Biên soạn: PGS. TS. HUỲNH XUÂN HIỆP (Chủ biên)
ThS. VÕ HUỲNH TRÂM
ThS. PHAN PHƯƠNG LAN
ThS. HUỲNH QUANG NGHI

GIÁO TRÌNH

KIẾN TRÚC VÀ THIẾT KẾ PHẦN MỀM

NHÀ XUẤT BẢN ĐẠI HỌC CẦN THƠ
2015

BIÊN MỤC TRƯỚC XUẤT BẢN THỰC HIỆN BỞI
TRUNG TÂM HỌC LIỆU TRƯỜNG ĐẠI HỌC CẦN THƠ

Giáo trình kiến trúc và thiết kế phần mềm / Huỳnh Xuân Hiệp … [et al.].– Cần Thơ : Nxb. Đại học Cần Thơ, 2015.

232 tr. : minh họa ; 24 cm

Sách có danh mục tài liệu tham khảo

ISBN: 9786049195242

1. Computer software management 2.Thiết kế phần mềm

I. Nhan đề. II. Huỳnh, Xuân Hiệp

005.3 – DDC 23 MFN 205657

Gi108

LỜI GIỚI THIỆU

Nhằm góp phần làm phong phú nguồn tư liệu phục vụ nghiên cứu, học tập cho bạn đọc và sinh viên khoa Công nghệ Thông tin và Truyền thông - Trường Đại học Cần Thơ, Nhà Xuất bản Đại học Cần Thơ ấn hành và giới thiệu cùng bạn đọc giáo trình "Kiến trúc và thiết kế phần mềm" do Phó Giáo sư, Tiến sĩ Huỳnh Xuân Hiệp, Thạc sĩ Võ Huỳnh Trâm, Thạc sĩ Phan Phương Lan và Thạc sĩ Huỳnh Quang Nghi biên soạn.

Giáo trình gồm 8 chương với các nội dung đi từ tổng quan đến chi tiết. Chương đầu tiên sẽ giới thiệu tổng quan về thiết kế phần mềm. Đi vào các chương sau, ta sẽ được tìm hiểu về kiến trúc của các mô hình thiết kế và một số hướng thiết kế quan trọng hiện nay như: thiết kế hướng mẫu, thiết kế ứng dụng web,... Thêm vào đó, cuối mỗi chương còn có nhiều bài tập thực hành hữu ích cho bạn đọc. Giáo trình là tài liệu học tập có giá trị cho sinh viên các ngành có liên quan đến kiến trúc và thiết kế phần mềm.

Nhà Xuất bản Đại học Cần Thơ chân thành cám ơn các tác giả và sự đóng góp ý kiến của quý thầy cô trong Hội đồng thẩm định trường Đại học Cần Thơ để giáo trình "Kiến trúc và thiết kế phần mềm" được ra mắt bạn đọc.

Nhà Xuất bản Đại học Cần Thơ trân trọng giới thiệu đến sinh viên, giảng viên và bạn đọc giáo trình này.

NHÀ XUẤT BẢN ĐẠI HỌC CẦN THƠ

LỜI NÓI ĐẦU

Giáo trình "Kiến trúc và Thiết kế phần mềm" cung cấp cho bạn đọc các nguyên lý, khái niệm, và phương pháp để tạo ra các mô hình thiết kế chất lượng cao. Qua giáo trình này, bạn đọc sẽ có kiến thức rộng về cách thức thiết kế một phần mềm nói chung và cách thức thiết kế một phần mềm theo hướng mẫu, Web hoặc dịch vụ nói riêng.

Giáo trình - được tổ chức thành 8 chương - bao gồm các nội dung tổng quan về từng phần tử của mô hình thiết kế (các chương 2, 3, 4, 5), và một số hướng thiết kế quan trọng hiện nay (các chương 6, 7, 8). Chương 1 giới thiệu tổng quan về thiết kế phần mềm. Những nội dung được cung cấp trong phần này gồm: thiết kế trong ngữ cảnh công nghệ phần mềm, tiến trình thiết kế, khái niệm thiết kế, và mô hình thiết kế. Chương 2 tập trung vào thiết kế dữ liệu và lớp với các nội dung chính: phân tích các yêu cầu và mô hình hóa dựa trên kịch bản, các cách mô hình hóa theo các hướng tiếp cận như lớp, luồng, hay hành vi. Chương 3 cung cấp một cách tiếp cận có hệ thống cho thiết kế kiến trúc - dựa trên những bản dựng sơ bộ mà từ đó phần mềm được tạo thành. Chương 4 trình bày các nguyên tắc quan trọng hướng dẫn việc thiết kế giao diện người sử dụng hiệu quả, quy trình phân tích và thiết kế giao diện người sử dụng. Ngoài ra, chương này còn giới thiệu đến bạn đọc một số lưu ý để thiết kế giao diện ứng dụng Web hiệu quả. Chương 5 tập trung vào thiết kế thành phần - thiết kế được bắt đầu khi vòng lặp đầu tiên của thiết kế kiến trúc đã hoàn thành. Chương 6 giới thiệu về thiết kế hướng mẫu. Mục đích của loại thiết kế này là tạo ra một ứng dụng mới thông qua một bộ các giải pháp đã được chứng minh (các mẫu thiết kế) cho một tập các vấn đề được mô tả rõ ràng. Chương 7 đề cập đến thiết kế cho các ứng dụng Web. Những thiết kế nội dung, thẩm mỹ, kiến trúc, giao diện, điều hướng, và mức thành phần cần phải được lưu ý trong suốt quá trình thiết kế ứng dụng Web. Chương 8 giới thiệu kiến trúc hướng dịch vụ - một kiến trúc phần mềm phân tán bao gồm nhiều dịch vụ độc lập - thông qua các nội dung: khái niệm, các mẫu cho kiến trúc hướng dịch vụ, thiết kế kiến trúc hướng dịch vụ, sự phối hợp và tái sử dụng dịch vụ, và mô tả một số hỗ trợ công nghệ cho loại kiến trúc này.

Giáo trình "Kiến trúc và Thiết kế phần mềm" được biên soạn trên cơ sở quyển sách "Software Engineering - A Practitioner's Approach" của tác giả Roger S. Pressman - một nguồn tài liệu được công nhận và sử dụng rộng rãi bởi các nhà chuyên môn, và Bài giảng Thiết kế phần mềm đã được các tác giả giảng dạy trong nhiều năm cho sinh viên đại học ngành Kỹ thuật phần.

Giáo trình "Kiến trúc và Thiết kế phần mềm" được sử dụng làm tài liệu chính của học phần Kiến trúc và Thiết kế phần mềm - một học phần bắt buộc đối với sinh viên năm thứ tư bậc đại học ngành Kỹ thuật phần mềm. Bên cạnh đó, giáo trình còn có thể dùng làm tài liệu tham khảo cho những ai tham gia trong các dự án phát triển và/hoặc bảo trì phần mềm. Để có thể tiếp nhận một cách hiệu quả những nội dung được trình bày trong giáo trình, bạn đọc cần có kiến thức nền tảng về công nghệ phần mềm.

