада. Произвольно принятое решение состояло в том, что это помещение должно иметь по ширине 16 футов. Предполагалось, что гостиная может иметь не только наружную дверь и окна, выходящие на фронтон дома, но и окна с других сторон дома, поэтому мы решили разместить гостиную в юго-восточной части дома, чтобы из нее открывался вид на юго-восток и устье реки Кейп-Фир, а также можно было следить за прохождением судов.

Западная веранда. Должна быть построена узкая веранда вдоль западной стороны дома, которая бы обеспечивала непосредственный выход на берег, не препятствовала проникновению свежего ветра в спальню хозяев и позволяла прямо с берега попасть внутрь дома, к душевым кабинам, чтобы люди не проходили с грязными ногами через весь дом.

Отличия в устройстве спален и столовой/кухни. В связи с тем, что принято решение не устанавливать систему кондиционирования, очень важно было обеспечить приток свежего воздуха в спальни. Требование по обеспечению прямого выхода на берег из спален остается вторичным, и такая возможность в большей степени становится приятным дополнением, особенно если учесть характеры наших детей. Поэтому все спальни передвинуты к фронтону и занимают всю оставшуюся часть фасада.

Спальня для девочек и спальня для мальчиков. Эти комнаты имеют самый высокий приоритет среди спален, поскольку они должны использоваться в каждом отпуске, проводимом семьей на берегу (в отличие от комнаты для гостей), а также могут служить как убежище для чтения и подготовки детей к экзаменам.

Комната для гостей. Спальня хозяев предназначена прежде всего для ночлега. Комната для гостей вполне может также использоваться как убежище для чтения, поэтому для нее зарезервировано место во фронтальной части дома.

Определение конфигурации спальни для мальчиков. Спальню для мальчиков было решено разместить в юго-западном углу дома, чтобы она находилась как можно дальше от других комнат. Одна кровать располагается под фронтальными окнами, а одна откидная койка устанавливается на западной стене, примыкая к ней в убранном состоянии. Вторую койку можно разместить на большей высоте (чтобы мальчикам было интересно ею пользоваться), если будет принято решение об ее установке на северной стене около конькового бруса. Поэтому шкаф устанавливается на восточной стене. Минимальная ширина этой комнаты должна определяться как сумма ширины дверного проема и длины кровати.

Определение конфигурации спальни для девочек. Высота кроватей должна быть ниже, чтобы можно было размещать над ними откидные койки, не создавая неудобства тем, кто занимает кровать. Одна кровать должна находиться под окном, а койка — на восточной стене. Вторую койку также можно установить на восточной стене и предусмотреть для нее лестницу. Вторую кровать можно установить у северной стены, а шкаф разместить на западной стене. Минимальная ширина этой комнаты должна определяться как сумма ширины дверного проема и длины кровати.

Определение конфигурации комнаты для гостей. Для этой комнаты достаточно предусмотреть лишь две односпальные кровати. Дверь с выходом на берег можно не предусматривать. Минимальная ширина комнаты определяется как сумма ширины кровати и прохода вокруг кровати. В глубину комната может быть сделана достаточно большой, поэтому шкаф можно разместить у северной стены.

Определение размеров дома в плане

Площадь в квадратных футах. Из расчета тех денежных средств, которые были в наличии, мы могли определить максимальные пределы площади, занимаемой домом, равными 2000 квадратных футов, учитывая то, что в доме должно быть много окон.

Конструкция крыши и ширина дома. Прогиб равномерно нагруженной однопролетной деревянной балки существенно зависит от длины пролета между опорами:

```
d = k l4 / w2 t
```

где `w` — ширина деревянной балки; `t` — толщина; `l` — эффективная длина. При расчете перекрытия эффективная длина балки определяется как расстояние между точками опоры, а общая длина балки равна сумме значения эффективной длины и значений длины двух консольных участков балки, которые определяют ширину карниза. Я выбрал карнизы шириной 4 фута с учетом того, на какой угол солнце поднимается летом и зимой. Общий расчет показал, что могут применяться сдвоенные балки 2×12, перекрывающие по горизонтали максимальную длину 16 футов и расположенные с шагом 4 фута. Это стало определяющим при вычислении значений ширины спален, расположенных вдоль фронтона дома, гостиной и башни.

Определение размеров столовой и кухни. Сложилось мнение, что в спальне хозяев необходимо предусмотреть одну дверь, выходящую прямо на кухню, и еще одну дверь, ведущую в холл. Важной составляющей при определении размеров дома стало вычисление длины западной стены кухни, которая должна была быть по меньшей мере равной сумме ширины одного рабочего стола, ширины плиты, ширины холодильника и ширины дверного проема спальни, с учетом возможности размещения еще одного рабочего стола между плитой и холодильником.

Для упрощения расчетов я решил принять ширину кухни равной максимально допустимой ширине помещения в тыльной части дома. В результате кухня приобрела большие размеры, очень удобные для работы. Но это также потребовало установки планок составной балки дополнительно к кровельным балкам для 17-футового промежутка.

Неудачные попытки

Крыша в форме волнолома. Первоначально я запланировал возвести над домом такую крышу, которая напоминала бы по форме волнолом (рис. 21.1).

Но Когсвелл настоятельно возражал против этого решения: “Еще во времена моей учебы наш преподаватель по архитектуре говорил, что можно считать хорошим только такой проект дома, который позволяет защититься от дождя. Возможно, вы сумеете найти в графстве Брунсвик подрядчика, который добьется

того, чтобы в доме с подобной крышей не было протечек от осадков, но я в этом сомневаюсь”. Я последовал его глубоко обоснованному совету.

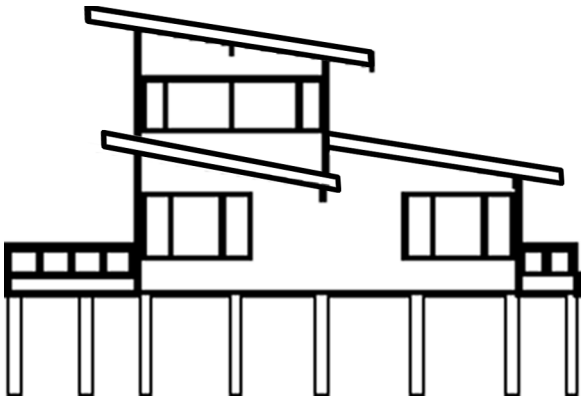


Рис. 21.1. Профиль крыш, предложенный Бруксом

Невыразительный симметричный профиль крыши башни, предложенный Бруксом. Я предложил для башни профиль крыши, который оказался примером невыразительности. Когсвелл существенно улучшил этот профиль, как показано на рис. 21.2.

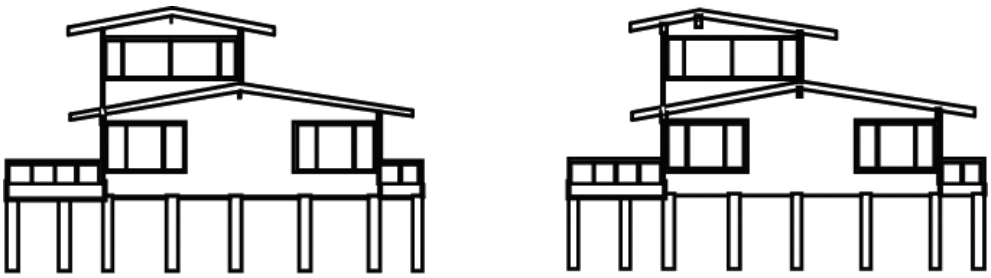


Рис. 21.2. Вид с востока на профили крыш, предложенные Бруксом и Когсвеллом

Изменения в проекте после проектных работ до начала строительства

В проекте были предусмотрены две выносные душевые кабины на первом этаже. Чтобы люди после купания не заносили песок, следы соленой воды и мокрые купальные костюмы в дом, мы решили спроектировать две выносные душевые кабины на первом этаже; в этих кабинах предусматривалась также возможность переодеться перед купанием и после него. Каждую раздевалку мы решили сделать достаточно большой по площади для размещения нескольких людей, чтобы родители могли помочь переодеться детям.

Установка в холле шкафов вместо первоначально задуманной душевой. Первоначально отведенное пространство для душевой заняли два шкафа — большой бельевой шкаф и шкаф общего назначения на всю высоту комнаты с шириной 12 футов, в котором на нижних полках должны храниться матрацы, а на верхних — гардины и занавеси.

Отказ от отдельного помещения для приема пищи и создание общей области со столовой и кухней. Изучение вопроса о том, где принимать пищу, показало, что ради повышения удобства обслуживания следует совместить столовую и кухню. В результате столовая превратилась в то место, где можно посидеть, поработать, поиграть в игры, заняться решением забавных задач.

Площадь кладовки на первом этаже была увеличена вдвое: с 8×16 до 16×16 футов.

Петли на двери с проволочной сеткой в спальню хозяев были переставлены на другую сторону. Если окно в ванной было открыто, то не открывалась дверь с проволочной сеткой — во время строительства был обнаружен конструктивный недостаток.

Изменения в проекте после возведения каркаса дома и первоначального заселения

В ходе строительства дома было решено отказаться от разделения зоны спален и зоны общедоступных помещений.

Установлены дополнительные диагональные растяжки. Растяжки на восточной и западной стенах гостиной, на северной стене комнаты для гостей и на западной стене спальни для мальчиков позволили усилить сопротивляемость каркаса дома против ветра. Эта работа была сделана после возведения каркаса и до выполнения каких-либо отделочных работ.

Установлены уголки для повышения устойчивости здания при ураганных порывах ветра. Подрядчик, ответственный за строительство каркаса здания, не установил указанные в проекте вертикальные анкерные тяги, поэтому мы заменили их уголками, чтобы удерживать крышу на месте при сильном ветре.

На верандах установлены водопроводные краны, чтобы было удобнее заниматься влажной уборкой.

Предусмотрен тент для широкой передней веранды. Тент позволяет создать тень на всей веранде, но по желанию может быть растянут только частично. Мы использовали прицепной тент, рассчитанный на максимальную скорость ветра 70 миль в час. Гораздо позже мы заменили тент стационарным навесом, перекрывающим по длине только половину веранды, чтобы дать возможность людям выбирать солнце или тень.

Замена и увеличение окон в башне. Ураган "Диана" (в 1984 году) выбил все окна, первоначально установленные в башне, включая стекла и рамы. Мы заменили

на передней стене несколько маленьких рам двумя одинарными большими оконными рамами. Эти рамы выглядят гораздо лучше по сравнению с прежними и способны выдерживать более сильные порывы ветра.

Установка дополнительной двери в западном холле. Эта дверь позволяет применять сидячую ванну, установленную в северной части дома, по одному из двух разных назначений: либо предоставить к ней доступ только из спальни хозяев как к отдельному удобству, либо включить в состав общедоступных туалетных помещений.

Подготовка съемных фанерных щитов. Для защиты окон на двух наветренных сторонах дома от зимних холодов и ураганов мы подготовили съемные фанерные щиты.

Удален дополнительный звукоизолирующий материал (и уменьшена толщина стены) между гостиной и комнатой для гостей после возведения каркаса и перед отделочными работами. Бабушка Брукс умерла в 1973 году, поэтому ее комната стала комнатой для гостей, в связи с чем специальная звукоизоляция стала ненужной.

Ограждение веранды с фронтальной стороны дома в восточной ее части заменено скамьей, что особенно нравится тем, кто любит дышать свежим воздухом и любоваться зарей.

Установлен горизонтальный брус (в 1997 году), чтобы стабилизировать усадку свай.

Установлены подпорки под западной стеной спальни для мальчиков (в 2000 году). В первоначальном плане сваи были предусмотрены под краем западной веранды, а не под западной несущей стеной, что оказалось ошибкой в строительном проекте.

Оценка достигнутых результатов (по прошествии 37 лет)

Положительные стороны

Башня. Как оказалось, башня представляет собой замечательную студию с видами до самого горизонта, на которых могут отдыхать глаза, с возможностью следить за тем, что происходит на берегу, с великолепным обзором в юго-восточном направлении, позволяющим наблюдать за движением судов в океанском канале, с превосходным видом на северо-восточное направление для наблюдения за судами, пересекающими реку, и с большим количеством света. Бо́льшая часть книги *Мифический человеко-месяц* была написана там.

Открытый план. Отказ от установки первоначально запланированной перегородки между кухней и холлом привел к существенному улучшению общего впечатления от внутренней части дома. В результате увеличилось видимое

пространство, поэтому после входа в дом стало создаваться ощущение объемности. В основной коридор стало поступать больше дневного света и свежего морского воздуха. Во время работы на кухне можно видеть океан через спальню для девочек и ее открытую наружную дверь, что способствует существенному повышению настроения. Основными декоративными элементами стали цветные двери спален, а белая стена холла оказалась идеальным местом для развешивания карт.

Винтовая лестница. Винтовая лестница была сделана из дуба настолько удачно, что вызывает восхищение почти как художественное произведение. При дневном свете она отражается в остеклении окон передней стены.

Обоснование проектирования. Консультирующий меня архитектор, Артур Когсвелл, однажды в шутку назвал проект как "View/360 — чертовски обоснованный логически дом на побережье", поскольку в его основу легло подробное обоснование, описанное в этой книге и согласованное с ним. У меня никогда не возникала мысль, что Артуру это не нравилось. Лично я неизменно испытываю удовольствие, находясь в этом доме, поскольку все в нем глубоко продумано.

Вид снаружи. Внешне дом интересен, но его нельзя назвать красивым. Дело в том, что он имеет форму, соответствующую назначению (рис. 21.3). Асимметричный профиль крыши башни, предложенный Когсвеллом, оказался абсолютно правильным; создается впечатление движения вперед.

Рис. 21.3. Вид дома "View/360" с юго-запада

Полезность

Задание выполнено. Дом очень удобен для проживания.

Изменения в проекте. Как оказалось, изменения, внесенные в первоначальный проект, привели к значительным улучшениям.

Отказ от некоторых внутренних стен. Отказ от установки первоначально запланированной перегородки между холлом и кухней позволил существенно увеличить объем свободного пространства для работы на кухне за счет присоединения пространства холла. Блюда подаются на столы прямо из этого рабочего пространства, как в буфете, благодаря чему создается дополнительное удобство, не предусмотренное в проекте. Приготовлением пищи могут заниматься сразу несколько человек.

Столовая. В первоначальный проект столовой было внесено изменение, позволяющее создать отдельную площадку, где можно посидеть, поговорить, поработать и поиграть в разные игры, поэтому столовая в большей степени приобрела вид жилого помещения. По существу, после реализации всех дополнений к проекту кухни/столовой это помещение превратилось во вторую гостиную. А иногда мы используем его как вторую студию, хотя здесь нельзя уединиться.

Спальня хозяев. Спальня хозяев неожиданно продемонстрировала возможность использовать ее как третью студию, расположенную отдельно и имеющую вид на болотистый берег реки.

Возможность приема больших групп. Дом не был специально спроектирован для приема больших групп отдыхающих, но оказался способным разместить на выходные до 25 человек, но не больше. Этот показатель был установлен нами по методу проб и ошибок. Количество мест для ночлега (в том числе на раскладных кроватях и на полу), за столами, в душевых и в гостиных ограничено, но остается достаточным при количестве людей не больше 25.

Размер семейства. Теперь наш дом хорошо подходит для сбора одновременно всех представителей семейства (хотя это происходит редко). Сюда входят трое наших детей, два племянника, девять внуков и мы. Очевидно, что состав семьи со времени начала проектирования дома существенно изменился.

Кухня. Кухня вполне может вместить целую бригаду поваров, но если работает один повар, то ему приходится пользоваться раковиной, расположенной довольно далеко от рабочих столов, плиты и холодильника.

Веранда. Главной ошибкой оказалось возведение веранды вдоль восточной стороны дома. На нее было израсходовано 42 дюйма важного лимитируемого ресурса — расположенной со стороны океана площади, пригодной для застройки. Эта веранда используется редко. Занимаемое ею пространство можно было бы гораздо лучше использовать по другому назначению.

Принятое незадолго до окончания проектирования решение поместить главные душевые кабины, обслуживающие отдыхающих на берегу, на первом этаже позволило исключить необходимость предоставлять проход от берега до главной

ванной комнаты. Обслуживание тех, кто хочет принять душ, стало еще проще благодаря тому, что вместо бельевого шкафа было решено установить внутри дома еще одну душевую кабину. Я должен был пересмотреть решение о возведении веранды после того, как было решено установить эту душевую кабину.

Спальня хозяев. Наружная дверь в спальню хозяев чаще всего бывает открыта ради поступления свежего воздуха, но не для прохода. Те, кто находится в спальне хозяев, могут беспрепятственно пройти на берег через кухню и заднюю дверь. Спальня хозяев обычно хорошо проветривается через западные окна и две внутренние двери. Вид на океан из этой спальни не слишком впечатляет, но это не столь важно, поскольку данная комната предназначена для ночлега, а не для постоянного пребывания. Таким образом, эту дверь можно было бы убрать, если бы была предусмотрена веранда.

Главная ванная. Наружная дверь в главную ванную комнату по большей части остается открытой, благодаря чему проветривается весь дом. Если бы здесь не было веранды, то пришлось бы предусмотреть голландскую дверь с нижней половиной, закрывающейся на щеколду.

Конфиденциальность. Гостиная, столовая и кухня представляют собой открытые помещения, и в этом состоит их недостаток, так как в них нельзя делиться секретами и найти убежище от суеты, которой наполнен дом. Башня изолирована от посторонних взглядов и ненужных посещений, но не обеспечивает защиту от шума.

Прочность

Дом устоял перед воздействием трех ураганов и различных штормов меньшей силы.

- **Шторм 1978 года.** С башни сорвало крышу (с рубероидной кровлей). Под воздействием ветра и эффекта Бернулли крыша поднялась в воздух и приземлилась на заднем дворе.
- **Ураган "Диана", который прошел в 1984 году.** Эпицентр урагана прошел от нас на расстоянии 10 миль, и при южном направлении ветра его максимальная скорость составляла 135 миль в час. Ураган поднял всю секционную крышу дома, за исключением крыши башни, и отпустил ее как единое целое на заднем дворе. Дом был залит дождем на высоту шестнадцать дюймов. Шторм выбил все стекла и оконные рамы в башне и унес все матрацы, коврики и лампы в болото.
- **Ураганы "Берта" и "Фрэн" 1996 года.** Центр урагана прошел на расстоянии 10 миль от дома. Повреждений не было, если не считать одного оконного щита и одного оконного стекла.

Из-за существенно различающихся нагрузок на сваи скорость усадки различных свай на песчаной почве значительно изменялась. В проекте не был учтен такой важный фактор. Данный недосмотр нельзя назвать иначе чем странным, поскольку от подобной проблемы страдает любой дом на сваях, который возведен на песчаной почве. Сваи под двумя южными углами башни несли более высокую нагрузку, поэтому дали более существенную усадку. Сваи под стенами, которые не столь нагружены, подверглись меньшей усадке. Указанный фактор в наибольшей степени проявился под центральной частью большой двойной двери в гостиную. Дверная коробка прогнулась вперед всередине, поэтому раздвижные двери перестали закрываться должным образом. Мы устанавливали в 1997 году под полом перемычку размерами 4×6 дюймов×16 футов, прикрепленную болтами ко всем трем сваям под южной стеной башни, поэтому в дальнейшем усадка уменьшилась и стала более равномерной.

Самый западный ряд свай был размещен под кромкой веранды, а не под западной стеной, на которую опирается коньковый брус и ложится вес крыши. Балки пола изогнулись под воздействием этого неподдерживаемого веса. На 28-м году со времени строительства дома пришлось дополнительно установить ряд свай под этой стеной. Очевидно, это был просто общий недосмотр. Ни Когсвелл, ни я не провели тщательное сравнение главного поэтажного плана и плана расположения свай.

Ошибочным было решение установить створчатые окна. Эти окна обеспечивали поступление свежего воздуха, как и было задумано. Но оконные рамы были сделаны из дерева, а запирающие механизмы — из стали, поэтому металлические конструкции приходилось заменять через каждые пять лет с той стороны дома, куда достигали брызги океана, и через каждые пятнадцать лет с противоположной стороны. Поэтому по прошествии тридцати пяти лет старые оконные рамы оказались полностью изношенными, и мы заменили большинство створчатых окон раздвижными.

Причина ошибки, допущенной при выборе створчатых окон. Ошибка была вызвана неправильной оценкой потребности в обслуживании на протяжении долгого периода эксплуатации дома, а также недостаточно внимательным выбором всех строительных материалов.

Если бы можно было начать сначала

Предположим, что сегодня я проектировал бы этот дом для этой площадки и для того состава семьи, который она имела в 1972 году. Что было бы сделано иначе с учетом полученного опыта? В приведенном выше разделе “Оценка” перечислены разные мелкие просчеты и ошибки, а теперь перейдем к анализу более крупных.

Первый важный урок, который я извлек бы из опыта, описанного в данной книге (глава 10), состоит в том, что следует еще внимательнее относиться к

распределению лимитируемого ресурса, каковым в данном случае является ширина прибрежной океанской полосы. Теперь, понимая важность этого фактора, я подробнее изучил бы все детали требований к отступу дома от границ соседних участков (включая площадь, покрываемую карнизом) и организовал бы проектирование так, чтобы использовался каждый доступный дюйм, пусть даже при этом пришлось бы выйти за пределы лимитируемой площади.

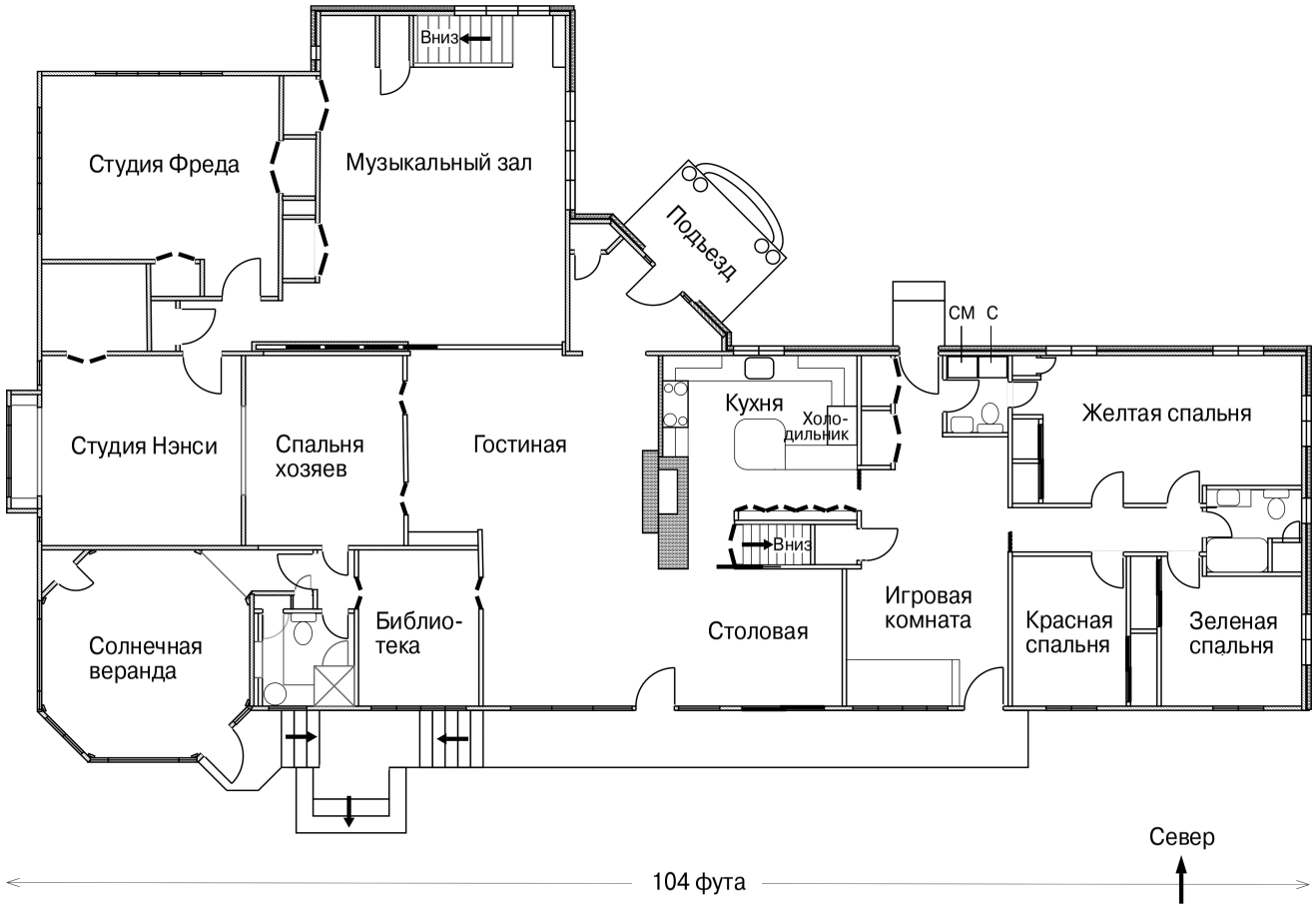
Вторым важным уроком (см. главу 11) стало то, что если бы я сразу предусмотрел душевые кабины вне здания, то в плане не пришлось бы учитывать пожелание и ограничение, связанные с доступом в главную ванную с внешней части дома. Поэтому я отказался бы от западной веранды и перераспределил отведенную ей ширину в 42 дюйма.

Общие усвоенные уроки

Изложенные здесь выводы, так или иначе, относятся к любому хоть в какой-то мере важному проекту, независимо от того, относится ли он к области оборудования, программного обеспечения или строительства.

1. Очень тщательно следуйте в своей работе указаниям профессионального проектировщика, но при этом требуйте обоснований. Даже честные, компетентные и добросовестные проектировщики допускают ошибки.
2. Достаточно часто и тщательно контролируйте ход реализации проекта. Даже честные, компетентные и добросовестные исполнители допускают ошибки.
3. Тщательно обдумывайте все аспекты технического обслуживания. Успешный проект гарантирует продолжительную эксплуатацию готового продукта.

TheDesignofDesign.indb 330



30.10.2012 15:58:38

22

Изучение практического примера. Пристройка крыла к дому

В действительности архитектурное проектирование можно рассматривать почти как образец процесса проектирования в проблемной области, отличающейся большой семантической насыщенностью.

См. [192]

◂ *План дома после пристройки крыла в 1991-1992 гг.*

Основные факты и особенности

Для чего предназначен этот практический пример. Мы разрабатывали этот проект на протяжении свыше 60 месяцев и одновременно вели журнал с описанием проблем проектирования, различных взглядов по поводу их решения и принятых решений, объем которого составил приблизительно 235 страниц. На основании этого журнала мы составили формальную документацию дерева проектирования, как описано в главе 16. Рассматриваемый в данной главе пример проектирования иллюстрирует взаимодействие процесса проектирования и процесса выявления требований (см. главу 3).

Решение в чистом виде. Отложить рассмотрение бюджетных ограничений; провести проектирование с учетом функционального назначения; затем заняться оценкой расходов. Это решение было принято после того, как процесс проектирования уже был начат, а сами строительные работы еще не проводились.

Решение в чистом виде. Переместить спальню хозяев в середину пространства с полным доступом для всех и частично ограниченным доступом в разное время. Это решение было принято на поздних этапах в процессе проектирования, после выявления до сих пор неучтенных требований по результатам анализа использования менее часто посещаемого пространства дома. В то время, когда было принято данное решение, казалось, что оно по существу означает, будто восточное крыло дома становится почти полностью ненужным. Однако впоследствии семья выросла и оказалось, что это крыло, которое мы рассматривали почти как "отель" с редкими постояльцами, весьма востребовано.

Ключевое решение. Выкупить полосу земли шириной 5 футов у соседа, чтобы решить сложную задачу проектирования. О предыстории этого решения см. в главе 3.

Разбивка по этапам. Осуществить полный пересмотр архитектурной концепции дома в два этапа, чтобы упростить задачу проектирования и надзора, как описано в этой и следующей главах.

Отказ от каких-либо ограничений по продолжительности проектирования. Мы решили продолжать проектирование до достижения полностью удовлетворяющего нас решения, поэтому не задавали какие-либо сроки строительства, которые могли бы нас ограничить. Это привело к тому, что проектирование проводилось свыше 60 месяцев (со значительными перерывами), тогда как само строительство заняло только 9 месяцев.

Введение и предыстория

Местоположение

413, Грэнвилл-Роуд, г. Чепел-Хилл, шт. Северная Каролина. Фасад дома обращен строго на север, поэтому в описаниях мы можем ссылаться на стороны света: север, юг, восток и запад.

Владельцы

Фредерик и Нэнси Брукс.

Проектировщики

Фредерик и Нэнси Брукс. Получение время от времени консультаций у Уэсли Макклюра (Wesley McClure), члена Американского института архитекторов, и Алекса Джонса (Alex Jones), члена Американского общества проектировщиков интерьера. Строительные чертежи выполнены другим чертежником.

Строительная компания

Строительная компания Additions Plus; подрядчик Стэнли Статтс (Stanley Stutts); бригадир плотников по строительному проекту Гэри Мэйсон (Gary Mason).

Даты

Проектирование — 1987–1992 гг. Строительство — 1991-1992 гг.

Вспомогательный веб-сайт

В процессе проектирования Фред и Нэнси вели подробный журнал с описанием проектных задач, проблем, идей, консультаций с друзьями и профессионалами и принятых решений, который составил приблизительно 235 страниц. Шариф Раззак (Sharif Razzaque) представил в условном виде наиболее существенные части этого журнала как граф для программы Compendium, показав дерево решений без подробных обоснований каждого решения. С этим деревом можно ознакомиться на веб-сайте книги по адресу `www.cs.unc.edu/brooks/DesignofDesign`.

Предыстория

1964-1965

Сам дом (рис. 22.1), в том виде, в каком он находился первоначально, был построен в 1960 г. и приобретен в 1964 г. Выбор на этот дом пал в значительной степени из-за наличия большого покрытого лесом участка, по границе которого протекает ручей, и удобного местоположения, несмотря на очевидные недостатки дома. Самым серьезным из всех недостатков проекта была плохо продуманная схема перемещений обитателей дома, особенно участок между столовой и кухней, по которому проходили все перемещения из восточного крыла в западное, и наоборот, где люди мешали друг другу. Кроме того, практика показала, что каждая комната в доме имеет недостаточные размеры с точки зрения ее назначения.

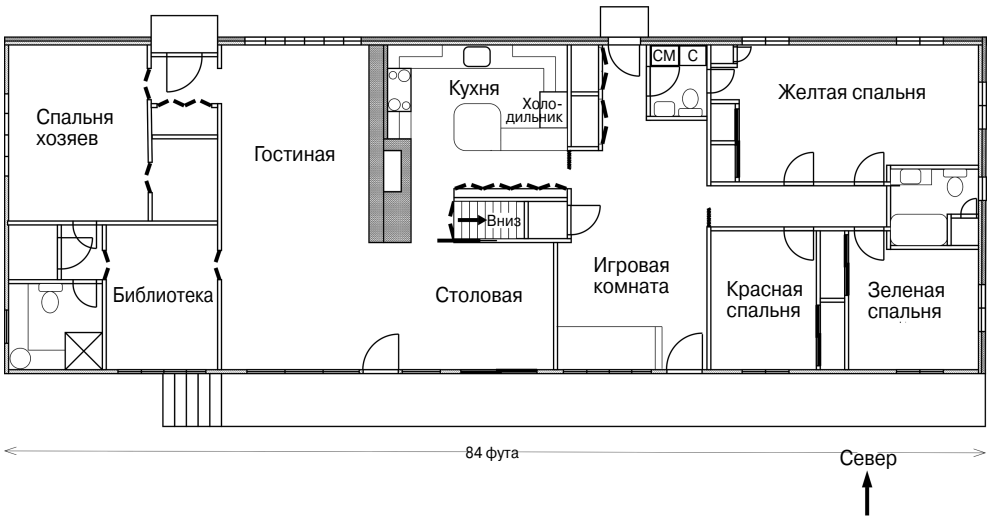


Рис. 22.1. План первого этажа по состоянию на 1987 г.

Ко времени нашего переезда в дом в 1965 г. у нас было два сына в возрасте 7 и 3 лет и 6-месячная дочь. Чтобы быть ближе к детям, мы использовали красную спальню в качестве спальни хозяев, а в трех других спальнях в восточном крыле поселили ребят. Комната в западном крыле, которая первоначально использовалась в качестве спальни хозяев, с 1965 по 1986 гг., стала служить комнатой для гостей.

1972

Мы закончили оформление первого этажа как квартиры и переселили туда нашего старшего сына; средний сын присоединился к нему позже. Это позволило нам снять перегородку и шкафы между двумя спальнями в северо-восточной части дома, создав одну большую спальню, желтую, для нашей дочери. Зеленая спальня стала студией Нэнси.

1987

Наша дочь закончила колледж в 1986 г. и поступила на службу в армию. Наши сыновья в свое время учились в аспирантуре или проходили профессиональную практику; один из них женился, но детей пока не было. После того как наши дети покинули дом, мы получили больше времени для других занятий, поэтому ощутили необходимость в дополнительном расширении определенных пространств. Нэнси давала уроки игры на скрипке у себя дома с середины 1960-х годов; теперь у нее появилась возможность взять больше учеников.

Овдовевший отец Нэнси, доктор Джозеф Гринвуд, только что переехал на постоянное жительство к нам и поселился в бывшей спальне хозяев (в комнате для гостей) в западном крыле. Нэнси унаследовала от своих родителей большой рояль. А поскольку у нас тоже был рояль, мы могли проводить фортепьянные дуэты.

Так или иначе, но по прошествии тридцати лет мы должны были обновить в доме все. Студия Фреда (в которую можно было попасть непосредственно снаружи) находилась на первом этаже, в той четверти дома, которую мы выделили для сыновей. Учитывая наш возраст, мы сочли необходимым обеспечить для себя комфортные условия жизни, располагаясь только на первом этаже, если это станет необходимым.

До сих пор, в прошлые годы, мы пытались определить, как сделать пристройку площадью 500 кв. футов, чтобы сделать дом более просторным, а также, в идеальном случае, переопределить назначение комнат так, чтобы в новом варианте распределения все жильцы дома получили немного более удобные условия проживания. Эти усилия в части проектирования оказались безуспешными.

В связи с этим мы всерьез задумались над возведением пристройки.

Задачи

Исходные задачи

- Улучшить схему передвижения по дому.
- Предусмотреть больше пространства почти в каждой комнате.
- Создать музыкальный зал (Music Room — MR), достаточно большой для того, чтобы в нем поместились два рояля, небольшой орган и площадка для струнного октета, вокруг которой должно быть свободное пространство шириной в 1 фут для удобства преподавателя. В этой комнате должно быть также предусмотрено место для музыкальной литературы. Выделение такой площади должно позволить вынести два рояля из гостиной, где они занимали всю северную часть и затрудняли проход через переднюю дверь. Музыкальный зал должен был также обеспечить проведение выступлений небольшой группы детей, которые посещают уроки музыки, в присутствии их родителей. В идеальном случае для этой комнаты было бы желательно предусмотреть отдельный вход.

- Перенести студию Нэнси (Nancy's Study — NS) из зеленой спальни в юго-восточной части дома, чтобы повысить удобство расположения музыкального зала и студии Фреда. Отвести для нее более значительную площадь.
- Переместить студию Фреда (Fred's Study — FS) на первый этаж.
- Предусмотреть увеличение пространства в соответствии с функциональным назначением: комнаты и (или) алькова.
- Пристроить переднюю веранду (Front Porch — FP), достаточно большую для размещения садовой качающейся скамейки и других мест для сидения.
- Предусмотреть застекленную заднюю веранду (мы ее стали называть просто застекленной верандой, Sun Porch — SP).
- Модернизировать кухню (Kitchen — K) и увеличить ее площадь.
- Увеличить площадь столовой (Dining Room — DR).
- Сделать главный вход более заметным для тех, кто поворачивает к дому со стороны шоссе (рис. 22.2).

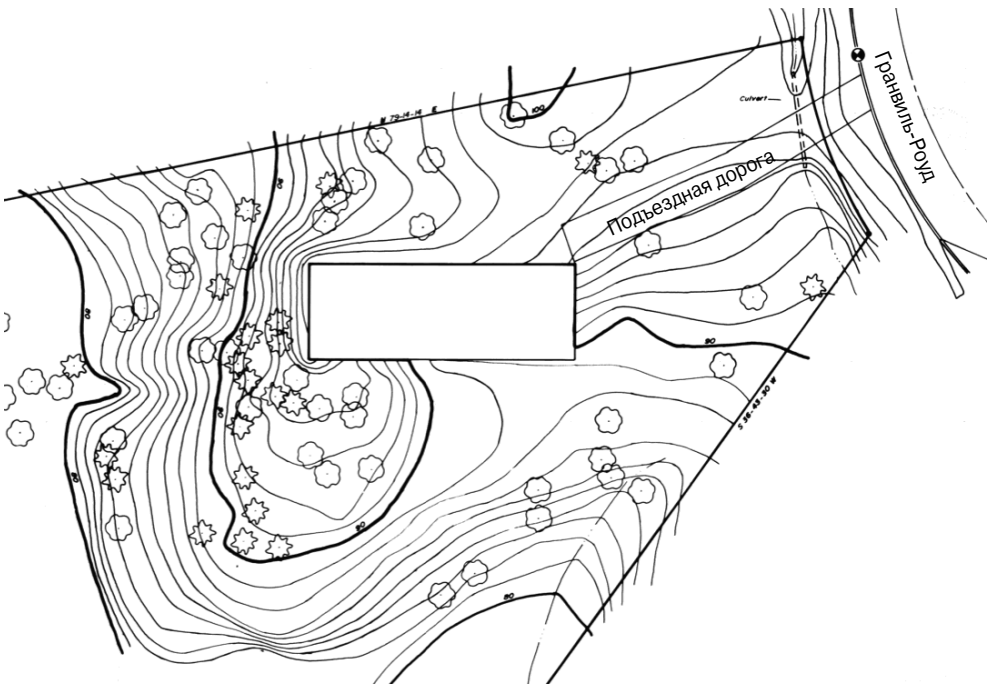


Рис. 22.2. Северная часть участка, на котором расположен дом

- Улучшить эстетику проекта, особенно внешнего вида здания, возможно, за счет применения более интересного профиля крыши.
- Сделать более привлекательным внешний вид двора и всего участка или, по крайней мере, постараться его не испортить.

