

ソフトウェアパターン

平成25年度シラバス

2013年 1月 4日

国立情報学研究所

トップエスイープロジェクト

代表者 本位田 真一

1. 講座名

ソフトウェアパターン

2. 担当者

鷲崎 弘宜、鄭 顕志、高橋 竜一ほか

3. 本講座の目的

本講座では、携帯端末およびWebサーバが協調動作するネットワーク情報システムを題材とした産業ソフトウェアのオブジェクト指向分析／設計／実装問題を扱う。近年、大規模化・複雑化するソフトウェアの分析／設計／実装において、変更への強さとトレーサビリティの向上を目的としてオブジェクト指向開発の適用が普及しつつある。しかし、オブジェクト指向開発の現状として、分析から実装へのギャップを埋める明示的プロセス、分析／設計／実装におけるコミュニケーション手段、熟練者のノウハウ・思考方法の再利用・移転手段がそれぞれ欠けているという問題がある。これらの問題を解決する技術として、開発上の定石（頻出する問題と解決および制約を抽象化した知識）を一定の形式に従って記述したソフトウェアパターンおよびその支援環境がある。ソフトウェアパターン技術は、上述の問題を解決し、その結果として、一貫したアーキテクチャの確立、品質の均一化、および、保守性／拡張性／生産性の向上をもたらす。しかし、ソフトウェアパターンを使いこなすには多くの難しさがあり、普及への障害となっている。そこで本講座では、携帯端末シミュレータおよびWebサーバ上でのオブジェクト指向ソフトウェア開発を通して、情報機器系のシステム開発において代表的な既存ソフトウェアパターン（主にアーキテクチャパターンおよびデザインパターン）の適用ノウハウ、設計原則／非機能的特性からの既存／新規ソフトウェアパターンの適用範囲と取捨選択、支援環境のメリット／デメリットおよび適用範囲を習得し、これらの習得を通じて、自身のシステム開発におけるパターン指向開発力を獲得する。また、グループ討議を通じた例題演習により、議論を通してソフトウェアパターン技術の理解を深め、ソフトウェアパターン活用プロセスの実用的ノウハウを体得できる効果が期待できる。

4. 本講座のオリジナリティ

ソフトウェアパターンおよび支援環境を用いたオブジェクト指向分析／設計／実装については、従来多くの同様な講座が開講されている。しかしそれらの講座にはさまざまな問題点があったため、受講生が習得した内容を速やかに開発現場で適用するための障害となっていた。本講座では、それらの問題点を解消し、本講座受講後に既存あるいは新規のソフトウェアパターンおよび支援環境を速やかに開発現場で適用できるように配慮している。

表 1 に、既存の講座の問題点と、本講座における解を示す。

表 1 既存の講座の問題点と、本講座における解

既存の講座の問題点	本講座における解
個々の既存ソフトウェアパターンの独立した扱いを中心とし、既存のソフトウェアパターン間の繋がりや、将来における新たなソフトウェアパターンを含めての取捨選択方法を習得するのが難しい	アーキテクチャパターンおよびデザインパターンの集合について、オブジェクト指向設計における設計原則や非機能的特性の観点から分析し整理する方法を実習するため、将来における新たなソフトウェアパターンを含めたソフトウェアパターンの関係分析や取捨選択の方法を習得可能
ソフトウェアパターン技術が解決する、現状のオブジェクト指向開発における問題点を体感できないため、ソフトウェアパターン技術の有用性を実感するのが難しい	同一の演習問題について、ソフトウェアパターン技術を活用しない場合と活用する場合の両方を実習し、その Before/After の比較を通じて効果を体感するため、従来の場合当たりのオブジェクト指向分析／設計／実装に比べての、ソフトウェアパターン技術の有用性を体感可能
ソフトウェアパターン支援環境（例えば適用ツール）について、有効に働く局面のみを取り上げるため、現状の支援環境の有効性が極めて限定的であることを実感するのが難しい	複数の演習問題について、現実的な開発プロセスの中で現状の支援環境が有効に働く局面と有効ではない局面をそれぞれ実習するため、支援環境のメリット／デメリットおよび適用範囲を習得可能
現実的な問題をそのまま使用せず、抽象化・簡略化されたトイ例題を題材にしているため、実際のシステム開発に対する有用性を実感するのが難しい	トイ例題に対するソフトウェアパターン技術を用いた分析／設計／実装プロセス、トイ例題を複合した現実的な例題に対する分析／設計／実装プロセス、そして、現実的な例題の骨格を抽象化したフレームワークの開発プロセスといった段階的な講座進行をとるため、実際のシステム開発に対する有用性を実感できるまで、無理のない習得が可能