Trong quá trình biên soạn chắc chắn còn nhiều thiếu sót. Nhóm biên soạn rất mong nhận được sự góp ý, phản hồi từ bạn đọc để giáo trình này có chất lượng ngày càng tốt hơn.

Cần Thơ, ngày 10 tháng 04 năm 2015

NHÓM TÁC GIẢ

MỤC LỤC

Chương 1. TỔNG QUAN VỀ THIẾT KẾ PHẦN MỀM **1**
1.1 THIẾT KẾ TRONG NGỮ CẢNH CÔNG NGHỆ PHẦN MỀM 1
1.1.1 Thiết kế là một chuỗi các quyết định 3
1.2 TIẾN TRÌNH THIẾT KẾ 4
1.2.1 Hướng dẫn và thuộc tính chất lượng phần mềm 4
1.2.2 Sự tiến hóa của thiết kế phần mềm 6
1.3 KHÁI NIỆM THIẾT KẾ 7
1.3.1 Trừu tượng hóa 7
1.3.2 Kiến trúc phần mềm 8
1.3.3 Các mẫu thiết kế 9
1.3.4 Sự phân tách mối quan tâm 9
1.3.5 Mô đun hóa 9
1.3.6 Sự che dấu thông tin 10
1.3.7 Sự độc lập chức năng 11
1.3.8 Sự tinh chỉnh 12
1.3.9 Các khía cạnh 12
1.3.10 Tái cấu trúc 13
1.3.11 Các khái niệm thiết kế hướng đối tượng 13
1.3.12 Các lớp thiết kế 13
1.4 MÔ HÌNH THIẾT KẾ 15
1.4.1 Các phần tử thiết kế dữ liệu 16
1.4.2 Các phần tử thiết kế kiến trúc 16
1.4.3 Các phần tử thiết kế giao diện 17
1.4.4 Các phần tử thiết kế mức thành phần 18
1.4.5 Các phần tử thiết kế mức triển khai 19
TỔNG KẾT 20
CÂU HỎI HƯỚNG DẪN ÔN TẬP 20
ĐỊNH HƯỚNG THẢO LUẬN 21
BÀI TẬP THỰC HÀNH 21
TÀI LIỆU THAM KHẢO 21
Chương 2. THIẾT KẾ DỮ LIỆU **23**
2.1 PHÂN TÍCH CÁC YÊU CẦU 23
2.1.1 Các mục tiêu tổng thể và triết lý 24
2.1.2 Phân tích quy tắc thông qua kinh nghiệm 24

2.1.3 Phân tích lĩnh vực 25
2.1.4 Các tiếp cận mô hình hóa yêu cầu 26
2.2 MÔ HÌNH HÓA DỰA TRÊN KỊCH BẢN 27
2.2.1 Tạo trường hợp sử dụng ban đầu 27
2.2.2 Tinh chỉnh trường hợp sử dụng ban đầu 28
2.2.3 Xây dựng trường hợp sử dụng chính thức 29
2.3 CÁC MÔ HÌNH UML HỖ TRỢ TRƯỜNG HỢP SỬ DỤNG 30
2.3.1 Sơ đồ hoạt động 30
2.3.2 Sơ đồ làn 30
2.4 KHÁI NIỆM MÔ HÌNH HÓA DỮ LIỆU 33
2.4.1 Các đối tượng dữ liệu 33
2.4.2 Các thuộc tính dữ liệu 34
2.4.3 Các mối quan hệ 34
2.5 MÔ HÌNH HÓA DỰA TRÊN LỚP 35
2.5.1 Xác định các lớp phân tích 35
2.5.2 Xác định các thuộc tính 36
2.5.3 Định nghĩa các phương thức 37
2.5.4 Mô hình hóa lớp-trách nhiệm-cộng tác 38
2.5.5 Kết hợp và phụ thuộc 41
2.5.6 Phân tích gói công việc 42
2.6 MÔ HÌNH HÓA HƯỚNG LUỒNG 43
2.6.1 Khởi tạo mô hình luồng dữ liệu 43
2.6.2 Khởi tạo mô hình kiểm soát luồng 45
2.6.3 Đặc tả kiểm soát 45
2.6.4 Đặc tả xử lý 47
2.7 MÔ HÌNH HÓA HƯỚNG HÀNH VI 47
2.7.1 Xác định sự kiện với trường hợp sử dụng 47
2.7.2 Biểu diễn trạng thái 48
2.8 MÔ HÌNH HÓA BẰNG MẪU 50
TỔNG KẾT 50
CÂU HỎI HƯỚNG DẪN ÔN TẬP 52
ĐỊNH HƯỚNG THẢO LUẬN 52
BÀI TẬP THỰC HÀNH 52
TÀI LIỆU THAM KHẢO 52
Chương 3. THIẾT KẾ KIẾN TRÚC 54
3.1 KHÁI NIỆM VỀ KIẾN TRÚC PHẦN MỀM 54
3.1.1 Kiến trúc phần mềm 54
3.1.2 Tầm quan trọng của kiến trúc phần mềm 56
3.1.3 Mục đích sử dụng kiến trúc phần mềm 56

3.1.4 Các thể loại kiến trúc (Architectural Genres) 57
3.2 CÁC PHONG CÁCH KIẾN TRÚC (ARCHITECTURAL STYLES) 58
3.2.1 Kiến trúc lấy dữ liệu làm trung tâm (data-centered architectures) 59
3.2.2 Kiến trúc luồng dữ liệu (data-flow architectures) 60
3.2.3 Kiến trúc gọi và trả về (call and return architectures) 61
3.2.4 Kiến trúc hướng đối tượng (object-oriented architectures) 61
3.2.5 Kiến trúc phân lớp (layered architectures) 61
3.3 THIẾT KẾ KIẾN TRÚC PHẦN MỀM 62
3.3.1 Biểu diễn hệ thống trong ngữ cảnh 63
3.3.2 Định nghĩa các archetype 64
3.3.3 Tinh chỉnh kiến trúc thành các thành phần 66
3.3.4 Mô tả các thể hiện của hệ thống 67
3.4 PHƯƠNG PHÁP THIẾT KẾ KIẾN TRÚC 68
3.4.1 Các kiểu luồng thông tin 68
3.4.2 Chuyển luồng biến đổi (transform mapping) 70
3.4.3 Chuyển luồng giao tác (transaction mapping) 77
TỔNG KẾT 80
CÂU HỎI HƯỚNG DẪN ÔN TẬP 80
ĐỊNH HƯỚNG THẢO LUẬN 81
BÀI TẬP THỰC HÀNH 81
TÀI LIỆU THAM KHẢO 81
Chương 4. THIẾT KẾ GIAO DIỆN NGƯỜI SỬ DỤNG 82
4.1 BỘ QUY TẮC VÀNG 82
4.1.1 Cho phép người sử dụng duy trì sự kiểm soát 83
4.1.2 Giảm tải việc phải nhớ của người sử dụng 83
4.1.3 Tạo sự nhất quán về giao diện 84
4.2 QUY TRÌNH PHÂN TÍCH VÀ THIẾT KẾ GIAO DIỆN NGƯỜI SỬ DỤNG 84
4.2.1 Các mô hình phân tích và thiết kế giao diện 84
4.2.2 Quy trình phân tích và thiết kế giao diện 86
4.3 PHÂN TÍCH GIAO DIỆN 87
4.3.1 Phân tích người sử dụng 88
4.3.2 Phân tích nhiệm vụ và mô hình hóa 89
4.3.3 Phân tích nội dung hiển thị 93
4.3.4 Phân tích môi trường làm việc 93
4.4 THIẾT KẾ GIAO DIỆN 94
4.4.1 Áp dụng các bước thiết kế giao diện 95
4.4.2 Các mẫu thiết kế giao diện người sử dụng 97
4.4.3 Các vấn đề thiết kế 97
4.5 THIẾT KẾ GIAO DIỆN ỨNG DỤNG WEB 101