- Сохранить деревья с юго-западной и юго-восточной стороны и цветники с северной.
- Повысить привлекательность вида на двор и сад из дома, особенно из тех комнат, которые посещают гости.

Дополнительно намеченные задачи

- Лучше приспособить здание к проведению два раза в месяц встреч со студенческой группой, которую мы консультируем (приблизительно 40 человек).
- Предусмотреть размещение в шкафах примерно 40 комплектов верхней одежды для присутствующих на выступлениях детей, которые берут уроки музыки, и на занятиях студентов.
- Обеспечить хранение домашнего имущества, находящегося в настоящее время на арендованном складе.

Ограничения

Существующее здание. Для определения объема работ по модернизации решающими факторами являются план, размещение и ориентация существующего здания.

Участок, на котором расположено здание. Расширение здания ограничивается тем, что оно должно отстоять на 15 футов от северной границы владения, а также тем, что необходимо отступить на 17 футов по остальным границам с учетом того, что тень от здания не должна выходить за пределы участка.

Деревья. Большой черный дуб в верхней части заднего двора был главной достопримечательностью участка, которую мы хотели сохранить.

Земельный участок. Ширина участка довольно резко уменьшается в западном направлении, начиная с торца существующего дома.

Бюджет. Мы планировали уложиться в 100 тыс. долл., а ожидаемая стоимость возведения пристройки составила 100 долл. за кв. фут.

Параметры, не рассматриваемые как ограничения

Бюджет. Намеченная нами сумма в 100 тыс. долл. не была абсолютным ограничением, поскольку мы полностью выплатили ипотеку на покупку.

Стоимость на случай перепродажи. Нам не требовалось рассматривать вопрос о том, приведут ли инвестиции в расширение здания к увеличению стоимости на случай перепродажи. Учитывая предполагаемый срок службы здания и то, что мы не планировали из него переезжать, можно было рассчитывать на то, что приблизительно за тридцать лет произойдет амортизация вложений на обновление здания, прежде чем настанет время его продавать. Но тогда, так или иначе, нужно будет снова делать ремонт.

Площадь земельного участка. Площадь земельного участка довольно велика и превышает 1,5 акра (0,61 га).

Время и трудозатраты. Мы располагали большим запасом времени на проектирование и были готовы заниматься проектированием, не жалея сил.

События

На ход проектирования повлияло несколько событий, происшедших за этот период времени.

- В конце 1988 г. умер доктор Гринвуд.
- Чертежник, нанятый архитектором для выполнения строительных чертежей, неправильно согласовал план фундамента с планом первого этажа. Это несоответствие было обнаружено после заливки фундамента. Для исправления этой ошибки пришлось расширить площадь первого этажа на 1 фут в западную сторону и на 1 фут в северную. На рисунке в этой главе фасад здания показан в том виде, в каком оно было построено, а не в том, в каком было спроектировано.

Проектные решения и доработки

Исследования

Пересмотр назначений комнат в обоих крыльях дома. Интерьер здания был пересмотрен с точки зрения назначения отдельных комнат коренным образом; в том крыле, где находились спальни, было решено разместить помещения общего пользования (музыкальный зал, гостиную, возможно, студию Нэнси) и, соответственно, спальню хозяев и спальни для гостей разместить там, где раньше находился музыкальный зал. **Плюсы.** Станет легче обустроить главный вход и вход для учащихся, а также, возможно, удастся разместить дополнительные студии в юго-восточной пристройке. **Минусы.** Такой пересмотр назначения помещений обойдется дорого; нам потребуется увеличить количество ванных комнат в западном крыле; больше не останется места для камина; а также придется отделить гостиную от столовой, поэтому все будут вынуждены проходить через кухню. Мы сразу же отказались от этого варианта.

Юго-восточное крыло. Построить новую спальню хозяев с удобствами в юго-восточной части дома, возможно, включив в нее зеленую спальню. Мы отказались от этой идеи, чтобы не пришлось выкорчевывать белый дуб и красные дубы с юго-востока от дома, а также сохранить доступ со стороны подъездной дороги к жилой части дома и к заднему двору.

Павильоны Макклюра (рис. 22.3). **Плюсы.** Музыкальный зал (северный павильон) становится почти полностью отдельным, в него можно легко попасть с

подъездной дороги, а также имеется хорошая акустическая изолированность от остальной части дома. Из гостиной (южный павильон) открывается превосходный вид на сад и двор. Освобожденная гостиная превращается в то место, где можно собраться вокруг камина, чтобы заняться чтением или просто поговорить. **Минусы.** Для южного павильона пришлось бы пожертвовать великолепным черным дубом с четырьмя ответвлениями от главного ствола. Мы не могли использовать другую полезную площадь, чтобы предоставить возможность входить в музыкальный зал тем, кто посещает выступления учеников, которые берут уроки музыки.

Мы довольно быстро отказались от идеи строительства южного павильона, поэтому снова пришли к решению о модернизации гостиной, чтобы в ней появилось привлекательное место у камина.

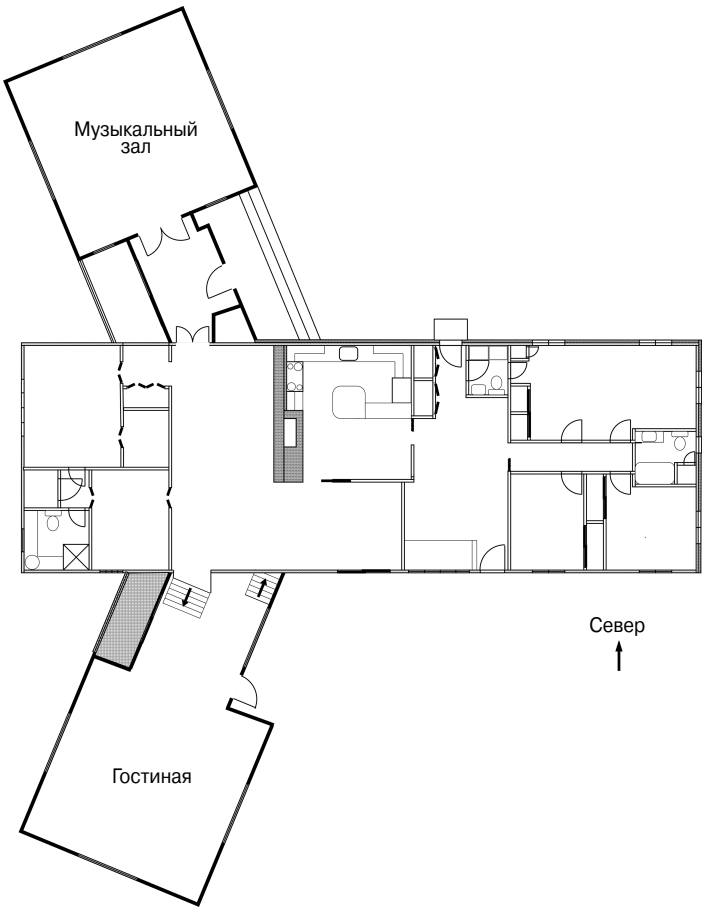


Рис. 22.3. Эскиз павильона Макклюра

Центр сосредоточения усилий по проектированию переместился на северный павильон. Мы рассмотрели много вариантов его формы и конфигурации, после

чего в конечном счете вместо этого павильона было возведено северное крыло, включающее музыкальный зал, студию Фреда, холл и переднюю веранду.

Поэтапное решение задач проектирования

В ходе дальнейшей работы стало очевидно, что мы можем разделить задачу модернизации дома на три задачи, почти не связанные друг с другом.

- Существующее восточное крыло: крыло со спальнями, а также, возможно, с игровой комнатой.
- Центральная часть: кухня, северный холл и его шкафы, игровая комната, прачечная/туалет, лестницы в подвал.
- Существующее западное крыло: музыкальный зал, столовая и западная спальня с удобствами наряду с вновь пристроенным новым западным крылом.

Как оказалось, такое распределение направлений работы по проектированию позволяет нам более свободно распоряжаться своими ресурсами; оно использовалось во всем дальнейшем проектировании.

Этапы работы. Довольно скоро мы решили предусмотреть два этапа проектирования и строительства, отстоящие на несколько лет, в основном для того, чтобы можно было сохранить контроль над решением задач проектирования и надзора за строительством. На этапе I все усилия должны были быть сосредоточены на существующем западном крыле и новом западном крыле; этап II предусматривал реконструкцию кухни и игровой комнаты. Описание работ на этапе II рассматривается как отдельный практический пример в главе 23.

Восточное крыло

В восточном крыле были выполнены различные доработки, причем все они были направлены на создание отдельных удобств рядом со спальней хозяев в виде собственного душа и туалета, а также на предоставление комфортных условий проживания для доктора Гринвуда. В некоторых довольно поверхностно проработанных наметках рассматривалось перемещение лестницы, ведущей с нижнего этажа, ближе к красной спальне, или установка этой лестницы там, где можно было ею пользоваться, проходя в красную спальню и игровую комнату. После смерти доктора Гринвуда требование по выделению еще одной спальни и ванной в дополнение к спальне хозяев стало не актуальным. В результате после перемещения спальни хозяев в западное крыло ближе к окончанию процесса проектирования мы оставили усилия по модернизации восточного крыла и сохранили его в том виде, в каком оно находилось.

Это крыло служит в качестве помещения для гостей, в котором отопление или кондиционирование включается только в тех случаях, когда в нем кто-то

проживает. В семье Роджера теперь пять детей, у Барбары — четверо. Если какая-то из этих семей наносит нам визит, мы поселяем их в этом помещении и в апартаментах на нижнем этаже.

Расположение комнат в западной половине дома с учетом их назначения

Больше всего времени мы отвели на исследование различных вариантов расположения комнат с учетом их назначения в старом западном и новом западном крыле, которые рассматривались как единое распределяемое пространство, хотя в большинстве вариантов неявно учитывалось требование по сохранению существующих стен между комнатами.

Мы исследовали эти варианты еще до принятия решения о том, что спальня хозяев должна остаться в западном крыле; предполагалось, что старая спальня хозяев должна быть частью распределимого пространства. Поэтому все варианты, которые рассматривались на первых порах, касались лишь размещения музыкального зала, гостиной, столовой, библиотеки, студий Нэнси и Фреда. План, приведенный на фронтисписе этой главы, на данном этапе не соответствует действительности.

После того как была исключена возможность расширения дома на юг, началось рассмотрение возможности достройки дома в северном, западном или в том и другом направлениях. Мы рассматривали следующие варианты расположения комнат с учетом их назначения.

- Музыкальный зал — в западной части; гостиная — в северной.
- Музыкальный зал — в северной части; гостиная — в западной.

Изменение подхода. Отказ от рассмотрения бюджета в качестве одного из проектных ограничений

По мере увеличения количества рассматриваемых вариантов стало очевидно, что необходимость постоянного учета бюджетных ограничений, выражающихся в определении полных затрат на возведение пристройки площадью в 1000 кв. футов, ограничивало нашу свободу мышления. Поэтому мы решили прежде всего разработать проект, направленный на достижение наших целей, и лишь затем произвести техническую оценку затрат и определить, следует ли предусмотреть дополнительные расходы, если это окажется оправданным. Как только было принято это решение, наше мышление стало намного более раскованным.

Я сам в свое время выступал в пользу этого подхода, когда речь шла о проектировании систем компьютерной графики. Изучая указанную проблемную область, я обнаружил, что лучший способ создать экономически эффективную прикладную систему состоит в том, чтобы вначале сделать ее технически эффективной, а затем заняться решением задачи сокращения ее стоимости. Это гораздо лучше

по сравнению с тем, чтобы вначале создать самую дешевую систему, а затем наращивать ее до тех пор, пока она не станет практически применимой. Я понял, пусть даже не сразу, что этот подход является применимым и при проектировании дома.

Вновь обнаруженное требование. Увеличение площади шкафов для верхней одежды

В ноябре 1990 г. мы проверили предварительный проект, рассмотрев несколько примеров ситуаций (импровизированных и не описанных), которые могут возникнуть при нашем проживании в доме. Подобные примеры рассматривались и ранее, но на сей раз мы провели проверку более тщательно, поскольку степень готовности проекта стала намного выше. В частности, был рассмотрен сценарий проведения два раза в месяц послеобеденной встречи с группой учащихся, которых мы консультируем и принимаем у себя, в количестве 30–40 человек.

Поскольку гости приходят к нам зимой, они должны куда-то поместить свою верхнюю одежду. Но куда именно? В шкафу для верхней одежды в холле она явно не поместится. Значит, нужно ее разложить на музыкальных инструментах? Разбросать по всей студии? Как же все-таки решить вопрос о размещении верхней одежды? Рассмотрим такой вариант: одежда будет разложена на кровати в комнате для гостей, смежной с гостиной. Однако в этом варианте проекта мы отказались от указанной комнаты.

Одно из решений могло состоять в том, чтобы значительно увеличить площадь шкафа для верхней одежды. Другой вариант мог предусматривать то, чтобы комната для гостей осталась в западном крыле, сменила свое назначение на спальню хозяев, а перепроектирование восточного крыла было бы прекращено. Суммарная стоимость: включает затраты на возведение пристройки в западном направлении большей площади, определяемой, по меньшей мере, существующей шириной комнаты для гостей. Чистая стоимость: суммарная стоимость за вычетом затрат, которые были запланированы на возведение восточного крыла, но так и не реализованы.

Задача. Если в пристройку в западной части дома, кроме спальни хозяев, войдут другие комнаты, необходимо определить суммарную площадь и проверить полученное значение на соответствие законодательным требованиям по обеспечению минимальной площади для спальных мест; между спальней и окнами не должна находиться закрывающаяся дверь.

Следствие 1. Если студия Нэнси будет помещена в западную пристройку, то соблюдение указанных требований не вызовет затруднений. Какие-либо другие комнаты не могут быть расположены в этом месте. (Фред часто работает в то время, когда Нэнси спит, но не наоборот, поэтому его студия также не может быть здесь расположена.)

Следствие 2. Если музыкальный зал не может быть расположен в западной пристройке, то его следует расположить в северной. В таком случае музыкальный зал будет изолирован от главных жилых помещений, а доступ к нему с подъездной дороги станет удобным.

Окончательное решение по размещению комнат с учетом их функционального назначения

После принятия описанного выше варианта процедура принятия окончательного решения существенно ускорилась.

Гостиная. Для выполнения требования по проведению концертов с выступлениями учащихся, которые берут уроки музыки, было бы желательно предусмотреть возможность время от времени соединять гостиную и музыкальный зал. Эта задача была решена путем установки 12-футовых раздвижных дверей (с четырьмя створками), которые убираются во вновь созданный карман с внешней стороны северной стены спальни хозяев.

Расширение гостиной было осуществлено за счет включения площадей существовавшего ранее холла, шкафа для верхней одежды в холле и шкафа для верхней одежды в комнате для гостей; холл и эти два шкафа должны были быть предусмотрены в новом крыле. Два рояля были перенесены из гостиной в музыкальный зал, в результате чего полезная площадь гостиной существенно увеличилась.

Студия Фреда. Стало очевидно, что после этого студию Фреда следует разместить так, чтобы она заполнила угол между северным и западным крылом. Площадь студии Фреда была определена с учетом размещения шкафа для новой спальни хозяев из студии Нэнси и других шкафов. В северном холле было также размещено копировальное устройство, к которому одинаково удобно можно пройти из музыкального зала, студии Нэнси и студии Фреда.

Застекленная веранда. Южная веранда, о которой мы мечтали, удобно разместилась в юго-западном углу нового западного крыла. Первоначально мы предполагали построить ее в виде веранды на уровне земли. Однако мы поняли, что при этом придется предусмотреть для спуска к ней ступеньки, ведущие с первого этажа, которые займут много места, поэтому оставили в силе вариант с размещением этой веранды на уровне первого этажа. Кроме того, после изучения сценариев использования веранды мы решили ее застеклить и предусмотреть много открывающихся окон, а не просто возвести над ней навес. В противном случае наружная веранда была бы мало востребована, поскольку на ней слишком холодно зимой и слишком жарко летом.

Передняя веранда. Проект передней веранды неоднократно подвергался пересмотру; в первоначальном замысле она должна была перекрывать весь восточный фасад северного крыла. В этом случае сохранялась бы исходная диагональная ориентация северного павильона Макклюра, который повернут прямо в сторону подъезда к дому от дороги. Веранду решено было сделать достаточно большой

для того, чтобы в ней поместились две расположенные напротив садовые качающиеся скамейки; в результате неожиданно появился уголок для бесед, который всем очень понравился.

Фронтон веранды был расположен с ориентацией в 45°, что позволило решить следующие две проблемы. Во-первых, весь общий вид фронтона стал свидетельствовать о том, что здесь находится вход в дом. Во-вторых, фронтон позволил изящно замаскировать разницу в высоте потолков в старом доме (8 футов) и в новом крыле (9 футов), а также выровнять соответствующие линии карнизов (рис. 22.4).

Рис. 22.4. Вид на перепроектированный дом с северо-восточного направления

Хранилище в подвале. Хотя и на довольно позднем этапе, но мы обнаружили возможность без особых затрат отвести пространство для хранилища под западным крылом, поскольку здесь находится достаточно большой уклон. После выполнения небольшого объема земляных работ и применения недорогой отделки появилось закрытое помещение для хранилища площадью 530 кв. футов; это помещение расположилось под музыкальным залом и позволило, кроме всего прочего, разместить небольшую мастерскую и отвести место для механического оборудования нового крыла. Это позволило нам отказаться от отдаленного хранилища, которое мы раньше арендовали, и получить более удобный доступ.

Выполнение ограничений, касающихся отступа от границ собственности. В главе 3 рассматривалась история одной из наиболее интересных встретившихся мне проблем проектирования и описано ее решение. Назначения,

которым служит музыкальный зал, весьма специфичны, поэтому при его создании приходилось учитывать довольно сложные ограничения. Любые варианты размещения музыкального зала с учетом этих ограничений, а также любые приемлемые варианты конфигурация студии Фреда не вписывались в принятые в Чапел-Хилл ограничения, согласно которым тень здания при любом положении солнца не должна находиться ближе 17 футов к границам собственности. Какие бы проектные решения мы ни рассматривали, данная проблема оставалась неразрешимой. Наконец, было решено купить полосу земли шириной в 5 футов у нашего соседа.

Столовая. В проекте было предусмотрено просто расширить помещение столовой в южном направлении. В результате длинной стороной этого помещения становится стена, сориентированная с севера на юг, а не с востока на запад. В связи с этим возникает необходимость реконструкции ската крыши, а также внесения изменений в конфигурацию навеса, проходящего вдоль южной стороны дома, которые не способствуют улучшению его внешнего вида. После оценки стоимости мы отказались от этого варианта, в основном опираясь на наши расчеты затрат.

Изменения, происшедшие в ходе строительства

Окна в студии Фреда. Согласно проекту, западные окна в студии Фреда должны были иметь высоту 4 фута и начинаться на высоте 3 футов от пола. Во время подготовки оконных проемов стало очевидно, что из таких окон будут видны только верхушки деревьев, поскольку в западном направлении уровень местности понижается. В связи с этим были выбраны другие окна, которые имеют высоту 6 футов и начинаются на высоте 1 фут от пола. Такое решение действительно привело к улучшению вида из комнаты и ее общего облика.

Это — проблема, о существовании которой нельзя было бы догадаться, изучая строительные чертежи и карты местности с обозначением высот; но если бы в ходе проектирования применялось моделирование в виртуальной среде, ее выявление не составило бы труда.

Окно в нише для органа. Наш сын, Кен Брукс, предложил поместить узкое окно за скамейкой перед новым органом. Из этого окна стал проникать свет в музыкальный зал с третьей стороны[1].

Оценка — успехи и неустраненные недостатки

После завершения работ по проекту этапа I прошло примерно семнадцать лет, а после перепроектирования игровой комнаты и кухни, выполненного на этапе II, — четырнадцать.

У нас не осталось ни одного замечания, по поводу которого мы могли бы сказать: “Жаль, что мы не сделали то-то или то-то иначе”. Наши проектно-

конструкторские работы продолжались необычайно долго, причем даже самым мелким деталям было уделено достаточное внимание, и это все окупилось, поскольку мы получили дом, удобный для проживания, отвечающий своему назначению и достойный восхищения. Безусловно, мы не воплотили в нем все свои идеалы, поскольку этого не позволили ограничения.

Дверь между кухней и гостиной. Наибольшую пользу принесло сооружение двери из кухни в гостиную. Это привело к существенному улучшению схемы передвижения по дому; кроме того, дом стал просматриваться от одного конца к другому.

Спальня хозяев. Наш план не совсем обычен в том, что спальня хозяев находится в окружении помещений другого назначения. Но в данном проекте мы не учитывали, повлияет ли такое решение на стоимость дома, которую он имел бы при перепродаже. И действительно, мало найдется таких семей, которым потребовался бы большой музыкальный зал, студия хозяина дома (Фреда) и студия хозяйки (Нэнси).

Музыкальный зал. Это помещение хорошо приспособлено для проведения уроков и упражнений с отдельными учениками, поскольку его можно преобразовать в закрытое помещение. В нем также очень удачно разместился большой орган. В музыкальном зале проводятся выступления учащихся, которые посещают уроки музыки, а также ежегодный симпозиум, поскольку между этим залом и гостиной может быть открыт широкий проем. Но в этот зал нет отдельного входа с улицы. Учащиеся оставляют обувь и инструменты в игровой комнате, а затем проходят в зал через кухню и гостиную.

Гостиная. После того как два рояля были вынесены из гостиной, она стала намного удобнее. Гостиная вполне может быть расширена за счет включения площади, занимаемой шкафом. Но это широкое помещение выглядело бы много лучше, если бы потолок имел бо́льшую высоту.

В нем неудобно демонстрировать фотографии и видеофильмы. Нельзя найти подходящее место для размещения экрана.

Столовая. Эта комната все еще остается ограниченной по своему назначению. Но наличие широкого проема между музыкальным залом и гостиной позволяет расположить в один длинный ряд несколько столов, а площади этих помещений дают возможность установить дополнительные столы, поэтому в доме можно собрать и рассадить много людей.

Передняя веранда. На веранде есть две расположенные напротив качающиеся садовые скамейки и создан уголок для встреч в узком кругу, который часто используется. Главный вход теперь виден издалека; внешний вид дома заметно улучшился.

Новые функции

Оказалось, что после перепроектирования у дома обнаружились новые функции, о которых мы не подозревали. Как было отмечено в предыдущих главах, такая ситуация на практике является обычной и не представляет собой нечто исключительное.

Застекленная веранда как место встреч. Стало очевидно, что застекленная веранда очень хорошо подходит для проведения встреч не более десяти человек. В расчете на использование этой веранды была основана новая христианская школа, и на ней же в течение нескольких лет проводились совещания членов ее правления.

Музыкальный зал как продолжение гостиной и место проведения встреч. Первоначальным требованием было то, чтобы гостиная могла служить продолжением музыкального зала, что способствовало бы повышению удобства проведения выступлений учащихся, которые берут уроки музыки. Это требование было успешно выполнено. Появилась возможность усадить учащихся и их родителей в количестве около 25 человек на рядах стульев, которые начинаются в музыкальном зале и заканчиваются в гостиной.

Однако неожиданно для нас обнаружилась возможность сделать музыкальный зал продолжением гостиной. Мы стали проводить встречи христианского братства выпускников InterVarsity (InterVarsity Christian Fellowship — IVCF), на которых обычно присутствуют 30–40 человек, но иногда количество приглашенных становится больше 50. На этих встречах все внимание сосредоточено на докладчиках, стоящих у камина в гостиной, а ряды стульев, которые не поместились в гостиной, продолжаются в музыкальном зале.

Навес, ступеньки и двор как место, где можно принять и угостить приглашенных. Навес вдоль южной наружной стены дома был продлен вплоть до верхнего заднего двора, а под ним были расположены широкие ступеньки и установлен откидной стол. Как оказалось, в этом месте нам удобно принимать и угощать на свежем воздухе руководителей IVCF. Большое количество людей может разместиться на ступеньках.

Общие усвоенные уроки

1. Не жалеть усилий на проектирование. Мы вложили больше трудозатрат в расчете на квадратный фут, чем было бы экономически оправдано в той ситуации, когда оплата труда проектировщика становится частью цены на готовый продукт. Несомненно, тоже можно сказать и в отношении основной части программного обеспечения Linux. Если бы нам при разработке операционной системы OS/360 было позволено затратить больше усилий на проектирование, прежде чем началась реализация, то конечный резуль-

тат был бы много лучше. Причем я не думаю, что из-за этого существенно повысилась бы стоимость конечного продукта.

2. Необходимо также, не жалея времени, проводить больше встреч с основными пользователями и демонстрировать для них прототипы, которые они могли бы понять.
3. Рассмотреть большее количество сценариев использования.
4. Проверять и перепроверять работу профессионалов, таких как архитекторы, чертежники и декораторы. В общении с профессионалами добиваться полного взаимопонимания и следить за тем, насколько точно они выполняют полученные указания.

Примечания и ссылки

1. В книге [192] дана такая рекомендация, чтобы в каждую комнату поступал дневной свет, по крайней мере с двух сторон или больше (что еще лучше). Мы тщательно следовали этой рекомендации, установив потолочные окна в студиях Нэнси и Фреда (но, безусловно, не в спальне хозяев), а также окно на внутренней стене между студией Нэнси и застекленной верандой. Таким образом, в каждую из новых комнат в той или иной степени поступает дневной свет с трех сторон.

TheDesignofDesign.indb 350 30.10.2012 15:58:39

23

Изучение практического примера. Реконструкция кухни

Если вы не можете вытерпеть жару, выйдите из кухни.

ГАРРИ С. ТРУМЕН

◂ *Модель виртуальной среды реконструированной кухни. Научно-исследовательский проект по использованию виртуальных сред. Северокаролинский университет, г. Чапел-Хилл*

Основные факты и особенности

Назначение данного практического примера. Простой практический пример, описанный в этой главе, позволяет показать возможности средств проектирования. В этом проекте мы использовали чертежи, программное обеспечение машинного проектирования (автоматизированного проектирования), масштабные модели, макеты в натуральную величину и средства исследования на основе виртуальной среды для достижения наивысшего качества проектирования. И применение виртуальной среды, и создание макетов способствуют улучшению проекта, как ни одно прочее средство проектирования.

Решение в чистом виде. Перемещение наружной стены. Это изменение привело к существенному пересмотру проекта.

Решение в чистом виде. Установка двери между кухней и гостиной. После появления этой двери вся схема передвижения по дому изменилась.

Потолочные окна. После установки двух потолочных окон темная, выходящая на север кухня преобразовалась в ярко освещенное, приятное для глаз помещение.

Общие сведения и предыстория

Местоположение

413, Грэнвилл Роуд, г. Чапел-Хилл, шт. Северная Каролина.

Владельцы

Фредерик и Нэнси Брукс.

Проектировщики

Фредерик и Нэнси Брукс.
Привлеченные в качестве консультантов Мэри Джун Маго (Mary June Mago) и Алекс Джонс (Alex Jones), член Американского общества проектировщиков интерьера (ASID).

Даты

1995-1996 гг.

Предыстория

Этот проект стал этапом II проводимой в 1990-х годах реконструкции дома, построенного в 1960-х годах. На этапе I была выполнена пристройка западного крыла, холла и веранды. Этот этап был описан в главе 22. Проведение этапа II

было запланировано на срок, отстоящий на несколько лет от этапа I, чтобы можно было отвести достаточно времени на проектирование для этапа II и учесть все недочеты, допущенные на этапе I.

Задачи

Основные задачи. Наши основные задачи состояли в том, чтобы расширить, реконструировать и сделать более привлекательной небольшую, темную, выходящую на север комнату, которая служила в качестве кухни и помещения для завтрака.

Другие задачи. Остальные задачи представлены ниже в порядке убывания значимости.

- Улучшить схему перемещения по дому, при которой до сих пор все перемещения из одной половины дома в другую осуществлялись через узкую лестницу на первый этаж. Нам также было необходимо приспособиться к приему учащихся, которые берут уроки игры на скрипке. Эти учащиеся входят через заднюю дверь, вынимают скрипки из футляров, оставляют футляры в игровой комнате и переходят в музыкальный зал. По окончании занятий те же действия осуществляются в обратном порядке.
- Перемещение кухонного стола к окну в сад для улучшения обзора.
- Реорганизация помещения таким образом, чтобы повар мог разговаривать с сидящим, неработающим посетителем.
- Сделать кухню удобной для каждого из следующих вариантов использования:
  - приготовление завтрака одним поваром;
  - обычная готовка, выпечка и консервирование с привлечением одного повара (работающего неполный день);
  - приготовление большого количество блюд с привлечением нескольких поваров (вплоть до трех);
  - обслуживание с удобством через буфет групп учащихся в составе 30–40 человек.
- Установка кухонной стойки со значительно большей площадью.
- Установка большой кухонной раковины.
- Проектирование кладовой с отдельным входом.
- Сохранение привлекательного внешнего вида дома.
- Расширение входа в дом с обратной стороны, который служит основным входом для членов семьи, учащихся и часто для близких друзей (расширение задней двери).
- Оформление задней двери в таком же стиле, как и весь внешний фасад.

- Установка изразцов с изображением птиц на унылой стене, где расположена кирпичная дымовая труба.
- Размещение всевозможных необходимых мелочей в шкафчиках.
- Выставление небольшого количества стеклянной посуды.

Возможности

Уменьшение количества членов семьи, постоянно проживающих в доме. Количество членов семьи, постоянно проживающих в доме, уменьшилось, поскольку дети выросли. Поэтому обычно в приеме пищи на кухне участвуют два человека, иногда три, а в редких случаях — четыре, тогда как в свое время за столом собирались пять человек.

Доступное пространство в игровой комнате. В 1992 г. была построена студия для обучения музыке в новом западном крыле, поэтому в игровой комнате освободилась площадь 5×5 футов, занимаемая прежде органом, а также площадь 2×6 футов, где раньше находилась высококачественная аудиотехника.

Карниз шириной три фута. Под влиянием идей Фрэнка Ллойда Райта (Frank Lloyd Wright) в 1960-х годах многие дома, включая наш дом, строились в стиле ранчо, поэтому имеют широкие карнизы.

Лимиты времени и сил, затрачиваемых на проектирование. Эти лимиты были почти полностью неограниченными.

Ограничения

Рост главного действующего лица. Основное действующее лицо на кухне имеет рост 5 футов 1 дюйм.

Денежные средства, запланированные на строительство. Ограничения на количество запланированных денежных средств не налагались, но не предусматривалось проведение в доме каких-либо крупных конструктивных изменений.

Внешний вид дома. Мы смогли наконец-то сделать внешний вид дома привлекательным благодаря возведению пристроек в 1991 г. (см. главу 22), поэтому больше не хотели ничего менять.

Существующая кухня. Размеры и форма новой кухни неизбежным образом определялись размерами и формой существующей кухни (рис. 23.1).

Стена, на которую выходит кирпичная дымовая труба. Эта стена имела толщину 8 дюймов и не позволяла улучшить схему перемещения по дому.

Лестница на первый этаж. Эту лестницу нельзя было перенести в другое место.

Наружная дверь игровой комнаты. Эта дверь играла важную роль, но мы никак не могли расположить ее наилучшим образом.

Задняя дверь. Эта дверь была пробита в кирпичной стене, поэтому попытка перенести ее обошлась бы очень дорого.

Существующая прачечная/туалет. В это помещение не планировалось вносить изменения.

Кладовая. Мы должны были сохранить назначение и вместимость кладовой, но не ее расположение.

Шкафы. Назначения и размеры шкафов в северном холле должны были быть сохранены, но их расположение нас не устраивало. Проектирование данного этапа реконструкции представляет собой интересный пример, подчеркивающий то, насколько важно тщательно проводить исследования существующих сценариев использования. Мы не смогли бы сразу же сформулировать требования к новому проекту, если бы просто изучали разработанные ранее проекты кухни. В основу новых требований лег тот факт, что мы могли проследить конкретные сценарии использования кухни, наблюдаемые на протяжении тридцати лет, и понять, насколько дорого нам обойдется "создание хаоса", если мы не сможем должным образом расположить шкафы в одном месте и рассредоточим их без учета назначения по всему дому.

Соображения, связанные с конструкцией дома. Крыша опирается на конструктивные вертикальные балки, расположенные вдоль наружной стены дома.

Использование остальной части дома. Во время реконструкции должна оставаться возможность постоянно использовать другие комнаты.

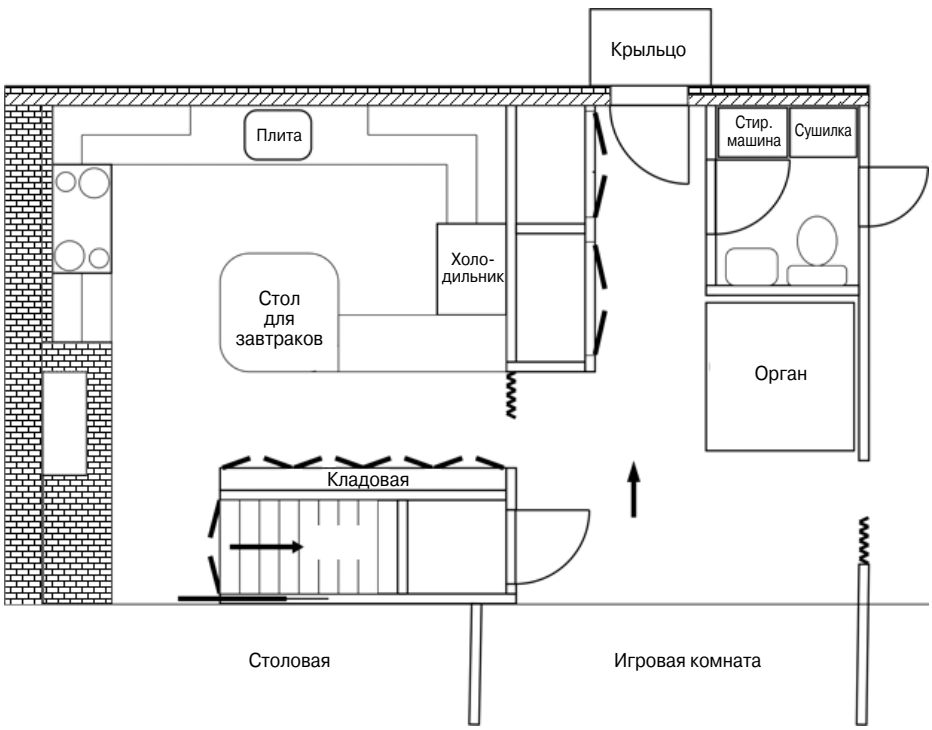


Рис. 23.1. План кухни перед реконструкцией

Регламентация проекта с учетом такого важного показателя, как ширина помещения

Необходимая ширина в направлении с севера на юг

Во всех предыдущих попытках создать осуществимый проект мы останавливались перед тем, что ширина кухни каждый раз оказывалась недостаточной. При этом рассматривались перечисленные ниже требования к новому проекту кухни.

- Рядом с окном должен находиться обеденный стол.
- Кухонная раковина должна находиться в таком месте, из которого, пользуясь ею, можно было смотреть из окна.
- К кухонной раковине должен быть предусмотрен доступ из той части кухни, где расположен обеденный стол.
- Через кухню должен быть обеспечен удобный проход для тех, кому нужно попасть в восточное или западное крыло.
- Должно быть предусмотрено достаточное пространство для кухонной стойки и шкафчиков.
- Треугольник, охватывающий плиту, кухонную раковину и холодильник, должен иметь минимальный периметр.

Предварительное проектирование

В предварительном проекте было решено создать островок с кухонной раковиной, который ориентирован в сторону окна и обеденного стола. Мы сочли необходимым установить кухонную стойку с плитой напротив южной стены.

В связи с этим нам пришлось распределять пространство по ширине, с севера на юг, с учетом кратчайших допустимых расстояний, следующим образом.

- Обеденный стол (шириной не менее 30 дюймов).
- Проход для перемещения из одной части дома в другую (шириной не менее 24 дюймов).
- Островок с кухонной раковиной (шириной не менее 24 дюймов).
- Пространство для работы и прохода между кухонной раковиной и плитой (согласно первоначальной оценке, его ширина должна быть не менее 36 дюймов).
- Плита и кухонная стойка (шириной не менее 27 дюймов).

Эти пространственные значения в общей сложности составляли не меньше 12 футов и 3 дюймов. Исходная ширина равнялась 12 футам.

