

コンポーネントベース開発

平成27年度シラバス

2015年 1月 9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

コンポーネントベース開発

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、ネットワーク家電と、サービスを提供する外部システムがネットワークによって接続され協調動作するネットワーク家電システムを題材とした産業ソフトウェアの分析/設計問題を扱う。実際にインターネット上に配備可能なネットワーク家電システムのソフトウェア開発を通して、多層アーキテクチャにおいて部分を再利用可能かつ全体が変更しやすいシステムを開発するためのソフトウェア分析/設計の難しさを体感した上で、早期の分割統治とインタフェース中心の分析/設計を系統立てて実施するためのコンポーネントベース開発手法とプロセスを習得する。代表的な 2 (3) つのコンポーネントベース開発方法論 (Catalysis, UML Components, Kobra) それぞれが提供する分析/設計プロセスを、プロダクトライン開発の文脈上で、ネットワーク家電システムのソフトウェア開発に適用した結果について比較評価を行い、分析/設計プロセスの有用性と適用性を議論する。議論を通してコンポーネントベース開発方法論の理解を深め、分析/設計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. 講座のオリジナリティ

近年、高品質なモデリングツール/実装環境が開発され利用されて、コンポーネントベース開発方法論を実用システムに適用した成功事例が報告されている。しかし、コンポーネントベース開発方法論について分析/設計/実装方針の違いから様々な方法論が存在しているのが実情である。実際のソフトウェア開発にこれらの方法論を適用していくためには、それぞれの開発方法論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論を取捨選択していくといった様々なノウハウが必要である。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、コンポーネントベース開発方法論を適用していく上での大きな妨げとなっている。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コンポーネントベース開発方法論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
コンポーネントベース開発方法論の定義が曖昧であり、コンポーネントベース開発方法論の本質を把握することが難しい	明確な最大公約数定義、種々のコンポーネントベース開発方法論に共通する特徴／プロセス、および、従来のオブジェクト指向開発方法論との違いを顕在化させて扱うため、コンポーネントベース開発方法論の本質を把握可能
様々なコンポーネントベース開発方法論が未整理、未比較であり、それぞれの特性や適用範囲を把握することが難しい	代表的な複数のコンポーネントベース開発方法論の特徴と適用範囲を顕在化させて演習・討論を実施するため、それらを把握可能
既存のコンポーネントベース開発方法論を構成する作業の具体的手順（特にコンポーネント、インタフェース識別の手順）が曖昧であり、適用にあたり経験に頼る部分が多い	顕在化された具体的手順（特にコンポーネント、インタフェース識別の手順）および適用ノウハウに沿った演習を行うため、既存のコンポーネントベース開発方法論を適用する具体的・実用的ノウハウを習得可能
扱われる幾つかのコンポーネントベース開発方法論は重厚長大であり、現実の開発への適用時のカスタマイズのポイントを把握することが難しい	既存のコンポーネントベース開発方法論の作業や成果物について必須な部分と必須でない部分、簡素化可能な部分を明確にした上で演習を行うため、現実の開発への適用時のカスタマイズ方法を習得可能

5. 本講座で扱う難しさ

近年、複数の高機能な情報家電がネットワークに接続され、相互に連携して動作することで 1 つの機能を実現する、ネットワーク家電市場が急速に立ち上がりつつある。従来の家電システムは単体で動作するように設計されていたのに対して、ネットワーク家電システムでは、システムへの接続を希望する多様な情報家電/外部サービスシステムへの対応や、技術発展に伴う家電/ネットワークの規格変化への対応が必要になるため、ソフトウェアは非常に複雑となる。このような環境に対して再利用性/変更容易性の高いソフトウェア分析/設計を実施するためには、ゼロベースからの開発や実装工程における遅れての分割統治に基づく従来型の開発方法論は膨大な工数を必要とし、もはや現実的手段ではない。

ネットワーク家電を外部ネットワーク上のサービスと連携動作させるソフトウェアシステムの複雑さの例を図 1 に示す。図 1 は、従来のオブジェクト指向開発方法論によって、ホームネットワーク上のネットワーク家電から会員認証を経て、様々な情報（例えばスーパーマーケットにおける食品情報、食品を組み合わせで作成可能な料理レシピ情報）を外部ネットワーク上のサービス事業者から取得するシステムの分析/設計/実装を行う場合に、以下に述べる様々な問題が発生することを示す例である。

- 追跡が困難：従来の開発方法論では、システム全体を最初に一枚岩として捉えて分析/設計を行い、後期の実装工程において分割統治を試みる。その結果、元々の要求や各工程における成果物間の対応関係が不明瞭となり、以下に述べる開発上/保守上の問題を引き起こす。
- 最適なアーキテクチャの選択が困難：設計の初期段階において全体で統一されたアーキテクチャを採用しなければならない。従って、部分ごとに最適なアーキテクチャを採用できない。
- 再利用が困難：システム内外における機能の再利用が困難となる。図 1 の例では、会員サービスにおけるユーザの認証機能と、スーパーマーケット/レシピ配信サービスにおける会員サービスの認証機能について機能的重複があるが、個別に設計/実装が行われている。
- 変更が困難：同様に、システム中の構成要素間の関係が密結合かつ複雑となり、ネットワーク規格や特定個所の仕様変更の影響が他の様々な個所に波及してしまう。