4.5.1 Nguyên tắc và hướng dẫn thiết kế giao diện Web 101
4.5.2 Luồng công việc thiết kế giao diện cho ứng dụng Web 104
4.6 ĐÁNH GIÁ THIẾT KẾ 106
TỔNG KẾT 107
CÂU HỎI HƯỚNG DẪN ÔN TẬP 108
ĐỊNH HƯỚNG THẢO LUẬN 109
BÀI TẬP THỰC HÀNH 109
TÀI LIỆU THAM KHẢO 109
Chương 5. THIẾT KẾ THÀNH PHẦN 111
5.1 GIỚI THIỆU 111
5.1.1 Quan điểm hướng đối tượng (An Object-Oriented View) 112
5.1.2 Quan điểm truyền thống (The Traditional View) 114
5.1.3 Quan điểm hướng tiến trình (A Process-Related View) 116
5.2 THIẾT KẾ THÀNH PHẦN DỰA TRÊN LỚP 117
5.2.1 Nguyên tắc thiết kế cơ bản 117
5.2.2 Hướng dẫn thiết kế cấp thành phần 120
5.2.3 Sự gắn kết (Cohension) 121
5.2.4 Sự nối kết (Coupling) 122
5.3 CÁC BƯỚC THIẾT KẾ THÀNH PHẦN CHO HỆ THỐNG HƯỚNG ĐỐI TƯỢNG 123
5.4 THIẾT KẾ CÁC THÀNH PHẦN TRUYỀN THỐNG 130
5.4.1 Các ký hiệu thiết kế đồ họa 131
5.4.2 Các ký hiệu thiết kế dạng bảng 132
5.4.3 Ngôn ngữ thiết kế chương trình (PDL) 133
5.5 PHÁT TRIỂN TRÊN CÁC THÀNH PHẦN 135
5.5.1 Công nghệ lĩnh vực ứng dụng (Domain Engineering) 135
5.5.2 Cơ cấu, sự tương thích và chất lượng của thành phần 136
5.5.3 Phân tích và thiết kế cho việc tái sử dụng 139
5.5.4 Phân loại và truy xuất các thành phần 140
TỔNG KẾT 141
CÂU HỎI HƯỚNG DẪN ÔN TẬP 142
ĐỊNH HƯỚNG THẢO LUẬN 143
BÀI TẬP THỰC HÀNH 143
TÀI LIỆU THAM KHẢO 143
Chương 6. THIẾT KẾ HƯỚNG MẪU (PATTERN) 145
6.1 THIẾT KẾ HƯỚNG MẪU 145
6.1.1 Các loại mẫu thiết kế 146
6.1.2 Khung làm việc 147
6.1.3 Mô tả mẫu thiết kế 148

6.1.4 Kho và Ngôn ngữ mẫu 148
6.2 THIẾT KẾ PHẦN MỀM HƯỚNG MẪU 149
6.2.1 Thiết kế hướng mẫu trong bối cảnh 149
6.2.2 Tư duy trong mẫu 150
6.2.3 Nhiệm vụ thiết kế 152
6.2.4 Xây dựng bảng tổ chức theo mẫu 153
6.2.5 Những sai lầm chung của thiết kế 154
6.3 MẪU KIẾN TRÚC 154
6.4 MẪU THIẾT KẾ MỨC THÀNH PHẦN 155
6.5 MẪU THIẾT KẾ GIAO DIỆN NGƯỜI SỬ DỤNG 157
6.5.1 Toàn bộ giao diện người sử dụng (Whole UI) 157
6.5.2 Dàn trang (PageLayout) 157
6.5.3 Các biểu mẫu và dữ liệu đầu vào (Forms and Input) 158
6.5.4 Các bảng 158
6.5.5 Thao tác dữ liệu trực tiếp 158
6.5.6 Sự điều hướng 159
6.5.7 Tìm kiếm 159
6.5.8 Các thành phần của trang 159
6.5.9 Thương mại điện tử 160
6.5.10 Miscellaneous 160
6.6 MẪU THIẾT KẾ ỨNG DỤNG WEB 160
6.6.1 Trọng tâm thiết kế 161
6.6.2 Mức độ chi tiết thiết kế 162
TỔNG KẾT 162
CÂU HỎI HƯỚNG DẪN ÔN TẬP 163
ĐỊNH HƯỚNG THẢO LUẬN 164
BÀI TẬP THỰC HÀNH 164
TÀI LIỆU THAM KHẢO 164
Chương 7. THIẾT KẾ ỨNG DỤNG WEB 166
7.1 CHẤT LƯỢNG THIẾT KẾ ỨNG DỤNG WEB 166
7.2 MỤC TIÊU CỦA THIẾT KẾ 169
7.2.1 Tính đơn giản 169
7.2.2 Tính nhất quán 169
7.2.3 Định danh 169
7.2.4 Sự mạnh mẽ 170
7.2.5 Khả năng điều hướng 170
7.2.6 Sự ấn tượng từ bên ngoài 170
7.2.7 Khả năng tương thích 170
7.3 THÁP THIẾT KẾ CHO ỨNG DỤNG WEB 170