5. 本講座で扱う難しさ

多様な機器がネットワークを介して接続されて協調動作する高度な情報システムの必要性が急速に増大している。このような複雑かつ大規模なシステムの開発にあたり、その分析や設計・実装をそのつど各開発者独自の方法で行うときに、以下の問題に直面する。

- 対象領域の熟練者でなければ、機能要求に加えて非機能要求（例えば保守性やパフォーマンス）を同時にきれいに満たすシステムの構築は困難である。
- システムが大規模であればあるほど、システム全体のアーキテクチャの一貫性を保つことが困難となり、結果として保守性が低下する。
- 特にオブジェクト指向開発の場合、オブジェクト指向特有の適用方法の難しさに起因して、変更に強いソフトウェアシステムを効率よく開発するためのノウハウは表しにくく、共有しにくい。

図 1 および図 2 は、ネットワークに接続された情報端末上で番組の予約を行うシステムの全体像およびオブジェクト指向分析結果を示す例である。ここで予約機能についてシステムを、画面部（バウンダリ）、制御部（コントロール）、およびデータ部（エンティティ）の 3 部構成によって実現すると分析したものとする。分析以降の工程において信頼性の高いシステムを設計・実装するためには、分析結果では明らかとなっていない(a)各部間の通信方法の設計や(b)制御部におけるセキュリティ認証方法の設計、(c)データ部の永続化方法の設計、(d)制御部／データ部における並行性処理の設計、および、(e)それらを一貫して実現するためのアーキテクチャの設計などが必要となる。さらにそれらの設計は、将来の予測可能な範囲における要求変更に対応するものでなくてはならない。各部について変更に対応した最適な設計を、一から考えて実施することは可能であるが、その実施は容易ではなく、また、システム全体として一貫性を失う可能性がある。

前掲の様々な設計問題は同様の文脈上で繰り返し発生するため、これまでに、高い拡張性や保守性を持ちつつそれぞれの機能要求を満たすような設計を促す解法が、種々のデザインパターンおよびアーキテクチャパターンとしてまとめられ再利用されつつある。例えば(a)については Observer パターンや Mediator パターン、(b)については Proxy パターンや Dependency Injection パターン、(c)については DAO パターン、(d)については Read/Write Lock パターン、(e)については Layers パターンをそれぞれ解法の候補として検討し、既知の設計ノウハウとして再利用することができる。

このように信頼性の高いシステムを効率よく実現するためには、ソフトウェア開発の上流工程において各種のソフトウェアパターンを適切に組み合わせて適用すること、および、実装工程において、ソフトウェアパターンの適用を支援する適切な開発支援環境を活用していくことが非常に重要となる。しかし、分析工程から実装工程までの一貫したソフトウェアパターンの活用を指向する開発方法論は不在であり、実問題に適用する為の実践的ノウハウとして整理されたカリキュラムは少なく、ソフトウェアパターン指向開発を実現していく上での大きな妨げとなっている。