Было решено предусмотреть возможность размещения за обеденным столом до трех человек, если же мы принимали также посетителей, то четвертого (и пятого) человека приходилось сажать в проходе, что фактически приводило к тому, что нельзя было пройти из одного крыла дома в другой. Это иногда допустимо, но не всегда.

Тем не менее исследования на макете показали, что пространство между плитой и кухонной раковиной в действительности должно составлять 44 дюйма, а не 36 дюймов, поскольку следует учитывать наличие шкафчиков с открывающимися дверцами и выдвижными ящиками, открывающейся дверцы плиты и выдвижного поддона посудомоечной машины. Таким образом, необходимо было, чтобы общая ширина кухни составляла не меньше 12 футов 11 дюймов.

Альтернативные решения по проекту кухни с учетом ширины

Устранение как такового прохода, соединяющего два крыла дома. В одном из отвергнутых проектов рассматривалась возможность отказаться от отдельного прохода и перенаправить людей, которым требуется перейти из восточного крыла в западное, в промежуток между кухонной раковиной и плитой; однако в связи с этим постоянно возникали бы помехи в работе повара. Причем это был единственный вариант, который позволил бы уменьшить необходимую ширину, которая при наличии отдельного прохода превышает 12 футов 11 дюймов. Иначе говоря, мы были вынуждены рассмотреть другие варианты, в частности, следующие.

- Перенести кладовую и лестницу на нижний этаж в другое место и расширить кухню за счет освободившегося пространства.
- Соорудить отдельный эркер для размещения обеденного стола.
- Передвинуть всю северную стену кухни настолько, насколько позволяет карниз.

В любом из двух последних случаев не исключалась также возможность переместить в другое место кладовую, чтобы присоединить к свободному пространству еще 9 дюймов.

Следует ли перемещать лестницу на нижний этаж? Мы провели всестороннее исследование, касающиеся того, следует ли перемещать лестницу в другое место. Полученные результаты показали, что единственной осуществимой альтернативой является возведение за пределами дома башни с винтовой лестницей. В башню должна также выходить южная наружная дверь игровой комнаты. Другие решения не позволяют предусмотреть удобную лестницу либо на верхний, либо на нижний этаж.

Мы работали над вариантом, предусматривающим возведение башни с лестницей, в течение нескольких месяцев. В конечном итоге этот вариант был

отклонен как дорогостоящий и неэстетичный, несмотря на то, что он мог бы улучшить схему перемещения по дому.

Вообще говоря, в процессе проектирования критическим моментом оказалось то, что все варианты организации перемещения из одного крыла дома в другое, предусматривающие применение лестниц, были признаны неосуществимыми или неприемлемыми. Это привело к резкому сужению пространства возможных проектных вариантов. Анализ указанного компонента общего проекта реконструкции кухни оказался наглядным примером процедуры "поиска в дереве", описанной Саймоном (Simon), поскольку я после исследования многочисленных вариантов проекта с винтовой лестницей каждый раз отказывался от них, а обнаружив, что ни один из них не является приемлемым, поднялся вверх на один уровень в дереве проектирования и полностью исключил возможность применения лестниц как средств сообщения между частями дома.

Выбор между вариантами возведения эркера и перемещения всей северной стены кухни наружу. Исследования экстерьера дома на макете показали, что после перемещения всей северной стены кухни наружу дом будет выглядеть много лучше, чем после постройки эркера, а затраты остаются примерно сопоставимыми. Поэтому был выбран вариант с перемещением стены. Те же исследования на макете показали, что после увеличения ширины кухни на 18–24 дюйма создается эстетически привлекательный внешний вид дома, учитывая то, что карниз имеет ширину 36 дюймов.

Итоговый проект, рассчитанный на большую ширину

Увеличение ширины кухни на 24 дюйма в результате перемещения стены и на 9 дюймов в результате переноса кладовой в другое место привело к существенному ослаблению ограничений по ширине. Ширина островка была увеличена с 24 до 36 дюймов, что позволило улучшить условия работы повара, размещения оборудования и хранения необходимых материалов. Северный проход был расширен до 39 дюймов, а южный — до 44 дюймов.

Распределение объектов в кухне с учетом длины помещения

Эталон, по которому планировалось распределение по длине. Эталоном, по которому планировалось распределение по длине, стала южная стена. Длина позволяла установить плиту размером 48 дюймов, справа от нее — одну кухонную стойку для работы, а слева — другую. Западная часть помещения, предназначенная для работы, была выбрана как место для размещения уголка готовки завтрака, поэтому в ней было решено установить микроволновую печь и тостер, что в общей сложности потребовало не меньше 36 дюймов.

Восточная кухонная стойка, предназначенная для работы, стала основным местом для приготовления блюд и выпечки, где были предусмотрены шкафчики для хранения сухих ингредиентов и специй, а также миксеров и других кухонных приспособлений. Исследования на макете показали, что для кухонной стойки желательно выбрать длину 48 дюймов.

Проектирование. Южная стена была продлена в восточном направлении на 18 дюймов, что привело к разделению кухни и игровой комнаты на два помещения. Исследования на макете в целях выбора места для новой кладовой показали, что для нее достаточным будет дверной проем шириной 5 футов (расположенный по диагонали), и в результате будет получен хороший визуальный эффект (рис. 23.2).

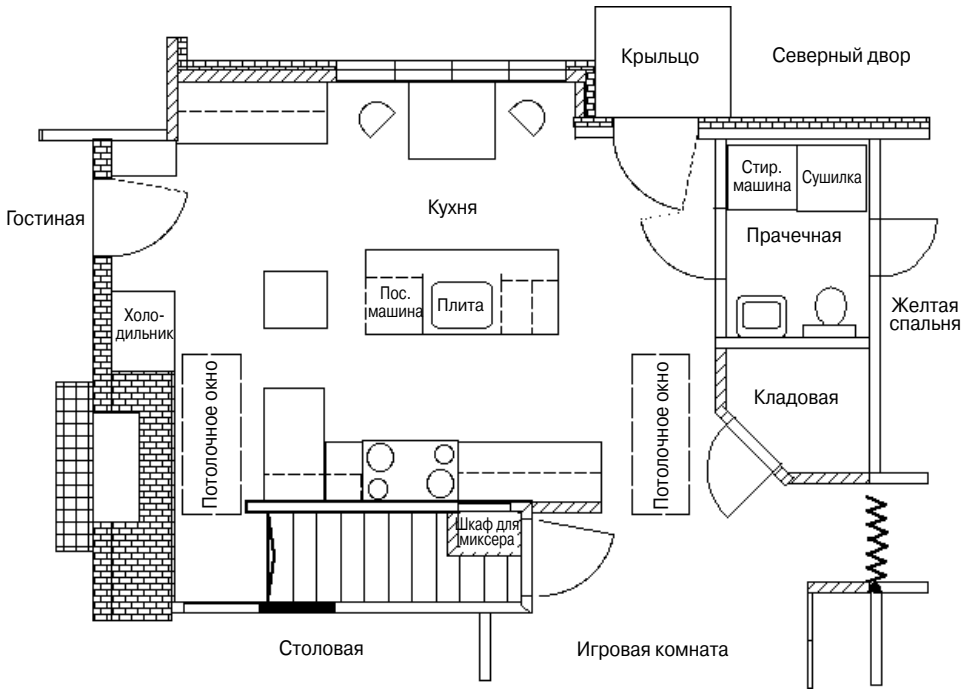


Рис. 23.2. План кухни после реконструкции

Другие проектные решения

Двери. Принятие решения в отношении того, следует ли прорезать дверь из гостиной в кухню. Несмотря на то, что для этого требовалось разобрать часть кирпичной стены толщиной 8 дюймов, т.е. пойти на определенные затраты, было принято положительное решение. Этого потребовала задача улучшения схемы перемещения по дому. Анализ возможного решения об установке двери из столовой в игровую комнату привел к тому, что это решение было отвергнуто, поскольку мы не могли терять ценное пространство стен в той и в другой комнате.

Шкафы. Еще одно рассматриваемое решение касалось того, куда переместить шкафы из северного холла. Мы решили передвинуть их к восточной стене игровой комнаты.

Полки кладовой. Мы перенесли полки кладовой (которые первоначально находились на южной стене) в новую кладовую, построенную рядом с северной стеной игровой комнаты, в то место, которое освободилось после перемещения органа.

Конфигурация промежутка между кухней и игровой комнатой. Размещение двери кладовой по диагонали увеличило визуально воспринимаемый промежуток.

Схема перемещения. Южный проход предназначен только для поваров, а северный сделан доступным для наших визитеров, а также для всех, кому нужно пройти по дому из восточного крыла в западное, или наоборот.

Шкафы. Один из вариантов предусматривал установку навесных шкафов над островком с кухонной раковиной, но просмотр создаваемого интерьера в виртуальной среде показал, что при этом визуальное впечатление от компоновки объектов в комнате становится менее благоприятным.

Периметр треугольника "кухонная раковина–плита–холодильник". Журнал *Small Homes Council* рекомендует, чтобы треугольник, в который вписываются кухонная раковина, плита и холодильник, имел периметр не больше 26 футов. В нашем окончательном проекте мы смогли добиться того, чтобы периметр составил 24 фута.

Приспособления для хранения тарелок и стаканов. Мы решили размещать тарелки и стаканы в выдвижных ящиках, поскольку это более удобно по сравнению с их размещением в шкафчиках для тех, кто имеет малый рост.

Изменение углового положения вспомогательного островка. Островок размером 26×26 дюймов с зеркально отраженными расширениями шириной 12 дюймов позволяет разместить серебро, стаканы и другие предметы сервировки. При этом также обеспечивается удобная установка холодильника и создается дополнительное расширение главного островка, которое позволяет организовать обслуживание по принципу буфета.

Низкая восточная кухонная рабочая стойка. Восточная кухонная рабочая стойка имеет меньшую высоту, которая выбрана для удобства в работе людей небольшого роста. Под эту стойку можно также убирать кухонное оборудование временного пользования, которое имеет относительно большую высоту.

Место для хранения оборудования. Пространство для хранения кухонного оборудования отведено в смежном шкафу на южной стене.

Освещение

Потолочные окна. Над обоими концами островка и над более темной южной частью комнаты установлены потолочные окна размерами 2×4 фута. Это

проектное решение было принято с учетом предложенного Александером шаблона проектирования: "Каждая комната должна иметь дневное освещение с двух или предпочтительно с трех сторон"[1].

Задняя дверь. Мы установили вместо сплошной задней двери стеклянную дверь.

Окна. Мы установили в кухне новые окна, соответствующие общему стилю уголка, в котором мы завтракаем.

Искусственное освещение. Предусмотрено семь контуров освещения, что позволяет формировать различные конфигурации для работы, создания общего настроения и подсветки прохода, предусмотренного в схеме перемещения по дому.

Цветовая схема. В качестве основного цвета применяется серовато-белый цвет, который позволяет повысить яркость освещения и подчеркнуть другие цвета. Мы сохранили старую деревянную обшивку в игровой комнате и на восточной стене кухни и покрыли кирпичную дымовую трубу белой сухой штукатуркой.

Оценка

Размер. Увеличение ширины на 2 фута 9 дюймов, достигнутое в результате переноса стены и перемещения кладовой, привело к изменению площади рабочей области и пространства для передвижения по дому. Благодаря тому что шкафы были сдвинуты в другое место, пространство увеличилось визуально на 2 фута 3 дюйма. Чистая площадь пола кухни выросла почти на 54 кв. фута.

Схема перемещения. Установка двери в гостиную, хотя и обошлась достаточно дорого, привела к радикальному преобразованию внутреннего пространства всего дома. Почти все передвижения людей из восточного крыла в западное, и наоборот, осуществляются через новую дверь. Из кухни видны восточные окна в желтой спальне и западные окна в студии Нэнси.

Яркость освещения. В кухне установлены потолочные окна, развешена стеклянная посуда и применена цветовая схема на основе серовато-белого цвета, поэтому условия освещения изменились и эта комната стала выглядеть лучше.

Изменения в игровой комнате. Игровая комната стала заметно уже вследствие того, что в ней были установлены дополнительные шкафы, но она все еще остается вполне применимой в следующих целях.

- Прием учащихся, которые берут уроки музыки, и размещение футляров инструментов.
- Размещение родственников учащихся, которые не должны присутствовать на уроках.
- Организация игр для наших внуков.

Южная дверь. Проход к южной внешней двери не совсем удобен.

Другие выполненные пожелания.

- Участок, на котором происходит прием пищи, хорошо подходит также для встреч с гостями. Кроме того, перед домом установлена кормушка для птиц, которая выглядит изнутри очень привлекательно.
- Уголок, в котором готовят завтрак, весьма удобен: не сходя с места, можно работать с микроволновой печью, электрической сковородой, тостером, выдвижными ящиками, плитой и брать приборы для сервировки.
- В случае необходимости на кухне можно с удобством организовать работу нескольких поваров.
- Новая кухня хорошо подходит для питания целых групп по принципу буфета следующим образом.
  - Посетители входят через столовую и выходят через гостиную.
  - Перед приемом пищи люди сами берут столовые приборы из выдвижного ящика в юго-западной секции кухонной стойки.
  - Подносы, тарелки и прочие предметы сервировки хранятся в шкафу рядом с юго-западной кухонной стойкой.
- Новая кладовая — намного просторнее и удобнее прежней.
- После реконструкции кухни внешний вид дома не стал хуже.

Использование чертежей, средств автоматизированного проектирования, моделей, макетов и виртуальной среды в проектировании

На проектирование реконструкции кухни были затрачены большие усилия по следующим причинам.

- Если кухня во всем устраивает жильцов дома, то достигается и общее удовлетворение от проживания в доме.
- Кухни интенсивно используются.
- Этот проект модернизации разрабатывался в условиях жестких ограничений, налагаемых наличием существующей структуры и потребностью учитывать схему передвижения по дому, поэтому в процессе проектирования приходилось решать трудные задачи.
- Проектировщики не были связаны какими-либо ограничениями, лимитирующими затраты времени на проектирование.

Ход проектирования показал, что в нем нельзя не воспользоваться всем диапазоном доступных средств проектирования.

Чертежи и средства автоматизированного проектирования. Разработка большинства проектных решений начиналась с создания эскизов, после чего проводилась рационализация решений для достижения согласованности и непротиворечивости по отношению к существующей структуре, для чего использовалась система архитектурного автоматизированного проектирования MiniCad на компьютере Macintosh. Результирующие файлы MiniCad служили в качестве проектных документов.

Основная часть работы в системе автоматизированного проектирования осуществлялась в масштабе 1/4 дюйма к 1 футу (1:48), а благодаря использованию системы автоматизированного проектирования и монитора, на котором могут размещаться сразу две страницы, упрощалась детализация в масштабах экрана вплоть до 1:6. Часто использовались также масштабы 1/2 дюйма к 1 футу и 1 дюйм к 1 футу.

Автоматизированное проектирование осуществлялось с применением слоев, в частности, отдельные слои были предусмотрены для исходного варианта кухни, удаляемых конструктивных компонентов, добавляемых конструктивных компонентов, оборудования и мебели.

Журнал проектирования. Обоснования, которые послужили причиной принятия наиболее важных проектных решений, а также варианты, исследованные на пути к достижению решений, по ходу работ фиксировались в журнале проектирования. Типичные страницы журнала проектирования в немного отредактированном виде показаны на веб-странице, посвященной этой книге.

Комплект изометрических чертежей. Мы также использовали комплекс средств проектирования кухонь, в котором предусмотрен набор изометрических сеток, а также набор изометрических чертежей оборудования, шкафов, кухонных стоек и пр., выполненных в надлежащем масштабе и напечатанных на электростатически активной пластиковой пленке. Этот комплекс оказался удобным и быстродействующим, а его применение позволило получить хорошие результаты. Главными ограничениями было то, что в наборе недоставало некоторых предметов мебели, а сам комплекс не обеспечивал представление цветов.

Модели. Нэнси изготовила на основе чертежей в масштабе 1/2 дюйма к 1 футу набор простых картонных моделей, чтобы было проще понять, как будет выглядеть проект в трехмерном пространстве. Эти модели оказались значительно более информативными по сравнению с изометрическими чертежами, поскольку предоставляли возможность рассматривать интерьер кухни под любым углом зрения, несмотря на то, что даже кухня была представлена в миниатюре.

Макеты. Для проверки наиболее важных проектных решений использовались макеты в натуральную величину. Они оказались просто неоценимыми.

Выдвинутая наружу внешняя стена моделировалась с помощью картонных упаковок для матрацев в целях прогнозирования того, как дом будет выглядеть извне. Для моделирования вариантов внутреннего размещения кухонных стоек применялись столы, строительный картон и козлы в более объемном внутреннем

пространстве другого здания. После создания очередной модели рассматривались конкретные сценарии использования кухни при различных расстояниях между предметами обстановки. Такая работа с моделями оказалась очень эффективным способом определения не только минимально допустимых расстояний между объектами, но и того, насколько проще становится организация функционирования кухни после лимитируемого снятия ограничений на использование ее площади.

В этом я сумел воплотить полученный прежде опыт представителя комитета по строительству церкви. После многих бесплодных усилий по планированию внутреннего пространства церкви мы пришли к тому, что его необходимо моделировать. Как оказалось, это был единственный удовлетворительный способ решения указанной задачи.

Визуализация виртуальной среды. В ходе того, как мой исследовательский коллектив в Университете штата Северная Каролина разрабатывал лабораторию виртуальной среды, мы с Нэнси проверяли саму лабораторию, тестируя с помощью этой лаборатории очередной предложенный нами проект кухни. На фронтисписе к этой главе показано одно из изображений, воспроизводимых на дисплее-шлеме наблюдателя, совершающего прогулку по виртуальной кухне. Наша технология отслеживания обеспечивала свободное передвижение в пространстве размером 15×18 футов, которое охватывало почти всю кухню.

Иллюзия присутствия, создававшаяся в сеансе проектирования кухни продолжительностью от 20 до 40 минут, была настолько впечатляющей, что наблюдатель забывал о том, что пребывает в виртуальной среде, и сосредоточивался только на обстановке кухни.

Данные, полученные в результате моделирования в виртуальной среде

- Наиболее важным открытием, полученным в сеансах моделирования в виртуальной среде, оказалось то, что подвесные шкафы над кухонной раковиной нерационально разделяют визуальное пространство, в результате чего кухня начинает казаться небольшой и тесной. Поэтому мы переделали проект, чтобы удалить эти шкафы, но все еще смогли сохранить необходимое пространство на полках.
- Подвесная лампа в уголке, где готовится завтрак, слишком бросалась в глаза, поэтому ее потребовалось заменить осветительной арматурой, закрепленной на потолке.
- Эксперименты в виртуальной среде подтвердили, что действительно необходимо разместить изразцы с изображениями птиц на стене, за которой расположен большой дымоход.
- Было также отмечено, что более эффективным является размещение деревянного настила кухни по диагонали.

- Другие полученные данные показали, какие улучшения требуются в техническом и программном обеспечении средств моделирования на основе виртуальной среды.

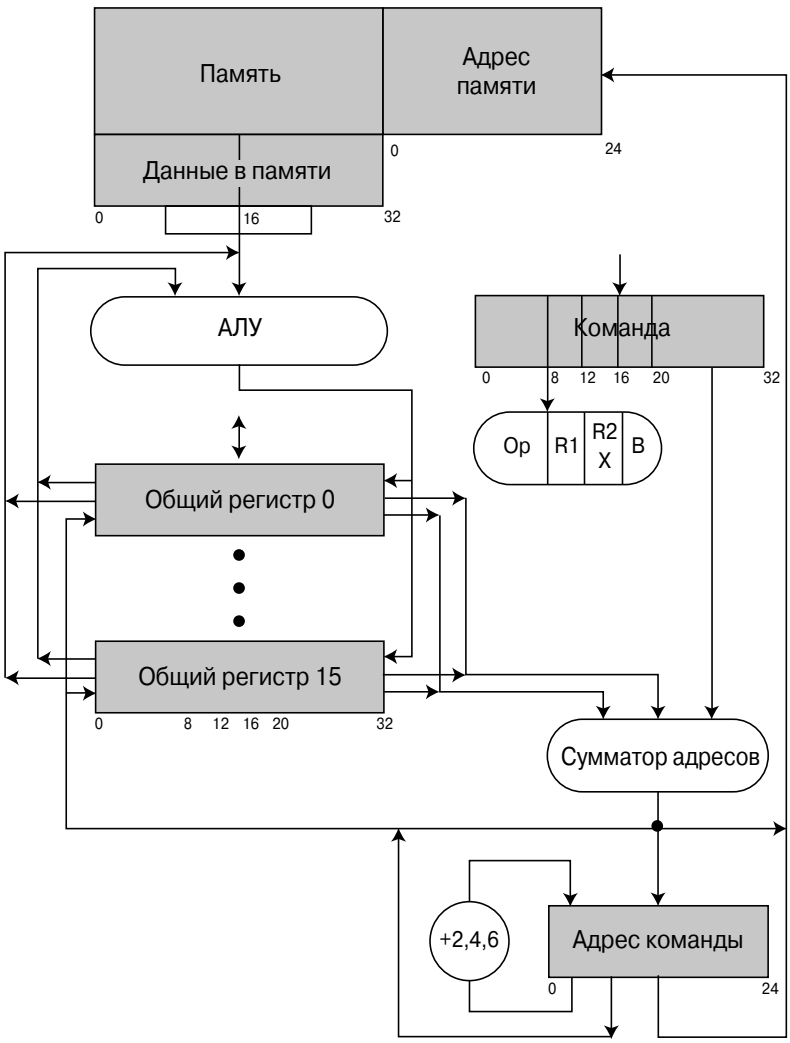
Общие усвоенные уроки

1. Кухня действительно является самым важным помещением в доме и заслуживает того, чтобы ее проектированию было посвящено много усилий.
2. Прошло уже четырнадцать лет со времени реализации нашего проекта кухни, но до сих пор всплывают лишь незначительные детали, которые можно было бы сделать иначе в этом проекте. Этот удачный результат отчасти достигнут благодаря тому, что все работы осуществлялись на компьютере с операционной системой Linux, а все проектировщики сами были пользователями, поэтому отработанные сценарии практического применения оказались реалистичными и представительными.
3. Другими важными факторами являются затраты времени и усилий на проектирование. Как и при проектировании архитектуры операционной системы OS/360 (глава 24), в нашем распоряжении было много времени. При создании программного обеспечения основным направлением использования дополнительного времени проектирования является проверка прототипов с участием реальных пользователей, а мы затрачивали время главным образом на проверку псевдопротипов в виде макетов и моделей в виртуальной среде на широком наборе сценариев практического использования.
4. Я убежден, что в большинстве проектов необходимо отводить более значительную часть общего графика на саму разработку.
5. Мы очень широко обсуждали все проблемы проектирования со своими друзьями, что привело к появлению чрезвычайно продуктивных идей, в том числе касающихся базовой конфигурации.
6. Неоценимую помощь оказало применение макетов в натуральную величину, на которых проверялись сценарии использования.
7. Технология моделирования в виртуальной среде позволила получить важную информацию, дополняющую те сведения, которые были получены с помощью поэтажных планов и даже макетов, особенно если дело касалось визуально воспринимаемого пространства и общего восприятия комнат.
8. Причем с точки зрения практики моделирование в виртуальной среде обходится дешевле и осуществляется проще. Это не относится к макетам. Поэтому ключевым становится не вопрос: “Позволяет ли моделирование в виртуальной среде получить результаты, не достижимые с помощью макетов?”, а вопрос: “Позволяют ли макеты получить ключевые результаты, не достижимые с помощью моделирования в виртуальной среде?”

9. Не только мой опыт проектирования пространственных объектов, но и научные результаты нашей лаборатории виртуальных сред подтверждают, что положительным является ответ на второй вопрос. Инско (Insko) пришел к выводу, что применение дополнительных макетов из пенополистирола, доступных для восприятия с помощью органов чувств (пусть даже они были видны лишь в составе других изображений в модели), в процессе моделирования в виртуальной среде способствовало значительному улучшению эффекта присутствия[2]. Участники экспериментов, которые прошли обучение в виртуальной среде, включающей осязаемые макеты наряду с визуальными изображениями, проходили по реальному лабиринту (вслепую) значительно быстрее и с существенно меньшим количеством ошибок, чем другие участники экспериментов, которые прошли обучение в той же виртуальной среде, но без макетов, лишь с визуальными изображениями.
10. Поэтому я полагаю, что применение макетов при проверке сценариев использования сохраняет свою значимость при проектировании интенсивно используемых помещений, таких как кухни, или помещений, которые будут создаваться в виде многочисленных копий, таких как офисы в здании с многими одинаковыми офисами, несмотря на то, что на создание макетов приходится затрачивать большие усилия и много денег.

Примечания и ссылки

1. См. [7].
2. См. [127]; [220].


Память
Адрес памяти
0
24
Данные в памяти
0
16
32
АЛУ
Команда
0
8
12
16
20
32
Op
R1
R2
X
B
Общий регистр 0
Общий регистр 15
0
8
12
16
20
32
Сумматор адресов
+2,4,6
Адрес команды
0
24

24

Изучение практического примера. Архитектура System/360

Ставка IBM в азартной игре на 5 млрд. долл. США

См. [226]

[Мэйнфрейм IBM System/360 и совместимые с ним последователи]... были рабочими лошадками в компьютерной отрасли в течение столь долгого времени и продолжают ими оставаться.

См. [17]

◂ *Основная модель программирования IBM System/360. См. [25], рис. 12.78*

Основные факты и особенности

Решение в наиболее чистом виде. Прекратить всю дальнейшую разработку каждой из шести существующих линеек продуктов IBM в пользу одной линейки продуктов, пусть даже это связано с риском перехода представителей существующей базы клиентов на компьютеры конкурентов, совместимые с имеющимися линейками продуктов IBM. Само собой разумеется, что это решение было принято главным администратором, Томасом Дж. Уотсоном-младшим (Thomas J. Watson, Jr).

Решение в чистом виде. Создать новую линейку продуктов из шести компьютеров, строго совместимую снизу вверх и сверху вниз, исключительно с одной архитектурой. Эта инициатива исходила от Дональда Сполдинга (Donald Spaulding), а Боб О. Эванс (Bob O. Evans) утвердил ее как решение.

Решение в чистом виде. Архитектура должна быть основана на восьмибитовом байте, в результате чего становятся устаревшими все существующие устройства ввода-вывода и вспомогательные устройства, даже предназначенные для обработки перфокарт.

Общие сведения и предыстория

Владелец

Корпорация IBM.

Проектировщики

Джин Амдал (Gene Amdahl), руководитель разработки архитектуры; Джерри Блау (Gerry Blaauw), заместитель руководителя разработки архитектуры и автор руководств; Ричард Кэйс (Richard Case), Джордж Гровер (George Grover), Уильям Хармс (William Harms), Дерек Эндерсон (Derek Henderson), Пол Эрвитц (Paul Herwitz), Грэм Джонс (Graham Jones), Андрис Падегс (Andris Padegs), Энтони Пикок (Anthony Peacock), Дэвид Рейд (David Reid), Уильям Стевенс (William Stevens), Уильям Райт (William Wright); Фредерик Брукс (Frederick Brooks), руководитель проекта

Даты

1961–1964 гг.

Предыстория

Трудно найти такие архитектуры вычислительных систем, обоснования которых бы так тщательно обсуждались, как семейство продуктов IBM System/360. В разделе “Примечания и ссылки” приведены некоторые из наиболее важных

обсуждений таких обоснований[1]. Поэтому в настоящей главе, которая представляет собой описание практического примера, затрагиваются только наиболее интересные моменты.

В 1960 г. стало ясно, что линейка компьютерных продуктов IBM второго поколения (в основе которой лежала дискретно-транзисторная технология) исчерпала архитектурные возможности (главным образом возможности адресации памяти). Существующие линейки продуктов IBM были несовместимыми, причем каждый из этих продуктов имел собственную программную и рыночную поддержку, как указано ниже.

- Компьютер модели IBM 650 (первое поколение, на электронных лампах) и несовместимая с ним последующая модель, выполненная на транзисторах, 1620.
- Компьютер модели IBM 1401 и несовместимая с ним последующая модель, 1410.
- Компьютер модели IBM 7070-7074.
- Компьютер модели IBM 702-705-7080.
- Компьютер модели IBM 701-704-709-7090.
- Компьютер модели IBM 7030 (компьютер Stretch, девять копий, дальнейшее распространение на рынке было прекращено).

Среди этих моделей первые две вошли в число примерно двух третей из общего количества проданных компьютеров; ответственность за их выпуск была возложена на отделение GPD (General Products Division); за остальные модели отвечало отделение DSD (Data Systems Division). Модели 1410 и 7070 непосредственно конкурировали друг с другом, как и модели 7080 и 7074. Эти разные линейки продуктов представляли весьма различные архитектурные подходы, и в их основе лежали разные основные решения.

Отделением DSD в 1959 г. была начата разработка новой линейки продуктов, так называемого "ряда 8000", основанной на втором поколении дискретной транзисторной технологии, что отражало архитектурный подход к разработке компьютера Stretch. Эта линейка продуктов была предназначена для замены моделей 7074, 7080, 7090 и Stretch. Была начата разработка этих технических моделей, и четыре модели ряда 8000 прошли этап оценки стоимости "нулевого уровня", рыночный прогноз и формирование цены к январю 1961 г. Ключевым направлением рыночного прогноза была оценка применимости для нового класса приложений, основанных на компьютеризированной телефонной связи.

В первой половине 1961 г. в отделении DSD развернулись яростные дискуссии среди разработчиков по поводу того, следует ли немедленно приступать к созданию ряда 8000, на чем я ошибочно настаивал, или подождать три года и приступить к проектированию новой линийки продуктов, которая должна быть основана

на новой перспективной технологии интегральных схем. В конечном итоге был принят последний вариант, на стороне которого выступал Боб О. Эванс. Все работы по созданию моделей ряда 8000 были остановлены и в июне в отделении DSD началась разработка новой линейки продуктов на базе интегральных схем. Эванс назначал меня ответственным за эту разработку, что для меня стало весьма неожиданным решением со стороны этого человека со столь высоким уровнем мышления.

Тем временем технический штатный сотрудник корпорации Дональд Сполдинг пришел к глубокому убеждению, что для IBM требуется унифицированная, охватывающая всю корпорацию новая линейка продуктов, а не исключительно одна лишь линейка DSD, которая охватывала только верхнюю половину рынка. Он убедил вице-президента Т. В. Лирсона (T. V. Learson), который возглавлял комитет по общей стратегии корпорации (комитет SPREAD), что нужно разработать такой план. В комитете огромным авторитетом пользовался внимательно следящий за всеми событиями Джон Хаанстра (John Haanstra), технический вице-президент отделения GPD, от которого можно было ожидать, что он будет наиболее энергично выступать против любых попыток ограничить автономию его отделения, особенно в связи с тем, что линейка продуктов 1401 этого отделения оказалась очень успешной (это был первый компьютер, проданный в количестве больше 10 000 экземпляров). Но комитет SPREAD подготовил к концу 1961 г. свой отчет, после чего Комитет управления корпорацией принял рекомендованную в нем новую линейку продуктов как идущую на смену всем существующим линейкам продуктов[2]. Этот потрясающе смелый шаг в дальнейшем был назван журналом *Fortune* "ставкой IBM в азартной игре на 5 млрд. долл. США" [3]. А Эванс отозвался об этом решении как о "ставке на всю компанию". Я был назначен корпоративным исполнительным руководителем, отвечающим за координацию всей деятельности по разработке. К счастью, кроме этих административных обязанностей в рамках всей корпорации, на меня была возложена ответственность за изучение требований рынка и контроль разработки архитектуры по всему проекту, а также ответственность за разработку вычислительных средств в отделении DSD и всю деятельность по программированию. Административная ответственность в большей степени требует подписания различных бумаг, а обязанности руководителя в основном касаются денег и людей.

В отчете SPREAD было указано, что вначале должны быть разработаны шесть компьютеров, что эти вычислительные устройства должны быть предельно дешевыми и что в течение нескольких последующих лет должен быть создан суперкомпьютер. Первым шести моделям были присвоены номера 30, 40, 50, 60, 64 и 70, а последним двум — номера 20 и 90. За разработку моделей 20 и 30 отвечало отделение GPD, а остальные вошли в сферу обязанностей отделения DSD.

Задачи

Основные цели

- Создание архитектуры вычислительной системы, строго совместимой снизу вверх и сверху вниз, на уровне двоичных кодов.
- Компьютеры должны быть пригодными и конкурентоспособными в области обработки бизнес-данных, научных и технических вычислений, а также для телеобработки.
- Возможности новых приложений должны быть расширены так, чтобы корпорация IBM имела постоянно растущий объем сбыта в денежном выражении, даже после падения цен на компьютеры в два раза. Мы не могли рассчитывать на то, что доля IBM на рынке существующих приложений значительно увеличится или что объемы использования этих приложений быстро удвоятся.
- Каждая модель должна быть рентабельной (конкурентоспособной) на своем собственном рынке, начиная с очень дешевых моделей и кончая самым быстрым суперкомпьютером.

Другие важные цели

- Разработка новой единой полной программной поддержки, которая опирается на совместимость на уровне двоичных кодов, для обеспечения создания общей развитой системы, заменяющей множество неполноценных систем второго поколения. Эта программная поддержка должна включать новую операционную систему, реализующую быстроразвивающиеся концепции, которая основана на опыте создания операционных систем для компьютеров второго поколения.
- Поиск способов оказания помощи клиентам в переходе на System/360 со своих систем второго поколения, включая даже системы компьютеров-последователей конкурентов, которые продолжили создание машин, совместимых с теми, которые относились к линейкам продуктов IBM, выпуск которых был прекращен.
- Предоставление архитектуры, которая иногда может быть реализована в виде аппаратной технологии, соответствующей требованиям отделения FSD (Federal Systems Division) корпорации IBM, занимающегося поставками для вооруженных сил и правительственных учреждений (таких как NASA).
- Достижение новых уровней надежности и удобства в эксплуатации, включая сверхнадежные мультипроцессорные системы.

Открывающиеся возможности по состоянию на июнь 1961 г.

Необходимость в создании новой архитектуры. Запоминающие устройства на магнитных сердечниках показали себя чрезвычайно надежными, а их стоимость существенно снизилась. Вследствие этого все клиенты стали выражать пожелания в увеличении объема памяти. Но во всех существующих линейках продуктов возможности адресации были исчерпаны, поэтому назрела необходимость в осуществлении одного или нескольких крупных пересмотров архитектуры. Это дало нам возможность применить обширный опыт, полученный по результатам эксплуатации и по отзывам людей, эксплуатирующих компьютеры первого и второго поколений. Этот опыт было бы труднее воплотить в рамках старой архитектуры.

Новая, более дешевая технология. Технологическое отделение IBM занималось интенсивной разработкой тематики интегральных микросхем, в результате чего в 1964 г. было создано важное подразделение SLT (Solid Logic Technology), готовое к массовому выпуску таких микросхем. При этом открывались возможности уменьшения стоимости элементной базы для компьютеров любой заданной сложности примерно в два раза; наряду с этим уменьшались размеры, снижалось энергопотребление и повышалась надежность. Достигаемое при этом резкое повышение показателя "производительность/стоимость" должно было стать достаточным стимулом для клиентов, чтобы они согласились пройти болезненный и дорогостоящий процесс преобразования используемого программного обеспечения с переходом на новую несовместимую систему.

Большой запас времени на проектирование. Сроки реализации новой технологии должны были наступить еще не скоро, а из этого следовало, что на этот раз системные архитекторы будут иметь в своем распоряжении больше времени (почти два года), чтобы в полной мере и тщательно выполнить свою работу.

Устройства ввода-вывода нового типа. Технология жестких дисков с произвольным доступом быстро прогрессировала, а это позволяло реализовать полностью новый подход к обработке данных и радикально пересмотреть принципы создания операционных систем.

Новые возможности телеобработки. Технология компьютерной связи, первоначально разработанная для ПВО, приобретала все большую привлекательность для коммерческих приложений, причем вначале нашла свое применение в системах резервирования билетов на авиалиниях.

Сложности и ограничения

Совместимость — размер адреса. Безусловно, наибольшие технические сложности заключались в необходимости строгой (на уровне двоичных кодов) совместимости снизу вверх и сверху вниз, что позволяло бы добиваться превосходства

в конкуренции компьютеров каждого класса на своем рынке с идущими с ними вровень продуктами конкурентов. Как выбрать архитектуру, обеспечивающую выпуск по небольшим ценам компьютеров самого низкого класса, но чрезмерно не ограничивающую возможности создания суперкомпьютера? Как обеспечить сверхбыстродействие суперкомпьютера, не применяя при этом элементную базу, которая бы оказалась слишком дорогостоящей для дешевых компьютеров? Принципиальная задача состояла в выборе размера адреса. Для компьютеров высшего класса из этой линейки требовалось большее количество адресных битов; но имело ли смысл предусматривать для относящихся к той же линейке компьютеров самого низкого класса (выпускаемых серийно) такое же количество битов адресации памяти и сталкиваться с опасностью снижения производительности в связи с наличием неиспользуемых адресных битов?

Совместимость — набор операций. Как обеспечить выполнение сложных математических операций, таких как вычисления с плавающей запятой для научных приложений, наряду с операциями со строками символов для обработки бизнес-данных, не подвергаясь опасности ненужного удорожания компьютеров?

Расширение сферы применения. Третья важная сложность, которая должна была быть преодолена, состояла в необходимости создания чрезвычайно разнообразных систем, необходимых для новых приложений (особенно для терминалов связи и удаленных терминалов), для систем с большой вычислительной нагрузкой и для систем, предназначенных для обработки значительных объемов данных.

