

# アスペクト指向開発

平成24年度シラバス

2012年10月23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. 科目名

アスペクト指向開発

## 2. 担当者

鷲崎 弘宜、鄭 顕志、小高 敏裕、高橋 竜一

## 3. 本科目の目的

本科目では、主にエンタープライズシステム分野における産業ソフトウェアを対象として、保守性の高い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。具体的には、従来のオブジェクト指向開発技術ではモジュール化/局所化が困難であった種々の関心事の分離を開発における上流工程から下流工程まで一貫して保つことで保守性の高いソフトウェアを高い生産性を伴って開発するために、対象問題のオブジェクト指向を発展させたアスペクト指向に基づく抽象化/モジュール化による分析能力、ならびにモジュールの分離/合成による設計/プログラミング能力を修得させることを目的とする。演習を通して、産業としてのソフトウェア開発におけるアスペクト指向に基づく開発プロセスを体験させ、アスペクト指向開発に必要な素養や知識を習得させる。

## 4. 本科目のオリジナリティ

アスペクト指向プログラミング (Aspect-Oriented Programming: AOP) および支援環境を用いたアスペクト指向実装技術については、従来幾つかの同様な講座・講義が開講されている。しかしそれら講座・講義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。本科目では、それらの問題点を解消し、本科目受講後に既存あるいは新規の分析／設計を含むアスペクト指向開発 (Aspect-Oriented Software Development: AOSD) 技術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・講義の問題点と、本科目における解を示す。

表 1 既存の講座・講義の問題点と、本科目における解

既存の講座・講義の問題点	本科目における解
実装段階で横断的関心事を分離/合成する様々な AOP があるが、特性や適用可能領域、有効性が十分にまとめられていない。	代表的な AOP 処理環境である AspectJ, JBossAOP, SpringAOP の演習と比較考察を実施し、それらに共通な部分/異なる部分を理解する。AOP を利用せずにオブジェクト指向分析設計およびプログラミングを実施する場合との比較実習により AOP の長所・短所を理解する。
AOP 処理系・動作フレームワークを主にブラックボックスなツールとして扱い、AOP が機能する具体的な仕組み・背景理論を把握しにくい。	JavaScript により AOP 動作フレームワークを独自に実装し、AOP 処理系やフレームワークに必要な要素技術、および、仕組みを理解する。
周辺技術としてオブジェクトコンテナ, DI (Dependency Injection), アノテーションがあるが、AOP との関係は明確に整理されていない。	DI コンテナを用いる場合の演習と比較考察を実施し、AOP との関係と併用ノウハウを習得する。アノテーションとの併用についても言及する。
上流の分析から AOP 実装に結びつけるプロセスはあるが、ノウハウ（標準環境における記述法、）がまとめられていない。完結した「例」がない。	ユースケースによるアスペクト指向開発手法（AOSD with Usecases）と AspectJ/ SpringAOP の併用により一貫した実用的例題を分析から実装に至るまで演習を実施し、プロセスと適用ノウハウを習得する。ICONIX と Java を併用した場合との比較実習により AOSD の長所・短所を理解する。

## 5. 本科目で扱う難しさ

近年、エンタープライズシステムは大規模化、複雑化が進んでいる。大規模化/複雑化しつつあるエンタープライズシステムでは、高い価値をもたらす機能やサービスを、複数のシステムやモジュールが内部的あるいはネットワーク越しに接続され、相互に連携して動作することで実現する。従来の小規模なシステムの制御ソフトウェア開発に課せられる非機能要求は一般に低く、また、機能要求も小さいものであり、機能もしくはデータ単位のモジュール分割と合成を基調として開発されていた。対して近年のネットワークシステムでは、プラットフォーム/環境変革の速さや企業/利用者ニーズの多様化および変化の速さ、および、価値増大のためのスケーラビリティや性能/可用性の高い要求に応じるため、機能やサービスを高い保守性および信頼性を持って効率よく実現する技術と利用する技術が必要となる。このような文脈において、高品質かつ高生産なソフトウェア開発を実施するために、機能もしくはデータ単位のモジュール組み合わせに代表される従来型の開発方法論は新規開発や保守/拡張に膨大な工数を必要とし、もはや現実的手段ではない。