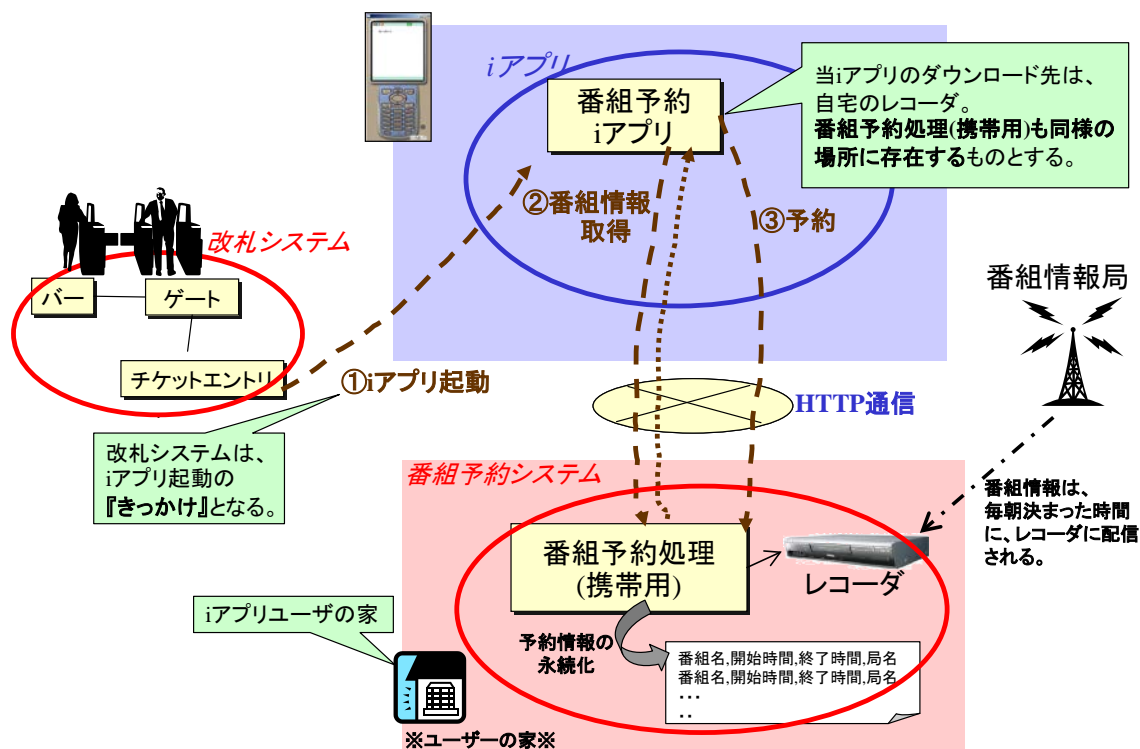


図1 ネットワーク対応型番組予約システムの全体像

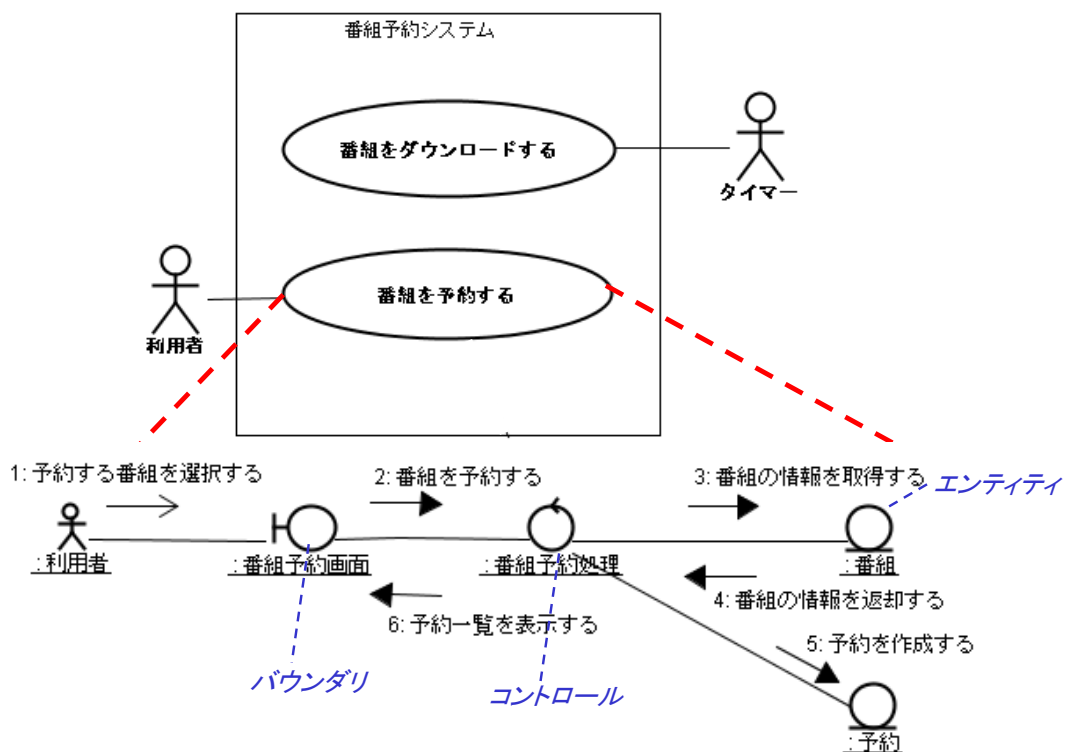


図2: 番組予約システムの分析

6. 本講座で習得する技術

本講座では、ネットワーク対応型番組予約システムを題材として産業ソフトウェアのオブジェクト指向分析／設計／実装問題を扱い、ソフトウェアパターンの必要性和、ソフトウェア開発の上流工程の重要性を習得する。具体的には、パターン指向開発技術、およびソフトウェアパターンを利用した分析／設計／実装工程について学び、信頼性の高いソフトウェアを開発するために有用な素養と知識を身に付ける。さらに、i アプリに代表される情報機器を組み合わせたネットワーク型システムのオブジェクト指向分析／設計／実装問題にソフトウェアパターンを用いた設計の演習を実施し、ソフトウェアパターン指向開発の知識と活用能力を体得する。

本講座で取り上げる手法およびツール、既知のパターンを以下にあげる。

- 手法
 - 上流から下流まで一貫してパターンの繋がりを意識したパターン指向開発
 - パターンマイニング手法
 - 品質特性シナリオに基づくアーキテクチャ設計/評価手法
 - ✧ 品質駆動型設計 (ADD)
 - ✧ アーキテクチャトレードオフ分析 (ATAM)
- ツール
 - UML モデリングツール: astah*
 - デザインパターン検出ツール: Tsantalis らの検出ツール
 - State パターン展開ツール: SMC (State Machine Compiler)
 - 実装環境: Java2, Eclipse
- ソフトウェアパターン
 - アナリシスパターン: [児玉 04]、[Fowler98] (もの-こと-こと、勘定など)
 - アーキテクチャパターン: [POSA] (Layers, MVC など)
 - デザインパターン: [GoF95]、[J2EE02]、[PofEAA03] (State, Observer など多数)
 - オブジェクト指向設計原則: [Martin04] (クラス設計原則、パッケージ設計原則)