Переход от существующих систем к новым системам. Мы знали, что переход от систем второго поколения к разрабатываемым системам будет связан с колоссальными сложностями, но в течение первого года проектирования не теряли время на размышления об этом.

Наиболее значительные проектные решения

Восьмибитовый байт. Было принято решение перейти к использованию байтов длиной 8 битов, в отличие от шестибитового байта, применением которого характеризовались все компьютеры первого и второго поколений (за исключением Stretch). Это решение было самым важным и наиболее горячо обсуждаемым. Переход к применению восьмибитового байта имел много следствий: достижение необходимой точности чисел с плавающей запятой требовало применения 48-битовых слов и 96-битовых двойных слов, следовательно, байты должны быть 6-битовыми. Длина команды, равная 24 битам, была слишком мала, а равная 48 — слишком велика. А что, если потребуется введение в поддерживаемый алфавит строчных букв, которые почти совсем не применялись в более ранних моделях компьютеров?

Для меня оказался самым убедительным аргумент, что в будущем в приложениях нельзя будет обойтись без применения строчных букв. Мы остановились на восьмибитовых байтах, 32-битовых словах данных и одноадресных командах, а также на 32- и 64-битовых словах с плавающей запятой.

Неудачные первые попытки разработки архитектуры стека. Атаку на проблему выбора длины адреса мы начинали с разработки архитектуры стека. Посвятив решению этой задачи шесть месяцев, мы обнаружили, что предложенная архитектура превосходно работает на компьютерах среднего и высокого классов, но ее применение приводит к резкому снижению производительности на компьютерах низкого класса, поскольку стек должен быть реализован в оперативной памяти, а не в регистрах.

Соревнование между проектировщиками. После провала первых попыток создания архитектуры стека Амдал предложил провести внутрифирменное соревнование по проектированию. Его идея сработала блестяще: коллектив Амдала и коллектив Блау, независимо друг от друга, придумали решение с базовым регистром, позволяющее решить задачу определения размера адреса. Таким образом, было принято следующее решение.

Адрес должен быть 24-битовым. Мы неохотно согласились принять этот размер, обеспечивающий адресацию до байта. Мы знали, и я открыто предсказал это в 1965 г., что в какой-то момент архитектура должна быть переведена на адресацию с применением 32 битов, но не могли позволить себе принять такое решение в 1964 г.[4] Безусловно, мы постарались предусмотреть различные меры предосторожности для перехода в дальнейшем на 32-битовый адрес, но, к сожалению, подпрограмма вызова инструкции перехода и ветвления (Branch and Link) была без долгих размышлений спроектирована так, что в ней использовались верхние 8 битов адреса, который нужно было бы оставить нетронутыми.

Это — наглядный пример того, насколько опасным может быть коллективное проектирование. Мне не удалось в достаточной степени довести до всего коллектива наши взгляды на необходимость расширения архитектуры в будущем; кроме того, эта ошибка не была замечена ни в одной из проверок.

Стандартный интерфейс ввода-вывода. Чтобы обеспечить применение широкого разнообразия специализированных прикладных систем, мы спроектировали стандартный логический, электрический и механический интерфейс для подключения всех устройств ввода-вывода, по аналогии с тем, что Бухгольц (Buchholz) впервые осуществил при проектировании компьютера Stretch. Это решение способствовало существенному сокращению затрат на настройку и программное обеспечение, а также упрощению инженерной разработки устройств ввода-вывода и блоков управления.

Условия супервизорного управления. Был спроектирован тщательно продуманный набор возможностей супервизора, поэтому всеми системами компьютера можно было управлять с помощью операционной системы, без физического вмешательства оператора. В состав этих средств входили система прерывания, защита памяти, привилегированный режим команд и таймер.

Обнаружение одиночной ошибки. Для всех реализаций S/360 обязательным было обеспечение полного сквозного обнаружения одиночных ошибок, несмотря на то, что клиенты явно выражали нежелание за это платить. Но такая

возможность существенно помогала в достижении постоянной надежности и удобства в эксплуатации.

Контроль ошибок предусматривался в коммерческих компьютерах, предназначенных для обработки данных, всех изготовителей, начиная с компьютера UNIVAC, который обеспечивал тщательную проверку. А в компьютерах научного назначения, начиная с тех, которые были описаны в статье Беркса, Голдстайна и фон Неймана (Burks, Goldstine, von Neumann), такая проверка не была предусмотрена. Безусловно, это выглядело как нелепость, ведь аппаратная ошибка, допущенная в вычислениях при моделировании атомного взрыва, гораздо опаснее по сравнению с ошибкой в счете на коммунальные услуги. Я думаю, что такое различие в подходах к обнаружению ошибок обусловлено тем, что представители науки всегда сами включают в свои программы глобальные проверки, например, основанные на законе сохранения энергии.

Мы заметили, что к 1961 г. люди начали доверять ответам, полученным от компьютеров, поэтому для нас стало задачей профессиональной ответственности и делом чести — включить аппаратную проверку (надеясь на то, что связанное с этим повышение цены не приведет к обвалу продаж наших компьютеров).

Десятичная арифметика. Чтобы стимулировать переход пользователей на огромном рынке средств обработки данных на наши компьютеры и упростить их обучение, мы решили включить не только двоичную, но и десятичную арифметику. (Вся адресация была двоичной, в отличие от выпускавшихся ранее систем 650, 1401, 1410, 7070, 7074 и 7080.)

Предусмотрев возможность применения десятичного типа данных, мы, по-видимому, допустили ошибку; мы вместо этого должны были заметить, что в COBOL и других языках проблема представления десятичных данных, например в денежных расчетах, была решена путем хранения денежных сумм с применением целочисленных значений в разменной монете (в центах, пенни, пфеннигах и т.д.), что позволяло предотвратить ошибки, связанные с преобразованием и выполнением других операций с дробными значениями. Насколько наш отказ от применения десятичного типа данных повлиял бы отрицательно на маркетинг, остается только гадать. Введение десятичного типа данных не привело к существенному удорожанию оборудования, но стоимость программного обеспечения возросла и увеличилась концептуальная сложность.

Многопроцессорная обработка. Было сделано все необходимое для возможности ввода в конфигурацию единственной системы нескольких процессоров, для обеспечения того, чтобы отказ одного процессора не приводил к нарушению работоспособности всей системы.

Микропрограммная реализация. В отчете SPREAD мы потребовали, чтобы применялась микропрограммная реализация системы команд; единственная возможность отказа от этого требования обусловливалась тем, что какой-то технический руководитель сумеет показать 33%-ное преимущество обычной логики по производительности/стоимости над этой реализацией. Благодаря этому

обеспечивалась возможность предусмотреть довольно широкий единообразный набор операций для всех процессоров, включая процессоры низкого класса, причем различия по стоимости процессоров разного класса определялись лишь объемом управляющей памяти. Разработка моделей 60 и 64 началась с применения обычной логики, а затем в ходе разработки произошел переход от этих моделей к единственной модели 65 с микропрограммной реализацией. В моделях 75 и 91 использовалась обычная логика.

Эмуляция более ранней архитектуры. Стюарт Такер (Stewart Tucker) заметил, что слово памяти и информационного канала в реализации модели 65 с длиной 32 бита и 4 битами контроля четности позволяло корректно представить слово с длиной 36 и без контроля четности модели 7090. Он изобрел эмулятор 7090 на микрокоде, в котором весьма эффективно использовались информационные каналы модели 65. Эта разработка привела к настоящему прорыву в решении задачи перехода клиентов с моделей 7090, 7074 и 7080 на новые модели[5].

В январе 1964 г. наступил критический момент, когда Уильям Хармс (William Harms), Джеральд Оттоуей (Gerald Ottoway) и Уильям Райт (William Wright) в очень короткие сроки изобрели микропрограммную эмуляцию 1401 на модели 30. Это стало решающим достижением по обеспечению перехода на новые модели самого большого массива клиентов.

Отказ от виртуальной памяти. В то время, как мы работали над определением архитектуры S/360, была изобретена виртуальная память и впервые реализована на компьютере Cambridge Atlas, после чего были разработаны использующие ее операционные системы в Кембриджском университете, Массачусетсском технологическом институте (MIT) и в Мичиганском университете. Мы проводили долгие и напряженные дискуссии о том, следует ли перейти к использованию виртуальной памяти и в нашем проекте. Было решено отказаться от этой идеи по соображениям производительности. Это была ошибка, которую удалось исправить в первом же из последующих поколений — System/370.

Новые устройства ввода-вывода с произвольным доступом. В этом проекте значительные усилия разработчиков были сосредоточены на создании новых устройств на магнитных барабанах для размещения операционной системы, а также на организации новых файловых систем на жестких дисках. Мы рассматривали эти усилия как чрезвычайно значимые с точки зрения создания новых приложений и обеспечения разнообразия конфигураций системы. Кроме того, были разработаны новые одно- и многоканальные контроллеры связи.

Каналы ввода-вывода. Ввод-вывод осуществлялся с помощью независимо действующих каналов, которые по существу представляли собой специализированные модули с хранимой в памяти программой; одни из этих модулей были оптимизированы для быстрой поблочной передачи данных, а другие — для мультиплексной передачи данных по каналам связи в количестве до 256.

Наиболее важные этапы

Лето 1961 г. В отделении DSD началась работа над архитектурой новой линейки продуктов. Амдал, Боэм и Кок (Amdahl, Boehm, Cocke) из лаборатории IBM Research присоединились к коллективу разработчиков архитектуры под руководством Блау, завершив работу над рядом 8000. Началась работа по реализации подхода к созданию стека.

Январь 1962 г. Организована работа над новыми моделями в рамках всей корпорации.

Весна 1962 г. Первые оценки производительности показали, что принятая архитектура стека является неконкурентоспособной. Проведенное соревнование между проектировщиками позволило сформулировать идею адресации с использованием базового регистра.

Лето 1962 г. Проведены дискуссии по определению размера байта.

Осень 1962 г. Подготовлен первый проект руководства по архитектуре.

Осень 1963 г. Руководство по архитектуре окончательно завершено.

Январь 1964 г. Развернулось серьезное соревнование между продуктами, моделью 30 семейства S/360 и моделью 1401S, разработанной отделением GPD, которая явилась в шесть раз более быстродействующим последователем модели 1401 и была разработана Хаанстрой (Haanstra), возглавившим теперь это отделение. Система S/360 стала победителем благодаря изобретению эмуляции 1401 для модели 30.

Апрель 1964 г. Опубликовано объявление о выпуске моделей 30, 40, 50, 65 и 75, с намеком на то, что следующей моделью будет модель 90.

Февраль 1965 г. Произведена поставка первого компьютера системы S/360 (модель 40).

Август 1972 г. Объявлено о создании виртуальной памяти для системы System/370.

Начало 1980 г. Объявлено о разработке 31-битовой архитектуры XA для System/370.

2000 г. Объявлено о разработке 64-битовой архитектуры ряда z[6].

Оценка

Устойчивость

Одним из синонимов определения устойчивости архитектуры вычислительной системы является ее долговременное существование. Я предсказывал, что эта архитектура сохранится в различных реализациях в течение 25 лет, с учетом модификаций, обеспечивающих применение адресов большей длины[4]. Прошло уже 45 лет после выхода первого объявления о выпуске системы S/360, но архитектура, постоянно совершенствуемая, продолжает применяться. Одной из недавних реализаций является система IBM z/90, о выпуске которой объявлено в

марте 2007 г. Эта система по-прежнему является обратно совместимой; на ней все еще выполняются программы системы S/360. Эти компьютеры, теперь именуемые мэйнфреймами, продолжают выполнять во всем мире большую часть работы по поддержке баз данных; на них применяется операционная система VM/360, потомок MVS/360, или, все более чаще, операционная система Linux.

Еще одним синонимом устойчивости можно считать признание в своей отрасли. Гордон Белл (Gordon Bell), который сам является крупным компьютерным архитектором, много сделавшим для корпорации DEC, недавно назвал System/360 компьютером, получившим наибольшее признание на протяжении всей истории развития вычислительной техники, подразумевая при этом признание среди специалистов, а не только присутствие на рынке, который в свое время ловко завоевал персональный компьютер[7]. Реализованный в системе S/360 переход на восьмибитовый байт привел к изменению компьютерной архитектуры полностью и окончательно. Кроме того, в системе S/360 большое внимание было уделено развитию конфигураций ввода-вывода на основе жестких дисков, что также привело к радикальному изменению системного проектирования[8].

Джин Амдал (Gene Amdahl) лицензировал архитектуру S/360 и весьма точно реализовал ее в чрезвычайно успешном семействе компьютеров Amdahl Corporation. Эта архитектура была лицензирована в RCA и использовалась в семействе Spectra 70 этой корпорации. Безусловно, в RCA была бережно реализована вся архитектура, относящаяся к режиму решения задач, но архитекторы этой корпорации все же решили реализовать особую версию архитектуры режима супервизора. Версия RCA была лицензирована, затем широко использовалась компаниями Siemens, Fujitsu и Hitachi, а также была скопирована советскими разработчиками.

Архитектура S/360 явно оказала влияние на семейство VAX корпорации DEC, а также на семейство компьютеров PDP-11, включая его многочисленных потомков — микрокомпьютеры типа Motorola 6800, 68000.

Полезность — конкурентоспособность на любом рынке

С коммерческой точки зрения рискованная ставка на создание System/360 привела к большому успеху. Годовые отчеты IBM показали средний ежегодный рост доходов на 21% в период с 1964 по 1968 гг., а также средний рост прибыли на 20% с 1964 по 1968 гг.

К 7 апреля 1964 г. было объявлено примерно о 144 новых продуктах. Многие из них характеризовались применением различных вариантов оперативной памяти. Но наибольшее впечатление произвело появление широкого набора восьмибитовых устройств ввода-вывода: многочисленных принтеров, включая принтеры с переменными наборами символов; многих типов дисков, в том числе со сменными картриджами; новых систем накопителей на магнитной ленте; целого спектра терминалов связи и сетевых устройств; новых перфораторов для перфокарт,

устройств чтения и специализированных принтеров; а также других разнообразных устройств, таких как аппараты для сортировки чеков и терминалы ввода производственных данных. Богатство этой коллекции, разработанной во многих лабораториях, разбросанных по всему миру, обеспечивало почти бесконечное разнообразие и широкие масштабы применения различных конфигураций системы. Наличие стандартного интерфейса ввода-вывода и его программной поддержки означало, что задача наращивания и изменения конфигурации становится проще. Совместимость процессоров позволяла проводить обновление компьютеров со сменой конфигурации, зависящей от процессора, с переходом на новую модель за считанные дни, не изменяя конфигурацию ввода-вывода или программное обеспечение.

Все модели получили признание на тех рынках, для которых они были предназначены. Быстрее всего был достигнут успех в реализации модели 30 благодаря применению в ней жестких дисков и принтеров. Появившаяся вскоре после нее модель 20, совместимая снизу вверх, получила очень широкое распространение.

Основной успех в распространении модели 65 был достигнут главным образом благодаря возможности перехода на нее пользователей приложений, которые перед этим эксплуатировались на моделях 7090, 7094, 7094 II, 7080, 7074 и др.

Кроме того, модель 65 хорошо подходила для поддержки новых методов работы с базами данных, поэтому она и ее потомки стали доминирующими в этой области, и хорошо показала себя при выполнении инженерных расчетов. Наиболее серьезными конкурентами оказались компании, которые сами производили компьютеры с архитектурой S/360, так называемые *аналоги с совместимостью до разъемов*, в которых обычно эксплуатировалось программное обеспечение OS/360.

Модели 75, 91 в этом семействе были спроектированы как научные суперкомпьютеры. После их появления рынок научных суперкомпьютеров был поделен почти поровну с современными компьютерами CDC и Cray, но во второй половине доминировали потомки семейства Cray. Четыре компьютера модели 75s применялись при наземных вычислениях для программы "Apollo", а на космических аппаратах в качестве бортовых компьютеров применялись более защищенные производные семейства System/360.

Положительные итоги

Исходная архитектура была довольно прозрачной, поскольку было тщательно продумано концептуальное разделение архитектуры, реализации и технологического осуществления[9]. Требование к обеспечению строгой совместимости снизу вверх и сверху вниз диктовало необходимость соблюдения жесткой дисциплины, которая защищала модели низкого класса от нехватки функциональных средств, а модели высокого класса — от их избытка. (Аналогично, любой писатель знает, что соблюдение строгого лимита на количество страниц часто приводит к созданию более лаконичных и впечатляющих произведений.) Блау оставил в списке рабочих команд хорошо продуманные места для заполнения в будущем новым

кодом. Разумеется, в дальнейшем были внесены некоторые дополнения, но в результате набор рабочих команд стал менее упорядоченным по сравнению с первоначальным.

С технической точки зрения нашей самой большой ошибкой был отказ от внедрения с самого начала виртуальной памяти. В этом состоит один из ярких примеров того, что неправильный путь могут выбрать даже опытные проектировщики (см. главу 14).

Еще одной нашей значительной принципиальной и концептуальной ошибкой была неспособность понять, что канал ввода-вывода по существу представляет собой отдельный компьютер. Превосходным воплощением этой изящной и мощной концепции оказались периферийные процессоры Cray, впервые введенные в модели CDC 6600. Каждый из многочисленных параллельных потоков ввода-вывода управляется отделенным от прочей архитектуры небольшим и простым двоичным компьютером, а каждый из этих компьютеров реализован с расчетом на обработку одного потока данных с разделением по времени.

Самым неудачным решением в оригинальной архитектуре процессора оказалось применение формата команд SS, в котором был предусмотрен базовый регистр, но не отдельный индексный регистр, как во всех других форматах. Кроме того, как отмечалось выше, в инструкции перехода и ветвления использовались старшие адресные биты, которые следовало бы зарезервировать для расширения

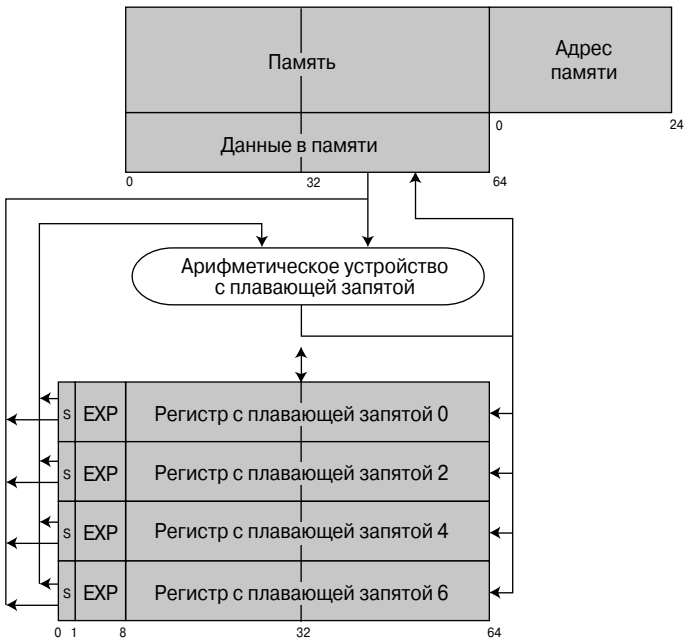


Рис. 24.1. Поток данных с плавающей запятой в системе System/360. См. [25], рис. 12.79

с переходом к 32-битовым адресам. При выполнении инструкции загрузки адреса те же старшие биты очищались.

Менее серьезной ошибкой было то, что мы с самого начала не предусмотрели защитный разряд в определениях операций с плавающей запятой. Для исправления этой ошибки нам пришлось дорабатывать на месте первые компьютеры S/360 после поставки.

По-видимому, наиболее ярким критическим замечанием по архитектуре, созданной в результате наших усилий, была шутка, согласно которой система S/360 фактически представляет три архитектуры под одной крышей: базовый 32-битовый двоичный компьютер, 64-битовый компьютер с плавающей запятой совсем с другим потоком данных и побайтовый процессор с полностью отличным потоком данных и даже с десятичной арифметикой (см. фронтиспис главы, рис. 24.1 и 24.2). В действительности, если добавить селекторные и мультиплексные каналы, можно насчитать даже пять архитектур. Но все эти компоненты нормально работают благодаря реализации с применением микрокодов.

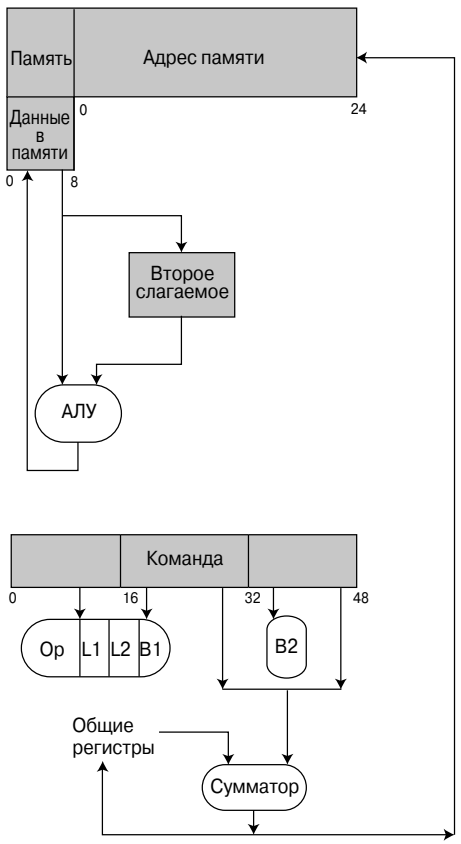


Рис. 24.2. Побайтовый поток данных в системе System/360. См. [25], рис. 12.80

Так или иначе, даже при наличии этих многочисленных параллельных архитектур, удалось создать семейство компьютеров, в полном смысле соответствующих определению универсальных, которые могут адаптироваться к конкретным условиям с применением подходящего процессора, памяти и особенно конфигурации ввода-вывода, а также могут применяться в приложениях всех разновидностей, с любыми потребностями в производительности.

Общие усвоенные уроки

1. Целесообразность предоставления проектировщикам большого количества времени на проектирование. Если проектировщики имеют в своем распоряжении много времени, то их продукт становится гораздо лучше и служит дольше; при этом могут даже сократиться сроки окончательной поставки, поскольку сокращается количество переделок.
2. Если одна и та же архитектура имеет множественные параллельные реализации, то обеспечивается строгая защита этой архитектуры от плохих компромиссов, поскольку всегда можно обнаружить, что некоторая реализация (обычно непредумышленно) отступает от основной архитектуры. При наличии только одной реализации всегда проще, дешевле и быстрее внести изменения в руководство, чем в сам компьютер. В главе 6 книги [40] более подробно рассматривается этот и другие методы обеспечения того, чтобы реализация соответствовала архитектуре (а не наоборот).
3. Предложенное Амдалом соревнование проектировщиков в нашем первом проекте, оказавшемся на мели, было весьма плодотворным. В этом соревновании по многим проблемам выявилось совпадение взглядов, а критически важные различия быстро вышли на передний план. Более того, это соревнование оказало мощный положительный эффект на моральный дух коллектива. В 2008 г., после того как прошло более сорока лет, со мной поделился своими впечатлениями Дуг Бэрд (Doug Baird), который в то время был младшим архитектором в одной из групп. Он все еще вспоминает с благодарностью, что группа довольно молодых разработчиков, к которой он принадлежал, получила шанс выдвинуть свой проект на тех же основаниях, что и все признанные архитекторы в коллективе.
4. Что касается полностью новых проектов, то в отличие от доработки существующих продуктов необходимо с самого начала направлять часть усилий проектировщиков на разработку показателей производительности и измерения других важных свойств, а также предлагать подходящие суррогаты стоимости (такие как количество битов регистров для компьютеров третьего поколения).
5. Методология маркетинговых прогнозов предназначена для изучения рыночного потенциала дорабатываемых продуктов, а не радикальных новшеств

(с этим уроком мы ознакомились в главе 19). Проектировщики полностью новых продуктов должны с самого начала приложить усилия к тому, чтобы в их коллективе появились собственные прогнозисты, способные выдвигать новые концепции.

Примечания и ссылки

1. Самыми важными трактовками обоснования архитектуры S/360 являются следующие:
   - Статья [10].
   - Книга [24].
   - Книга [25], раздел 12.4.
   - Статья [91].
   - Доклад [126].
   - Доклад [125].
   - Выпуск *IBM Systems Journal* 3, № 2 (весь номер).
   - Книга [170].
2. См. [126].
3. См. [226].
4. См. [39].
5. См. [208]. В данном случае он не отображал одно из слов значения с плавающей запятой компьютера 7090 в одно слово системы S/360, а проводил разбиение на части.
6. См. страницу `http://en.wikipedia.org/wiki/System_370`, которая содержит превосходный обзор истории развития архитектуры и подчеркивает основные особенности основной архитектуры (в книге взята за основу версия этой страницы за декабрь 2008 г.). Не менее интересна страница `http://www.answers.com/topic/ibm-system-360` (за август 2009 г.).
7. См. [17].
8. В [18], раздел 3, с. 561–637, дана другая оценка и приведено весьма подробное обсуждение.
9. См. [25], где это важное трехстороннее различие подробно рассматривается в разделе 1.1.

TheDesignofDesign.indb 386 30.10.2012 15:58:40

25

Изучение практического примера. IBM Operating System/360

Основная сложность в процессе разработки программного обеспечения обусловлена тем фактом, что мы должны перейти от неформально обозначенной потребности, которая обнаруживается в реальном мире, к формальной модели, которая работает в компьютере.

См. [26]

◀ *Персик и вишни, метафора OS/360: большая программа контроля и несколько меньших и независимых языковых компиляторов и утилит, которые дополняют пакет поддержки OS/360. © P. Desgrieux/photocuisinet/CORBIS. Все права защищены*

TheDesignofDesign.indb 387 30.10.2012 15:58:41

Основные факты и особенности[1]

Решение в чистом виде. Разработка единого пакета программ с одной операционной системой и набором компиляторов и утилит для всего диапазона компьютеров и конфигураций ввода-вывода. Этот пакет программ может быть разработан так, чтобы он подходил для всех моделей и мог эксплуатироваться на компьютерах с различными размерами памяти и конфигурациями ввода-вывода.

Решение в чистом виде. Для размещения операционных систем должно обязательно применяться устройство с произвольным доступом.

Решение в чистом виде. Повседневная эксплуатация операционной системы не должна требовать вмешательства оператора. Операционная система должны быть спроектирована так, чтобы ее можно было эксплуатировать для текущего управления компьютерной системой без ввода команд вручную или осуществления каких-либо действий со стороны оператора. Операторы выполняют для компьютера лишь функции рук и ног, устанавливая диски, магнитные ленты, колоды перфокарт и бумагу для принтеров. Еще один вариант состоит в том, что та же операционная система может быть сконфигурирована в расчете лишь на полное управление оператором-человеком.

Решение в чистом виде. Должен быть предусмотрен многозадачный режим для параллельного безопасного выполнения заданий и программ, которые не были специально разработаны для одновременного выполнения.

Не зависящий от устройства ввод-вывод. Программы должны быть написаны для абстрактных типов данных, применяемых для ввода-вывода, называемых *методами доступа*. Типы устройств ввода-вывода, конкретные устройства и пространство на них должны автоматически распределяться при планировании работы на выполнение. Программы должны быть универсальными; например, одна и та же программа сортировки с объединением должна быть применимой для записи данных с диска на диск в одном прогоне и с магнитной ленты на магнитную ленту в другом. Должна быть предусмотрена возможность определять, следует ли сохранить вывод в файле или напечатать во время выполнения, без перехода на другую программу.

Способность выполнять задачи в масштабах реального производства. OS/360 проявила себя как операционная система, способная выполнять задачи в масштабах реального производства, предназначенная для повседневной и круглосуточной эксплуатации, обеспечивающая ведение журналов и автоматический перезапуск после сбоев. От одного поколения к другому характеристики этой операционной системы становились все лучше и лучше, поэтому ее последние версии все еще широко используются для повседневной и круглосуточной эксплуатации систем баз данных.

Дистанционная обработка. Система должна обеспечивать мощную поддержку удаленного доступа для работы с базами данных в реальном времени и выполнения пакетных заданий.

Разделение времени в простейшем варианте. Система не предназначена для интерактивного терминального программирования и отладки, поэтому не поддерживает эти виды работ достаточно эффективно.

Немного запоздалое внедрение виртуальной памяти. Первые поставляемые версии компьютеров S/360 и пакета OS/360 не поддерживали виртуальную память. Но в компьютеры и в программное обеспечение были внесены существенные изменения в первом же последующем поколении, поэтому к 1970 г. во всех версиях System/360 была предусмотрена поддержка виртуальной памяти.

Применение ассемблера наряду с языками высокого уровня. Безусловно, к 1961 г. основной объем работы по программированию выполнялся на языках высокого уровня Fortran, COBOL и RPG (Report Program Generator), но в некоторых частях проекта OS/360 еще нашло свое отражение стремление многих программистов работать на языке ассемблера. Поэтому в качестве одного из языковых пакетов был предусмотрен мощный макроассемблер, в котором были реализованы весьма разнообразные традиции использования макрокоманд в сообществах разработчиков, занимающихся научными вычислениями и обработкой коммерческих данных. Но проведенные к 1966 г. на некоторых крупных вычислительных центрах измерения показали, что из общего объема израсходованного машинного времени прикладные программы, написанные на ассемблере, занимали лишь примерно 1%.

Общие сведения и предыстория

Один раз в жизни мне довелось управлять таким крупным проектом, как создание семейства компьютеров IBM System/360. На первых порах я руководил разработкой оборудования, а затем пакета программ операционной системы OS/360. Я занимался этой работой от начала проекта в 1961 г. до середины 1965 г. О создании этого семейства компьютеров было объявлено 7 апреля 1964 г., а первая поставка произошла в феврале 1965 г. Компьютеры System/360 были определены как относящиеся к компьютерам третьего поколения и оказались первыми серийно выпускаемыми компьютерными продуктами, в которых была применена технология (полу)интегральных микросхем.

Операционные системы первого поколения были разработаны для компьютеров второго поколения, а OS/360 оказалась первым из пакетов поддержки программного обеспечения второго поколения, который был разработан для первых компьютеров третьего поколения. До сих пор такие всеобъемлющие операционные системы еще не создавались.

В компьютерах System/360 была предусмотрена строгая совместимость на уровне двоичных кодов, что позволило нам спроектировать единый пакет поддержки программного обеспечения, способный поддерживать все семейство продуктов, который позволил распределять расходы, связанные с разработкой, по всему семейству, а также разрабатывать большой объединенный рыночный прогноз для

всей линейки продуктов. Это, в свою очередь, позволяло заняться созданием пакета поддержки программного обеспечения, характеризующегося полнотой и беспрецедентно широким набором возможностей. Я не описываю пакет программ OS/360 как относящийся к прошлому, поскольку многие его прямые потомки все еще занимают ведущие роли в мире мэйнфреймов.

Термин *Operating System/360*, или *OS/360*, используется в двух смыслах, во-первых, для описания всего пакета поддержки программного обеспечения (собственно операционной системы, языковых компиляторов и утилит) и, во-вторых, для обозначения лишь самой операционной системы. Как показано на фронтисписе к данной главе, в нашем коллективе разработчиков не раз применялось сравнение всего пакета с большим персиком и несколькими вишенками. Я сам чаще всего под термином *OS/360* подразумеваю только операционную систему.

Кроме пакета поддержки OS/360, первоначально были запланированы разработка и поставка пакета базовой поддержки накопителей на магнитной ленте, включая компилятор Fortran, для систем с небольшим объемом памяти, не имеющих жестких дисков, и пакета базовой поддержки устройств ввода-вывода перфокарт. Пакет OS/360 с самого начала предназначался для систем с объемом оперативной памяти 16 Кбайт или больше. Но нам не удавалось разместить в столь малом объеме код поддержки даже крайне ограниченного набора функций, поэтому требование к минимальному объему оперативной памяти было пересмотрено и стало равным 64 Кбайт. Корпорация IBM стремилась обеспечить поддержку заказчиков, использующих малые системы, а также предоставить альтернативный вариант операционной системы в связи с задержкой выпуска OS/360. Поэтому был разработан полностью отдельный пакет поддержки, оптимизированный для меньших объемов памяти, который получил название Disk Operating System/360, или DOS/360[2]. Этот пакет также получил развитие, и его современные версии до сих пор эксплуатируются.

Семейство компьютеров System/360

В главе 24 описано, для каких рынков разрабатывалось все семейство компьютеров, и перечислены главные особенности архитектуры этих компьютеров. В семействе System/360 было реализовано беспрецедентное концептуальное новшество, заключающееся в том, что все модели (кроме самой дешевой, Model 20) разрабатывались как логически идентичные, совместимые снизу вверх и сверху вниз реализации единой архитектуры. Мы вместе с Блау дали точное определение архитектуры компьютера как совокупности свойств компьютера, видимых для программиста, не включая быстродействие[3]. В терминах разработки программного обеспечения архитектура компьютера в нашем точном ограниченном толковании эквивалентна абстрактному типу данных. Архитектура определяет наборы допустимых данных, их абстрактные представления, а также синтаксис и семантику набора операций, применимых к указанным наборам данных. В таком случае каждая реализация

компьютера представляет собой экземпляр соответствующего типа. Таковым же является каждый эмулятор, или каждое моделирующее устройство. На практике в наших первых аппаратных реализациях применялись потоки данных шириной от 8 до 64 битов, были предусмотрены разные объемы памяти, кроме того, различались характеристики быстродействия на уровне электрических схем.

Контекст, в котором разрабатывалось программное обеспечение, по состоянию на 1961 г.

Операционные системы. Между различными операционными системами первого поколения проводилось резкое различие по тому признаку, предназначались ли они для научных расчетов или для обработки коммерческих данных. Это были пакетные операционные системы, разработанные в целях управления последовательной обработкой потока независимых заданий.

В каждой из этих операционных систем были предусмотрены три компонента, которые развивались отдельно друг от друга. Супервизор, который представляет собой постоянно находящуюся в памяти часть операционной системы, стал результатом развития применявшихся на первых порах программ обработки прерываний. Развитие исходного компонента управления данными привело к созданию стандартной библиотеки программ ввода-вывода, которые в ходе редактирования связей встраиваются в прикладную программу. Планировщик, который было принято размещать на магнитной ленте, стал загружаться в оперативную память во время перехода от одного задания к другому для определения необходимых операций установки файлов, расположенных на магнитной ленте (и перфокартах), и указания направления вывода, а затем выгружаться. Операционные системы предоставляли возможность использования файлов на дисках, но обычно сама операционная система находилась на магнитной ленте.

В операционных системах IBM конца первого поколения была предусмотрена возможность одновременной оперативной эксплуатация периферийных устройств (Simultaneous Peripheral Operation On Line — SPOOL), что позволяло выполнять в любое время на компьютере второго поколения одно главное приложение и несколько утилит передачи данных с перфокарт на магнитную ленту или диск, с магнитной ленты или диска на перфокарты, с магнитной ленты или диска на принтер. В дальнейшем все подобные утилиты стали разрабатываться как "программы, заслуживающие доверия", написанные столь тщательно, чтобы ни в коем случае не могло произойти нарушение работы или вмешательство в работу главного приложения, которое обычно эксплуатировалось в условиях ввода с магнитной ленты и вывода на магнитную ленту или ввода с диска и вывода на диск. Таким образом, эксплуатация компьютера состояла в том, что осуществлялась подготовка магнитных лент к следующему заданию, выполнялось главное задание и одновременно происходила печать вывода предыдущего задания, причем все эти действия происходили одновременно.

Языковые компиляторы. Клиентам IBM предоставлялась возможность использовать широкий набор языков высокого уровня, причем корпорация IBM взяла на себя обязательства по предоставлению компиляторов для этих языков. Самыми популярными были Fortran и COBOL. ALGOL был популярен в Европе. Язык RPG (Report Program Generator), относящийся к низшей категории языков высокого уровня, использовался главным образом в тех приложениях, которые стали результатом развития программ, первоначально основанных на применении перфокарт.

С технической точки зрения представляет интерес развитие программирования на ассемблере. Работа классического двухпроходного ассемблера была основана на применении двух предварительных проходов, что позволяло применять генератор макроопераций с развитыми возможностями времени компиляции, включая ветвление и обработку циклов. Такие макроассемблеры использовались двумя весьма различными способами.

В сообществе, занимающемся научными расчетами, как правило, использовались написанные программистами макроопределения в качестве открытых подпрограмм для часто применяемых операций высокого уровня, таких как матричные операции. Применение макроопределений способствовало не только упрощению кодирования, но и повышению быстродействия во время выполнения, поскольку позволяло исключить издержки, связанные с вызовом подпрограмм.

В отличие от этого, во многих вычислительных центрах, занимающихся обработкой бизнес-данных, распространилась практика, характеризующаяся тем, что к работе привлекается небольшая группа специалистов высокого уровня, которые пишут "внутрифирменные" библиотеки макрокоманд, фактически определяющие новые типы данных, со структурами данных и операциями, которые задают своего рода специализированный язык программирования для практики деловых отношений конкретной компании. Основная часть программистов лишь использует эту библиотеку макрокоманд, как правило, не создавая какие-либо собственные макроопределения.

Утилиты. Пакет программ для каждого компьютера дополнял целый ряд утилит, едва заметных для постороннего взгляда, но необходимых и нетривиальных: генераторы программ сортировки, трансляторы программ переноса с одних носителей на другие, трансляторы формата, средства отладки, программы формирования дампов памяти.