7.4 THIẾT KẾ GIAO DIỆN 171
7.5 THIẾT KẾ THẨM MỸ 172
7.5.1 Vấn đề bố trí 172
7.5.2 Vấn đề thiết kế đồ họa 173
7.6 THIẾT KẾ NỘI DUNG 173
7.6.1 Đối tượng nội dung 173
7.6.2 Vấn đề thiết kế nội dung 174
7.7 THIẾT KẾ KIẾN TRÚC 175
7.7.1 Kiến trúc nội dung 175
7.7.2 Kiến trúc ứng dụng Web 178
7.8 THIẾT KẾ ĐIỀU HƯỚNG 179
7.8.1 Ngữ nghĩa của sự điều hướng 179
7.8.2 Cú pháp của sự điều hướng 181
7.9 THIẾT KẾ MỨC THÀNH PHẦN 182
7.10 PHƯƠNG PHÁP THIẾT KẾ SIÊU PHƯƠNG TIỆN HƯỚNG ĐỐI TƯỢNG 182
7.10.1 Thiết kế khái niệm của OOHDM 183
7.10.2 Thiết kế điều hướng của OOHDM 184
7.10.3 Thiết kế giao diện trừu tượng và Thực hiện 185
TỔNG KẾT 185
CÂU HỎI HƯỚNG DẪN ÔN TẬP 186
ĐỊNH HƯỚNG THẢO LUẬN 186
BÀI TẬP THỰC HÀNH 187
TÀI LIỆU THAM KHẢO 187
Chương 8. THIẾT KẾ KIẾN TRÚC HƯỚNG DỊCH VỤ 189
8.1 GIỚI THIỆU 189
8.1.1 Các mẫu cho SOA 189
8.1.2 Nguyên tắc thiết kế dịch vụ 189
8.2 MẪU MÔI GIỚI 190
8.2.1 Mẫu Đăng ký dịch vụ 191
8.2.2 Mẫu Chuyển tiếp môi giới 192
8.2.3 Mẫu Xử lý môi giới 193
8.2.4 Mẫu Phát hiện dịch vụ 194
8.3 MẪU GIAO DỊCH 195
8.3.1 Mẫu Giao thức cam kết hai giai đoạn 195
8.3.2 Mẫu Giao dịch tổ hợp 198
8.3.3 Mẫu Giao dịch muôn năm (Long-Living Transaction) 199
8.4 MẪU ĐÀM PHÁN 200
8.5 PHỐI HỢP VÀ TÁI SỬ DỤNG DỊCH VỤ TRONG SOA 201
8.5.1 Phối hợp dịch vụ 201

8.5.2 Tái sử dụng dịch vụ 202
8.5.3 Thiết kế các SOA 202
8.6 HỖ TRỢ CÔNG NGHỆ CHO SOA 202
8.6.1 Các giao thức dịch vụ Web 203
8.6.2 Các dịch vụ Web 203
8.6.3 Dịch vụ đăng ký 204
8.6.4 Dịch vụ phát hiện và môi giới 204
TỔNG KẾT 205
CÂU HỎI HƯỚNG DẪN ÔN TẬP 205
ĐỊNH HƯỚNG THẢO LUẬN 206
BÀI TẬP THỰC HÀNH 206
TÀI LIỆU THAM KHẢO 206

DANH MỤC HÌNH

Hình 1.1	Chuyển đổi từ mô hình yêu cầu sang mô hình thiết kế (1)	2
Hình 1.2	Chuyển đổi từ mô hình yêu cầu sang mô hình thiết kế (2)	2
Hình 1.3	Chuỗi các quyết định thiết kế	3
Hình 1.4	Mô đun hóa và chi phí phần mềm	10
Hình 1.5	Lớp thiết kế cho FloorPlan và sự kết tập tổng hợp cho lớp	14
Hình 1.6	Các chiều của mô hình thiết kế	15
Hình 1.7	Giao diện biểu diễn cho ControlPanel	18
Hình 1.8	Một sơ đồ thành phần dạng UML	19
Hình 1.9	Một sơ đồ triển khai dạng UML	19
Hình 2.1	Mô hình phân tích yêu cầu như một cầu nối giữa mô tả hệ thống và mô hình thiết kế	24
Hình 2.2	Đầu vào và đầu ra cho phân tích lĩnh vực	26
Hình 2.3	Các phần tử của mô hình yêu cầu	27
Hình 2.4	Sơ đồ trường hợp sử dụng ban đầu/sơ bộ cho hệ thống SafeHome	29
Hình 2.5	Sơ đồ hoạt động cho camera giám sát truy cập thông qua Internet - chức năng hiển thị các bức ảnh	31
Hình 2.6	Sơ đồ làn cho camera giám sát truy cập thông qua Internet - chức năng hiển thị các bức ảnh	32
Hình 2.7	Bảng biểu diễn các đối tượng dữ liệu	33
Hình 2.8	Các mối quan hệ giữa các đối tượng dữ liệu	35
Hình 2.9	Sơ đồ lớp cho lớp Hệ thống System	37
Hình 2.10	Sơ đồ lớp cho lớp Mặt bằng sàn FloorPlan	38
Hình 2.11	Một mô hình thẻ chỉ mục (CRC)	39
Hình 2.12	Đa dạng/bản số	41
Hình 2.13	Các phụ thuộc	42
Hình 2.14	Các gói	42
Hình 2.15	DFD mức ngữ cảnh cho chức năng bảo mật SafeHome	44
Hình 2.16	DFD mức 1 cho chức năng bảo mật SafeHome	44
Hình 2.17	DFD mức 2 tinh chỉnh xử lý Giám sát bộ cảm biến	45
Hình 2.18	Sơ đồ trạng thái cho chức năng bảo mật SafeHome	46
Hình 2.19	Bảng kích hoạt xử lý (PAT) cho chức năng bảo mật của SafeHome	46
Hình 2.20	Sơ đồ trạng thái cho lớp ControlPanel	49
Hình 2.21	Sơ đồ tuần tự bộ phận của chức năng bảo mật SafeHome	49
Hình 3.1	Kiến trúc lấy dữ liệu làm trung tâm	60
Hình 3.2	Kiến trúc luồng dữ liệu (mẫu pipe and filter)	60