7. 前提知識

本講座の受講生は、TopSE 講座「ソフトウェア工学入門」の履修等を通じて、以下の項目を習得済みであることが望ましい。

- オブジェクト指向プログラミング
- オブジェクト指向分析／設計手法
- UML

8. 講義計画

- ・ 概要（予定、変更の可能性あり）

第1回 ソフトウェアパターン導入

第2回 パターンの抽出

第3回 パターンの利用

第4回 パターン指向開発プロセス

第5回 パターン指向開発の実践

第6回 アーキテクチャパターン実践

第7回 アーキテクチャ設計評価(1)

第8回 アーキテクチャ設計評価(2)

第9回 デザインパターン実践(1)

第10回 デザインパターン実践(2)

第11回 デザインパターンと設計原則

第12回 オブジェクト指向フレームワークとデザインパターン

第13回 ソフトウェアパターンの広がり

第14回 組織へのパターン導入事例

第15回 パターン技術の展望と全体のまとめ

- ・ 詳細（予定、変更の可能性あり）

第1回 ソフトウェアパターン導入

座学:

- パターンの概要、要求との関係、歴史的背景
- パターン記法（フォーマット）
- OOプログラミング、モデリング概要

演習: 記法に基づく身近なパターンライティング

- 何がパターンで、何がパターンではないのか
- フォース、非機能要求との関係
- パターンの抽象度と適用範囲のトレードオフ

作業形態: 個人実習

宿題: 演習結果の清書

第2回 パターンの抽出

座学:

- パターンのライフサイクル
- パターン抽出活動とパターンマイニング
- 手法: H&L パターンマイニング手法

演習:

- 身近なソフトウェア体験からのパターン抽出

作業形態: グループ実習

宿題: 演習結果の清書

第3回 パターンの利用

座学:

- パターン適用プロセス
- ツール: SMC

演習:

- 鉄道改札システムの設計、実装への State パターン適用
- 非機能的特性の変化考察

作業形態: 個人実習

宿題: 演習結果の清書

第4回 パターン指向開発プロセス

座学:

- パターン指向開発プロセス
- パターンの縦・横の繋がり、組み合わせ例
- 代表的分析パターン、アーキテクチャパターン、デザインパターン

演習:

- パターン活用経験の振り返り、効果について討論

作業形態: グループ演習

宿題: (なし)

第5回 パターン指向開発の実践

座学:

- 代表的イディオム、その他

演習:

- 番組予約システムのドメイン分析、システム分析、アーキテクチャ設計

作業形態: 個人実習

宿題: 演習結果の清書

第6回 アーキテクチャパターン実践(1)

座学:

- 代表的アーキテクチャパターンと非機能的特性

- パッケージレベル設計原則
- アーキテクチャの定性的評価と定量的評価

演習:

- アーキテクチャパターンの整理と設計原則との関係分析

作業形態: グループ実習

宿題: 演習結果の清書

第 7, 8 回 アーキテクチャ設計評価 (1), (2)

座学:

- 品質シナリオとパターンによるアーキテクチャ設計
- 手法: 品質駆動型設計 (ADD)、アーキテクチャトレードオフ分析 (ATAM)

演習:

- アーキテクチャの設計と評価

作業形態: グループ実習

宿題: 演習結果の清書

第 9, 10 回 デザインパターン実践 (1), (2)

座学:

- デザインパターンの仕組みと必要性
- 代表的デザインパターン
- パターン連続適用プロセス

演習:

- 番組予約システムのデータ構造周辺部詳細設計への複数デザインパターン連続適用

作業形態: 個人実習

宿題: 演習結果の清書

第 11 回 デザインパターンと設計原則

座学:

- クラスレベル設計原則
- デザインパターンと設計原則の関係

演習:

- State パターンによって満たされる設計原則分析
- 番組予約番組予約システムの永続化詳細設計へのパターン・設計原則連続適用

作業形態: 個人実習

宿題: 演習結果の清書

第 12 回 オブジェクト指向フレームワークとデザインパターン

座学:

- オブジェクト指向フレームワーク (OO-FW)
- OO-FW 開発/利用へのパターン活用

演習:

- 番組予約システムのフレームワーク設計へのパターン適用

作業形態: グループ演習

宿題: 演習結果の清書

第 13 回 ソフトウェアパターンの広がりと活用ツール

座学:

- 各種ソフトウェアパターン適用領域
- 種々のパターン: GoF, JavaEE, PoFEEA など
- ソフトウェアパターンとツール
- ツール: Tsantalis らの検出ツール

演習:

- 番組予約番組予約システム実装からのデザインパターン検出

宿題: (なし)

第 14 回 組織へのパターン導入事例

座学:

- パターン指向開発の組織への導入方法
- 組織固有パターンの定義、適用方法・ツール
- パターン指向開発の導入事例

演習:

- パターンの組織導入について討論

作業形態: グループ演習

宿題: (なし)

第 15 回 パターン技術の展望と全体のまとめ

座学:

- 全講義内容の振り返り
- 全演習の振り返り
- パターン技術の研究動向

演習:

➤ パターン技術の有用性について討論

作業形態：グループ演習

宿題：（なし）

9. 教育効果

本講座を受講することにより、実際の情報機器システムの開発に、既存のアーキテクチャ／デザインパターンを適用できるようになる。具体的には、既存パターンの適用範囲の見極め、機能要求・非機能要求に合致した既存パターンの取捨選択の方法、既存パターンの組み合わせ・連続適用方法、一貫したアーキテクチャ設計・詳細設計力を習得できる。また、実際のシステム開発経験から新たなアーキテクチャ／デザインパターンを発見して組織へ展開し、パターン指向開発への取り組みを開始できるようになる。具体的には、パターン適用ツールの効果と限界、パターン記法、非機能的特性・設計原則のパターンとの関係について習得できる。

10. 使用ツール、および、ソフトウェアパターン

SMC: 状態表コンパイラ、State パターン適用

- 困難さ: 事前補助クラス群作成が難しい、適用後保守が難しい
- 必要ノウハウ: 事前準備手順、適用後の読解
- 選択理由: フリー、State パターンを実直に適用
- 実用性: 組込み等での迅速な実装、適用ルーチンワーク排除