Бесплатное программное обеспечение. В то время изготовители предоставляли операционные системы и компиляторы бесплатно, чтобы стимулировать продажу и использование оборудования. В связи с этим приходилось включать стоимость пакетов программ в цены на оборудование[4].

Новые направления, получившие дальнейшее развитие

В связи с открывшимися возможностями создания принципиально новых пакетов программной поддержки определились направления будущего развития, которые должны были стать следующим шагом программной поддержки. Некоторые из этих направлений получили свое развитие, а другие остались нереализованными.

Универсальная применимость. Пакеты поддержки программного обеспечения предыдущего поколения строго различались по своей прикладной области и уровню производительности, тогда как пакет OS/360 предназначался для представления всего спектра применений. Само название System/360 было выбрано для указания на то, что эта компьютерная система "охватывает все вокруг". Кроме того, компьютеры System/360 разрабатывались для поддержки чрезвычайно широкого диапазона характеристик производительности, начиная со скромной системы с оперативной памятью 64 Кбайта и кончая наиболее сложной суперкомпьютерной системой или базой данных с мощной конфигурацией.

Для обеспечения таких возможностей потребовалось многое сделать в части языков программирования и предназначенных для них компиляторов. В сотрудничестве с ассоциациями пользователей IBM, занимающихся научными расчетами и обработкой бизнес-данных, был разработан новый, универсальный язык программирования PL/I. Для каждого из языков — Fortran, COBOL, Assembler и PL/I— был разработан целый ряд компиляторов, оптимизированных для различных объемов памяти. Группы разработчиков, ответственные за создание компиляторов или утилит, анализировали идеи нескольких взаимодействующих с ними сообществ пользователей и включали предложенные усовершенствования в продукты следующего поколения. Далее будет рассматриваться только самый новаторский компонент — сама операционная система.

Операционная система, располагаемая на диске. В нашем распоряжении оказался новый недорогой дисковод, IBM 2311, который имел емкость 7 Мбайт, казавшуюся в то время огромной. Это означало, что появилась возможность спроектировать операционную систему, располагаемую на так называемом "устройстве с произвольным доступом", а не на магнитной ленте. В связи с этим нам пришлось кардинальным образом пересмотреть все концепции проекта операционной системы. Модули операционной системы могли быть быстро развернуты в памяти по мере необходимости, что позволяло сделать их небольшими и специализированными по функциональному назначению.

Было разработано новое устройство ввода-вывода на магнитных барабанах с параллельной передачей слов, обеспечивающее низкое время задержки и позволяющее размещать операционные системы с высокой скоростью передачи данных для компьютерных систем более высокого класса.

Мультипрограммирование. При создании OS/360 было сделано очень многое для обеспечения параллельной эксплуатации независимых друг от друга программ, относящихся к категории так называемого программного обеспечения, "не заслуживающего доверия" (для которого необходимо предусматривать контроль доступа). Это стало возможным в результате реализации аппаратных супервизорных средств архитектуры System/360. В ранних версиях OS/360 обеспечивалась поддержка многочисленных задач фиксированного размера, для которых не приходилось применять сложные процедуры распределения памяти. А через два года была выпущена версия MVS операционной системы, в которой поддержка мультипрограммирования была осуществлена в полной степени. Решение этой задачи оказалось намного сложнее, чем мы ожидали.

Эксплуатация компьютера под управлением операционной системы, а не оператора. Ключевым достижением при создании новой операционной системы была передача функций управления компьютером от оператора операционной системе. Теперь такой принцип эксплуатации применяется повсеместно. Вплоть до 1987 г. под ручным управлением оператора эксплуатировались даже некоторые суперкомпьютеры, такие как ETA 10, одна из моделей ряда ETA корпорации CDC (Control Data Corporation). Управление компьютером с помощью операционной системы, которое теперь применяется повсеместно, было впервые реализовано в компьютере Stretch. Внешне это выражается в том, что оператор в основном работает с клавиатурой или консолью, а в качестве средств непосредственного управления компьютером сохранилась лишь пара кнопок (например, кнопки включения/выключения питания и перезапуска).

Дистанционная обработка без разделения времени. OS/360 была с самого начала спроектирована как система дистанционной обработки, но первые версии этой операционной системы еще нельзя было назвать в полном смысле слова системами с разделением времени на основе терминалов. Любопытно сравнить OS/360 с современной системой MIT Multics System. OS/360 была предназначена для использования в качестве системы производственного назначения, обеспечивающей выполнение научных расчетов и обработку данных с помощью специализированных приложений. С другой стороны, Multics была спроектирована в целях применения в качестве исследовательской системы, прежде всего для разработки программ.

Каждодневная и круглосуточная надежная эксплуатация. В проекте OS/360 было предусмотрено применение контрольных точек. Это позволяло устанавливать точки автоматического перезапуска, осуществлять контроль за сбоями оборудования и выполнять перезапуск после сбоя аппаратного или программного обеспечения. Диагностические средства, предусмотренные в многопроцессорной конфигурации, позволяют предусматривать взаимное резервирование процессоров и в случае отказа одного из процессоров передавать его нагрузку на другие процессоры. OS/360 с самого начала была предназначена для круглосуточной эксплуатации, однако для достижения этой цели потребовалось выполнение еще нескольких эволюционных шагов.

Проектные решения[5]

Структура системы

По мере развития вычислительной техники развивались и программы, применяемые для управления ею, в основном по трем независимым направлениям. В OS/360 эти три потока развития были объединены. На основе применявшихся ранее программ обработки прерываний был создан супервизор; планировщик стал преемником тех первых планировщиков заданий, которые входили в состав операционных систем, размещенных на магнитной ленте; система управления данными была разработана на основе существующих пакетов подпрограмм ввода-вывода. Иначе говоря, OS/360 имеет богатую родословную, и этим определяется ее облик.

Супервизор. Первые версии супервизоров обеспечивали только обработку программных и системных прерываний, а также распределяли процессорное время между задачами, а в современных мультипрограммных системах на супервизор возложена также задача распределения основного пространства памяти. Супервизор OS/360 распределяет блоки памяти и машинное время между задачами с учетом их приоритетов.

Управление компьютером в супервизоре OS/360 осуществляется путем контроля над счетчиком команд. В любой момент времени счетчик команд находится в распоряжении только одной программы. Возникновение любого прерывания, в том числе вызванного ошибкой в программе или попыткой нарушения правил, установленных любым из системных механизмов защиты, вызывает прерывание, после чего управление счетчиком команд снова переходит к супервизору. Прерывания возникают также в ходе выполнения синхронных и асинхронных операций устройствами ввода-вывода, например, по завершении записи или чтения. Кроме того, супервизор следит за показаниями защищенного таймера, который отсчитывает заданные интервалы времени, что позволяет ему перехватить управление по истечении указанного интервала. Это, в частности, дает возможность предотвратить возникновение бесконечных циклов, которые являются следствием ошибки в программе. Только супервизор может устанавливать правила защиты памяти и других ресурсов и выполнять прочие привилегированные операции, такие как управление вводом-выводом.

Обычные прикладные программы запрашивают обслуживание со стороны супервизора, например, для получения дополнительных блоков памяти, выполняя операцию вызова супервизора (SVC — Supervisor Call). Вызов этой операции приводит к прерыванию, после чего супервизор обрабатывает параметры, указанные в команде прерывания. Таким образом, приложения имеют весьма ограниченный доступ к супервизору, осуществляемый полностью под его контролем.

Супервизор предоставляет также механизмы для обеспечения взаимодействия программ, поскольку обычные программы не могут без его помощи организовать взаимообмен данными.

Планировщик. Планировщик OS/360 обеспечивает параллельное выполнение не связанных друг с другом программ в виде процессов, которые принято называть *заданиями*, а в пределах заданий обеспечивает выполнение последовательных действий, именуемых задачами, например, таких, как компиляция, связывание с библиотеками, выполнение и преобразование вывода. После того как задание становится подготовленным для выполнения, планировщик проверяет приоритет задания, выделяет для задания все необходимые устройства ввода-вывода, выдает оператору инструкции, касающиеся установки носителей или томов, отсутствующих в системе, и ставит задание в очередь на выполнение. Затем супервизор распределяет начальный объем памяти и инициализирует первую задачу в составе задания. По мере формирования выходных данных планировщик управляет их размещением на внешних устройствах, а после окончания задания выдает команду на снятие томов с данными, которые больше не требуются.

В OS/360, в отличие от предыдущих версий операционных систем, большее значение придается планированию работы, учитывая то, что во время компиляции должны быть доступны все необходимые информационные ресурсы, а во время выполнения — все необходимые устройства ввода-вывода. Планировщик, с одной стороны, должен обеспечить подключение к компилируемой программе всех необходимых модулей, а с другой, предоставить приложению во время выполнения все необходимые наборы данных с учетом их расположения на определенных устройствах. Все условия выполнения приложения определяются в пакете на языке управления заданиями (Job Control Language — JCL), выполняемом планировщиком.

Управление данными. Одной из наиболее важных отличительных особенностей семейства компьютеров System/360 стала реализация новой концепции строгой совместимости программ. Но не менее важно то, что в этих компьютерах предусмотрен широкий набор устройств ввода-вывода. Это позволяет расширить спектр применимых приложений, повысить гибкость определения конфигурации и увеличить производительность. Был принят единый стандартный механический, электрический и логический интерфейс ввода-вывода, что привело к существенному уменьшению стоимости разработки новых устройств ввода-вывода. Кроме того, стало намного проще вносить изменения в конфигурацию системы, а также обеспечивать наращивание конфигурации и модернизацию ее компонентов.

В связи с этим были также внесены важные новаторские изменения в программную поддержку аппаратных средств, поэтому в системе стало возможным обеспечение единого управления вводом-выводом и данными для всех устройств ввода-вывода на основе стандартного аппаратного интерфейса обмена данными. Я считаю это наиболее важным новшеством в OS/360.

В итоге операционная система стала обладать принципиально новым свойством — обеспечивать ввод-вывод независимо от устройств. Появилась возможность разрабатывать прикладные программы в терминах имен наборов данных.

Соответственно, программист мог отложить выполнение операции связывания данных с конкретными наборами данных, будь то определенные катушки с магнитной лентой, магнитная лента или диск, диск или канал передачи данных, или даже принтер, до этапа планирования задания.

В связи с появлением накопителей на жестких дисках новых типов были разработаны четыре метода доступа, позволяющие в полной мере использовать возможности этих дисков в приложениях. В этих методах реализованы различные операции буферизированной или блочной передачи данных, поэтому они различаются между собой по своим возможностям, в частности, по производительности, как описано ниже.

- Метод последовательного доступа (Sequential Access Method — SAM). Доступ осуществляется по такому же принципу, как к файлам на магнитной ленте, с применением буферизации.
- В качестве примера применения этого метода можно назвать сортировку (сортировка может применяться для магнитных лент, принтеров и колод перфокарт, а также для дисков).
- Метод прямого доступа (Direct Access Method — DAM). В полном смысле произвольный доступ к любой записи файла.
- В качестве примера применения этого метода можно назвать системы резервирования авиабилетов.
- Библиотечный метод доступа (Partitioned Access Method — PAM). Этот метод обеспечивает быструю передачу блоков данных постоянной длины.
- В качестве примера применения этого метода можно назвать хранение модулей операционной системы.
- Индексно-последовательный метод доступа (Indexed Sequential Access Method — ISAM). Этот метод обеспечивает последовательный и буферизированный доступ, но практически не требует поиска, поэтому позволяет быстро обрабатывать произвольные запросы.
- В качестве примера применения этого метода можно назвать выставление счетов за коммунальные услуги.

Два последних метода доступа были специально предназначены для обеспечения особой гибкости и простоты в использовании и в терминальном режиме, и при быстродействующей передаче данных.

В свое время мы обнаружили курьезный факт — из всех устройств ввода-вывода наибольшие требования к производительности операционной системы исходили от устройств сортировки чеков, поскольку между поступлением бумажного чека к считывающей головке и его передачей в один из приемных карманов сортировочного устройства проходило очень мало времени. Дело в том, что в отделах банков, занимающихся распределением и обработкой чеков, были установлены

машины, считывающие цифры, написанные вдоль нижнего края чека магнитными чернилами, а затем распределяющие чеки по 24 карманам со скоростью до 40 чеков в секунду[6].

Оценка

Преимущества

Полнофункциональность, универсальная применимость. С выходом OS/360 были установлены новые стандарты полнофункциональной операционной системы. И действительно, эта система поддерживала чрезвычайно широкий диапазон приложений, конфигураций системы и требований к производительности.

Надежность. Степень надежности системы стала буквально непревзойденной. Это — операционная система, способная выполнять задачи в масштабах реального производства, которая стала стандартной для крупных приложений доступа к базам данных, потребляющих основную часть машинного времени современных мэйнфреймов.

Система управления данными. Поддержка ввода-вывода, независимая от устройства, способствовала существенному упрощению работы по программированию и расширению возможностей центров регистрации и обработки данных по повышению гибкости и дальнейшему развитию. Стало привычным то, что в каждый выходной день устанавливаются новые конфигурации процессоров и устройств ввода-вывода. После любых подобных реконфигураций эксплуатация большинства приложений продолжается без перекомпиляции.

Поддержка дистанционной обработки. OS/360 стала основой для создания широких сетей терминалов для банков, предприятий розничной торговли, а также для многих других отраслей.

Дополнительная поддержка виртуальной памяти. После того как корпорация IBM обеспечила поддержку виртуальной памяти в своей следующей линейке компьютерных продуктов, System/370, сама OS/360 послужила основой для создания расширения OS/360 MVS (Multiple Virtual Systems), но полная модернизация этой операционной системы не произошла.

Компании Amdahl, Hitachi и Fujitsu. Большинство изготовителей компьютеров, совместимых с S/360 на уровне разъемов, не стали брать на себя разработку новых программных систем, а воспользовались пакетом OS/360.

Недостатки проекта

Система. Было отмечено, что OS/360 имеет слишком широкий набор функций. В связи с тем что появилась возможность разместить операционную систему на диске, были исключены ограничения по размерам, которые раньше вынуждали проектировщиков операционных систем придерживаться жестких требований,

поэтому в систему было введено много функциональных средств, польза от которых является довольно сомнительной[7]. Стремление к погоне за ненужными новинками и в наши дни не преодолено в сообществе программистов.

Были предусмотрены две совершенно разные системы отладки. Одна из них опиралась на интерактивное использование терминалов и обеспечивала быструю перекомпиляцию, а другая была рассчитана на пакетную обработку. Это была самая лучшая система отладки с применением пакетов, когда-либо спроектированная, но оказалась полностью устаревшей с самого начала.

Процесс генерации системы OS/360 является невероятно гибким и вместе с тем невероятно трудоемким. Нам пришлось определить конфигурации в виде некоторого числа стандартных пакетов, позволяющие выполнить требования большинства пользователей в ходе формирования общей конфигурации системы, чтобы создать достаточно гибкий процесс настройки конфигурации.

Контрольные блоки. Связь между модулями в масштабах всей системы обеспечивалась с помощью общих контрольных блоков, каждый из которых содержал структурированный набор переменных, предназначенный для чтения и записи всеми взаимодействующими модулями. Каждый программист имел доступ ко всем контрольным блокам. К сожалению, свою стратегию сокрытия информации Парнас (Parnas) выдвинул только в 1971 г., поэтому в 1963 г. мы были вынуждены применять указанную концепцию контрольных блоков, с чем были связаны значительные недостатки в оригинальной конструкции системы, а также вызваны все последующие доработки. Современным воплощением стратегии сокрытия информации явилось объектно-ориентированное программирование; мы все признаем превосходство этого подхода к программированию.

Виртуальная память. Как уже было сказано в главе 24, мы не сумели сразу понять возможности поддержки виртуальной памяти, впервые реализованные в процессорах, поэтому были вынуждены провести соответствующую доработку операционной системы несколько лет спустя. Связанное с этим расширение OS/360 было более трудоемким и дорогостоящим по сравнению с тем, что можно было реализовать в первоначальном проекте.

Язык управления заданиями планировщика. Язык управления заданиями — это самый плохой язык программирования, когда-либо и где-либо спроектированный кем-либо еще, причем он был спроектирован под моим руководством. Неправильной была сама концепция; мы не рассматривали JCL как язык программирования, а подразумевали под ним лишь "несколько перфокарт с командами управления, предшествующих заданию". Я подробно рассмотрел недостатки JCL в главе 14.

Недостатки в системе управления данными. Нам пришлось отказаться от блочной структуры переменной длины с данными ключа и количества, которая применялась ранее в дисководах производства IBM, и разработать небольшое количество форматов блоков с фиксированной длиной для всех устройств с произвольным доступом.

Дерево подключения блоков управления каналами в системе управления устройствами ввода-вывода оказалось излишне сложным[8]. Мы были вынуждены задавать по одному (чаще всего виртуальному) каналу для каждого устройства и по одному (также виртуальному) блоку управления для каждого устройства.

Полагаю, что мы были обязаны разработать единый последовательный метод доступа к диску, который сочетал бы в себе оптимизированные варианты всех трех методов доступа: SAM, PAM, ISAM.

Недостатки в поддержке процессов

Я подробно осветил эту тему в книге [40]. Хочу отметить лишь следующее.

Я твердо убежден в том, что если бы мы встроили все функции управления процессами в PL/I, который был в то время наилучшим из всех имеющихся языков высокого уровня, то наша операционная система осталась бы такой же быстродействующей, но была бы намного проще, надежнее, а ее создание произошло бы быстрее. Фактически эти функции были затем встроены в язык PLS — синтаксически обогащенный язык ассемблера. Но применение PL/I (или любого другого языка высокого уровня) потребовало бы провести интенсивное обучение наших сотрудников в части того, как писать хорошие программы на языке PL/I, чтобы исходные файлы PL/I компилировались в машинный код, показывающий высокое быстродействие во время выполнения.

Мы должны были бы обеспечить более строгое управление реализацией архитектуры во всех интерфейсах, настаивая на том, чтобы все объявления внешних переменных импортировались из библиотек, а не задавались вручную снова и снова в каждом экземпляре модуля. Это способствовало бы предотвращению многих ошибок.

Проектировщики

Над разработкой всего пакета программ OS/360 работали примерно 1 000 человек. Отмечу в этой главе отдельные группы и отдельных людей, которые внесли наибольший вклад в концептуальную структуру.

Ключевые исполнители

Лаборатории. Пукипси, Эндикотт, Сан-Хосе, Нью-Йорк, Херсли (Великобритания), Ла-Год (Франция).

Архитектор OS/360. Мартин Белски (Martin Belsky).

Ключевые исполнители. Берни Витт (Bernie Witt), Джордж Мили (George Mealy), Уильям Кларк (William Clark).

Руководитель разработки управляющей программы. Скотт Локен (Scott Locken).

Руководитель разработки компиляторов и утилит. Дик Кэйс (Dick Case).

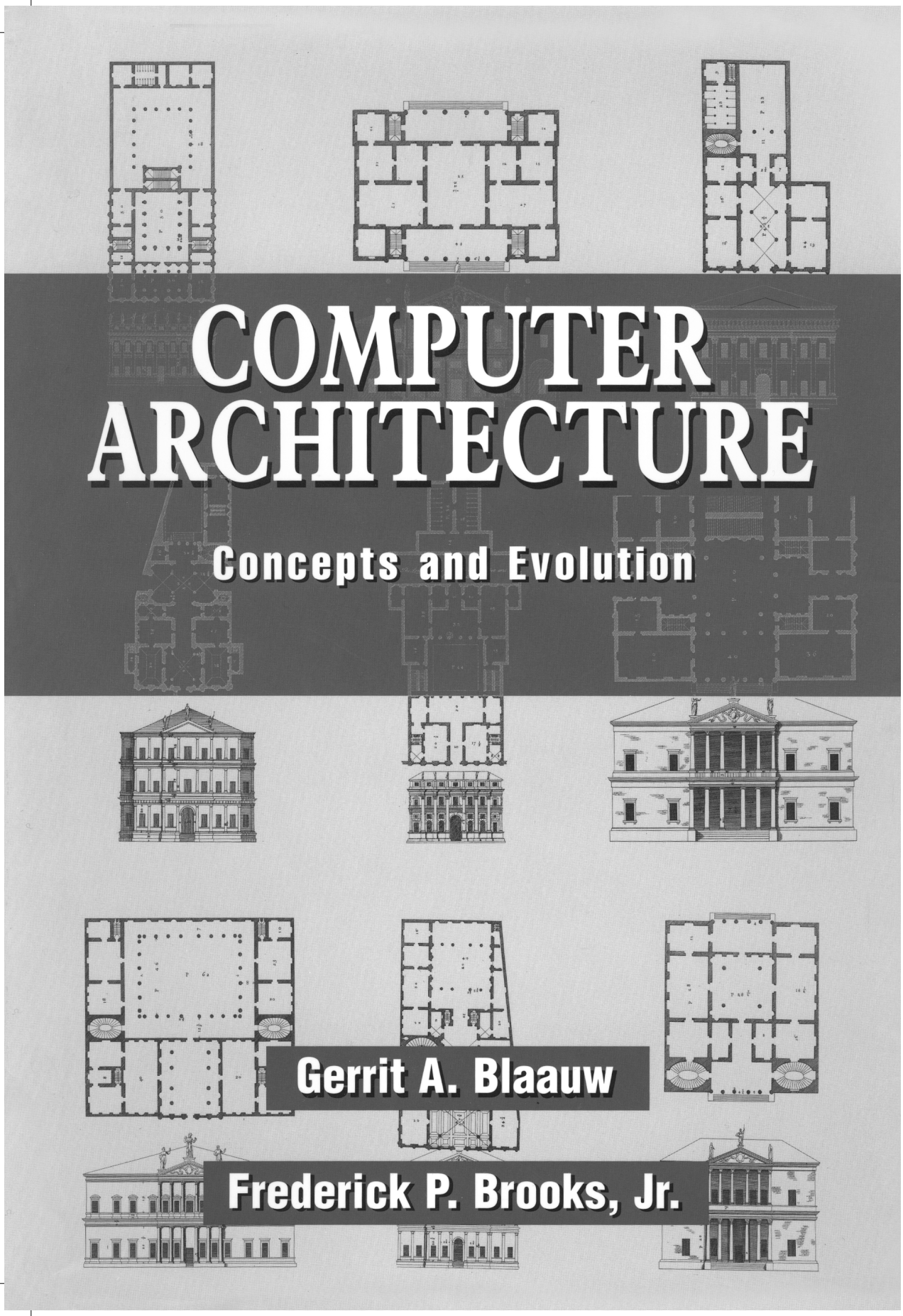
Руководитель проекта помощника по OS/360. Дик Кэйс (Dick Case).
Руководитель разработки OS/360 с 1965 г. Фриц Трэпнелл (Fritz Trapnell).
Лучший в своем роде документ. Руководство *Concepts and Facilities*, написанное Бернардом Виттом[9,10].

Общие усвоенные уроки

1. Дайте системному архитектору полную власть над проектированием (глава 19). Эта "многомиллионная ошибка" более подробно обсуждается в книге [40], с. 47 и далее.
2. Отведите достаточно времени на то, чтобы выполнить качественное проектирование и макетирование, невзирая на то, насколько жесткие сроки определены планом. Благодаря этому проект может быть закончен раньше, а не позже, поскольку время будет распределено более рационально. В главах 21–24 иллюстрируются преимущества предоставления достаточного количества времени на проектирование; здесь демонстрируется противоположное.

Примечания и ссылки

1. Эти заметки приведены в статье [46] из книги [49]. Материал взят в значительной степени из *IBM Systems Journal* 5, № 1 [1966].
2. См.страницуhttp://en.wikipedia.org/wiki/DOS/360_and_successors; в книге взята за основу версия этой страницы, полученная в августе 2009 г.
3. См. [25], раздел 1.1.
4. В статье [107] описано, как в 1969 г. была решена задача раздельного представления программного и аппаратного обеспечения.
5. В книге [170] подробно описана история разработки первых версий OS/360.
6. Подробная информация и фотография следующей версии компьютера приведены по адресу http://www.thegalleryofoldiron.com/3890.HTM; в книге взята за основу версия этой страницы, полученная в августе 2009 г.
7. См. [40], глава 5.
8. См. [25], раздел 8.22.
9. См. [124].
10. В книге [227] подробно рассказано о концепциях и подходах в проекте.



COMPUTER ARCHITECTURE

Concepts and Evolution

Gerrit A. Blaauw

Frederick P. Brooks, Jr.

TheDesignofDesign.indb 402 30.10.2012 15:58:41

26

Изучение практического примера. Проектирование книги Computer Architecture: Concepts and Evolution

Если бы записаны были слова мои! Если бы начертаны были они в книге!

ЙОВ 19:23

Задача написания книги имеет логарифмическую сходимость.

◂ *Суперобложка книги Blaauw and Brooks [1997], Computer Architecture*

Основные факты и особенности

Решение в чистом виде. Придерживаться узкого, но весьма точного определения компьютерной архитектуры как предмета данной работы. Мы предложили термин *компьютерная архитектура* в 1962 г. и дали его весьма точное определение в 1964 г., но со временем он стал использоваться в намного более широком смысле. Мы тщательно определяем и различаем архитектуру, реализацию и осуществление. Мы рассматриваем только архитектуру, точно определяемую как те свойства компьютера, от которых зависит, какие программы могут на нем выполняться и какие результаты они будут производить, но не касаемся вопроса о том, насколько быстро это будет осуществляться. Столь точное определение компьютерной архитектуры позволяет дать определение понятия совместимости программ.

Решение в чистом виде. В книгу должен быть включен набор примеров 30 компьютерных архитектур, описанных в стандартизированном формате. Этот формат включает текстовое описание основных фактов и особенностей, краткое представление исторического и технического контекста, рассмотрение поддерживаемой модели программирования, перечисление проектных решений и точное описание представлений данных, форматов и наиболее важных операций с помощью схем и фрагментов на языке APL.

Решение в чистом виде. Создание, проверка и публикация исполняемых эмуляторов для набора компьютерных архитектур; все эмуляторы должны быть написаны на языке APL. Описание каждого компьютера из набора примеров должно включать исполняемую программу APL, моделирующую выполнение в компьютере выборки, декодирования и вызова команд, относящихся к соответствующим программам чтения и обработки данных. Наиболее важные операции также должны быть описаны с помощью исполняемых функций APL. При решении задачи создания этих эмуляторов мы были вынуждены еще внимательнее рассмотреть моделируемые компьютеры и существенно уточнить их описание. Но нам не известно, использовались ли в дальнейшем эти эмуляторы достаточно широко.

Матричная организация. Проектные решения, касающиеся каждой исследуемой компьютерной архитектуры, рассматривались дважды. Во-первых, проводилось описание в рамках определенного перечня проблемных областей, и, во-вторых, архитектуры рассматривались в контексте всех взаимосвязанных решений, касающихся каждого конкретного компьютера.

Деревья решений. Мы использовали деревья решений как формальное инструментальное средство для представления вариантов, рассматриваемых в ходе проектирования. Около 80 деревьев были связаны вместе в единственное обширное унифицированное дерево решений для компьютерной архитектуры. Разумеется, такое формальное описание наталкивает на мысль, что проектирование представляет собой решение задач путем поиска в полностью определенном пространстве, но именно против этой модели я приводил убедительные аргументы в

настоящей книге! Дело в том, что со времени написания книги о компьютерной архитектуре мои представления о проектировании стали шире и глубже.

Развитие компьютерной архитектуры: анализ и синтез. Мы рассмотрели развитие компьютерной архитектуры от самого момента зарождения этого понятия (начиная с вычислительной машины Бэбиджа — Babbage) до 1985 г. и показали, насколько широко проводился экспериментальный анализ в этой области и насколько быстро произошел в дальнейшем синтез всех достижений, приведший к созданию на удивление стандартной архитектуры. Указанные тенденции отразились в документации не только многих первых моделей компьютеров, но и в документации некоторых современных компьютеров.

Намеченная к написанию книга о компьютерной архитектуре должна была также представлять собой исследовательскую монографию. Наша с Блау работа над созданием System/360 и даже над написанием книги, посвященной компьютерной архитектуре, позволила получить многочисленные, но небольшие научные результаты, которые по отдельности не были пригодны для печати. Поэтому мы впервые опубликовали многие результаты в книге *Computer Architecture* и привели их перечень в разделе "Предисловие". Разумеется, такое не часто встретишь в книге, которая предназначена для широкого круга представителей компьютерной отрасли и для студентов.

Полный справочник. В книге приведены тщательно выверенные определения терминов. К этим определениям терминов и их подробным описаниям ведет обширный предметный указатель, в котором также указаны все вхождения соответствующих терминов. Предусмотрены также отдельные указатели для имен и названий компьютеров, упоминающихся в книге. Библиография включает больше 500 наименований. Книга может применяться как справочник еще очень долго после того, как приведенный в ней дидактический материал перестанет быть полезным для обучения.

Введение и предыстория

Авторы введения и основного текста

Джеррит А. Блау и Фредерик Брукс

Даты

~1971–1997 гг.

Предыстория

Мы оба оставили практическую работу по разработке компьютерной архитектуры и стали вести курсы по этой теме. Мы постоянно увеличивали объем сведений, преподаваемых на курсах, а эти новые материалы включали в книгу, поэтому

она постепенно увеличивалась. После проведения примерно двух циклов лекций в рамках своих курсов мы решили начать подготовку к печати нашей книги, предназначая ее для использования главным образом не в качестве учебника для студентов, а в качестве систематического описания данной проблематики для широкого круга представителей компьютерной отрасли. Мы включили в книгу развернутые примеры, предназначенные не только для выполнения под руководством преподавателя, но и для самостоятельного изучения.

Задачи

Из предисловия к книге Computer Architecture:

"В этой книге мы поставили перед собой цель всесторонне рассмотреть проблематику проектирования компьютерной архитектуры. Эта работа предназначена для использования в основном не в качестве учебника, а в качестве руководства и справочника для специалистов, работающих в сфере проектирования компьютерной архитекторы, а также в качестве исследовательской монографии, в которой формулируется новая концептуальная структура для компьютерной архитектуры. Мы обрисовали достаточно широкий контекст, в котором происходило развитие данной проблематики, чтобы дать возможность читателю не только ознакомиться с современной практикой, но и узнать, как было достигнуто текущее состояние, а также узнать о том, какие попытки уже были сделаны и отвергнуты. Наша цель состоит в том, чтобы, с одной стороны, показать оставшиеся неизвестными для широкой публики альтернативы проектирования, а с другой — проанализировать и систематизировать известные достижения.

Мы сочли необходимым рассмотреть всю совокупность проблем, связанных с проектированием компьютерной архитектуры, а также обсудить положительные и отрицательные факторы, связанные с принятием многих известных решений задач проектирования. Таким образом, мы предоставляем каждому специалисту в области компьютерной архитектуры возможность по-новому взвесить указанные факторы исходя из реализуемых им приложений и применяемой технологии, а также на основе его личных предпочтений и опыта".

Возможности

Мы с Блау тесно сотрудничали в корпорации IBM в течение восьми лет, нам было легко общаться и мы понимали друг друга с полуслова. Поэтому нам удалось организовать процесс написания этой книги в режиме дистанционного взаимодействия, как описано в главе 7.

Мы совместно работали над проектированием трех компьютерных архитектур, причем каждый из нас принимал участие в решении тех задач проектирования,

которыми занимался другой. Эти проекты потребовали от нас изучения проектов наших предшественников, а после того как мы перешли в сферу преподавания, для нас стали еще более ясными внутренние пружины выполненных проектов и их значимость. К тому же мы были авторами руководств по программированию для большинства рассматриваемых компьютеров.

Проектирование архитектуры System/360 осуществлялось без спешки, поскольку предназначенная для применения в этой системе технология полуинтегральных схем была намечена к реализации в 1964 г. Поэтому мы смогли тщательно обсудить подготавливаемые проектные решения и изучить в этом контексте все аргументы за и против при выборе многих решений в области компьютерной архитектуры.

Что касается рассматриваемой книги, то бюджет времени и усилий на ее подготовку был почти безграничным. По крайней мере, мы так думали. Но в этом мы очень серьезно ошиблись. Чтобы стать наиболее влиятельной и полезной, эта книга должна была выйти раньше.

Ограничения

Каждый из нас принимал участие в напряженных исследовательских программах, уделял много времени преподаванию, а также не мог оставлять без внимания подрастающих детей. К тому же, приступив к подготовке данной книги, каждый из нас занялся публикацией собственной книги по указанной теме. Поэтому в расписании подготовки нашей совместной книги по компьютерной архитектуре часто возникали задержки.

Проектные решения

Последовательность. Одним из самых трудных проектных решений при подготовке любого описательного литературного произведения является определение последовательности изложения. Необходимо построить общий граф взаимосвязанных концепций и преобразовать его в древовидную структуру, чтобы иметь возможность затем отобразить это дерево на линейную структуру текста.

Мы рассматривали две основные возможные последовательности, причем и та, и другая была одинаково значимой. В конечном итоге мы сформировали структуру книги исходя из обеих последовательностей и привели в книге большое количество перекрестных ссылок. В части I систематически рассматриваются проектные решения в той последовательности, которая соответствует заложенным в них концепциям. Но каждое реальное решение должно быть принято в контексте всех других решений, относящихся к одному и тому же компьютеру. Поэтому мы проиллюстрировали проектные решения в контекстах нескольких образцов компьютеров в части II нашей совместной книги.

Выше в данной главе, в разделе "Основные факты и особенности", приведены другие важные проектные решения, которые не требуют дополнительного рассмотрения.

Оценка

Устойчивость. Если устойчивость проекта определять как его постоянство, то данный проект соответствует этому определению. По истечении тринадцати лет со времени выхода книги она не стала менее полезной, а изложенные в ней трактовки не устарели, хотя и не мешало бы дополнить ее материалом, относящимся к более современным разработкам.

Рыночная пригодность. Книга вышла слишком поздно, поэтому не удалось реализовать весь потенциал ее использования. Вместо нее в качестве учебника для одного или нескольких курсов по компьютерной архитектуре стала, в частности, применяться превосходная и постоянно обновляемая книга Хеннесси (Hennessy) и Паттерсона (Patterson) *Computer Architecture: A Quantitative Approach*.

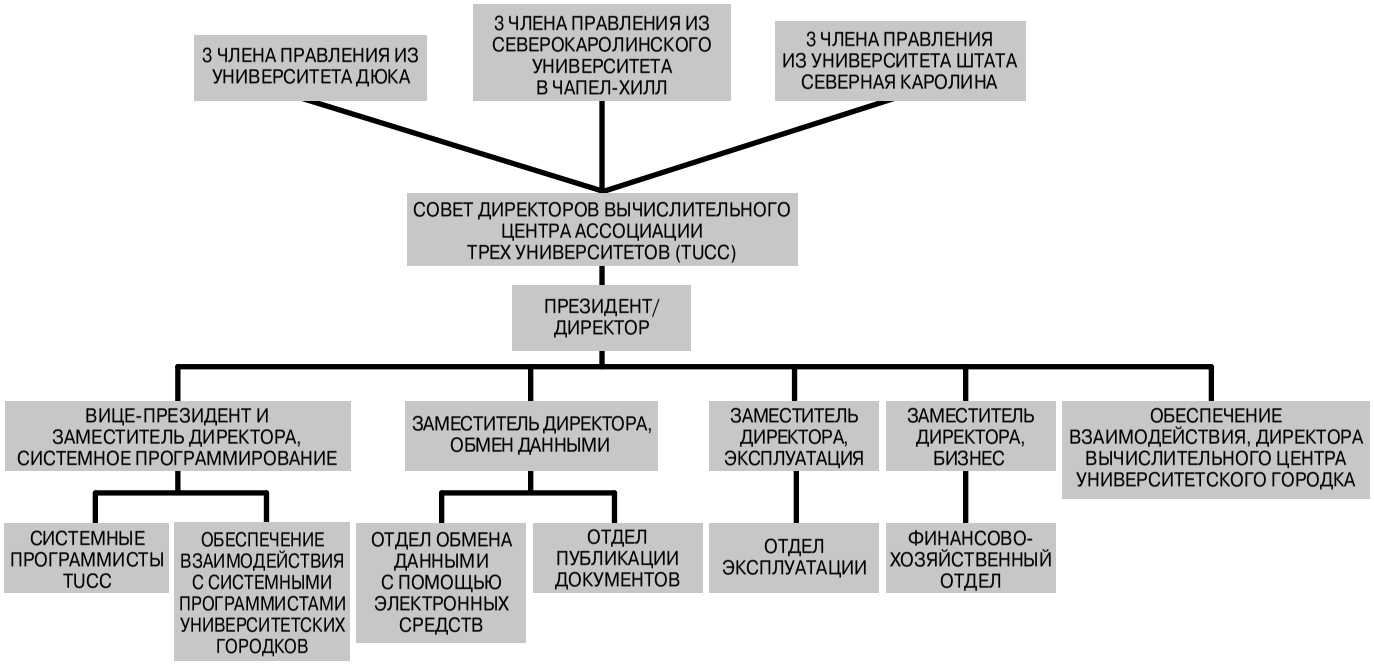
Но профессионал в области компьютерной архитектуры должен изучить и нашу работу, поскольку она позволяет ознакомиться с работами его предшественников, а также может послужить в качестве руководства и справочника. У нашей книги есть небольшой, но постоянный круг приверженцев, в основном среди специалистов по компьютерной архитектуре.

Общая польза. Это должны оценить другие.