高い品質が要求される多機能なエンタープライズシステムを制御するソフトウェアの複雑さの例（抜粋）を図1に示す。図1は、ネットワーク越しに顧客がWebクライアントを用いてホテルの部屋を予約し、また、顧客のホテル到着時に管理者がチェックイン処理する機能を、ユーザ認証による高信頼性を伴って実行可能なシステム制御ソフトウェアのオブジェクト指向に基づく要求分析および設計の結果を示す例である。ここで、顧客による部屋の予約要求を従来のオブジェクト指向開発方法論に基づいて分析/設計するならば、ソフトウェアを顧客画面、予約処理、部屋、予約の4つのモジュール（クラス）によって実現する設計結果を得る。一方、他の機能要求として管理者によるチェックイン処理要求について同様に分析/設計すると、管理者画面、チェックイン処理、部屋、予約の4つのクラスによって実現することとなり、部屋および予約のクラス群内部において保持するデータや処理の「もつれ合い」が発生し、また、各機能要求を実現するためのデータや処理の多数のクラスへの「散らばり」が発生する。さらに、高い信頼性を満たすためユーザ認証の仕組みを各機能に伴ってオブジェクト指向により実現するならば、その仕組みを実現するためのデータや処理も複数のクラスに「散らばる」と同時に他のデータや処理と「もつれ合う」結果となる。これらのソフトウェア構造における「もつれ合い」や「散らばり」は、各機能要求/非機能要求の変更に対する変更容易性や将来の拡張に対する拡張性および部分の再利用性の著しい低下につながり、さらには開発における生産性の低下を引き起こす。

上述の分析/設計問題に対する解法として、アスペクト指向開発技術を用いることができる。アスペクト指向開発では、各機能要求/非機能要求をそれぞれ異なり分離された「関心事」として捉えて、各関心事を実現する分析/設計レベルのクラスを分離されたアスペクトもしくはアスペクトを包含するモジュール単位に局所化し、分析/設計/実装/運用の任意の時点において合成する。ここでアスペクトとは、従来のオブジェクト指向開発においてクラス集合からなるソフトウェア構造に「散らばり」、他と「もつれ合う」データや処理を 1

つにまとめあげて後の合成を可能とする新しいモジュール単位である。アスペクト指向技術の適用により、上流から下流まで一貫した関心事およびその実現結果の分離と追跡可能性の維持を実現し、その実現により高い保守性/再利用性/生産性を達成する。具体的には、上流工程においてユースケースによるアスペクト指向開発手法を代表とするアスペクト指向分析/設計手法を適用し、下流工程において AspectJ や JBossAOP を代表とするアスペクト指向プログラミング技術/環境（フレームワークおよびコンテナ）を適用することができる。また、下流工程において Spring Framework を代表とする依存性注入（Dependency Injection）の仕組みを併用可能なアスペクト指向プログラミング環境の適用も重要である。

このように高い品質が要求される多機能なエンタープライズ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アスペクト指向に基づく分析/設計方法論を適用すること、下流工程においてアスペクト指向プログラミングを適用すること、および、それら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重要となる。実際のソフトウェア開発にこれらの手法/技術を適用していく為には、それぞれの開発方法論/プログラミング技術が前提としている条件を理解し、現在抱えている現実問題の特性に合わせた方法を取捨選択して組み合わせるといった様々なノウハウが必要である。しかし、現在では、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、アスペクト指向を一貫して現実に適用していく上での大きな妨げとなっている。

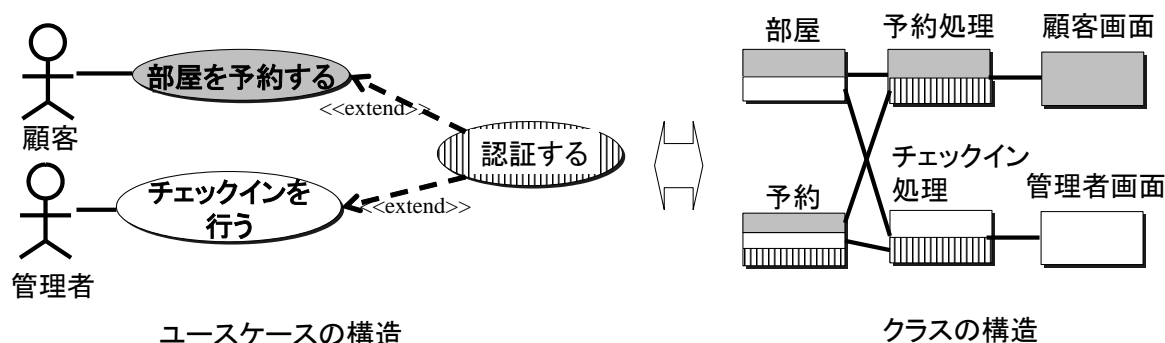


図1 オブジェクト指向によるホテル予約システム分析設計結果における複雑さ

## 6. 本科目で習得する技術

本科目では、エンタープライズシステムを題材として産業ソフトウェアの分析/設計/実装問題を扱い、オブジェクト指向技術の限界と、アスペクト指向技術の必要性を学ばせる。アスペクト指向分析/設計技術、アスペクト指向プログラミング技術、および、それらを一貫して扱うためのアスペクト指向開発方法論について学ばせ、品質の高い多機能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させる。さらに、エンタープライズシステムのソフトウェア分析/設計/実装問題にアスペクト指向技術を用いた分析/設計/実装演習を実施し、アスペクト指向開発の知識と活用能力を体得させる。