Hình 3.3 Kiến trúc chương trình chính/chương trình phụ 62
Hình 3.4 Kiến trúc phân lớp 62
Hình 3.5 Biểu đồ ngữ cảnh kiến trúc ACD 63
Hình 3.6 Biểu đồ ngữ cảnh kiến trúc cho chức năng bảo vệ nhà của SafeHome 64
Hình 3.7 Quan hệ UML cho các archetype của chức năng bảo vệ nhà SafeHome 65
Hình 3.8 Kiến trúc tổng thể của SafeHome với các thành phần ở mức cao 67
Hình 3.9 Một minh họa cho SafeHome với các thành phần chi tiết 68
Hình 3.10 Kiểu luồng biến đổi 69
Hình 3.11 Kiểu luồng giao tác 70
Hình 3.12 Mô hình ngữ cảnh DFD cho chức năng bảo vệ nhà SafeHome 70
Hình 3.13 DFD mức 1 cho chức năng bảo vệ nhà SafeHome 71
Hình 3.14 DFD mức 2 cho hệ thống con *giám sát cảm biến* 72
Hình 3.15 DFD mức 3 cho hệ thống con *giám sát cảm biến* với đặc tả biên luồng 73
Hình 3.16 Phân tách mức độ một cho hệ thống con *giám sát cảm biến* 74
Hình 3.17 Phân tách mức độ hai cho hệ thống con *giám sát cảm biến* 75
Hình 3.18 Lược đồ cấu trúc cho quá trình giám sát các cảm biến 76
Hình 3.19 Lược đồ cấu trúc đã tinh chỉnh cho quá trình giám sát các cảm biến 76
Hình 3.20 DFD mức 2 cho hệ thống con *tương tác người sử dụng* 78
Hình 3.21 Chuyển luồng giao tác cho hệ thống con *tương tác người sử dụng* 79
Hình 3.22 Phân tách mức độ một cho hệ thống con tương tác người sử dụng 79
Hình 4.1 Quy trình thiết kế giao diện người sử dụng 86
Hình 4.2 Lược đồ cho chức năng bán thuốc theo toa lần nữa 92
Hình 4.3 Bố trí màn hình sơ bộ 96
Hình 4.4 Ánh xạ mục tiêu của người sử dụng vào hoạt động giao diện 104
Hình 4.5 Chu kỳ đánh giá giao diện 107
Hình 5.1 Phác thảo chi tiết một thiết kế thành phần 113
Hình 5.2 Lược đồ cấu trúc cho một hệ thống theo quan điểm truyền thống 115
Hình 5.3 Thiết kế cấp thành phần của *ComputePageCost* 116
Hình 5.4 Thiết kế theo nguyên tắc OCP 118
Hình 5.5 Gắn kết mức lớp 121
Hình 5.6 Biểu đồ hợp tác với thông báo 124
Hình 5.7 Các giao diện và lớp tái cấu trúc được định nghĩa cho PrintJob 125
Hình 5.8 Sơ đồ hoạt động UML cho hoạt động *computePaperCost()* 127
Hình 5.9 Biểu đồ trạng thái phân đoạn cho lớp PrintJob 129
Hình 5.10 Các cấu trúc lưu đồ 132
Hình 6.1 Thiết kế hướng mẫu theo bối cảnh 150
Hình 7.1 Cây yêu cầu chất lượng 167
Hình 7.2 Tháp thiết kế cho các ứng dụng Web 171
Hình 7.3 Biểu diễn thiết kế đối tượng nội dung 174
Hình 7.4 Cấu trúc tuyến tính 176

Hình 7.5 Cấu trúc dạng lưới 176
Hình 7.6 Cấu trúc phân cấp 177
Hình 7.7 Cấu trúc được nối kết mạng 178
Hình 7.8 Kiến trúc MVC 179
Hình 7.9 Tạo một đơn vị ngữ nghĩa điều hướng NSU 181
Hình 7.10 Sơ đồ khái niệm thành phần của SafeHomeAssured.com 184
Hình 8.1 Đăng ký dịch vụ với bộ môi giới 191
Hình 8.2 Mẫu chuyển tiếp môi giới (trang trắng) 192
Hình 8.3 Mẫu xử lý môi giới 193
Hình 8.4 Mẫu phát hiện dịch vụ (trang vàng) 194
Hình 8.5 Ví dụ về giai đoạn đầu của giao thức cam kết hai giai đoạn: chuyển ngân 196
Hình 8.6 Ví dụ về giai đoạn thứ hai của giao thức cam kết hai giai đoạn: chuyển ngân 197
Hình 8.7 Ví dụ về mẫu giao dịch tổ hợp 198
Hình 8.8 Ví dụ về mẫu giao dịch muôn năm 200
Hình 8.9 Web client và dịch vụ Web trong một ứng dụng các dịch vụ WWW 204
Hình 8.10 Ví dụ về một bộ môi giới các dịch vụ Web 205

DANH MỤC BẢNG

Bảng 5.1	Ký pháp bảng quyết định	133
Bảng 6.1	Bảng tổ chức theo mẫu	153
Bảng 7.1	Tóm tắt phương pháp OOHDM	183

DANH MỤC KÝ HIỆU VIẾT TẮT

ADV	Abstract Data View
ACD	Architectural Context Diagram
ADL	Architectural Description Language
CBSE	Component-Based Software Engineering
CCP	The Common Closure Principle
CSPEC	Control Specification
CRP	The Common Reuse Principle
DFR	Design for Reuse
ERD	Entity-Relationship Diagram
GUI	Graphical User Interface
HCI	Human Computer Interaction
MVC	Model-View-Controller
NSU	Navigation Semantic Unit
OOHDM	Object Oriented Hypermedia Design Method
PDL	Program Design Language
PAT	Process Activation Table
PSPEC	Process Specification
REST	Representationak State Transfer
SOA	Service Oriented Architecture
SQA	Software Quality Assurance
UI	User Interface
WON	Ways of Navigating
WSDL	Web Services Description Language

DANH MỤC MỘT SỐ THUẬT NGỮ

Abstract Data View - ADV	Khung nhìn dữ liệu trừu tượng
Abstractions	Các trừu tượng hóa
Architectural Context Diagram - ACD	Sơ đồ ngữ cảnh kiến trúc
Activity diagram	Sơ đồ hoạt động
Actor	Tác nhân
Architectural Description Language - ADL	Ngôn ngữ mô tả kiến trúc
Alternative behavior	Hành vi thay thế
Architectural description	Mô tả về kiến trúc
Architectural design	Thiết kế kiến trúc
Architectural styles	Các phong cách kiến trúc
Artificial intelligence	Trí tuệ nhân tạo
Aspect-oriented method	Phương pháp hướng khía cạnh
Aspects	Các khía cạnh
Associations	Các kết hợp
Atomicity	Tính nguyên tử
Behavioral elements	Các phần tử hành vi
Behavioral model	Mô hình hành vi
Boundary classes	Các lớp ranh giới
Bueprint	Kế hoạch chi tiết
Category	Phạm trù
Component-Based Software Engineering - CBSE	Công nghệ phần mềm dựa trên thành phần
The Common Closure Principle - CCP	Nguyên tắc đóng chung
Characteristic	Đặc điểm
Class-based	Dựa trên lớp
Class-oriented models	Các mô hình hướng lớp
Class-responsibility-collaboration model	Mô hình lớp-trách nhiệm-cộng tác