Усвоенные уроки

1. Если бы мы осуществили менее амбициозный замысел, но сделали бы это быстрее, то принесли бы больше пользы нашей профессии. Я и теперь занимаюсь преподаванием на одном курсе, который основан на книге *Computer Architecture*. При этом я демонстрирую студентам описанные разновидности компьютеров и их образцы, показывая конкретные проектные решения на ярких примерах, а не разбирая их систематически. Возможно, нам с Блау следовало бы написать и опубликовать свои части книги отдельно, быстрее и раньше всех. Хотя и это было бы непросто, поскольку в большей части обсуждения конкретных образцов компьютерных архитектур подразумевалось знакомство с концепциями, введенными и описанными в части I.
2. Задача написания книги имеет логарифмическую сходимость. Мы слишком много потратили времени на уточнение в конце написания книги некоторых недостаточно твердо установленных фактов, исправления нескольких последних искаженных рисунков и проверку отдельных неточно заданных ссылок. Просто многие мелкие, но неприятные задачи были отложены на время ближе к завершению книги.

TheDesignofDesign.indb 410



30.10.2012 15:58:42

27

Изучение практического примера. Организация объединенного вычислительного центра. Вычислительный центр ассоциации трех университетов

Цель вычислений — получение большего объема знаний, а не чисел.

См. [110]

◂ *Организационная структура вычислительного центра ассоциации трех университетов (Triangle Universities Computation Center — TUCC) по состоянию на 1980 г.*

Основные факты и особенности

Решение в чистом виде. Организовать вычислительный центр как объединенный. Три университета должны объединить свои ресурсы, обеспечить совместное владение и управление единым высокоэффективным вычислительным центром.

Создание пула ресурсов. Поскольку кривая "производительность/цена" имеет форму квадратичной функции, необходимо стремиться к объединению ресурсов. На данное время, а также на многие годы вперед должна сохраниться зависимость, согласно которой увеличение объема доступного процессорного времени в n раз приводит к увеличению по меньшей мере в n^2 раз производительности вычислительной техники. Этим фактом обусловлены мощные экономические стимулы к преодолению сложностей, с которыми приходится сталкиваться в настоящее время и которые могут возникнуть в будущем, на пути к организации совместного владения объединенным вычислительным центром.

Отсутствие организационных моделей. Насколько нам было известно, теоретические основы создания объединенного академического вычислительного центра еще не были исследованы, поэтому отсутствовали и модели для такой организации.

Право принятия решений. В этом проекте лимитируемым ресурсом было право принятия решений. Как защитить несовпадающие интересы отдельных владельцев, вместе с тем обеспечивая эффективное принятие решений?

Разнообразие приложений. Некоторые из владельцев намеревались использовать вычислительный центр для академических и административных вычислений, а другие ставили перед собой цель проведения только академических вычислений.

Выбор площадки, не предоставляющей никому преимуществ. Здание для центра было приобретено в бизнес-парке "Исследовательский треугольник" (Research Triangle Park) в том месте, которое находится примерно на одинаковом расстоянии от всех трех университетских городков.

Выдвижение в качестве крайне важного, но не единственного требования обеспечение телеобработки. Приобретаемое оборудование IBM System/360 было спроектировано и оснащено программным обеспечением для дистанционного ввода заданий и проведения интерактивных вычислений. Вначале вычислительный центр эксплуатировался только в режиме дистанционного ввода заданий, но под управлением системы с приоритетами, которая позволяла ускорить выполнение небольших заданий. В качестве своего рода "широкополосной" передачи больших наборов данных применялась курьерская служба, с помощью которой осуществлялась доставка магнитных лент и пакетов дисков в автомобиле-универсале между университетами и вычислительным центром.

Влияние, распространяющееся на весь штат. В 1964 г. лишь немногие учреждения в Северной Каролине, предоставляющие высшее образование (кроме владельцев TUCC), обладали вычислительными ресурсами или ноу-хау в области

компьютеризации. Поэтому была создана отдельная организация, занимающаяся осуществлением проекта компьютеризации штата Северной Каролины под эгидой комитета по высшему образованию Северной Каролины. Эта организация оплачивала услуги TUCC и предоставляла доступ к ним другим университетам и колледжам шт. Северной Каролины на следующих условиях.

- Предоставление бесплатного машинного времени в течение года для выполнения до 100 пакетных заданий в месяц.
- Бесплатное использование в течение года телетайпа, установленного в университетском городке.
- Бесплатное обслуживание со стороны своего рода "миссионеров", которые выполняли важную роль в компьютеризации штата, посещая университетские городки, проводя занятия с преподавателями, организовывая симпозиумы, предоставляя консультации по телефону и устраняли неисправности.

Услугами этих активных пропагандистов воспользовались свыше ста учреждений, причем многие из этих учреждений затем продолжали пользоваться услугами TUCC за свой счет в течение многих лет.

Продолжительное существование. Организация TUCC оставалась достаточно эффективной в течение 18 лет, даже несмотря на то, что потребности совладельцев вычислительного центра TUCC иногда вступали во взаимное противоречие. В результате революционных изменений, вызванных появлением миникомпьютеров, вычислительный центр утратил свою прежнюю роль. Тем не менее многие организации продолжали эксплуатировать объединенным суперкомпьютерный центр, специализирующийся на научных приложениях, еще в течение многих последующих лет, но в рамках другой организационной модели.

Общие сведения и предыстория

Местоположение

Бизнес-парк "Исследовательский треугольник".

Владельцы — резиденты штата Северная Каролина

- Университет Дьюка, частный университет (Дарем, шт. Северная Каролина).
- Университет штата Северная Каролина (Роли, шт. Северная Каролина).
- Северокаролинский университет в Чапел-Хилл.

Разработчики организационной структуры

Совет директоров TUCC.

Даты

1964–1992 гг.

Предыстория

Во всех трех университетах ассоциации, Университете Дьюка, Университете штата Северная Каролина и Северокаролинском университете в Чапел-Хилл, были установлены компьютеры первого поколения, которые эксплуатировались в централизованных вычислительных центрах. Каждый из этих компьютеров нуждался в замене более мощными компьютерами, обладающими более высокой производительностью. Представители всех трех университетов желали иметь в своем распоряжении больше компьютеров, чем могли себе позволить.

Поэтому эти три университета решили объединить свои ресурсы и организовать эксплуатацию единого современного высокоэффективного оборудования (в объеме, гораздо большем, чем каждый мог бы позволить себе самостоятельно).

Для этой цели Национальный научный фонд США (National Science Foundation) выделил существенный грант, поскольку перед фондом, в частности, стояла задача провести исследование новой организационной модели предоставления академической поддержки исследований с использованием ЭВМ.

Весьма существенную помощь в обеспечении финансирования в течение первых трех лет оказывала корпорация IBM, арендуя ночные смены для своего нового подразделения по разработке и производству компьютерных продуктов в бизнес-парке "Исследовательский треугольник".

Основная проблема при создании объединенного вычислительного центра заключалась в том, как организовать совместное административное управление этим предприятием.

Цели[1]

Основные цели

Основное назначение центра. Предоставление оперативного, высококачественного обслуживания с использованием ЭВМ клиентам для эксплуатации широкого круга приложений разной степени сложности.

Основная цель разработки проекта организации. Разработка бесперебойно функционирующего плана руководства для объединенного вычислительного центра, которым на равных правах владеют три учреждения с разными потребностями и целями.

Другие задачи

- Поддержка финансовой стабильности центра.
- Обеспечение эффективного и своевременного принятия решений.
- Получение пользователями от каждого владельца равной доли всех ресурсов, без какого-либо ущемления прав.
- Защита финансовых инвестиций каждого владельца.
- Предоставление гарантий, согласно которым ни один владелец не должен понести финансовые потери в результате какого-либо судебного иска.
- Обеспечение функционирования центра как единой организации, а не трех подразделений, поскольку можно добиться экономии, лишь создав единый коллектив, установив общий комплект оборудования и выполняя общий поток заданий.
- Предоставление владельцам возможности изменять объем своего вклада в объединенный вычислительный центр, с учетом того, что от доли каждого владельца зависит, будет ли он получать больший или меньший объем услуг.

Возможности

Экономия за счет повышения общей производительности. Во время проектирования было установлено, что экономия за счет повышения общей производительности, связанного с тем, что создается один большой вычислительный центр вместо трех значительно меньших, должна быть существенной. Эта экономия была связана с сокращением количества персонала, что особенно важно при круглосуточной работе, а также с повышением экономической эффективности эксплуатации оборудования, поскольку вся техника, включая компьютеры, оперативную память, жесткие диски и другие устройства ввода-вывода, подчиняется закону квадратичной зависимости между производительностью и ценой.

Возможности использования в дистанционном режиме. Новые технологии, применение которых стало возможным в связи с внедрением новых компьютеров третьего поколения, впервые позволили дистанционно передавать на выполнение задания на вычисления и получать результаты, а также дистанционно проводить интерактивные сеансы вычислений.

Прототип общегосударственного значения. Стремительно развивающийся вычислительный центр TUCC мог стать образцом объединенного регионального вычислительного центра. Крупный вычислительный центр должен был получить известность в масштабах всей страны и дать возможность многим узнать о новом бизнес-парке "Исследовательский треугольник" в штате Северная Каролина. Новая организация, которая могла бы послужить в качестве

прототипа, должна была повысить значимость инноваций и преодолеть нередко встречающуюся разобщенность между местными университетами.

Привлечение правительственной поддержки. Экономия за счет повышения общей производительности могла способствовать привлечению дополнительной поддержки со стороны правительства США, поскольку предлагаемая новаторская концепция могла стать источником достижения улучшенного соотношения цены и качества для агентств, занимающихся финансированием. Кроме того, интерес у финансирующих органов мог вызвать сам принципиально новый характер модели.

Привлечение поддержки промышленных кругов. Масштаб и общегосударственная известность объединенного центра должны были вывести его в число вероятных кандидатов на получение существенной поддержки от промышленных кругов.

Ограничения

Быстрота принятия оперативных решений. Руководство повседневной эксплуатацией должно было осуществляться жестко и эффективно.

Необходимость в более частом наращивании мощностей. Предполагалось, что будет происходить стремительный рост потребностей в услугах центра и его финансировании, поэтому необходимо было добиться того, чтобы процедура внесения изменений в конфигурацию стала хорошо организованной и полностью контролируемой.

Защита жизненных интересов всех владельцев, которые могут быть разными. Университет Дьюка, как наименьшее из участвующих в проекте учреждений, должен был иметь гарантии, что от него не потребуют больший взнос, чем он может себе позволить. Университет штата Северная Каролина как потребитель самого большого объема ресурсов должен был иметь возможность рассчитывать на получение требуемых вычислительных мощностей. Для Университета Дьюка требовалась гарантия того, чтобы два других университета, которые представляли собой принадлежащие штату филиалы системы Северокаролинского университета, не отдавали свои голоса в пользу решений, которые бы противоречили интересам Университета Дьюка, находящегося в частной собственности.

Стабильность бюджета TUCC. Сам центр TUCC был обязан взять на себя, хотя и не слишком обременительные, но долгосрочные обязательства (например, заключение арендных договоров, контрактов со служащими), поэтому должен был гарантировать для себя стабильность бюджета.

Стабильность бюджета владельцев. Для успешного исполнения собственных бюджетных процессов владельцев требовалось, чтобы любое увеличение финансирования TUCC планировалось заблаговременно.

Руководители университетов, наделенные правом решающего голоса. Интересы каждого из университетов, в совместном владении которых находился

центр, были таковы, что их руководители стремились воспользоваться расширением бизнес-парка "Исследовательский треугольник". Безусловно, каждый руководитель стремился отстаивать интересы собственного учреждения.

Некоторые руководители университетов не могли уделять много времени организации работы центра, но не делегировали свои полномочия. Не во всех университетских городках были назначены руководители информационной службы, обладающие правами возглавлять необходимые мероприятия, поэтому иногда возникала задержка в принятии решений.

Значительное влияние личностного фактора. Приходилось считаться с тем, что в совместном руководстве центром принимали участие некоторые представители университетов, которых иногда было трудно в чем-то переубедить.

Проектные решения

Тщательное разделение решений, относящихся к сферам общего руководства и повседневной эксплуатации. Общее руководство должно было осуществляться советом директоров, заседания которого проходили ежемесячно, а повседневная эксплуатация осуществлялась персоналом под руководством главного администратора, директора TUCC.

Состав совета. Совет должен был быть достаточно небольшим, чтобы он мог работать, и вместе с тем включать достаточное количество членов, чтобы представлять различные направления деятельности каждого университета. Планируя состав совета, мы остановились на том, что он должен состоять из десяти членов, в том числе включать по три представителя от каждого университета-владельца и директора TUCC. Кроме того, место за столом совета было предоставлено директору проекта компьютеризации штата Северная Каролина (North Carolina Computer Orientation Project — NCCOP). Сам проект NCCOP базировался в здании TUCC, поэтому его директор имел существенное неформальное влияние на все происходящее.

Правила голосования, предусмотренные для совета

- Единодушное согласие членов (консенсус).
- Простое большинство голосов членов.
- Единодушное согласие университетов-владельцев; при таком голосовании решение каждого университета определялось большинством членов его правления.
- Большинство голосов от университетов.

Единодушное согласие не требовалось. Мы с самого начала остановились на том, что не следует по всем вопросам требовать единодушного согласия членов

совета, поскольку утверждение многих решений станет слишком затруднительным. Отказ от требования достижения консенсуса способствовал значительному упрощению процедуры принятия решений.

Принятие за основу большинства голосов членов совета, а не голосов самих учреждений. Мы пришли к выводу, что необходимо стимулировать работу совета как единого органа и противодействовать тенденциям к тому, чтобы представители отдельных университетов выступали исключительно в интересах своих учреждений. Поэтому было определено, что рядовые решения должны приниматься простым большинством голосов членов совета, присутствующих на заседании и имеющих право голоса.

"Вопросы первоочередной значимости". Была выделена категория вопросов первоочередной значимости, требующих принятия более ответственных решений по сравнению с обычным консенсусом. Эти вопросы перечислены в уставе TUCC следующим образом.

- Назначение или снятие с должности директора TUCC.
- Увеличение ежегодного бюджета больше чем на 10%.
- Внесение изменений в свидетельство о регистрации центра или в его устав.

По вопросам первоочередной значимости требовалось единодушное голосование со стороны университетов-владельцев. Следует отметить, что для этого не требовалось согласие всех членов совета; достаточно было, чтобы общее мнение возобладало среди представителей каждого университета. Голосом от каждого университета-владельца считалось мнение, выработанное большинством его представителей (двух из трех).

Постановка на голосование важного вопроса. Представители каждого университета-владельца могли объявить важный для себя вопрос как относящийся к категории первоочередной значимости и потребовать единодушного голосования со стороны университетов. Соответственно, эта процедура была намеренно сделана обременительной — в случае отрицательного решения представители университета могли повторно ставить свой вопрос на голосование в течение месяца. Затем главный администратор университета получал право перевести неразрешенный вопрос на более высокий уровень — в категорию вопросов первоочередной значимости. Таким образом, любой университет, затратив определенные усилия, мог противодействовать любым решениям, которые рассматривались как противоречащие его жизненным интересам.

Ротация председательства. С интервалом в два года право выдвинуть своего представителя на пост руководителя совета TUCC последовательно передавалось от одного университета-владельца к другому.

Распределение полномочий

Разрабатывая эту организационную схему, мы уделяли также внимание вопросам распределения полномочий по следующим направлениям.

- Между советом и рядовыми сотрудниками.
- Между большинством и меньшинством, включая университеты, вместе взятые, и их отдельных представителей.
- Между пользователями академических приложений и пользователями административных приложений.

Оценка

Устойчивость

Продолжительное существование. Организация TUCC успешно работала в течение 18 лет, при этом сменились два директора, три поколения компьютеров-мэйнфреймов, а сама модель с тремя равноправными пользователями-владельцами стала во многом другой.

Постановка на голосование важного вопроса. Насколько я помню, эта модель и с самого начала не применялась в чистом виде. Но то, что она была принята, стало источником значительного психологического комфорта и, как мне кажется, позволило избежать таких ситуаций, что представители какого-либо университета чувствовали себя загнанными в угол и были вынуждены бороться за существование.

Эксплуатация как главное направление деятельности. Сотрудники центра, как и предполагалось, чувствовали себя как работающие в едином коллективе. В этом большая заслуга совета, деятельность которого оказывала благоприятное воздействие. Лишь очень редко приходилось замечать проявление противоречий между разными университетами. Различия во мнениях членов совета чаще всего проходили по линии "ученый–администратор" или "новатор–консерватор".

Положительные результаты

Гибкость модели. Объем ресурсов, потребляемых Университетом штата Северная Каролина, постоянно увеличивался, в связи с чем университет вкладывал больше денег в расширение определенных мощностей (например, приобретал больше памяти) взамен на увеличение доли в общем объеме используемых ресурсов. Но приобретение нового оборудования приводило к изменению лишь внешней формы, но не сущности этого центра как предприятия с тремя равноправными владельцами, в рамках которого центр TUCC функционировал с самого начала. С наступлением революции, связанной с внедрением миникомпьютеров, степень участия Университета Дьюка в использовании ресурсов

центра уменьшилось. В Университете Дьюка многие направления обслуживания с использованием ЭВМ были переориентированы на мини-компьютеры, установленные на факультетах.

Наконец, принцип участия совладельцев в неравных долях был формализован. Труднее всего было перейти от равного представительства в совете к пропорциональному. Эта проблема была решена путем определения доли использования ресурсов центра, от которой зависит количество представителей в совете.

Распределение работ между вычислительными средствами самих университетов и центра TUCC. Каждый совладелец TUCC с самого начала имел также свой университетский вычислительный центр, сотрудники которого оказывали помощь пользователям из этого университета и эксплуатировали оборудование, позволяющее осуществлять ввод данных и вывод результатов, полученных от TUCC. Кроме того, это оборудование (что вполне естественно) позволяло выполнять часть заданий своего университета.

Таким образом, директору каждого университетского вычислительного центра приходилось постоянно следить за тем, чтобы часть выделенного ему бюджета направлялась в центр TUCC, а другая часть затрачивалась на оборудование для самого университета.

В Университете штата Северная Каролина, как правило, основные потребности пользователей удовлетворялись в центре TUCC, поэтому для данного университета вполне приемлемым был путь удовлетворения своих растущих потребностей путем непрерывного наращивания ресурсов TUCC и потребления все большей их части. В университете Дьюка наметилась тенденция к постепенному снижению потребления доли используемых ресурсов TUCC при удовлетворении большинства своих потребностей, учитывая то, что начальный взнос этого университета, который был, как и у двух других, равен одной трети, составлял значительную часть бюджетных расходов университета на вычислительную технику. Северокаролинский университет продолжал использовать свою треть растущего объема ресурсов TUCC, но в нем наметилась также тенденция восполнять излишки своих растущих потребностей за счет создания вычислительных мощностей на территории самого университета, а не увеличения своей доли в капитале TUCC.

Усвоенные уроки

1. Структуру организации удалось согласовать довольно быстро, и этому способствовало тщательное и открытое выявление с самого начала жизненно важных потребностей каждого из трех партнеров-университетов и самого правления вычислительного центра.
2. Была предусмотрена такая процедура окончательного утверждения, которая гарантировала соблюдение интересов каждого участника, пусть даже ее проведение было достаточно сложным.

3. Оправдалось то, что было признано наличие во многом противоречивых интересов каждого партнера, а это обусловило необходимость обеспечить представительство всех партнеров в руководящих органах. Но в дальнейшем было обнаружено, что различия во мнениях по многим проблемам (и это оказалось самым неожиданным) в основном касались областей ответственности, а не интересов самих университетов. По всем прочим вопросам часто одинаково голосовали не только представители финансовых органов всех трех университетов, но и представители университетов в вычислительном центре, и представители пользователей от факультетов.
4. Я не припомню такой ситуации, при которой различные представители непосредственно получали бы указания, касающиеся соблюдения собственных интересов университетов, но администраторы, назначая каждого представителя, по-видимому, были достаточно мудрыми, чтобы дать ему понять, как действовать.
5. Мы хорошо знали, что в подобной организации руководящий совет вполне может превратиться в орган, который лишь утверждает решения, принятые на более высоком уровне. Поэтому было решено ежемесячно проводить заседания совета, чтобы избежать такой опасности.
6. Некоторые главные администраторы взяли себе за правило отправлять на заседания своих подчиненных, а не заниматься лично обсуждением реальных проблем. Возможно, некоторые главные администраторы слишком опасались того, что после рассмотрения на заседании совета какой-то реальной проблемы будет принято решение, противоречащее их мнению.
7. Кроме того, мой опыт показал, что многие главные администраторы не стремились даже в случае необходимости воспользоваться знаниями других членов правления. Я думаю, что это — большое упущение.

Примечания и ссылки

1. Устав вычислительного центра ассоциации трех университетов можно найти по адресу `http://www.cs.unc.edu/~brooks/DesignofDesign`.

TheDesignofDesign.indb 422 30.10.2012 15:58:42

28

Рекомендуемое чтение

Литературные источники, упомянутые в настоящей книге, а также другие книги, статьи и документы, представляющие интерес с точки зрения изучения процесса проектирования, с полными выходными сведениями, перечислены в приложении Б. А в этой главе я решил отдельно указать некоторые из этих работ, которые нахожу исключительно ценными для тех, кто интересуется процессом проектирования. Они приведены в алфавитном порядке фамилий авторов с краткими комментариями.

1. Blaauw, G. A., and F. P. Brooks, Jr. [1997]. Computer Architecture: Concepts and Evolution. (См. [25].)

   В разделе 1.1 проводятся различия между архитектурой, внедрением и реализацией. Раздел 1.2 содержит краткий обзор проектов архитектур вычислительных систем. В нем также дано формальное определение концепции дерева проектирования, которое иллюстрируется на примере отдельных проектных решений. В разделе 1.4 предпринята попытка определить и охарактеризовать отличительные особенности качественной архитектуры.

2. Boehm, B. [2007]. Software Engineering: Barry Boehm's Lifetime Contributions to Software Development, Management and Research. (См. [29].)

   Ряд статей, охватывающих многие аспекты проектирования программного обеспечения, с которыми обязательно следует ознакомиться.

3. Brooks, F. P., Jr. [1975, 1995]. The Mythical Man-Month: Essays on Software Engineering, Anniversary edition. (См. [40].)

   В главе 16 трудности проектирования рассматриваются как относящиеся к категории сущностей и акциденций. Глава 19 представляет ретроспективу, охватывающую период с 1975 по 1995 гг.

4. Burks, A. W., H. H. Goldstine, and J. von Neumann [1946]. "Preliminary discussion of the logical design of an electronic computing instrument". (См. [52].)

◀ *В. Бенгоф, "Сценка в старой Библиотеке Конгресса". 1897 © CORBIS. Все права защищены*

Самая важная статья о компьютерах из всех, которые когда-либо были написаны. Потрясающе разносторонняя. Доступна в Интернете.

5. Cross, N., K. Dorst, et al., eds. [1992]. Research in Design Thinking. (См. [74].)

   Содержит проведенный Кроссом уничтожающий критический анализ взглядов Саймона: в действительности проектировщики действуют совсем иначе, и это подтверждают исследования. Другие статьи в книге также представляют ценность.

6. DeMarco, T., and T. Lister [1987]. Peopleware: Productive Projects and Teams, 2nd edition. (См. [76].)

   Важные результаты исследований и мысли о том, какие факторы, не относящиеся к сфере техники, влияют на качество проектирования.

7. Hales, C. [1987, 1991]. An Analysis of the Engineering Design Process in an Industrial Context. (См. [109].)

   По-видимому, наиболее полная из всех опубликованных документация с описанием реального, значимого процесса проектирования, которое составлено одним из проектировщиков, одновременно выполнявшим функции научного наблюдателя. Эта работа основана на диссертации на степень доктора философии, которую Хэйлс защитил в Кембриджском университете.

8. Hennessy, J., and D. A. Patterson [1990, 2006]. Computer Architecture: A Quantitative Approach, 4th edition. (См. [112].)

   Исключительно важная работа, посвященная проектированию архитектур вычислительных систем. Наглядно показан процесс сходимости к стандартной архитектуре.

9. Hoffman, D., and D. Weiss, eds. [2001]. Software Fundamentals: Collected Papers by David L. Parnas. (См. [119].)

   Еще один ряд статей, требующий внимательного ознакомления, которые охватывают много аспектов проектирования программного обеспечения.

10. Mills, H. D. [1971]. "Top-down programming in large systems". In Debugging Techniques in Large Systems, ed. R. Rustin. (См. [153].)

    Учебные материалы и доводы в пользу инкрементного проектирования и программирования.

11. Royce, W. [1970]. "Managing the development of large software systems". Proceedings of IEEE Wescon. (См. [180].)

    Классическая статья с описанием модели водопада, которая отвергается как непригодная. В этой статье приведены аргументы в пользу альтернативной модели.

12. Schoen, D. [1983]. The Reflective Practitioner. (См. [185].)

13. Simon, H. A. [1969, 1996]. The Sciences of the Artificial, 3rd edition. (См. [192].)

Наиболее влиятельное и обоснованное предложение по использованию рациональной модели для проектирования.

14. Winograd, T., et al., eds.[1996]. Bringing Design to Software. (См. [225].)

Очень полезный сборник статей, в который включены важные работы.

15. Wozniak, S. [2006]. iWoz: From Computer Geek to Cult Icon: How I Invented the Personal Computer, Co-Founded Apple, and Had Fun Doing It. (См. [229].)

Замечательная автобиография, написанная инженером для инженеров, в которой можно найти много интересных идей по проектированию.

Приложение А Благодарности

Хочу выразить свою признательность лицам, перечисленным в табл. А.1.

Таблица А.1.

Имя	Область проектирования	Основные место работы
Беседы и интервью		
Дэвид Эндрюс (David Andrews)	Теория кораблестроения	Британская королевская корпорация инженеров-кораблестроителей, Университетский колледж Лондона
Марко Аурисиккио (Marco Aurisicchio)	Программное обеспечение — обоснование проектирования, DRed	Кембриджский университет
Руй Бастос (Rui Bastos)	Проектирование графических микросхем	nVidia
Джеррит Блау (Gerrit Blaauw)	Компьютерное оборудование, проектирование книжной продукции	IBM, Университет Твенте
Барри Боэм (Barry Boehm)	Программное обеспечение — калькулятор орбиты ROCKET	RAND, TRW Systems, Университет штата Южная Калифорния
Роберт Брейсуэлл	Программное обеспечение — обоснование проектирования, DRed	Кембриджский университет
Джон Кларксон (John Clarkson)	Обучающие тренажеры	PA Consulting Group, Кембриджский университет
Сэр Дэвид Давис (Sir David Davies)	Безопасность железных дорог	Министерство обороны, Королевская инженерная академия наук Великобритании
Нейл Доджсон	Учебные планы по информатике	Кембриджский университет
Сэр Джон Фэркло (Sir John Fairclough)	Компьютерное оборудование	IBM, UK Chief Scientific Advisor
Кен Фаст (Ken Fast)	Теория кораблестроения — ядерная субмарина	General Dynamics Electric Boat
Стив Фербер (Steve Furber)	Компьютерное оборудование — BBC Microcomputer, ARM	Acorn Computers Ltd., Манчестерский университет

Продолжение табл. А.1

Имя	Область проектирования	Основные место работы
Гордон Глегг (Gordon Glegg)	Машиностроение	Кембриджский университет
Дональд Гринберг (Donald Greenberg)	Архитектура — проектирование жилья, автоматизации проектирования	Корнеллский университет
Билл Хиллир	Городское планирование — пространственный синтаксис	Университетский колледж Лондона
Джеффри Джапп (Jeffrey Jupp)	Авиастроение — Airbus 380, распределенная разработка	Airbus UK British Aerospace
Джули Джапп (Julie Jupp)	Финансовые модели проектирования и создания в рамках одного договора	Кембриджский университет
Джо Лоде (Joe Lohde)	Разработка аттракционов для парка отдыха	Disney Entertainment
Джанет Макдоннелл (Janet McDonnell)	Исследования в области проектирования — практика проектирования, организация совместной работы	Центральный колледж искусства и дизайна им. Святого Мартина
Крэйг Мадж (Craig Mudge)	Интегральные схемы	Digital, PARC, Pacific Challenge
Сэр Алан Мюр Вудс (Sir Alan Muir Woods)	Туннели — первые исследования в части строительства тоннеля под Ла-Маншем	William Halcrow and Partners
Брэдфорд Паркинсон (Bradford Parkinson)	Системотехника — GPS	Военно-воздушные силы США, Стэнфордский университет
Дэвид Паттерсон (David Patterson)	Компьютерное оборудование — RISC, RAID	Калифорнийский университет — Беркли
Шариф Раззак (Sharif Razzaque)	Методология проектирования — прототипы, реконструкция	Lockheed Martin, Университет штата Северная Каролина, InnerOptic Technology
Ричард Ризенфелд (Richard Riesenfeld)	Программное обеспечение — геометрическое проектирование, цифровое управление	Университет штата Юта
Джеймс Робертсон (James Robertson)	Программное обеспечение — процесс выдвижения требований	Atlantic Systems Guild
Сюзанна Робертсон (Suzanne Robertson)	Программное обеспечение — процесс выдвижения требований	Atlantic Systems Guild
Дональд Шён (Donald Schoen)	Архитектура, теория проектирования	Массачусетский технологический институт
Альберт Сегарс (Albert Segars)	Новаторские технологии	Северокаролинский университет
Мэри Шоу (Mary Shaw)	Разработка программного обеспечения — проектирование ПО с учетом значимости	Университет Карнеги-Меллона

Окончание табл. А.1

Имя	Область проектирования	Основные место работы
Герберт Саймон (Herbert Simon)	Теория проектирования, искусственный интеллект	Университет Карнеги-Меллона
Малком Саймон (Malcolm Simon)	Программное обеспечение — тесселяция	AVEVA, Кембриджский университет
Уильям Свартоут	Программное обеспечение — самодокументирующие программы	Институт творческих технологий
Кен Уоллис	Исследования проектирования	Кембриджский университет
Мэри Уиттон (Mary Whitton)	Компьютерное оборудование — устройства моделей Ikonas; программное обеспечение — виртуальные среды	Ikonas Graphics, Trancept, Северокаролинский университет
Сэр Морис Уилкс (Sir Maurice Wilkes)	Компьютерное оборудование — EDSAC	Кембриджский университет
Мартин Уильямс	Эксплуатационно-техническая служба — виртуальные среды в проектировании нефтяных платформ	Kellogg Brown & Root
Стив Возняк	Компьютерное оборудование — Apple II	Apple Computer, Inc.
Обзоры		
Анонимный собеседник		
Гордон Белл (Gordon Bell)	Компьютерное оборудование	DEC, Microsoft
Джеррит Блау (Gerrit Blaauw)	Компьютерное оборудование	IBM, Университет Твенте
Гради Буч (Grady Booch)	Программное обеспечение	Rational Software Corp., IBM Rational
Кеннет Брукс (Kenneth Brooks)	Программное обеспечение	DEC, Sparklight.com
Роджер Брукс	Юриспруденция	Cravath, Swaine & Moore
Ричард Кэйс (Richard Case)	Компьютерное оборудование, операционные системы	IBM
Мэри Шоу (Mary Shaw)	Программное обеспечение	Университет Карнеги-Меллона
Иван Сазерленд	Компьютерное оборудование	Evans and Sutherland, Sun Microsystems, Университет штата Орегон
Эоин Вудс (Eoin Woods)	Программное обеспечение	Artechra, Barclays Global Investors
Уильям Райт (William Wright)	Компьютерное оборудование	IBM, Северокаролинский университет

Приложение Б Библиография

1. Aiken, Howard H. [1937]. "Proposed automatic calculating machine". In *Perspectives on the Computer Revolution* [1989], eds. Z. W. Pylyshyn and L. J. Bannon. Norwood, NJ: Ablex Publishing Corp., p. 29–37.
2. Air Force Studies Board, Committee on Pre-Milestone A Systems Engineering, and Paul Kaminsky, Chairman [2008]. *Pre-Milestone A and Early-Phase Systems Engineering*. Washington, DC: National Research Council.
3. Akin, Omer [1988]. "Expertise of the architect". In *Expert Systems for Engineering Design*, ed. M. D. Rychener. New York: Academic Press, p. 173–196.
4. Akin, Omer [2008]. "Variants and invariants of design cognition". In *Design Thinking Research Symposium 7*, eds. J. McDonnell and P. Lloyd. London.
5. Alexander, Christopher [1964]. *Notes on the Synthesis of Form*. Cambridge, MA: Harvard University Press.
6. Alexander, Christopher [1979]. *The Timeless Way of Building*. New York: Oxford University Press.
7. Alexander, Christopher, Sara Ishikawa, and Murray Silverstein [1977]. *A Pattern Language: Towns, Buildings, Construction*. New York: Oxford University Press.
8. Allen, Frances, and John Cocke [1971]. *Design and Optimization of Compilers*. Englewood Cliffs, NJ: Prentice Hall.
9. Allen, Frances, and John Cocke [1972]. "A catalog of optimizing transformations". In *Design and Optimization of Compilers*, ed. R. Rustin. Englewood Cliffs, NJ: Prentice Hall.
10. Amdahl, Gene M., Gerrit A. Blaauw, and F. P. Brooks, Jr. [1964]. "Architecture of the IBM System/360". *IBM Journal of Research and Development* 8 (2), p. 87–101.
11. Arden, Bruce W., Bernard A. Galler, T. C. O'Brien, et al. [1966]. "Program and addressing structure in a time-sharing environment". *Journal of the ACM* 13 (1), p. 1–16.
12. Arthur, K., T. Preston, R. M. Taylor II, et al. [1998]. "Designing and building the PIT: A head-tracked stereo workspace for two users". In *Proceedings of the 2nd International Immersive Projection Technology Workshop*, eds. B. Froelich, J. Deisinger, H.-J. Bullinger, et al. Vienna: Springer Computer Science.
13. Aurisicchio, M., M. Gourtovaia, R. H. Bracewell, et al. [2007]. "Evaluation of how DRed design rationale is interpreted". In *Proceedings of the 16th International*

Conference on Engineering Design (ICED '07), ed. J.-C. Bocquet. Glasgow: The Design Society, p. 63-64.
14. Bacon, Sir Francis [1605]. *The Two Books of the Proficience and Advancement of Learning*.
15. Bacon. Novum Organum, Sive Indicia Vera De Interpretatione Naturae, L., p. 1620.
16. Barkstrom, Bruce R. [2004, updated Jan. 29, 2004]. "The standard Waterfall Model for systems development". Retrieved April 11, 2008, from http://web.archive.org/web/20050310133243/; http://asd-www.larc.nasa.gov/barkstrom/public/The_Standard_Waterfall_Model_For_Systems_Development.htm.
17. Bell, C. Gordon [2008]. "Q & A: IT vet Gordon Bell talks about the most influential computers". *ComputerWorld*, April 29.
18. Bell, C. Gordon, and Allen Newell [1971]. *Computer Structures: Readings and Examples*. New York: McGraw-Hill.
19. Bell, C. Gordon, J. Craig Mudge, and John E. McNamara [1978]. *Computer Engineering: A DEC View of Hardware Systems Design*. Maynard, MA: Digital Press.
20. Bergin, Thomas J., and Richard G. Gibson, eds. [1996]. *History of Programming Languages*, vol. 2. Reading, MA: Addison-Wesley (ACM Press).
21. Billington, David P. [2003]. *The Art of Structural Design: A Swiss Legacy*. Princeton, NJ: Princeton University Art Museum.
22. Blaauw, G. A. [1965]. "Door de vingers zien". Inaugural address at Twente Technical University. Enschede, Netherlands: Technische Hogeschool Twente.
23. Blaauw, G. A. [1970]. "Hardware requirement for the Fourth Generation". In *Fourth Generation Computers*, ed. F. Gruenberger. Englewood Cliffs, NJ: Prentice Hall, p. 155–168.
24. Blaauw, Gerrit A., and Frederick P. Brooks, Jr. [1964]. "Outline of the logical structure of System/360". *IBM Systems Journal* 3 (2), p. 119–135.
25. Blaauw, Gerrit A., and Frederick P. Brooks, Jr. [1997]. *Computer Architecture: Concepts and Evolution*. Reading, MA: Addison-Wesley.
26. Blum, Bruce I. [1996]. *Beyond Programming*. Oxford: Oxford University Press.
27. Boedker, S., P., J. Ehn, M. Kammersgaard, et al. [1987]. "A utopian experience: on design of powerful computer-based tools for skilled graphic workers". In *Computers and Democracy: A Scandinavian Challenge*, eds. G. Bjerknes, P. Ehn, M. Kyng, et al. Avebury, UK: Aldershot, p. 251–278.
28. Boehm, Barry [1988]. "A spiral model of software development and enhancement". *Computer* 21 (5), p. 61–72.
29. Boehm, Barry [2007]. *Software Engineering: Barry Boehm's Lifetime Contributions to Software Development, Management and Research*, ed. R. Selby. New York: John Wiley/IEEE Press.
30. Boehm, Barry W., Terence E. Gray, and Thomas Seewaldt [1984]. "Prototyping versus specifying: A multiproject experiment". *IEEE Transactions on Software Engineering*. SE-10 (3), p. 290–303.