本科目で取り上げる手法およびツールを以下にあげる。

- 手法
  - 上流から下流まで一貫したユースケースによるアスペクト指向開発手法（AOSD with Usecases）
- ツール
  - アスペクト指向プログラミング環境：AspectJ
  - アスペクト指向プログラミング環境：SpringAOP
  - アスペクト指向プログラミング環境：JavaScript に基づく AOP 処理フレームワークの独自実装 JavaScriptAOP
  - DI コンテナツール：Spring Framework
  - UML モデリングツール：astah\*

## 7. 前提知識

本科目の受講生は、トップエスイー開講科目「ソフトウェア工学入門」「基礎理論」「コンポーネントベース開発」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。

- 集合論
- 論理学
- UML
- 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



## 8. 講義計画

- ・ 概要（予定、変更の可能性あり）

第1回 全体構成

第2回 オブジェクト指向プログラミングの限界

第3回 アスペクト指向プログラミング

第4回 AspectJ・振る舞いへの作用

第5回 AspectJ・振る舞いへの作用の実践

第6回 AspectJ・構造への作用

第7回 アスペクト指向分析/設計とユースケース

第8回 ユースケースによるアスペクト指向開発：ピアユースケース

第9回 ユースケースによるアスペクト指向開発：拡張ユースケース

第10回 Web 開発におけるオブジェクト指向の限界

第11回 DI コンテナと Spring

第12回 SpringAOP

第13回 JavaScriptAOP フレームワークの実装と利用

第14回 事例

第15回 まとめ

- ・ 詳細（予定、変更の可能性あり）

第1回 全体構成

- はじめに ～本科目の概要～
- アスペクト指向とは？
- OOP の限界
- AOP の概念
- AOP 処理系：拡張言語
- OOAD の限界
- AOSD の概念
- まとめ

第2回 オブジェクト指向プログラミングの限界

- オブジェクト指向
- オブジェクト指向開発プロセス：ICONIX 系
- 課題 P1：AOP/AOSD による開発の理解
- 演習 S1：ホテル予約システム OOAD+Java による実装
- まとめ

### 第3回 アスペクト指向プログラミング

- <解答例>演習 S1:ホテル予約システム OOAD+Java による実装
- アスペクト指向プログラミング
- AOP 処理系:ライブラリ
- AOP 処理系:フレームワーク
- AOP 処理系:拡張言語
- ツール演習:AspectJ/AJDT プラグイン体験
- まとめ

### 第4回 AspectJ・振る舞いへの作用

- AspectJ とは
- 振る舞いへの作用
- まとめ

### 第5回 AspectJ・振る舞いへの作用の実践

- 演習 S2:ホテル予約システム AspectJ の利用
- まとめ

### 第6回 AspectJ・構造への作用

- 構造への作用
- アスペクトの高度な定義
- アノテーションの活用
- 演習: AspectJ の構造への作用と高度な定義
- まとめ

### 第7回 アスペクト指向分析/設計とユースケース

- ユースケースとは
- ユースケースと AOP (AspectJ) との親和性
- ユースケースによるアスペクト指向開発
- まとめ

### 第8回 ユースケースによるアスペクト指向開発: ピアユースケース

- ユースケースによるアスペクト指向開発
- AOP の手法によるユースケース記述
- ユースケーススライス
- ピアユースケースの拡張

- ユースケースモジュール
- ユースケースによるアスペクト指向モデリング
- モデリングの道具立て／ツール
- 演習 S3:ホテル予約システム AOSD+AspectJ による実装

#### 第 9 回 ユースケースによるアスペクト指向開発：拡張ユースケース

- <解答例>演習 S3:ホテル予約システム AOSD+AspectJ による実装
- 演習 S3' :ホテル予約システム AOSD+AspectJ による実装(機能追加)
- まとめ
- 補足資料：ユースケースによるアスペクト指向モデリング～応用編～

#### 第 10 回 Web 開発におけるオブジェクト指向の限界

- <解答例>演習 S3' :ホテル予約システム AOSD+AspectJ による実装(機能追加)
- Web アプリケーション開発
- Java EE 5
- Java Servlet
- Servlet の利用例
- JSP
- JSP の利用例
- 課題 P2:Web アプリ開発への DI、AOP 利用とその理解
- 演習 S4:ホテル予約システム(Web アプリケーション) DAO を Mock オブジェクト化
- まとめ

#### 第 11 回 DI コンテナと Spring

- <解答例>演習 S4:ホテル予約システム(Web アプリケーション) DAO を Mock オブジェクト化
- DI
- DI と AOP の違い
- DI コンテナ
- Spring 概要
- Spring の DI コンテナ
- 演習 S5:ホテル予約システム(Web アプリケーション) Spring DI の利用
- まとめ

