

目次

1. はじめに	13
2. デザインパターン入門.....	15
2.1 ソフトウェア開発の現状と問題点.....	15
2.1.1 動かないソフトウェア	15
2.1.2 保守できないソフトウェア	16
2.1.3 再利用できないソフトウェア	17
2.1.4 品質の悪いソフトウェア	17
2.2 よい設計とは?	18
2.2.1 外的品質要因	18
2.2.2 内的品質要因	20
2.2.3 モジュール性の原則	21
2.2.4 再利用可能なモジュール構造の要件	22
2.3 なぜデザインパターンか?	24
2.4 デザインパターンとは?	26
2.5 デザインパターンの例.....	27
2.6 デザインパターンの詳細を記述するための仕様記述言語について	30
3. デザインパターン解説.....	33
3.1 パターンの種類.....	33
3.2 デザインパターンの分類	35
3.3 生成パターン	35
3.3.1 抽象工場 (Abstract Factory)	36
3.3.2 工場操作 (Factory Method)	41
3.3.3 一枚札 (Singleton) パターン	43
3.4 構造パターン	44
3.4.1 変換器 (Adapter) パターン	45
3.4.2 混成 (Composite) パターン	48

3.5	振る舞いパターン	51
3.5.1	命令 (Command) パターン	52
3.5.2	通訳 (Interpreter) パターン	54
3.5.3	観察者 (Observer) パターン	59
3.5.4	状態 (State) パターン	62
3.5.5	戦略 (Strategy) パターン	64
3.6	分析パターン	66
3.6.1	当事者 (Party) パターン	66
3.6.2	組織構造 (Organization Structure) パターン	67
3.6.3	責任 (Accountability) パターン	69
3.7	アーキテクチャパターン	70
3.7.1	階層 (Layer) パターン	70
3.7.2	クライアント / サーバー・パターン	72
3.8	算法 (Algorithm) パターン	79
3.8.1	辞書 (Dictionary) パターン	79
4.	デザインパターンによる OOA/OOD 開発技法	85
4.1	デザインパターンを使ったシステム開発の概要	85
4.2	ドメイン分析	87
4.2.1	ビジネスゴールの作成	88
4.2.2	用語集の作成	89
4.2.3	UseCase 作成および制約の発見	90
4.2.4	ドメインオブジェクトの発見	108
4.2.5	オブジェクトの状態変化の記述	115
4.2.6	要求仕様記述	117
4.2.7	ドメインモデルの検証	131
4.3	システム分析	135
4.3.1	クラス図の作成	135
4.3.2	操作仕様の記述	143
4.3.3	状態遷移図の作成	145
4.3.4	分析モデルの検証	147
4.4	設計	149
4.4.1	アーキテクチャの決定	150
4.4.2	デザインパターンを考慮したクラス図の修正	151
4.4.3	効率の検討	156
4.4.4	操作仕様の変換	161
4.4.5	状態遷移の実装方法決定	169
4.4.6	関連の実装方法決定	174

4.4.7	実装図の作成	175
4.4.8	再利用性・保守性分析	177
4.4.9	設計モデルの検証	179
5.	制御系システムの分析と設計	181
5.1	分析	181
5.1.1	用語集の作成	182
5.1.2	UseCase の作成	183
5.1.3	制約の発見と記述	186
5.1.4	制約を考慮したシナリオの見直し	188
5.1.5	状態遷移図の作成	188
5.1.6	状態遷移図の検証	190
5.1.7	クラス図の作成	191
5.1.8	オブジェクトを分類する	195
5.2	設計	195
5.2.1	アーキテクチャの決定	195
5.2.2	デザインパターンの適用	196
5.2.3	状態遷移図の修正	197
6.	導入の課題と今後の動向	199
6.1	デザインパターン導入の課題	199
6.1.1	デザインパターンの前にやっておくべきこと	199
6.1.2	体制の変革	200
6.1.3	デザインパターン作成・管理チームの必要性	200
6.1.4	デザインパターンだけでは、問題は解決しない	201
6.2	今後の動向	201
6.2.1	形式技術	202
6.2.2	オブジェクト指向技術	203
7.	付録	205
7.1	UML1.1 概要	205
7.1.1	UseCase 図	205
7.1.2	クラス図	206
7.1.3	状態遷移図	210
7.1.4	順序図	213
7.1.5	配置図	214
7.2	仕様記述言語 (OCL) 概要	215

7.2.1	RSL から追加した構文	216
7.2.2	OCL を使う場所	217
7.2.3	OCL の型	218
7.2.4	演算子の優先順位	219
7.2.5	注釈	219
7.2.6	未定義の値	219
7.2.7	特性 (Property)	219
7.2.8	すべてのオブジェクトに共通の定義済みの操作	222
7.2.9	Collection	223
7.2.10	Collection の既定義操作	223
7.3	参考文献	227
7.3.1	オブジェクト指向	227
7.3.2	デザインパターン	228
7.3.3	仕様記述	229
7.3.4	アルゴリズムとデータ構造	229
7.3.5	その他	230
7.4	関連 URL	231
7.4.1	オブジェクト指向	231
7.4.2	デザインパターン	231
7.4.3	仕様記述	232
7.4.4	ビジネスオブジェクト	232