Client-server	Khách-chủ
Cohesion	Sự gắn kết
Collaboration	Sự cộng tác/hợp tác
Combinatorial specification of behavior	Đặc tả tổ hợp của hành vi
Commercial and nonprofit	Thương mại và phi lợi nhuận
Communication	Truyền thông
Compatibility	Khả năng tương thích
Component-level design	Thiết kế mức thành phần
Components	Các thành phần
Composability	Tính hợp thành
Compound transaction	Giao dịch tổ hợp
Concept	Khái niệm
Consistency	Nhất quán
Constrained dependencies	Phụ thuộc ràng buộc
Control Specification - CSPEC	Đặc tả kiểm soát
Controlled autonomy	Tính tự chủ được kiểm soát
Controller classes	Các lớp điều khiển
Convergence	Hội tụ
Counteroffers	Các đơn chào giá cạnh tranh
Counterproposal	Đề xuất phản hồi
Coupling	Sự nối kết
The Common Reuse Principle - CRP	Nguyên tắc tái sử dụng chung
Data abstraction	Trừu tượng hóa dữ liệu
Data attributes	Các thuộc tính dữ liệu
Data flow	Luồng dữ liệu
Data models	Các mô hình dữ liệu
Data structure	Cấu trúc dữ liệu
Data/class design	Thiết kế dữ liệu/lớp
Data-centered architecture	Kiến trúc lấy dữ liệu làm trung tâm
Data-flow architecture	Kiến trúc luồng/dòng dữ liệu
Decision table	Bảng quyết định

Dependencies	Các phụ thuộc
Deployment diagram	Sơ đồ triển khai
Deployment-level design	Thiết kế mức triển khai
Design patterns	Các mẫu thiết kế
Design for Reuse - DFR	Thiết kế để tái sử dụng
Diagram	Sơ đồ
Discoverability	Tính có thể phát hiện
Diversification	Đa dạng hóa
Domain-specific	Lĩnh vực cụ thể
Dynamic model	Mô hình động
E-commerce	Thương mại điện tử
Efficiency	Tính hiệu quả
Elaborative	Xây dựng chi tiết
Entity classes	Các lớp thực thể
Entity-Relationship Diagram- ERD	Sơ đồ thực thể-quan hệ
Ergonomically sound	Thân thiện với người sử dụng
Error handling	Xử lý lỗi
Flat transaction	Giao dịch phẳng
Flexibility	Linh hoạt
Flowchart	Sơ đồ khối
Flow-oriented	Hướng luồng
Framework	Khung làm việc
Framework model	Mô hình khung làm việc
Functional model	Mô hình chức năng
Functionality	Chức năng
Genres	Thể loại
Graphical User Interface - GUI	Giao diện người sử dụng đồ họa
Hierarchical structure	Cấu trúc phân cấp
Human Computer Interaction - HCI	Tương tác người máy
Hypermedia	Siêu phương tiện
Identifier	Định danh

Implementation model	Mô hình thực hiện
Informal use cases	Các trường hợp sử dụng chưa chính thức
Inheritance	Kế thừa
Instance	Thể hiện
Integration	Tích hợp
Interface design	Thiết kế giao diện
Layered architectures	Các kiến trúc phân lớp
Learnability	Tính có thể học
Linear structure	Cấu trúc tuyến tính
Main program/subprogram	Chương trình chính/chương trình phụ
Mental model	Mô hình tư duy
Menu	Trình đơn
Metaphor	Phép ẩn dụ
Middleware	Tầng trung gian
Model-driven development	Phát triển hướng dẫn dắt theo mô hình
Model-View-Controller Architecture - MVC	Kiến trúc mô hình - khung nhìn - bộ điều khiển
Navigability	Khả năng điều hướng
Navigation Semantic Unit - NSU	Đơn vị ngữ nghĩa điều hướng
Object elaboration	Xây dựng đối tượng
Object Oriented Hypermedia Design Method - OOHDM	Phương pháp thiết kế siêu phương tiện hướng đối tượng
Object-oriented architecture	Kiến trúc hướng đối tượng
Objects	Các đối tượng
Operations	Các phương thức/toán tử/thao tác
Paradigm	Dạng thức
Program Design Language - PDL	Ngôn ngữ thiết kế chương trình
Peer-level systems	Các hệ thống ngang cấp
Performance	Hiệu suất
Persistent classes	Các lớp bền vững
Plaform	Nền tảng
Principles	Các nguyên tắc

Problem-oriented	Hướng vấn đề
Procedural abstraction	Trừu tượng hóa thủ tục
Process Activation Table - PAT	Bảng kích hoạt tiến trình
Process Specification - PSPEC	Đặc tả tiến trình
Process view	Khung nhìn tiến trình
Processing rule	Quy tắc xử lý
Properties	Các đặc tính
Pseudocode	Ngôn ngữ giả
Readability	Khả năng đọc
Refactoring	Tái cấu trúc
Reliability	Độ tin cậy
Requirements engineering	Kỹ thuật/phân tích yêu cầu
Requirements model	Mô hình yêu cầu
Response time	Thời gian đáp ứng
Responsibilities	Các trách nhiệm
Reuse library	Thư viện tái sử dụng
Ripple effect	Hiệu ứng gợn sóng
Robustness	Sự mạnh mẽ
Scenario-based models of requirements	Các mô hình yêu cầu dựa trên kịch bản
Secondary scenarios	Các kịch bản thứ cấp
Sequential specification of behavior	Đặc tả tuần tự của hành vi
Simplicity	Tính đơn giản
Site map	Bản đồ trang Web
Service Oriented Architecture - SOA	Kiến trúc hướng dịch vụ
Software domain analysis	Phân tích lĩnh vực phần mềm
Software process	Quy trình/tiến trình phần mềm
Software Quality Assurance - SQA	Đảm bảo chất lượng phần mềm
Stage diagram	Sơ đồ trạng thái
Statelessness	Tính phi trạng thái
Stepwise refinement	Tinh chỉnh từng bước
Subassemblies	Các cụm lắp ráp

Subordinate systems	Các hệ thống phía dưới
Superordinate systems	Các hệ thống phía trên
Supportability	Khả năng hỗ trợ
Swimlane diagrams	Sơ đồ làn
Symbol	Ký hiệu
Template	Biểu mẫu
Test-driven development	Phát triển hướng dẫn dắt kiểm thử
Track state	Trạng thái theo vết
Tranational View	Quan điểm truyền thống
Universal Description	Mô tả tổng quát
Usability	Khả năng sử dụng
Use case	Trường hợp sử dụng
User Interface - UI	Giao diện người sử dụng
User model	Mô hình người sử dụng
Utilities	Tiện ích
Visible navigation	Sự điều hướng trực quan
Visual appeal	Ấn tượng từ bên ngoài
Ways of Navigating - WON	Cách điều hướng
Web Services Description Language - WSDL	Ngôn ngữ mô tả các dịch vụ Web
Workflow analysis	Phân tích luồng công việc