ATL: デザインパターン適用 (展開)

- 困難さ: パターンの定義および適用方法
- 必要ノウハウ: パターンの定義および適用手順
- 選択理由: オープンソース、モデル駆動開発の普及の兆し
- 実用性: 分析・設計・実装時の既知パターン迅速適用、新パターンの定義と組織展開

その他、パターン関係および一般的なツール

- Tsutalis らのデザインパターン検出ツール
- Eclipse: 統合開発
- astah*: UML モデリング

アーキテクチャパターン [POSA]

- 困難さ: 各アーキテクチャパターン取捨選択、適用方法
- 必要ノウハウ: 非機能的特性や設計原則に基づく取捨選択、適用方法
- 選択理由: 基本的、代表的
- 実用性: アーキテクチャ設計時の基本的アーキテクチャ設計に有用

デザインパターン [GoF][J2EE02][PofEAA]

- 困難さ: 各デザインパターン取捨選択、連続適用方法
- 必要ノウハウ: 非機能的特性や設計原則、GRASP、メタパターンに基づく関係、取捨選択、連続適用方法
- 選択理由: 基本的、代表的
- 実用性: 詳細設計時に既知/新アーキテクチャパターン活用は有用、

オブジェクト指向設計原則 [Martin04]

- 困難さ: 他との併用が難しい、何に効くのか分かりにくい
- 必要ノウハウ: アーキテクチャ/デザインパターン併用手順、非機能的特性との関係
- 選択理由: 基本的、代表的

- 実用性: アーキテクチャ設計時に既知/新デザインパターン活用に有用

その他、触れる基本的パターン

- アナリシスパターン [児玉 04][Fowler98]

11. 実験及び演習

個人単位、および、2～3 名程度の少人数で構成されたグループ単位で作業を行う。ネットワーク型番組予約システムを構成するソフトウェアを題材にして、パターン指向開発に基づくソフトウェアの分析/設計/実装を行う。非機能要求、特に拡張性・保守性・一貫性を維持した分析/設計/実装の難しさを体感しながら、パターンを活用した分析/設計モデリングおよび実装コーディングプロセスを体得する。グループ内で、パターンの適用前後について比較評価を行い、パターンおよび設計原則適用の有用性と適用性を議論する。議論を通してパターン指向開発の理解を深めながら、分析/設計/実装プロセスの実用的ノウハウを会得する。

12. 評価

演習課題レポート、プレゼン発表、出席日数を総合して評価する。

13. 教科書/参考書

- 深澤良彰（監修），鷺崎弘宜，丸山勝久，山本里枝子，久保淳人，“ソフトウェアパターン”，近代科学社，2007.
- 羽生田栄一（監修），金澤典子，井上健，森下民平，鷺崎弘宜，佃軍治，細谷竜一，瀬戸川教彦，山野裕司，沖田直幸（著），“ソフトウェアパターン入門～基礎から応用へ”，ソフトリサーチセンタ，2005.

パターン指向開発を構成する基盤技術としての各種パターン、パターン体系、パターンとフレームワークおよび設計原則について述べられており、この講義に最適である。

- Martin Fowler（著），堀内一（監訳），児玉公信，友野晶夫（訳），"アナリシスパターン：再利用可能なオブジェクトモデル"，アジソン・ウェスレイ・パブリシャーズ・ジャパン，1998.
 - Mary Shaw and Garlan David, "Software Architecture: Perspectives on an Emerging Discipline", Prentice Hall, 1996.
 - Franc Busshman ほか（著），金澤典子ほか（訳），"ソフトウェアアーキテクチャ：ソフトウェア開発のためのパターン体系"，近代科学社，2000.
 - Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides（著），本位田真一，吉田和樹（監訳），"オブジェクト指向における再利用のためのデザインパターン 改訂版"，ソフトバンクパブリッシング，2000.
 - Robert C. Martin（著），瀬谷啓介（訳），"アジャイルソフトウェア開発の奥義：オブジェクト指向開発の秘伝書"，ソフトバンクパブリッシング，2004.
- 代表的なアナリシスパターン、アーキテクチャパターン、デザインパターン、オブジェクト指向設計原則について詳細に解説されており、各パターン/設計原則の習得に最適である。