31. Booch, Grady [2009]. "Handbook of software architecture". Retrieved July 22, 2009, from http://www.handbookofsoftwarearchitecture.com/index.jsp?page=Main.
32. Bracewell, R. H., and K. M. Wallace [2003]. "A tool for capturing design rationale". *Proceedings of the 14th International Conference on Engineering Design (ICED '03)*. Stockholm: The Design Society.
33. Britton, Edward, James S. Lipscomb, Michael Pique, et al. [1981]. *The GRIP-75 Man-Machine Interface.* Invited videotape presented at 1981 SIGGRAPH conference. ACM SIGGRAPH.
34. Brooks, F. P., Jr. [1956]. "The analytic design of automatic data processing systems". PhD dissertation, Harvard University Computation Laboratory, Cambridge, MA.
35. Brooks, F. P., Jr. [1972]. "Brooks beach house design". From http://www.cs.ucl.ac.uk/staff/S.Stumpf/DR.html.
36. Brooks, F. P., Jr., and Kenneth E. Iverson [1969]. *Automatic Data Processing: System/360 Edition*. New York: John Wiley.
37. Brooks, F. P., Jr., and Michael Pique [1985]. "Computer graphics for molecular studies". In *Molecular Dynamics and Protein Structure*, ed. J. Hermans. Chapel Hill, NC: University of North Carolina (distributed by Polycrystal Book Service), p. 109.
38. Brooks, F. P., Jr. [1964]. *"NPL Announcement Sprint": Letters to W. C. Hume, B. O. Evans, H .D. Ross, Jr.* Poughkeepsie, NY: IBM Processor Office.
39. Brooks, F. P., Jr. [1965]. "The future of computer architecture". In *Proceedings of IFIPS Congress '65.* Amsterdam: Elsevier North Holland.
40. Brooks, F. P., Jr. [1975, 1995]. *The Mythical Man-Month: Essays on Software Engineering*. Reading, MA: Addison-Wesley.
41. Brooks, F. P., Jr. [1977]. "The computer 'scientist' as toolsmith: Studies in interactive computer graphics". *Proceedings of International Federation of Information Processing Congress '77*, ed. B. Gilchrist. Amsterdam: Elsevier North Holland.
42. Brooks, F. P., Jr. [1986]. "No silver bullet: Essence and accident in software engineering" (reprinted in Brooks [1995]). In *Information Processing 1986, Proceedings of the IFIPS Tenth World Computer Conference*, ed. H.-J. Kugler. Amsterdam: Elsevier Science, p. 1069–1076.
43. Brooks, F. P., Jr. [1996a]. "Keynote address: Language design as design". In *History of Programming Languages*, vol. 2, eds. T. J. Bergin and R. G. Gibson. Boston: Addison-Wesley (ACM Press), p. 4–16.
44. Brooks, F. P., Jr. [1996b]. "The computer scientist as toolsmith II". (Keynote/Newell Award address at SIGGRAPH 94.) *Communications of the ACM* 39 (3), p. 61–68.
45. Brooks, F. P., Jr. [1999]. "What's real about virtual reality?" *IEEE Computer Graphics and Applications* 19 (6), p. 16–27.
46. Brooks, F. P., Jr. [2002]. "The history of IBM Operating System/360". In *Software Pioneers: Contributions to Software Engineering*, eds. M. Broy and E. Denert. Berlin: Springer, p. 170–178.

47. Brooks, Kenneth P. [1988]. "A two-view document editor with user-definable document structure". PhD dissertation, Stanford University, Palo Alto, CA.
48. Brooks, Kenneth P. [1991]. "A two-view document editor". *Computer* 24 (6), p. 7–19.
49. Broy, M., and Ernst Denert, eds. [2002]. *Software Pioneers: Contributions to Software Engineering*. Berlin: Springer.
50. Buchholz, Werner, ed. [1962]. *Planning a Computer System: Project Stretch*. Hightstown, NJ: McGraw-Hill.
51. Burge, J., and D. C. Brown [2008]. "Software engineering using RATionale". *Journal of Systems and Software* 81 (3), p. 395–413.
52. Burks, Arthur W., Herman H. Goldstine, and John von Neumann [1946]. "Preliminary discussion of the logical design of an electronic computing instrument". In *Collected Works of John von Neumann* [1963], vol. 5, ed. A. H. Taub. New York: Macmillan, 5, p. 34–79. Also at http://research.microsoft.com/en-us/um/people/gbell/Computer_Structures_Readings_and_Examples/00000112.html.
53. Buschmann, Frank, Regine Meunier, Hans Rohnert, et al. [1996]. *Pattern-Oriented Software Architecture: A System of Patterns*. New York: John Wiley.
54. Bush, Vannevar [1945]. "That we may think". *Atlantic Monthly* 176 (1), p. 101–108.
55. Buxton, William, and B. Myers [1986]. "A study in two-handed input". *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* New York: ACM, p. 321–326.
56. Chen, Kuohsiang, and Charles L. Owen [1997]. "Form language and style description". *Design Studies* 18 (3), p. 249–274.
57. Chesterfield, Lord [1774]. *Lord Chesterfield's Letters*.
58. Clark, Nicola [2006]. "The Airbus saga: Hubris and haste snarled the A380". *International Herald Tribune*, December 11.
59. Clarkson, John, and Mari Huhtala, eds. [2005]. *Engineering Design: Theory and Practice—A Symposium in Honour of Ken Wallace*. Cambridge, UK: University of Cambridge Engineering Design Centre.
60. Cockburn, Alistair [2000]. *Writing Effective Use Cases*. Boston: Addison-Wesley.
61. Cockburn, Alistair, and Laurie Williams [2001]. "The costs and benefits of pair programming". In *Extreme Programming Examined*, eds. G. Succi and M. Marchesi. Boston: Addison-Wesley, p. 223–248.
62. Cocke, John, and Harwood Kolsky [1959]. "The virtual memory in the STRETCH computer". In *AFIPS Eastern Joint Computer Conference.* New York: ACM, 16, p. 82–93.
63. Cocke, John, and Jacob T. Schwartz [1970]. *Programming Languages and Their Compilers: Preliminary Notes*. New York: Courant Institute of Mathematical Sciences.
64. Codd, E. F., E. S. Lowry, E. McDonough, et al. [1959]. "Multiprogramming STRETCH: feasibility considerations". *Communications of the ACM* 2 (11), p. 13–17.
65. Conklin, J., and M. L. Begeman [1988]. "gIBIS: A hypertext tool for exploratory policy discussion". *ACM Transactions on Information Systems* 6 (4), p. 303–331.

66. Conner, Brookshire D., Scott S. Snibbe, Kenneth P. Herndon, et al. [1992]. "Three-dimensional widgets". In *Proceedings of the 1992 Symposium on Interactive 3D Graphics*. Cambridge, MA: ACM, p. 183–188.
67. Cross, Nigel [1962]. "Research in design thinking". In *Research in Design Thinking*, eds. N. Cross, K. Dorst, and N. Roozenburg. Delft: Delft University Press.
68. Cross, Nigel [1989, 1994, 2000]. *Engineering Design Methods: Strategies for Product Design*. Chichester, UK: John Wiley.
69. Cross, Nigel [2006]. *Designerly Ways of Knowing*. London: Springer.
70. Cross, Nigel, and Anita Clayburn Cross [1996b]. "Winning by design: The methods of Gordon Murray, racing car designer". *Design Studies* 17 (1), p. 91–107.
71. Cross, Nigel, and Kees Dorst [1999]. "Co-evolution of problem and solution spaces in creative design". In *Computational Models of Creative Design*, vol. 4, eds. J. S. Gero and M. L. Maher. Sydney: Key Centre of Design Computing and Cognition, University of Sydney, p. 243–262.
72. Cross, Nigel, ed. [1984]. *Developments in Design Methodology*. Chichester, UK: John Wiley.
73. Cross, Nigel, K. Christiaans, and K. Dorst, eds. [1996a]. *Analysing Design Activity*. Chichester, UK: John Wiley.
74. Cross, Nigel, Kees Dorst, and Norbert Roozenburg, eds. [1992]. *Research in Design Thinking*. Delft: Delft University Press.
75. Davies, Sir David [2000]. *Automatic Train Protection for the Railway Network in Britain: A Study; Report to the Deputy Prime Minister.* London: Royal Academy of Engineering.
76. DeMarco, Tom, and Tim Lister [1987, 1999]. *Peopleware: Productive Projects and Teams*. New York: Dorset House.
77. DeMarco, Tom, Peter Hruschka, Tim Lister, et al. [2008]. *Adrenaline Junkies and Template Zombies: Understanding Patterns of Project Behavior*. New York: Dorset House.
78. Denning, Peter, and Pamela Dargan [1996]. "Action-centered design". In *Bringing Design to Software*, ed. T. Winograd. Reading, MA: Addison-Wesley, p. 110–120.
79. Dennis, Jack B. [1965]. "Segmentation and the design of multiprogrammed computer systems". *Journal of the ACM* 12 (4), p. 589–602.
80. Descartes, Rene [1628]. "Rules for the direction of the mind". In *Philosophical Writings*. London: Thomas Nelson and Sons Ltd., p. 153–180.
81. Dijkstra, Edsger W. [1968]. "A constructive approach to the problem of program correctness". *BIT* 8, p. 174–186.
82. Dijkstra, Edsger W. [1982]. *Selected Writings on Computing: A Personal Perspective*. Berlin: Springer-Verlag.
83. Dornburg, Courtney C., S. M. Stevens, S. M. L. Hendrickson, et al. [2007]. *Improving Human Effectiveness for Extreme-Scale Problem Solving — Final Report (Assessing the Effectiveness of Electronic Brainstorming in an Industrial Setting)*. Albuquerque, NM: Sandia National Laboratories.

84. Dorst, Kees [2006]. "Design problems and design paradoxes". *Design Issues* 22, p. 4–17.
85. Dorst, Kees, and Judith Dijkhuis [1995]. "Comparing paradigms for describing design activity". *Design Studies* 16 (2), p. 261–274.
86. Dorst, Kees, and Nigel Cross [2001]. "Creativity in the design process: Coevolution of problem–solution". *Design Studies* 22 (5), p. 425–437.
87. Eastman, Charles [1997]. [Review of] "Analyzing Design Activity". *Design Studies* 18 (4): 475–476.
88. Economist [2009a]. "Grounded—the airlines and business travel". Economist.com.
89. Economist [2009b]. "Harvest moon: Artificial satellites are helping farmers boost crop yields". *Economist*, November 5.
90. Ettlinger, Steve [2007]. *Twinkie, Deconstructed.* New York: Hudson Street Press.
91. Evans, Bob O. [1986]. "System/360: A retrospective view". *Annals of the History of Computing* 8 (2), p. 155–179.
92. Ferguson, Eugene S. [1992]. *Engineering and the Mind's Eye*. Cambridge, MA: MIT Press.
93. Fowler, H. W. [1926, 1944]. *A Dictionary of Modern English Usage*. Oxford: Oxford University Press.
94. Galle, Per, and Laslo Bela Kovacs [1992]. "Introspective observations of sketch design". *Design Studies* 13 (3), p. 229–272.
95. Gamma, Erich, Richard Helm, Ralph Johnson, et al. [1995]. *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley.
96. Garner, Steve [2001]. "Comparing graphic actions between remote and proximal design teams". *Design Studies* 22 (4), p. 365–376.
97. Garner, Steve [2005]. "Revealing design complexity: Lessons from the open University". *CoDesign* 1 (4), p. 267–276.
98. Gelernter, David H. [1998]. *Machine Beauty: Elegance and the Heart of Technology*. New York: Basic Books.
99. Gerstner, Louis V., Jr. [2002]. *Who Says Elephants Can't Dance? Inside IBM's Historic Turnaround*. New York: Harper Business.
100. Ghemawat, Pankaj [2007]. *Redefining Global Strategy: Crossing Borders in a World Where Differences Still Matter*. Cambridge, MA: Harvard Business School Press.
101. Gibbon Ed. Memoirs of My Life and Writings // Miscellaneous Works of Edward Gibbon, Esq., with Memoirs of his Life and Writings, Composed by himself, Illustrated from his Letters, with occasional Notes and Narrative by the Right Honorable John, Lord Sheffield. In 5,h Vol. 2nd ed. L., 1814. Vol. I. P., p. 225.
102. Glegg, Gordon L. [1969]. *The Design of Design*. Cambridge, UK: Cambridge University Press.
103. Goel, Vinod [1991]. "Sketches of thought: A study of the role of sketching in design problem-solving and its implications for the computational theory of the mind". PhD dissertation, University of California at Berkeley, Berkeley, CA.

104. Goel, Vinod [1995]. *Sketches of Thought*. Cambridge, MA: MIT Press.
105. Goldschmidt, Gabriela [1995]. "The designer as a team of one". *Design Studies* 16 (2), p. 189–210.
106. Gould, John D., and Clayton Lewis [1985]. "Designing for usability: Key principles and what designers think". *Communications of the ACM* 28 (3), p. 300–311.
107. Grad, B. [2002]. "A personal recollection: IBM's unbundling of software and services". *IEEE Annals of the History of Computing* 24 (1), p. 64–71.
108. Greenbaum, Joan, and Morten Kyng, eds. [1991]. *Design at Work: Cooperative Design of Computer Systems*. Hillsdale, NJ: Lawrence Erlbaum Associates.
109. Hales, Crispin [1991]. *An Analysis of the Engineering Design Process in an Industrial Context*. Eastleigh, UK: Gants Hill.
110. Hamming, Richard W. [1963, 1973]. *Numerical Methods for Scientists and Engineers*. New York: McGraw-Hill.
111. Heath, Tom [1989]. "Lessons from Vitruvius". *Design Studies* 10 (3), p. 246–253.
112. Hennessy, John L., and David A. Patterson [1990, 1996, 2002, 2006]. *Computer Architecture: A Quantitative Approach*. San Mateo, CA: Morgan Kaufmann.
113. Herbsleb, James D., Audris Mockus, Thomas A. Finholt, et al. [2000]. "Distance, dependencies, and delay in a global collaboration". In *CSCW '00: Proceedings of the 2000 ACM Conference on Computer-Supported Collaborative Work*. Philadelphia, PA: ACM, p. 319–328.
114. Hickling, Allen [1982]. "Beyond a linear iterative process?" In *Changing Design*, eds. B. Evans, J. A. Powell, et al. Chichester, UK: John Wiley, p. 275–293.
115. Highet, Gilbert [1950]. *The Art of Teaching*. New York: Vintage.
116. Hillier, Bill, and Alan Penn [1995]. "Can there be a domain-independent theory of design?—a comment." Short comment on accepting the *Design Studies* best paper award. They doubt that there can be such a theory.
117. Hinds, P., and S. Kiesler, eds. [2002]. *Distributed Work*. Cambridge, MA: MIT Press.
118. Hoff, Marcian E. (Ted) [1972]. "The one-chip CPU—computer or component?" In *Proceedings of the Computer Systems Design Conference [WESCON]* 16.
119. Hoffman, Daniel M., and David M. Weiss, eds. [2001]. *Software Fundamentals: Collected Papers by David L. Parnas*. Boston: Addison-Wesley.
120. Holson, Laura M. [2009]. "Putting a bolder face on Google". *New York Times*, February 28.
121. Holt, J. E., D. F. Radcliffe, and D. Schoorl [1985]. "Design or problem solving—a critical choice for the engineering profession". *Design Studies* 6 (2), p. 107–110.
122. Howard, Hugh [2006]. *Dr. Kimball and Mr. Jefferson: Rediscovering the Founding Fathers of American Architecture*. New York: Bloomsbury USA.
123. IBM Corp. [1965]. *IBM Operating System/360, Job Control Language. Form C28-6539-0*. Armonk, NY: IBM Corp.
124. IBM Corp. and Bernard Witt [1965]. *IBM Operating System/360, Concepts and Facilities, Form C28-6535-0*. Armonk, NY: IBM Corporation.

125. IBM Corp., Gerrit Blaauw, and Andris Padegs [1964]. *IBM System/360 Principles of Operation. Poughkeepsie, NY, Form A22-6821-0*. Armonk, NY: IBM Corp.
126. IBM Corp., John W. Haanstra, and SPREAD Task Force [1961]. "Processor products—final report of SPREAD Task Group, Dec. 28, 1961". Reprinted in *IEEE Annals of the History of Computing* 5 (January 1983), p. 6–26.
127. Insko, Brent [2001]. "Passive haptics significantly enhances virtual environments". PhD dissertation, University of North Carolina at Chapel Hill, Chapel Hill, NC.
128. Janlert, Lars-Erik, and Erik Stolterman [1997]. "The character of things". *Design Studies* 18 (3), p. 297–314.
129. Jupp, Julie R., and C. M. Eckert [2007]. "A unified framework for analysing decision-making in design: A multi-perspective approach". In *Proceedings of the Conference on Knowledge and Information Management.* New York: ACM.
130. Klein, Gerwin [2009a]. "Operating system verification—an overview". *Sadhana (India)* 34 (1), p. 27–69.
131. Klein, Gerwin, Kevin Elphinstone, Gernot Heiser, et al. [2009b]. "seL4: Formal verification of an OS kernel". In *Proceedings of the 22nd ACM Symposium on Operating Systems Principles*. New York: ACM.
132. Kruchten, Philippe [1999]. "The software architect and the software architecture team". In *Software Architecture*, ed. P. Donohoe. Dordrecht, Netherlands: Kluwer Academic Publications, p. 565–583.
133. Lansdown, John [1987]. "The creative aspects of CAD: A possible approach". *Design Studies* 8 (2), p. 76–81.
134. Lee, Jintai [1993]. "The 1992 workshop on design rationale capture and use". *AI Magazine* 14, p. 24–26.
135. Lee, Jintai [1997]. "Design rationale systems: Understanding the issues". *IEEE Intelligent Systems* 12 (3), p. 78–85.
136. Lehman, Manny M., and Laszlo A. Belady [1971]. "Programming system dynamics". In *ACM SIGOPS Third Symposium on Operating System Principles.* New York: ACM.
137. Lehman, Manny M., and Laszlo A. Belady [1976]. "A model of large program development". *IBM Systems Journal* 3, p. 225–252.
138. Leverett, B. W., R. G. G. Cattell, S. o. Hobbs, et al. [1980]. "An overview of the production-quality compiler-compiler project". *Computer* 13 (8), p. 38–49.
139. Lewis, C. S. [1961]. *An Experiment in Criticism*. Cambridge, UK: Cambridge University Press.
140. Lewis, C. S. [1947]. *Miracles: A Preliminary Study*. San Francisco: Harper Collins.
141. Locke, John [1690]. *An Essay Concerning Human Understanding*. Oxford: Oxford University Press.
142. Lohr, Steve [2009]. "The crowd is wise (when it's focused)". *New York Times*, July 19.
143. Luck, Rachael [2009]. "Does this compromise your design? Socially producing a design concept in talk-in-interaction". Reprinted in McDonnell [2009]. *CoDesign* 5 (1), p. 21–34.

144. MacLean, A., R. M. Young, and T. P. Moran [1989]. "Designing rationale: The argument behind the artifact". In *Proceedings of CHI'89 Conference on Human Factors in Computing Systems.* New York: ACM, p. 247–252.
145. Madison, James [1787]. *Notes on the Debates in the Federal Convention of 1787*.
146. Maher, Mary L., J. Poon, and S. Boulanger [1996]. "Formalising design exploration as co-evolution: A combined gene approach". In *Advances in Formal Design Methods for CAD*, eds. J. S. Gero and F. Sudeweks. London: Chapman and Hall.
147. Maher, Mary Lou, and Hsien-Hui Tang [2003]. "Co-evolution as a computational and cognitive model of design". *Research in Engineering Design* 14 (1), p. 47–63.
148. Margolin, Victor, and Richard Buchanan, eds. [1995]. *The Idea of Design*. Cambridge, MA: MIT Press.
149. McDonnell, Janet, and Peter Lloyd, eds. [2008]. *About Designing: Analysing Design Meetings.* Leiden: CRC Press/Balkema.
150. McManus, John, and Trevor Wood-Harper [2003]. *Information Systems Project Management: Methods, Tools and Techniques*. London: Financial Times Management.
151. Meehan, Michael, Brent Insko, Mary C. Whitton, et al. [2002]. "Physiological measures of presence in stressful virtual environments". *ACM Transactions on Graphics, Proceedings of ACM SIGGRAPH 2002* 21 (3), p. 645–652.
152. Menn, Christian [1996]. "The place of aesthetics in bridge design". *Structural Engineering International* 6 (2), p. 93–95.
153. Mills, Harlan D. [1971]. "Top-down programming in large systems". In *Debugging Techniques in Large Systems*, ed. R. Rustin. Englewood Cliffs, NJ: Prentice Hall.
154. Mills, Harlan D., M. Dyer, and R. Linger [1987]. "Cleanroom software engineering". *IEEE Software* 4 (5), p. 19–25.
155. Moran, Thomas P., and John M. Carroll, eds. [1996]. *Design Rationale: Concepts, Techniques, and Use*. Mahwah, NJ: Lawrence Erlbaum Associates.
156. Mosteller, Frederick, and D. L. Wallace [1964]. *Inference and Disputed Authorship: Federalist Papers*. Reading, MA: Addison-Wesley.
157. Muir Wood, Sir Alan [2007]. "Strategy for risk management". In *Tunneling 2007*.
158. Murray, Charles J. [1997]. *The Supermen: The Story of Seymour Cray and the Technical Wizards Behind the Supercomputer*. New York: John Wiley.
159. Naur, Peter, and Brian Randell [1968]. "Software engineering: Report of a conference sponsored by the NATO Science Committee". NATO Software Engineering Conference, Garmisch, DE. Scientific Affairs Division, NATO.
160. Noble, Douglas, and Horst W. J. Rittel [1988]. "Issue-based information systems for design". In *Proceedings of the ACADIA '88 Conference.* Ann Arbor, MI: Association for Computer Aided Design in Architecture.
161. Osborn, Alexander F. [1963]. *Applied Imagination: Principles and Procedures of Creative Problem Solving*. New York: Charles Scribner's Sons.
162. Pahl, G., and W. Beitz [1984, 1996, 2007]. *Engineering Design: A Systematic Approach*. Berlin: Springer-Verlag.

163. Pahl, Gerhardt [2005]. "VADEMECUM—recommendations for developing and applying design methodologies". In *Engineering Design: Theory and Practice—A Symposium in Honour of Ken Wallace*, eds. J. Clarkson and M. Huhtala. Cambridge, UK: University of Cambridge Engineering Design Centre, p. 126–135.
164. Parnas, David L. [1979]. "Designing software for ease of extension and contraction". *IEEE Transactions on Software Engineering* 5 (2), p. 128–138.
165. Parnas, David L. [2001]. *Software Fundamentals: Collected Papers by David L. Parnas*, eds. D. Hoffman and D. Weiss. Boston: Addison-Wesley.
166. Patterson, David [1981]. "RISC I: A reduced instruction set architecture". *Computer Architecture News* 9 (3), p. 443–458.
167. Paulk, Mark C. [1995]. "The evolution of the SEI's capability maturity model for software". *Software Process: Improvement and Practice* pilot issue (1), p. 3–15.
168. Petroski, Henry [2008]. *Success through Failure: The Paradox of Design*. Princeton: Princeton University Press.
169. Pique, Michael, Jane S. Richardson, and F. P. Brooks, Jr. [1982]. *What Does a Protein Look Like?* Invited videotape presented at 1982 SIGGRAPH Conference.
170. Pugh, Emerson W., Lyle R. Johnson, and John H. Palmer [1991]. *IBM's 360 and Early 370 Systems*. Cambridge, MA: MIT Press.
171. Radin, George [1982]. "The 801 minicomputer". *ACM SIGPLAN Notices* 17 (4), p. 39–47.
172. Radin, George [1983]. "The IBM 801 minicomputer". *IBM Journal of Research and Development* 27 (3), p. 237–246.
173. Raskar, R., G. Welch, M. Cutts, et al. [1998]. "The office of the future: A unified approach to image-based modeling and spatially immersive displays". In *SIGGRAPH '98: The Twenty-fifth Annual Conference on Computer Graphics and Interactive Techniques*. New York: ACM, p. 179–188.
174. Raymond, Eric S. [2001]. "The golden cauldron". In *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol, CA: O'Reilly Media.
175. Raymond, Eric S.[2001]. *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol, CA: O'Reilly Media.
176. Risen, Isadore L. [1970]. "A theory on meetings". *Public Administration Review* 30 (1), p. 90–92.
177. Rittel, Horst, and Melvin Webber [1973]. "Dilemmas in a general theory of planning". *Policy Sciences* 4, p. 155–169.
178. Robertson, Suzanne, and James Robertson [2005]. *Requirements-Led Project Management: Discovering David's Slingshot*. Boston: Addison-Wesley.
179. Robertson, Suzanne, and James Robertson [2006]. *Mastering the Requirements Process*. Boston: Addison-Wesley.
180. Royce, Winston [1970]. "Managing the development of large software systems". In *Proceedings of IEEE Wescon*. New York: IEEE Press.

181. Rybczynski, Witold [1989]. *The Most Beautiful House in the World*. New York: Penguin Group.
182. Salton, Gerald [1958]. "An automatic data processing system for public utility revenue accounting". PhD dissertation, Harvard University Computation Laboratory, Cambridge, MA.
183. Sammet, Jean E. [1969]. *Programming Languages: History and Fundamentals*. Englewood Cliffs, NJ: Prentice Hall.
184. Sayers, Dorothy [1941]. *The Mind of the Maker*. New York: Harcourt Brace Jovanovich.
185. Schoen, Donald [1984]. *The Reflective Practitioner: How Professionals Think in Action*. New York: Basic Books.
186. Schoen, Donald [1986]. *Educating the Reflective Practitioner*. San Francisco: Jossey-Bass.
187. Schoen, Donald A., and Glenn Wiggins [1992]. "Kinds of seeing and their functions in designing". *Design Studies* 13 (2), p. 135–156.
188. Selby, Richard, ed. [2007]. *Software Engineering: Barry Boehm's Lifetime Contributions to Software Development, Management and Research*. New York: John Wiley/IEEE Press.
189. Shannon, Claude, and Warren Weaver [1949]. *The Mathematical Theory of Communication*. Urbana, IL: University of Illinois at Urbana.
190. Shum, Simon J. B., Albert M. Selvin, Maarten Sierhuis, et al. [2006]. "Hypermedia support for argumentation-based rationale: 15 years on from gIBIS and QOC". In *Rationale Management in Software Engineering*, eds. A. H. Dutoit, R. McCall, I. Mistrik, et al. Berlin: Springer-Verlag.
191. Sibly, P. G., and A. C. Walker [1977]. "Structural accidents and their causes". *Proceedings of the Institution of Civil Engineers, Part 1*, 62, p. 191–208.
192. Simon, H. A. [1969, 1981, 1996]. *The Sciences of the Artificial*. Cambridge, MA: MIT Press.
193. Smethurst, Canon A. F. [1967]. *The Pictorial History of Salisbury Cathedral*. London: Pitkin Pictorials.
194. Sonnenwald, Diane H., Mary C. Whitton, and Kelly L. Maglaughlin [2003]. "Evaluating a scientific collaboratory: Results of a controlled experiment". *ACM Transactions on Computer-Human Interaction* 10 (2), p. 150–176.
195. Squires, Arthur [1986]. *The Tender Ship: Governmental Management of Technological Change*. Boston: Birkhauser.
196. Stillinger, Jack [1991]. *Multiple Authorship and the Myth of Solitary Genius.* New York: Oxford University Press.
197. Stoakley, Richard, Matthew Conway, and Randy Pausch [1995]. "Virtual reality on a WIM: Interactive worlds in miniature". In *SIGCHI Conference on Human Factors in Computing Systems*. Denver, CO: ACM Press/Addison-Wesley, p. 265–272.
198. Strassen, Volker [1969]. "Gaussian elimination is not optimal". *Numerische Mathematik* 13, p. 354–356.

199. Sullivan, William G., Pui-Mun Lee, James T. Luxhoj, et al. [1994]. "Survey of engineering design literature: Methodology, education, economics, and management aspects". *Engineering Economist* 40 (1), p. 7–40.
200. Sumner, F. H., G. Haley, and E. C. Y. Chen [1962]. "The central control unit of the 'Atlas' computer". In *Information Processing 1962, Proceedings of the IFIP Congress '62.* Amsterdam: Elsevier North Holland.
201. Svensson, U. P., and U. R. Kristiansen [2002]. "Computational modelling and simulation of acoustic spaces". In *Proceedings of the AES 22nd International Conference: Virtual, Synthetic, and Entertainment Audio*. New York: Audio Engineering Society, p. 1–20.
202. Teasley, S., L. Covi, M. S. Krishnan, and Judith S. Olson [2000]. "How does radical collocation help a team succeed?" In *CSCW '00: Proceedings of the ACM 2000 Conference on Computer Supported Cooperative Work*. New York: ACM, p. 339–346.
203. Thornton, J. E. [1964]. *Design of a Computer—The CDC 6600*. Glenview, IL: Scott, Foresman.
204. Tolkien, John R. R. [1964]. "On fairy-stories". In *Tree and Leaf*. London: George, Allen & Unwin, Ltd., p. 3–84.
205. Torrance, E. Paul [1970]. "Dyadic interaction as a facilitator of gifted performance". *Gifted Child Quarterly* 14 (3), p. 139–143.
206. Tovey, Sir Donald [1950]. "Johann Sebastian Bach". *Encyclopedia Britannica*, vol. 2. Chicago: Encyclopedia Britannica, p. 868–875.
207. Towles, Herman, Wei-Chao Chen, Ruigang Yang, et al. [2002]. "3D telecollaboration over Internet2". In *Proceedings of the International Workshop on Immersive Telepresence (ITP2002)*. New York: ACM.
208. Tucker, Stewart G. [1965]. "Emulation of large systems". *Communications of the ACM* 8 (12), p. 753–761.
209. Tyree, Jeff, and Art Akerman [2005]. "Architecture decisions: Demystifying architecture". *IEEE Software* 22 (2), p. 19–27.
210. Ullman, David G. [1962]. "The foundations of the modern design environment: An imaginary retrospective". In *Research in Design Thinking*, eds. N. Cross, K. Dorst, and N. Roozenburg. Delft: Delft University Press.
211. van der Poel, W. L. [1959]. "zEBRA, a simple binary computer". Reprinted in Bell and Newell [1971], p. 200–204. In *Proceedings of ICIP*. Paris: UNESCO, p. 361–365.
212. van der Poel, W. L. [1962]. *The Logical Principles of Some Simple Computers*. Amsterdam: Excelsior.
213. VDI, Verein Deutscher Ingenieure [1986, 1987]. *VDI-2221: Systematic Approach to the Design of Technical Systems and Products*. Duesseldorf, DE: VDI Verlag.
214. Vincenti, Walter G. [1990]. *What Engineers Know and How They Know It: Analytical Studies from Aeronautical Engineering*. Baltimore, MD: Johns Hopkins University Press.
215. Visser, Willemien [2006]. *The Cognitive Artifacts of Designing*. Mahwah, NJ: Lawrence Erlbaum Associates.

216. Vitruvius, Marcus Vitruvius Pollo [22 BC, 1960]. *De Architectura (The Ten Books on Architecture).* Rome: Dover. Русский перевод: Витрувий (2006) Десять книг об архитектуре. Архитектура-С, 2006.
217. Waldron, Manjula B., and Kenneth J. Waldron [1988]. "A time sequence study of a complex mechanical system design". *Design Studies* 9 (2), p. 95–106.
218. Weisberg, R. W. [1986]. *Creativity: Genius and Other Myths*. New York: Freeman.
219. Wexelblat, Richard L., ed. [1981]. *History of Programming Languages*. New York: Academic Press (ACM Monograph Series).
220. Whitton, Mary C., Benjamin Lok, Brent Insko, et al. [2005]. "Integrating real and virtual objects in virtual environments". In *Proceedings of HCI International 2005*. Berlin: Springer-Verlag, 9.
221. Wilkes, Maurice V. [1985]. *Memoirs of a Computer Pioneer*. Cambridge, MA: MIT Press.
222. Wilkes, Maurice V., and W. Renwick [1949]. "The EDSAC". In *Report of a Conference on High Speed Automatic Calculating Machines*. Cambridge, UK: University Mathematics Laboratory, 9–12.
223. Williams, F. C., and T. Kilburn [1948]. "Electronic digital computers". *Nature* 162, p. 487.
224. Williams, Laurie, Robert R. Kessler, Ward Cunningham, et al. [2000]. "Strengthening the case for pair-programming". *IEEE Software* 17 (4), p. 19–25.
225. Winograd, Terry, John Bennett, Laura De Young, et al., eds. [1996]. *Bringing Design to Software*. New York: ACM Press.
226. Wise, T. A. [1966]. "I.B.M.'s $5,000,000,000 Gamble". *Fortune* 74 (September), p. 118–123; 224–228; (October), p. 138–143; 199–212.
227. Witt, Bernard I., F. Terry Baker, and Everett W. Merritt [1994]. *Software Architecture and Design: Principles, Models, and Methods*. New York: Van Nostrand Reinhold.
228. Wolff, Christoff [2000]. *Johann Sebastian Bach: The Learned Musician*. New York: W. W. Norton.
229. Wozniak, Steve, and Gina Smith [2006]. *iWoz: From Computer Geek to Cult Icon: How I Invented the Personal Computer, Co-Founded Apple, and Had Fun Doing It*. New York: W. W. Norton.
230. Ziman, John M., ed. [2000]. *Technological Innovation as an Evolutionary Process*. Cambridge, UK: Cambridge University Press.
231. Андреа Палладио (1508–1580) *Четыре книги об архитектуре. М., 1936*.
232. Веб-страница http://en.wikipedia.org/wiki/Design_methods.
233. Веб-страница http://www.cs.unc.edu/~brooks/DesignofDesign.
234. Веб-страница http://www.english-heritageshop.org.uk/mall/productpage.cfm/EnglishHeritage/_03567/-/Guidebook%3A%20Kenwood.
235. Леонардо да Винчи (2006) *Записные книжки. Серия: Антология мудрости*. ООО "Издательство "Эксмо".
236. Платон. Сочинения в 4-х томах. М.: Мысль, 1964.