Chương 1
TỔNG QUAN VỀ THIẾT KẾ PHẦN MỀM

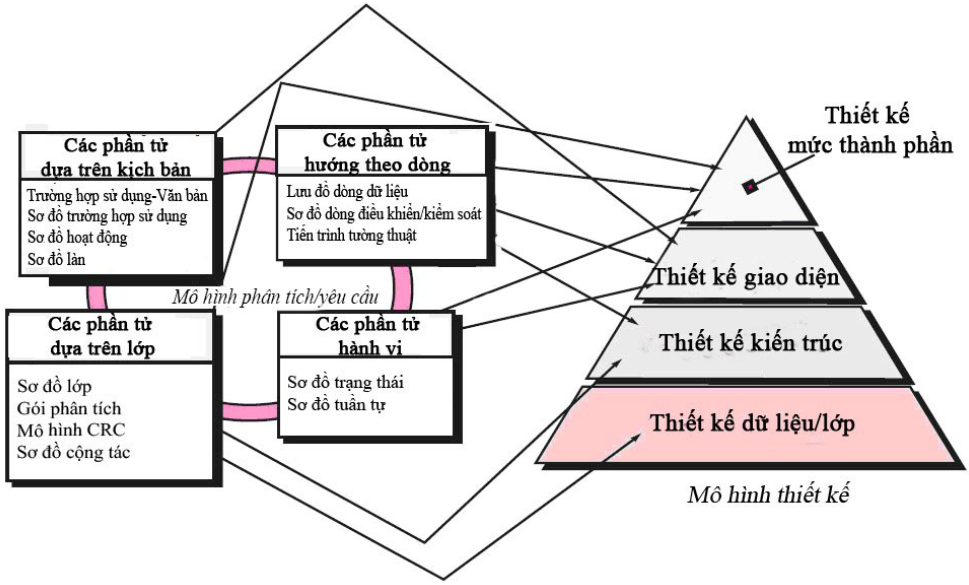
Thiết kế phần mềm (software design engineering) bao gồm tập hợp các nguyên tắc (principles), khái niệm (concepts) và thực hành (practices) dẫn đến sự phát triển của một hệ thống hoặc sản phẩm phần mềm chất lượng cao. Các nguyên tắc thiết kế thiết lập một triết lý quan trọng hướng dẫn việc thiết kế. Khái niệm thiết kế phải được hiểu trước khi cơ chế thực hành thiết kế được áp dụng. Thực hành thiết kế bản thân nó sẽ dẫn đến việc tạo ra các thể hiện/phiên bản khác nhau của phần mềm phục vụ như một hướng dẫn cho hoạt động xây dựng phần mềm về sau. Mục đích của thiết kế là tạo ra một mô hình hoặc một mô tả thể hiện độ vững chắc, tính thương phẩm và sự thích thú. Để thực hiện điều này, cần phải thực hành đa dạng hóa (diversification) và sau đó hội tụ (convergence). Chương 1 sẽ cung cấp cho người đọc kiến thức tổng quan về thiết kế phần mềm. Phần 1 và 2 của chương nhấn mạnh tầm quan trọng của giai đoạn thiết kế phần mềm trong quy trình phát triển phần mềm, và những hướng dẫn mà hoạt động thiết kế nên tuân thủ để đạt được các mục tiêu chất lượng. Phần 3 của chương sẽ cung cấp các khái niệm thiết kế phần mềm làm nền tảng cho việc tiếp nhận kiến thức các chương sau. Cuối cùng, Phần 4 giới thiệu các phần tử trong mô hình thiết kế: thiết kế dữ liệu, thiết kế kiến trúc, thiết kế giao diện, và thiết kế mức thành phần - chi tiết về những thiết kế này sẽ được lần lượt trình bày trong 4 chương kế tiếp.

1.1 THIẾT KẾ TRONG NGỮ CẢNH CÔNG NGHỆ PHẦN MỀM

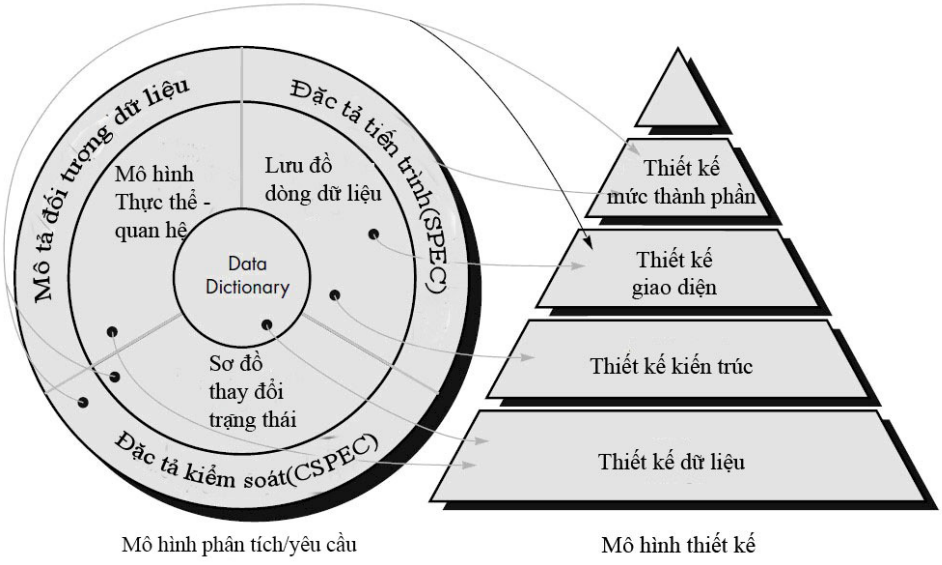
Thiết kế phần mềm nằm ở lõi kỹ thuật (technical kernel) của công nghệ phần mềm và được áp dụng không phụ thuộc vào mô hình quy trình/tiến trình phần mềm (software process) được sử dụng. Thiết kế phần mềm, được bắt đầu từ mỗi khi yêu cầu phần mềm đã được phân tích và mô hình hóa, là hành động công nghệ phần mềm cuối cùng trong hoạt động mô hình hóa và thiết lập ra giai đoạn xây dựng phần mềm (tạo mã lệnh và kiểm thử).

Mô hình yêu cầu (requirements model) còn được gọi là mô hình phân tích. Để thuận tiên cho việc trình bày, từ đây thuật ngữ mô hình yêu cầu sẽ được sử dụng. Mỗi một phần tử/yếu tố của mô hình yêu cầu cung cấp thông tin cần thiết để tạo ra bốn mô hình thiết kế cần có cho một đặc tả thiết kế hoàn chỉnh. Luồng thông tin trong quá trình thiết kế phần mềm được minh họa trong Hình 1.1 và Hình 1.2. Mô hình yêu cầu được thể hiện dựa trên kịch bản (scenario-based), dựa trên lớp (class-based), hướng luồng/dòng (flow-oriented) và các phần tử hành vi (behavioral element) cung cấp các nhiệm vụ thiết kế. Ký hiệu

thiết kế và phương pháp thiết kế được thảo luận trong các chương sau sẽ bao gồm thiết kế dữ liệu/lớp (data/class design), thiết kế kiến trúc (architectural design), thiết kế giao diện (interface design) và thiết kế thành phần (component design).