#### 第 12 回 AOP フレームワーク(SpringAOP)

- <解答例>演習 S5:ホテル予約システム(Web アプリケーション)Spring DI の利用

- AOP の復習
- AOP フレームワーク
- AOP 処理系
- SpringAOP
- SpringAOP の利用例
- 演習 S6: ホテル予約システム (Web アプリケーション) Spring AOP の利用
- まとめ

#### 第 13 回 AOP フレームワークの実装と利用

- JavaScript の基本
- JavaScript による Before アドバイスの実現
- 演習: JavaScript の基本的な扱い
- 演習: JavaScript による After、Around アドバイスの実現
- 演習: ダブルクリック防止等の AOP フレームワークによる追加実装
- まとめ

#### 第 14 回 事例と効果

- オブジェクト指向の問題
- アスペクト指向の効果
- 企業事例

#### 第 15 回 まとめ

- 全課題を振り返って
- 技術比較
- ポストオブジェクト指向
- 全体のまとめ

## 9. 教育効果

アスペクト指向開発技術を活用した以下のスキルを身につけることができ、複雑および大規模なシステムを効率よく分析/設計/実装/保守できるようになる。

- 上流工程におけるユースケースによるアスペクト指向開発手法（AOSD with Usecases）を代表とするアスペクト指向分析/設計手法の適用能力を習得できる。
- 下流工程における AspectJ を代表とするアスペクト指向プログラミング技術/環境（フレームワークおよびコンテナ）の適用能力を習得できる。
- 下流工程における Spring Framework を代表とする依存性注入（Dependency Injection）の仕組みを併用可能なアスペクト指向プログラミング環境の適用能力を取得できる。

- 異なるアスペクト指向プログラミング環境およびDI環境の比較検討能力を習得できる。

以上の知識およびスキルから、エンタープライズシステムのような複雑かつ大規模なソフトウェアシステムの開発に関して、上流から下流まで一貫した関心事およびその実現結果の分離と追跡性が維持された形でソフトウェアをアスペクト指向分析/設計/実装する能力を身に付けることができる。

さらに、AspectJ, Spring Framework（SpringAOP）の2つのアスペクト指向プログラミング/DI環境を使い分けて、ソフトウェア実装の作業を効率的に進めることができるようになる。こうした環境活用のスキルにより、新たなアスペクト指向プログラミング環境の利用が必要になったときに、より迅速かつ的確な環境活用能力を身に付けることができる。

## 10. 使用手法およびツール

- アスペクト指向プログラミング AspectJ および SpringAOP
- アスペクト指向開発手法 AOSD with Usecases によるアスペクト指向分析/設計モデリング
- DI コンテナ/フレームワーク Spring Framework を活用したオブジェクト指向プログラミング
- JavaScript に基づく AOP 処理フレームワークの独自実装 JavaScriptAOP

## 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。エンタープライズシステムを題材として産業ソフトウェアの分析/設計/実装問題を扱い、オブジェクト指向技術の限界と、アスペクト指向技術の必要性、アスペクト指向による分析/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、アスペクト指向技術の適用前後について比較評価を行い、アスペクト指向技術および周辺技術の有用性と適用性を議論する。議論を通してアスペクト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

- Robert Filman, Tzilla Elrad, Siobhan Clarke and Mehmet Aksit, “Aspect-Oriented Software Development,” Addison-Wesley, 2004.
- 千葉滋, アスペクト指向入門, 技術評論社, 2005.

アスペクト指向開発を構成する基盤技術および AOP について述べられており、この科目に最適である。

- 長瀬嘉秀, 天野まさひろ, 鷺崎弘宜, 立堀道昭 著, "AspectJ によるアスペクト指向プログラミング入門", ソフトバンクパブリッシング, ISBN 4797326387, 2004.4.
- arton, Seasar2 で学ぶ DI と AOP アスペクト指向による Java 開発, 技術評論社, 2006.
- 長谷川裕一, Java・J2EE・オープンソース Spring 入門 より良い Web アプリケーションの設計と実装, 技術評論社, 2005.
- Ivar Jacobson, Pan-Wei Ng 著, “Aspect-Oriented Software Development with Use Cases”, Addison-Wesley, 2005. (邦訳) 鷺崎弘宜, 太田健一郎, 鹿糠秀行, 立堀道昭 訳, "ユースケースによるアスペクト指向ソフトウェア開発", 翔泳社, ISBN 4798108960, 2006.

アスペクト指向プログラミング技術としての AspectJ、DI+AOP 技術としての Spring、および、ユースケースによるアスペクト指向開発手法についてそれぞれ詳細に解説されており、各技術・手法の習得に最適である。