Предметный указатель

A
Allen, Fran, *301*
Amdahl, Gene, *109*; *370*
Aurisicchio, Marco, *244*

B
Babbage, *405*
Baird, Doug, *384*
Baruch, Bernard, *93*
Beitz, *39*
Bell, Gordon, *201*; *380*
Belsky, Martin, *400*
Blaauw, Gerrit, *370*
Blaauw, Gerry, *37*
Bloch, Erich, *21*
Boehm, Barry, *63*; *88*
Booch, Grady, *202*
Boulanger, *83*
Bracewell, Rob, *244*
Brooks, Frederick, *370*
Brooks, Ken, *276*
Buchholz, *376*
Bush, Vannevar, *230*

C
Case, Dick, *400*
Case, Richard P., *90*
Case, Richard, *370*
CFD
  Computational Fluid Dynamics, *96*
Clark, William, *400*
CMM
  Capability Maturity Model, *285*
Cocke, John, *293*; *300*
Conklin, *242*
Corbis, *33*
Cray, Seymour, *67*
Cross, Nigel, *25*

D
DAM
  Direct Access Method, *397*
Davies, David, *290*
Dijkstra, Edsger, *141*
Dorst, Kees, *62*
DSD
  Data Systems Division, *371*

E
Evans, Bob O., *370*
Evans, Bob, *21*

F
Fagg, Peter, *133*
Freeman, Peter A., *28*
FSD
  Federal Systems Division, *373*
Fuchs, Henry, *131*
Fuller, Margaret, *293*

G
Garcia, James, *69*
Gerstner, Lou, *119*
Glegg, Gordon, *28*
Gould, *225*
GPD
  General Products Division, *371*
Grover, George, *370*

H
Haanstra, John, *372*
Harms, William, *370*; *378*
Henderson, Derek, *370*
Hennessy, *201*; *408*
Herwitz, Paul, *370*
Hickling, *84*
Humphrey, Watts, *285*

I
IBIS
  Issue-Based Information System, *242*
Insko, *366*
ISAM
  Indexed Sequential Access Method, *397*
Iverson, Ken, *162*
Iverson, Kenneth, *104*

J
JCL
  Job Control Language, *211*; *396*
Jobs, Steve, *279*

Johnson, Kelly, *102*
Jones, Alex, *333*; *352*
Jones, Graham, *370*
Jupp, Jeffrey, *123*

K

Kaminsky, Paul, *78*
Kinkade, Thomas, *271*
Kolsky, Harwood, *300*
Kruchten, Philippe, *223*

L

Lewis, *225*
Locken, Scott, *400*
Luck, Rachael, *36*

M

Mago, Mary June, *352*
Maher, *83*
Malone, Thomas W., *133*
Mason, Gary, *333*
Mayer, Marissa, *104*
McClure, Wes, *235*
McClure, Wesley, *333*
McManus, John, *79*
Mealy, George, *400*
Mills, Harlan, *119*; *142*; *223*
Mitchell, Joe, *102*
Moss, Michael, *244*
Murray, Gordon, *290*

N

NCCOP
  North Carolina Computer Orientation Project, *417*
Newell, Allen, *201*

O

Opel, John, *302*
Ottoway, Gerald, *378*

P

Padegs, Andris, *370*
Pahl, *39*
PAM
  Partitioned Access Method, *397*
Parkinson, Brad, *118*
Parnas, *54*
Patterson, *201*; *408*
Peacock, Anthony, *370*
PERT
  Program Evaluation and Review Technique, *241*
Petroski, Henry, *210*
Poon, *83*
Prins, Jan, *28*

R

Radin, George, *301*
Redtenbacher, Ferdinand, *39*
Reid, David, *370*
Research Triangle Park, *412*
RISC
  Reduced Instruction Set Computer, *307*
Robinson, Peter, *28*
Royce, Winston, *46*
RPG
  Report Program Generator, *389*
Ruthrauff, Robert, *161*

S

SAM
  Sequential Access Method, *397*
Sayers, Dorothy, *34*
Schoen, Donald, *56*
Schwartz, Jack, *301*
SEI
  Software Engineering Institute, *279*
Shaw, Mary, *77*
Simon, Herbert, *30*
Slater, Mel, *28*
Spaulding, Donald, *370*
SPOOL
  Simultaneous Peripheral Operation On Line, *391*
Squires, *77*
Stevens, William, *370*
Stutts, Stanley, *333*
Sutherland, Ivan, *28*
SVC
  Supervisor Call, *395*

T

Taylor, James, *165*
Tolkien, J. R. R., *24*
Torvalds, Linus, *87*
Trapnell, Fritz, *401*
TUCC
  Triangle Universities Computation Center, *411*

V
Van der Poel, *179*

W
Wallace, Ken, *63*
Weiss, Stephen, *28*
Whitted, Turner, *251*
Wilkes, Maurice, *198*
WIMP
  window, icon, menu, pointing device, *194*
Witt, Bernie, *400*
Wood-Harper, Trevor, *79*
Woods, Eoin, *118*
Wren, *100*
Wright, Frank Lloyd, *185*
Wright, William V., *141*
Wright, William, *124*; *370*; *378*

A
Автоматизации
  проектирования, *46*
Айверсон, Кеннет, *104; 162*
Акустика
  помещения, *274*
Алгоритм
  поиска в глубину, *231*
Александер, Кристофер, *39*
Аллен, Френ, *301*
Амдал, Джин, *109*; *370*
Аристотель, *140*
Архитектор, *76*; *250*
  системный, *103*
Архитектура, *404*
  RISC, *200*
  качественная, *181*
  компьютера, *390; 404*
  программная, *250*
  стека, *376*
  строительная, *250*
Ассемблер, *389*
Ассоциация
  немецких инженеров, *63*
Атака
  мозговая, *106*
Атрибут
  проектный, *47*
Аурисиккио, Марко, *244*

Б
Байт
  восьмибитовый, *375*
  шестибитовый, *375*
Барух, Бернард, *93*
Барьер
  временной, *125*
  пространственный, *125*
  языковой, *126*
Бах, *166*
Башня, *316*
Бейц, *39*
Бекон, Френсис, *33*
Белл, Гордон, *201*; *380*
Белски, Мартин, *400*
Библиотека, *231*
  моделей, *252*
  образцов, *252*
  программ ввода-вывода, *391*
  стартовая, *275*
Бизнес-парк
  "Исследовательский треугольник", *412*
Блау, Джерри, *37; 370*
Блок
  контрольный, *399*
Блох, Эрих, *21*
Боэм, Барри, *63*; *88*
Брейсуэлл, Роб, *243*
Бригада
  хирургическая, *119*
Брукс, Кен, *276*
Брукс, Фредерик, *370*
Брунеллески, *107*
Буанже, *83*
Букстехуде, *194*
Бухгольц, *376*
Буч, Гради, *202*
Буш, Ванневар, *230*
Бэбидж, *405*
Бэрд, Дуг, *384*
Бюджет
  TUCC, *416*

В
Ван дер Пол, *179*
Ввод
  голосовой, *255*
  команд с клавиатуры, *255*
  с помощью голоса, *254*
  с помощью меню, *254*

Веб-адрес
сопровождающего сайта, *190*
Вейс, Стивен, *28*
Веранда, *319*
Вертолет
Comanche, *70*; *77*
Ветвление, *213*
Ветвь
альтернативная, *47*
атрибута, *47*; *234*
Взаимодействие, *34*
двухстороннее, *268*
человеко-машинное, *276*
Видеоконференция, *127*; *130*
Видеосвязь, *130*
Вид
снабжения, *315*
Визуализация
виртуальной среды, *364*
молекул, *223*
Виссер, *66*
Витрувий, *38*; *178*
Витт, Бернард, *401*
Возможности
стиля, *183*
Возняк, Стив, *189*
Волнолом, *320*
Вопрос
проектирования, *234*
Восприятие
глубины, *262*
Встреча
личная, *126*
Вуд-Харпер, Тревор, *79*
Вудс, Эон, *118*
Выработка
требований, *72*
Вычисления
научно-технические, *373*
Выявление
контекста, *269*

Г

Галилей, *140*
Гармоничность, *181*
Гарсия, Джеймс, *69*
Гибкость
проектирования, *65*
Гидрогазодинамика
вычислительная, *96*
Гипотезы
о пользователях, *150*
Глагол
общепринятый, *255*
Глегг, Гордон, *28*
Гомори, Ральф, *300*
Гостиная, *318*
Гоулд, *225*
Графика
компьютерная, *251*
График
PERT, *246*
Гринвуд, Джозеф, *335*
Гровер, Джордж, *370*

Д

да Винчи, Леонардо, *39*
Дейкстра, Эдсгер, *141*
Декарт, Рене, *140*
Декомпозиция
функций, *183*
Дерево
оглавления, *231*
проектирования, *45*; *54*; *232*; *239*
решений, *239*
Джапп, Джеффри, *123*
Джерстнер, Лу, *119*
Джефферсон, Томас, *210*
Джобс, Стив, *279*
Джонс, Алекс, *333*; *352*
Джонс, Грэм, *370*
Джонсон, Келли, *102*
Дисплей
контекстный, *275*
проектирования, *275*
рабочей книги, *275*
тестовых примеров, *275*
трехмерный, *270*
Доказательство
правильности, *142*
формальное, *142*
Документ, *128*
разрабатываемый совместно, *129*
Документация
эксплуатационная, *197*
Документирование
проекта, *188*
хода развития проекта, *230*
Допущение, *152*
Дорст, Кис, *62*

Доска
виртуальная, *131*; *244*
Доставка
по запросу, *132*
принудительная, *132*
Достижение
удовлетворительных результатов, *48*
хорошего стиля, *189*
Доступ
удаленный, *388*
Дэвис, Дэвид, *290*
Дюны, *314*

Е

Единица
измерения вспомогательная, *159*
Единообразие, *183*
микрорешений, *185*
стилей, *100*
стиля, *183*
Естествознание, *251*

Ж

Журнал, *272*
проектирования, *363*

З

Задание, *44*; *396*
глаголов, *255*
техническое, *159*
Задача, *396*
акцидентная, *35*
второстепенная, *35*
существенная, *35*
сущностная, *35*
таксономическая, *253*
Заключение
контракта, *79*
Закон
квадратичной зависимости, *415*
Заметка, *272*
Защита
интеллектуальной собственности, *288*
Значимость
достигнутого успеха, *205*
Зоркость
глаз, *269*
Зрение, *268*

И

Избыточность, *182*
Изделие
LHX, *70*
Изображение
белка виртуальное, *224*
двухмерное, *257*
трехмерное, *270*
Изучение
образцов, *304*
рисков, *90*
Индикатор
визуальный, *268*
Инско, *366*
Институт
программной инженерии, *285*
Инструментарий
интеллектуальный, *232*
Интеллект
искусственный, *25*
коллективный, *132*
Интерфейс
WIMP, *194*
ввода, *255*
ввода-вывода, *376*
ввода двумя руками, *254*
между подзадачами, *99*
оконный интерактивный, *194*
пользовательский, *104*
стандартный, *125*
человеко-машинный, *250*
Интерьер
здания, *271*
Использование
контекста, *256*
Исследование
образцов, *197*
проектирования, *25*
теоретическое, *203*
Исследователь
проблем проектирования, *23*
Истребитель
F-117, *102*

К

Кадастр, *315*
Камински, Пол, *78*
Канал
ввода-вывода, *378*
Карьера
академическая, *307*

Качество
концептуального проектирования, *242*
Кинкэйд, Томас, *271*
Клавиатура
алфавитно-цифровая, *258*
цифровая, *259*
Кларк, Уильям, *400*
Книга
Computer Architecture, *405*
Когсвелл, Артур, *313*
Кодд, Тэд, *223*
Кок, Джон, *293*; *300*
Коллекция
Sweets File, *273*
Sweets Network, *273*
образцов, *201*
Колода
перфокарт, *214*
пыльная, *212*
Колски, Харвуд, *300*
Команда
голосовая, *255*
Комитет
SPREAD, *372*
Коммуникатор
iPhone, *282*
Компания
Google, *104*
Компонуемость, *183*
Компьютеры
System/360, *124*
Компьютер
Baby, *198*
Cambridge Atlas, *378*
EDSAC, *198*
Harvest, *21*
Stretch, *21*
UNIVAC, *377*
Whirlwind, *196*
первого поколения, *198*
третьего поколения, *199*
Конклин, *242*
Контекст, *256*
Контракт, *75*
на проектирование, *76*
на строительство, *76*
Контроль
доступа, *394*
ошибок, *377*
Концепция
проекта, *36*
проекта структурная, *180*
Корбис, *33*
Корректировка
атрибутов, *251*
документа в режиме совместного использования, *129*
Краткость, *179*
Крачтен, Филипп, *223*
Крейслер, Фриц, *189*; *191*
Критерий
краткости, *180*; *182*
проектирования, *230*
уместности, *182*
Кромвель, Оливер, *209*
Кросс, Найджел, *25*
Крэй, Сеймур, *67*
Кули, Майкл, *219*
Купол
Брунеллески, *108*
Кэйс, Дик, *400*
Кэйс, Ричард П., *90*; *370*

Л

Ла Гардиа, Фиорелло, *209*
Лак, Рэйчел, *36*
Лестница
винтовая, *312*
Линеаризация
представления знаний, *230*
пространственного графа, *231*
Локализация, *259*
Локен, Скотт, *400*
Локк, Джон, *139*
Льюис, *225*

М

Маго, Мэри Джун, *352*
Майер, Марисса, *104*
Макет, *363*
в натуральную величину, *352*
Макклюр, Вес, *235*
Макклюр, Уэсли, *333*
Макманус, Джон, *79*
Макроассемблер, *389*
Макрокоманда, *389*
Манипулятор, *261*
EyeBall, *261*
Машина
вычислительная, *405*
Маэр, *83*
Метафора, *180*

Методика
работы, *233*
Методология
PERT, *241*
проектирования, *141*
Метод
"стерильных условий", *142*
CFD, *97*
анализа критического пути, *241*
доступа, *388*
доступа библиотечный, *397*
доступа индексно-последовательный, *397*
итерации целей, *53*
коробчатой конструкции, *146*
нисходящий, *250*
последовательного доступа, *397*
прямого доступа, *397*
разработки адаптивный, *225*
Механизм
рынка, *85*
Микеланджело, *165*
Микрокомпьютер, *200*
Микрорешение, *151*; *185*
Мили, Джордж, *400*
Миллс, Харлан, *119*; *142*; *223*
Мини-компьютер
DEC, *196*
Мир
виртуальный, *135*
виртуальный Second Life, *135*
Митчелл, Джо, *102*
Моделирование, *274*
акустическое, *274*
Модель
CMM, *285*
Паля и Бейца, *60*
Саймона, *60*
водопада, *46*; *60*; *424*
детализированная, *251*
естественная, *60*
использования, *150*; *154*
каркасная, *82*
карусели, *84*
компьютерная, *253*
компьютерная графическая, *82*
коэволюционная, *74*; *83*
ленточная, *82*
масштабная, *352*
неопределенная, *153*
неправильная, *153*
организационная, *412*
подвижная, *82*
пользователя, *150*
приложения, *150*
продукта, *111*
проектирования рациональная, *46*
процесса проектирования, *44*
процесса проектирования формальная, *89*
развития, *88*
разработки Реймонда, *84*
рациональная, *60*; *425*
спиральная, *74*
спиральная Боэма, *88*
структурированная, *252*
твердотельная, *82*
технологической зрелости, *285*
эволюционная, *86*
явная, *152*
Мосс, Майкл, *244*
Моцарт, Вольфганг Амадей, *35*
Моцарт, Леопольд, *35*
Мультипрограммирование, *394*
Мышление
правополушарное, *64*
Мэдисон, *246*
Мэйнфрейм, *380*
System/360, *196*
Мэйсон, Гэри, *333*
Мэлоун, Томас В., *133*
Мюррей, Гордон, *290*

Н

Наблюдатель, *260*
виртуальный, *260*
Наблюдение, *25*
Набор
операций, *375*
требований минимальный, *72*
Надежность, *398*
Название
System/360, *393*
Накопление
опыта, *55*
Нарушение
графика, *73*
Наставник, *296*
Наука
проектирования, *24*
Недостатки
JCL, *212*

Непротиворечивость, *100*
Новизна, *194*
Ньюэлл, Аллен, *201*

О

Обеспечение
  телеобработки, *412*
Обзор
  проекта, *110*
Обмен
  структурированными данными, *100*
Обоняние, *268*
Обоснование
  проектирования, *246*
Обработка
  бизнес-данных, *373*
  дистанционная, *388*
  журнала, *233*
  многопроцессорная, *377*
Образец, *194*
  творчества, *305*
Образование
  академическое, *295*
  архитектурное, *295*
  медицинское, *294*
Обслуживание
  техническое, *328*
Обстоятельство, *259*
Обучение
  проектированию, *294*
  проектировщиков, *197*; *225*; *294*
  систематическое, *298*
Обшивка
  деревянная, *317*
Общность, *100*; *182*
Объект
  искусственный, *251*
  математический, *141*
  проектирования, *230*
Оглавление, *231*
Ограничение
  бюджетное, *341*
  проектное, *341*
Ограничения, *44*; *166*; *314*
  действительные, *167*
  искусственные, *167*
  неактуальные, *167*
  неправильно воспринятые, *167*
  устаревшие, *167*
Опель, Джон, *302*
Операция
  вызова супервизора, *395*
  позиционирования, *258*
Описание, *25*
  стиля, *187*
Определение
  наречий, *259*
  объектов, *256*
  по имени, *256*
  понятия дерева проектирования, *234*
  понятия стиля, *184*
  проектирования, *34*
  текста, *258*
  требований, *285*
Организация
  военная, *296*
  матричная, *404*
  проектная, *270*
  промышленная, *296*
Оригинальность, *205*
Ортогональность, *100*; *182*
Освещение
  внутреннее, *271*
  рассеянное, *264*
  резкое, *264*
  сцены, *264*
Осмотр
  внешний, *271*
  снаружи, *271*
Осуществимость
  идеальной системы, *275*
Осуществление, *37*; *404*
Осязание, *268*
Отделение
  DSD, *371*
  FSD, *373*
  GPD, *371*
Открытость, *183*
Отображение
  дерева на линейный граф, *231*
  трехмерного пространства, *131*
Отопление, *316*
Отпуск
  творческий, *300*
Отражение, *25*
Отступ
  здания от соседних участков, *314*
Отсутствие
  итерации, *213*
Оттоуей, Джеральд, *378*

Отчет
текстовый, *244*
Оценка
производственных затрат, *285*
Очки
стереоскопические, *270*

П

Падегс, Андрис, *370*
Пакет
Disk Operating System/360, *390*
DOS/360, *390*
Палладио, Андреа, *39*
Паль, *39*
Память
виртуальная, *199*; *378*
Парадокс
проектирования, *171*
Параметр
возвышения, *262*
динамический, *260*
статический, *260*
Паркинсон, Брэд, *118*
Парнас, *54*
Паттерсон, *201*; *408*
Пентагон, *70*
Переключатель
режима просмотра, *270*
Перфокарта, *215*
управляющая, *215*
Петроски, Генри, *210*
Пикок, Энтони, *370*
Планировщик, *391*
Планшет
графический, *269*
План
общий, *270*
Платон, *36*
Площадь
здания, *159*
Поверхность
рабочая, *269*
Подготовка
специалиста, *296*
Подсказка
визуальная, *131*
Позитивизм, *66*
Позиционирование, *258*
Позиция
договорная, *88*
Поле
обзора, *130*
Полнота
наборов функций, *183*
Последовательность
изложения, *407*
Постановка
целей, *106*
Почта
электронная, *128*
Право
принятия решений, *412*
Предложение
повелительное, *254*
Предотвращение
ошибок, *282*
Предписание, *25*
Предположения, *152*
неявные, *151*
явные, *151*
Представительство
пропорциональное, *420*
равное, *420*
Представление
двухмерное, *257*
детализированное, *269*
знаний, *232*
контекстное, *269*
модели проектирования визуальное, *89*
пространственное, *270*
рабочей книги, *271*
спецификации, *273*
трехмерное, *262*; *270*
Презентация
PowerPoint, *119*
Преимущества
образцов, *203*
сотрудничества, *94*
Премия
Тьюринга, *104*
Престиж, *85*
Приближение
последовательное, *251*
Пример
использования, *154*
Принс, Ян, *28*
Принцип
“разных лестниц”, *297*
возведения собора, *84*
идентификации компонентов, *100*

качества, *181*
ортогональности, *182*
проектирования, *182*; *225*
функционирования базара, *84*
эволюционный, *86*
Проведение
конкурса, *107*
Прогнозирование
оптимальной продажной цены, *285*
Прогноз
маркетинговый, *384*
рыночный, *285*; *389*
Программа
"сэндвичевая", *220*
1-3-1, *220*
Compendium, *233*; *241*
DRed, *243*
Excel, *181*
Fetchmail, *85*
gIBIS, *242*
Lotus, *181*
MacDraw, *104*
MacPaint, *104*
Microsoft Project, *241*
OmniPlan, *241*
SmartDraw, *241*
Task Architect, *241*
VisiCalc, *181*
заслуживающая доверия, *391*
компьютерная, *141*
проектирования, *58*
распознавания речи, *255*
Программирование
инкрементное, *424*
объектно-ориентированное, *399*
Прогресс
научно-технический, *84*
Проектирование, *35*; *219*
автоматизированное, *352*
адаптивное, *40*
высокотехнологичное, *54*
зданий, *75*; *250*
инкрементное, *424*
интуитивное, *23*
итерационное, *57*
коллективное, *94*; *150*
комитетом, *114*
концептуальное, *76*
массовое, *40*
машинное, *352*
механическое, *250*
оригинальное, *40*
программного обеспечения, *173*
систематическое, *23*
системное, *40*
строительное, *173*
художественное, *40*
целенаправленное, *84*
Проектировать, *35*
Проектировщик, *23*
языков программирования, *179*
Проект
NCCOP, *417*
System/360, *283*
View/360, *37*
выдающийся, *281*
детализированный, *252*
концептуальный, *76*
модульный, *240*
новаторский, *281*
самообразования, *306*
сложный, *151*
эскизный, *159*
Проекция
двухмерная, *257*
Прозрачность, *182*
Пространство
проектирования, *45*; *171*; *232*
рабочее общее, *134*
функциональное, *251*
Протоколы
Delft II, *67*
Прототип, *76*; *194*; *251*
Процедура
передачи ответственности, *128*
Процесс, *281*
организованный, *286*
повышения квалификации, *296*
проектирования, *23*; *232*; *251*; *285*
разработки продуктов, *281*
согласования, *283*
Пун, *83*
Пуньяни, *189*

Р

Работа
бесперебойная, *301*
круглосуточная, *122*
на дому, *122*
научно-исследовательская, *275*

совместная, *94*
Радин, Джордж, *301*
Развитие
навыков, *303*
стиля, *187*
Разделение
времени, *389*
задач проектирования на подзадачи, *99*
Раззак, Шариф, *229*
Размер
адреса, *374*
байта, *379*
дисплея, *234*
Разнообразие
приложений, *412*
Разработка
моделей, *251*
новаторская, *194*
параллельная, *225*
плана дома, *251*
программы, *251*
прототипа быстрая, *53*
с открытым исходным кодом, *84*
Райт, Уильям В., *124*; *141*; *370*; *378*
Райт, Фрэнк Ллойд, *185*
Раскадровка, *38*
Расписание
учебное, *295*
Распределение
бюджета, *45*
критического бюджета, *45*
масс, *262*
ресурсов, *45*
Расстояние
просмотра изображения, *269*
Расширение
Common Lisp, *179*
сферы применения, *375*
Ратрофф, Роберт, *161*
Рационализм, *140*
Рационалист, *140*
Рациональность
техническая, *66*; *294*
Реализация, *34*; *37*; *404*
микропрограммная, *377*
проекта, *328*
Регистр
базовый, *379*
Редтенбахер, Фердинанд, *39*
Режим
дистанционный, *94*; *415*
многозадачный, *388*
отслеживания изменений, *129*
Рейд, Дэвид, *370*
Рекомендации
проектные, *202*
Рен, *100*
Респиги, *189*
Ресурс
критический, *45*; *162*
лимитируемый, *158*; *312*; *328*; *412*
нормированный, *159*
ограниченный, *158*
оптимизируемый, *158*
проекта, *161*
процесса проектирования, *161*
Речь, *268*
Решение
в чистом виде, *312*; *332*; *370*
ключевое, *332*
на микроуровне, *185*
Рисование
эскизов, *257*
Ритмика
глагола–существительного, *254*
Робертсон, Джеймс, *64*
Робертсон, Сюзанна, *64*
Робинсон, Питер, *28*
Ройс, Уинстон, *46*
Росселлини, Антонио, *166*
Рост
наблюдателя, *262*
Руководитель
дизайн-проекта, *23*
проекта, *161*
Руководство
по проектированию, *202*
по стилю, *188*
Рыбчински, Витольд, *311*

С

Сазерленд, Айвэн, *28*
Саймон, Герберт, *30*
Самолет-шпион
U-2, *102*
Самолет
Airbus 380, *123*
Spitfire, *102*
Самонадеянность, *216*
Сатисфакция, *206*

Семейство
IBM System/360, *124*
компьютеров System/360, *390*
Сетка
координатная, *268*
Систематизация
образцов, *202*
процессов проектирования, *24*
Система
GPS, *118*
GRIP, *257*
IBIS, *242*
Lilac, *276*
Memex, *230*; *246*
MiniCad, *363*
OS/360, *37*
System/360, *37*
военного назначения, *72*
графическая UNC GRIP, *250*
десятичная Дьюи, *231*
дистанционной обработки, *394*
инкрементной разработки и итерационной поставки, *223*
нумерации библиотеки Конгресса США, *231*
операционная, *391*
операционная пакетная, *391*
операционная Linux, *85*
операционная Mac, *104*
операционная MVS, *39*
операционная UNIX, *87*
отладки, *399*
проектирования, *250*; *253*
с разделением времени, *394*
управления данными, *395*; *398*
Ситтерсон, *207*
Сквайрс, *77*
Слейтер, Мэл, *28*
Служба
курьерская, *412*
Слух, *268*
Собственность
интеллектуальная, *246*
Совещание, *283*
Совместимость, *374*
на уровне двоичных кодов, *373*
программ, *404*
Согласование
междисциплинарное, *103*
Согласованность, *100*
Соглашение
письменное, *75*
Создание
архитектуры, *37*
Сокращение
дерева проектирования, *55*
Сопровождение
звуковое, *273*
Соревнование
по проектированию, *376*
Сотрудничество
в проектировании, *132*
дистанционное, *122*
Сохранение
контекста, *256*
Специализация, *122*
инженерной деятельности, *221*
Спецификация
текстовая, *273*
Сполдинг, Дональд, *370*
Способность
разрешающая, *131*; *269*
Способ
решения задачи проектирования, *230*
Справочник, *405*
Среда
виртуальная, *222*; *352*
Средство
автоматизированного проектирования, *363*
организации связи, *128*
Ссылка
перекрестная, *231*
Стандарт, *202*
IEEE/EIA 12207.0, *90*
IEEE/EIA 12207.1, *90*
IEEE/EIA 12207.2, *90*
JCL, *216*
Становление
проектирования, *220*
Статтс, Стэнли, *333*
Стевенс, Уильям, *370*
Стиль, *183*
проектирования, *187*; *296*
Стоимость
бюджетная, *45*
изучения и обучения, *99*
исправления ошибок, *127*
проектирования, *24*
разделения на подзадачи, *99*
управления изменениями, *100*

Стол
рабочий, *181*
Стратегия
поиска в глубину, *46*
сокрытия информации, *399*
Структура
многомерная, *232*
Структурирование
библиотеки, *253*
Супервизор, *391*
Суперкомпьютер
Cray, *199*
Stretch, *198*
Схема
заключения контрактов, *79*
преобразования, *233*; *243*
Сценарий
JCL, *211*
использования, *348*
практической эксплуатации, *222*
эксплуатации, *251*
Сэйерс, Дороти, *34*

Т

Таблица
определения требований, *73*
электронная, *181*
Телеобработка, *373*
Телеработа, *123*
Телефон, *129*
мобильный, *282*
Термин
Operating System/360, *390*
OS/360, *390*
Технология
Stealth, *102*
дискретно-транзисторная, *371*
дистанционного сотрудничества, *122*
интегральных схем, *372*
передачи данных, *126*
Тип данных
десятичный, *39*
Тип
данных абстрактный, *388*; *390*
Толкиен, Дж. Р. Р., *24*
Торвальдс, Линус, *87*
Торренс, *115*
Точка
контрольная, *394*
Требования, *44*
к образцам, *197*
Трумен, Гарри С., *351*
Трэпнелл, Фриц, *401*
Тэтчер, Маргарет, *119*

У

Узел
вопроса проектирования, *234*
листовой, *234*
описания, *235*
решения, *234*
Уилкс, Морис, *198*
Уиттед, Т„урнер, *251*
Указатель, *257*
авторов, *231*
визуальный, *261*
названий, *231*
предметный, *231*
тематический, *231*
Ульман, *250*
Уместность, *100*; *182*
Уоллис, Кен, *63*
Уолф, *166*
Управление
данными, *396*
изменениями, *100*
риском, *88*
супервизорное, *376*
Усиление
конкурентной борьбы, *98*
Условия
совместной работы, *105*
Устройство
EyeBall, *260*
Утилита, *392*
Участие
в реализации, *224*
Участок
строительный, *314*

Ф

Фагт, Питер, *133*
Файл
MiniCad, *363*
Фасад, *318*
Формат
команд SS, *382*
Формула
оценки, *55*
Формулировка, *34*

Фримен, Питер А., *28*
Фукс, Генри, *131*
Фуллер, Маргарет, *293*
Функция
  качества, *44*; *55*
  ортогональная, *182*
  полезности, *44*; *106*

Х

Хаанстра, Джон, *372*
Хармс, Уильям, *370*; *378*
Хеннесси, *201*; *408*
Хиклинг, *84*
Холт, *66*
Хэмфри, Уотс, *285*

Ц

Целостность
  концептуальная, *94*; *289*
  концепции, *38*
Цель, *313*
  основная, *44*; *313*
  проектирования, *236*
Центр
  TUCC, *416*
  вычислительный, *412*
  вычислительный ассоциации трех университетов, *411*

Ч

Чертеж, *352*
  двухмерный, *268*
  изометрический, *363*
Четкость
  стиля, *185*

Ш

Шварц, Джек, *301*
Шеннон, Клод, *180*
Шоу, Мэри, *77*
Штурм
  мозговой, *107*
Шён, Доналд, *56*

Э

Эванс, Боб О., *21*; *370*
Эволюционность
  модели, *83*
Эволюция, *83*
  биологическая, *84*
Эксплуатация
  компьютера, *391*
  периферийных устройств, *391*
Экстерьер, *315*
  здания, *271*
Эмпиризм, *140*
Эмпирик, *140*
Эмулятор, *404*
  исполняемый, *404*
Эндерсон, Дерек, *370*
Эпистемология, *144*
Эрвитц, Пол, *370*
Эскиз, *363*
Эстетика, *178*
Эффект
  глубины, *263*
  кинетический, *263*
  побочный, *182*
  присутствия, *131*

Я

Ядро, *142*
  операционной системы, *142*
  языка Lisp, *179*
Язык
  APL, *180*; *404*
  COBOL, *389*
  Fortran, *389*
  JCL, *211*
  Lisp, *179*
  PL/I, *393*
  RPG, *389*
  Visual Basic, *179*
  высокого уровня, *392*
  планирования, *212*
  программирования, *389*
  программирования APL, *104*
  сценариев, *211*
  управления заданиями, *396*
  управления заданиями MVS, *211*

АЛХИМИЯ ФИНАНСОВ

Джордж Сорос

www.dialektika.com

В своей книге *Алхимия финансов,* выдержавшей несколько изданий и пользующейся непреходящим успехом, знаменитый финансист и спекулянт Джордж Сорос раскрывает свои секреты принятия решений при игре на бирже. Кроме того, он предлагает читателю собственную философию, согласно которой не только игроки влияют на рынок, но и рынок влияет на игроков. Этот процесс Джордж Сорос называет рефлексивным взаимодействием. Кроме того, в книге дается анализ общесивенно-политических событий, происходящих в конце 1985 — начале 1986 гг. Книга рассчитана на финансистов, менеджеров инвестиционных фондов, просто игроков на бирже и, наконец, для всех, кого интересует деятельность и идеи Джорджа Сороса.

ISBN 978-5-8459-1649-5

в продаже

TheDesignofDesign.indb 459 30.10.2012 15:58:44

ФИНАНСЫ

Зви Боди
Роберт Мертон

w w w.williamspublishing.com

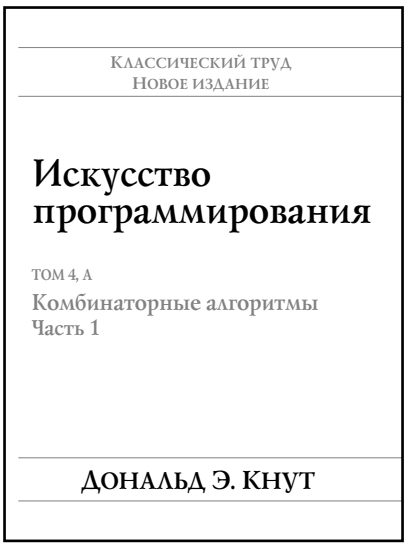
Авторы книги — университетские профессора: Зви Боди и Роберт Мертон (Нобелевский лауреат по экономике 1997 г.) — детально проанализировали проблемы, с которыми все мы сталкиваемся дома и на работе. В первой части книги дается общий обзор финансовой системы, а также рассматриваются структура и применение финансовых отчетов корпораций для нужд хозяйственного анализа и прогнозирования. Части II, III и IV посвящены концептуальным основам финансов. Упор в них делается на применении финансовых принципов в ходе принятия решений как домохозяйствами (финансовое планирование и вопросы инвестирования на всех этапах жизненного цикла семьи), так и фирмами (экономический анализ эффективности капиталовложений). В части V описана теория и практика оценки активов. Часть VI посвящена базовым вопросам корпоративных финансов. Теоретические подходы и практические примеры, рассматриваемые в учебнике, применимы для любой страны с рыночной экономикой. Особенно полезной эта книга является для подготовки студентов экономических специальностей в России, Украине и других странах СНГ.

ISBN 978-5-8459-0946-6

в продаже

ИСКУССТВО ПРОГРАММИРОВАНИЯ
ТОМ 4, А. КОМБИНАТОРНЫЕ АЛГОРИТМЫ
Часть 1

Дональд Э. Кнут



www.williamspublishing.com

В этом томе рассматриваются методы, позволяющие компьютерам эффективно работать с задачами гигантского размера. Рассматриваемый материал начинается с булевых функций и технологий и трюков работы с битами, затем всесторонне рассматривается генерация всех кортежей и перестановок, всех сочетаний и разбиений, и всех деревьев. Комбинаторный поиск — богатая и важная тема, и Кнут приводит слишком много нового, интересного и полезного материала, чтобы его можно было разместить в одном или двух (а может быть, даже в трех) томах. Одна эта книга включает около 1500 упражнений с ответами для самостоятельной работы, а также сотни полезных фактов, которые вы не найдете ни в каких других публикациях. Том 4А определенно должен занять свое место на полке рядом с первыми тремя томами этой классической работы в библиотеке каждого серьезного программиста.

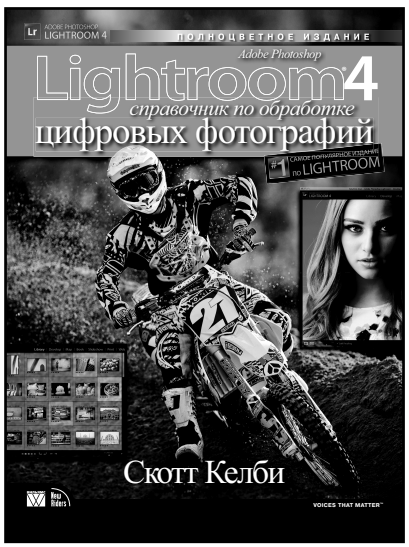
ISBN 978-5-8459-1744-7

в продаже

TheDesignofDesign.indb 461 30.10.2012 15:58:44

ADOBE PHOTOSHOP LIGHTROOM 4
СПРАВОЧНИК ПО ОБРАБОТКЕ ЦИФРОВЫХ ФОТОГРАФИЙ

Скотт Келби



www.williamspublishing.com

Это полноцветное практическое руководство составлено по принципу пошагового выполнения процесса обработки цифровых фотографий в Lightroom 4, начиная с их импорта и заканчивая выводом на печать или публикацией в веб. Работа с фотографиями в каждом модуле Lightroom разделена на отдельные упражнения, где в краткой и наглядной форме демонстрируются приемы наиболее эффективного для пользователей выполнения операций импорта, отбора и правки фотографий, составления из них коллекций, слайд-шоу, фотоальбомов, веб-галерей и получения фотоотпечатков.
Книга адресована фотографам, дизайнерам и всем, кто стремится основательно разобраться в обработке цифровых фотографий средствами Lightroom.

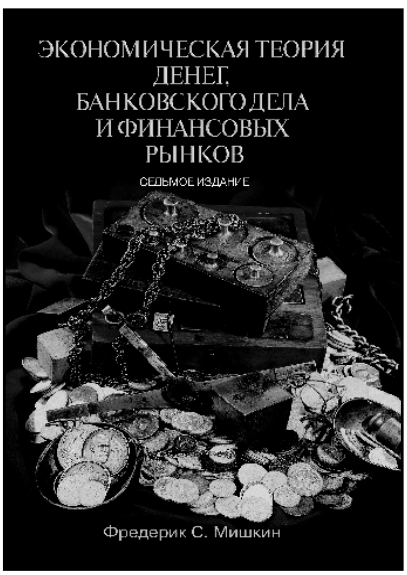
ISBN 978-5-8459-1810-9

в продаже

TheDesignofDesign.indb 462 30.10.2012 15:58:44

ЭКОНОМИЧЕСКАЯ ТЕОРИЯ ДЕНЕГ, БАНКОВСКОГО ДЕЛА И ФИНАНСОВЫХ РЫНКОВ
7-Е ИЗДАНИЕ

Фредерик С. Мишкин



www.williamspublishing.com

Один из самых популярных зарубежных учебников в области финансов. В центре внимания автора — влияние денежно-кредитной политики на деятельность финансовых рынков и финансовых учреждений, объем производства товаров и услуг в экономике страны, развитие международной торговли, прибыль предприятий и благосостояние граждан. Книга содержит много практических примеров, богатый справочно-методический материал, вопросы и задания к каждой главе, а также ответы и решения ко многим из них. Особое внимание уделено активному использованию возможностей Интернета при изучении данной проблематики.
Для студентов, изучающих теорию денег и финансовых рынков, преподавателей экономики, аналитиков и бизнесменов.

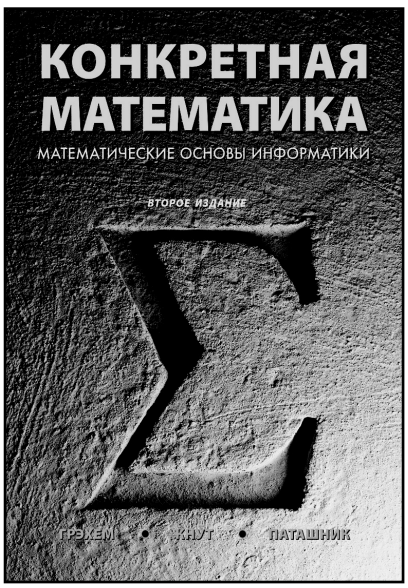
ISBN 5-8459-0918-X

в продаже

TheDesignofDesign.indb 463 30.10.2012 15:58:45

КОНКРЕТНАЯ МАТЕМАТИКА
2-е издание

Рональд Л. Грэхем,
Дональд Э. Кнут,
Орен Паташник



www.williamspublishing.com

В основу данной книги положен одноименный курс лекций Станфордского университета. Название “конкретная математика” происходит от слов “КОНтинуальная” и “дисКРЕТНАЯ” математика. Назначение данной книги — обеспечить читателя техникой оперирования с дискретными объектами, что совершенно необходимо для математиков, работающих в области информатики. Книга ориентирована в первую очередь на практиков (хотя и теоретики найдут в ней много полезного), и изобилует массой конкретных примеров и упражнений. Конкретность изложения абстрактного материала — еще одно пояснение названия книги. Широта охвата столь различных тем в одной книге могла бы вызвать подозрения в некоторой легковесности, если бы не имена ее авторов — известных американских математиков. Тем не менее слово “легкий” к книге вполне применимо, так как стиль изложения достаточно далек от сухого академизма. Как признаются сами авторы, они считают математику развлечением, и они сделали все, чтобы читатели книги получили от ее прочтения не только знания, но и удовольствие.

Книгу можно рекомендовать всем математикам, но в первую очередь она предназначена для студентов, обучающихся математике и информатике.

ISBN 978-5-8459-1588-7

в продаже

TheDesignofDesign.indb 464 30.10.2012 15:58:45