Hình 1.1 Chuyển đổi từ mô hình yêu cầu sang mô hình thiết kế (1)



Hình 1.2 Chuyển đổi từ mô hình yêu cầu sang mô hình thiết kế (2)

Thiết kế dữ liệu/lớp chuyển đổi mô hình lớp vào việc thực hiện thiết kế lớp và cấu trúc dữ liệu cần thiết được yêu cầu để cài đặt phần mềm. Các đối tượng và các mối quan hệ được xác định trong sơ đồ CRC (class-responsibility-collaboration) và nội dung dữ liệu chi tiết được mô tả bởi các thuộc tính lớp và ký hiệu khác tạo cơ sở cho hoạt động thiết kế dữ liệu. Một phần của thiết kế lớp có thể xảy ra kết hợp với thiết kế kiến trúc phần mềm. Thiết kế lớp chi tiết hơn xảy ra khi mỗi một thành phần phần mềm được thiết kế.

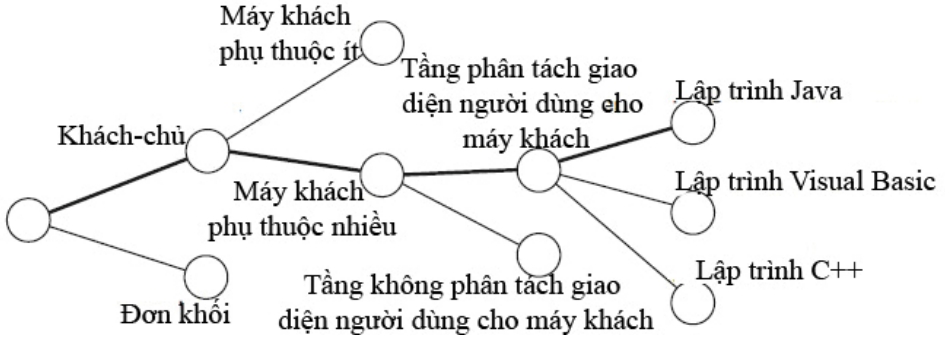
Thiết kế kiến trúc xác định mối quan hệ giữa các phần tử cấu trúc chính của phần mềm, các phong cách kiến trúc (architectural styles) và các mẫu thiết kế (design patterns) có thể được sử dụng để đạt được các yêu cầu được xác định cho hệ thống, và những ràng buộc ảnh hưởng đến cách thức mà kiến trúc có thể được cài đặt. Các biểu diễn của thiết kế kiến trúc, có nguồn gốc từ mô hình các yêu cầu, chính là khung nền của một hệ thống dựa trên máy tính.

Thiết kế giao diện mô tả cách thức phần mềm giao tiếp với các hệ thống tương thích với nó, và với con người người sử dụng nó. Một giao diện bao hàm một luồng thông tin (ví dụ như dữ liệu và/hoặc kiểm soát) và một loại hình cụ thể của hành vi. Vì vậy, kịch bản sử dụng và mô hình hành vi cung cấp nhiều thông tin cần thiết cho thiết kế giao diện.

Thiết kế mức thành phần chuyển đổi các phần tử cấu trúc của kiến trúc phần mềm thành một mô tả về thủ tục của các thành phần phần mềm. Thông tin thu được từ các mô hình dựa trên lớp, các mô hình luồng, và mô hình hành vi phục vụ như là cơ sở để thiết kế thành phần.

1.1.1 Thiết kế là một chuỗi các quyết định

Có thể xem quyết định thiết kế (xem Hình 1.3) với các thiết kế thay thế từ những lựa chọn khác nhau. Trong đó, các đường thẳng biểu diễn cho các tùy chọn và các đường đậm nét là tập hợp các quyết định được đưa ra.



Hình 1.3 Chuỗi các quyết định thiết kế

Trong quá trình thiết kế, các quyết định được đưa ra cuối cùng sẽ ảnh hưởng đến sự thành công của việc xây dựng phần mềm và rất quan trọng, tạo ra sự dễ dàng mà phần mềm có thể được bảo trì sau này. Do đó giai đoạn thiết kế là rất quan trọng!

Tầm quan trọng của thiết kế phần mềm có thể được ghi bằng một từ duy nhất là chất lượng. Thiết kế là nơi mà chất lượng được "nuôi dưỡng" trong công nghệ phần mềm. Thiết kế cung cấp các mô tả của phần mềm có thể được đánh giá về chất lượng. Thiết kế là cách duy nhất có thể dịch chính xác yêu cầu của các bên liên quan vào một sản phẩm hoặc hệ thống phần mềm hoàn chỉnh. Thiết kế phần mềm phục vụ như nền tảng cho tất cả các hoạt động kỹ thuật phần mềm và hỗ trợ phần mềm theo sau. Nếu thiết kế không có, nguy cơ xây dựng một hệ thống không ổn định sẽ xuất hiện, và sự thất bại có thể xảy ra ngay cả khi một thay đổi nhỏ được thực hiện; sự kiểm thử có thể gặp khó khăn; chất lượng không thể được đánh giá cho đến gần thời điểm cuối trong tiến trình phần mềm khi mà thời gian còn lại ngắn và nhiều kinh phí đã được sử dụng.

1.2 TIẾN TRÌNH THIẾT KẾ

Thiết kế phần mềm là một quá trình lặp (iterative process) mà thông qua đó yêu cầu được chuyển thành một "kế hoạch chi tiết" ("bueprint") để xây dựng phần mềm. Ban đầu, kế hoạch chi tiết mô tả một cái nhìn toàn diện của phần mềm. Đó là, các thiết kế được thể hiện ở một mức độ trừu tượng cao - một mức độ mà nó có thể được truy vết một cách trực tiếp đến mục tiêu cụ thể của hệ thống và và các yêu cầu về hành vi chức năng, và dữ liệu chi tiết hơn. Khi việc thiết kế được lặp lại, các tinh chỉnh tiếp theo dẫn đến những thể hiện của thiết kế ở các cấp thấp hơn so với cấp trừu tượng, những thiết kế này vẫn có thể được truy nguồn từ yêu cầu, nhưng kết nối là tinh tế hơn.

1.2.1 Hướng dẫn và thuộc tính chất lượng phần mềm

Trong suốt quá trình thiết kế, chất lượng của các thiết kế phát triển được đánh giá bằng một loạt các đánh giá kỹ thuật. Trong đó, ba đặc điểm được đề nghị như những hướng dẫn cho việc đánh giá một thiết kế tốt là:

- Việc thiết kế phải thực hiện tất cả các yêu cầu tường minh có trong mô hình yêu cầu, và nó phải chứa tất cả các yêu cầu tiềm ẩn mong muốn của các bên liên quan.

- Việc thiết kế phải là một hướng dẫn dễ hiểu, dễ đọc đối với người viết các mã lệnh, người kiểm thử, và sau đó là người hỗ trợ phần mềm.

- Việc thiết kế nên cung cấp một bức tranh hoàn chỉnh về phần mềm, giải quyết các dữ liệu, chức năng, và các lĩnh vực hành vi từ quan điểm cài đặt.