上述の開発上の問題（内的問題）は、ユーザに見える形で、システムの誤動作、システムのアップグレードの困難さ、システムに接続できない端末の存在、システムの高価/リリースの遅れといった様々な問題（外的問題）を引き起こす。

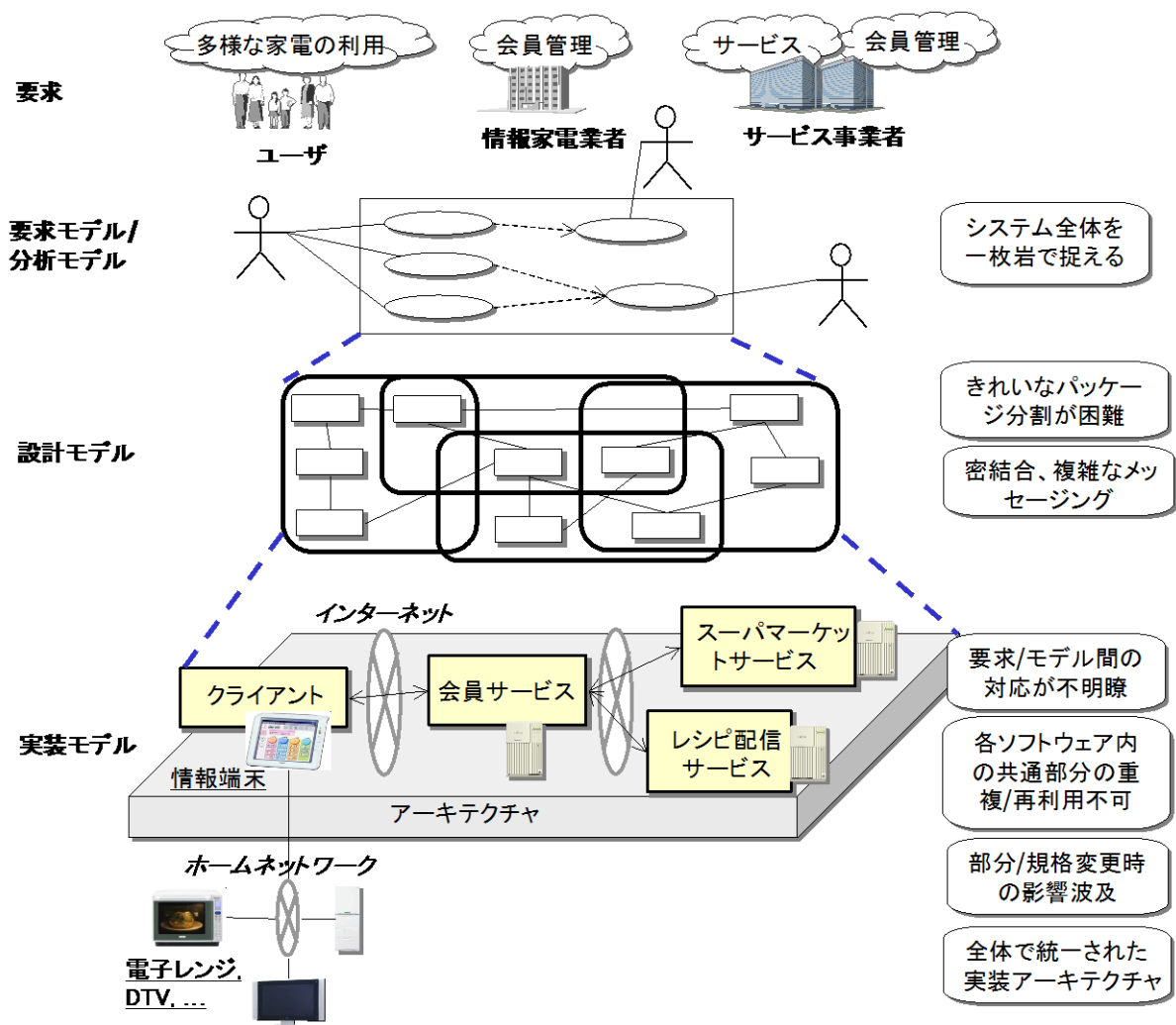


図 1: ネットワーク家電と外部サービスを組み合わせる場合の複雑さの例

6. 本講座で習得する技術

ネットワーク家電システムでは、多様な情報家電/外部サービスシステムへの対応や、技術発展に伴う家電/ネットワークの規格変化への対応が必要なため、ソフトウェアは非常に複雑となり、従来の開発方法論では再利用性/変更容易性の高いソフトウェアの開発が困難である。

コンポーネントベース開発方法論は、早期の分割統治に基づくインタフェース中心の分析/設計およびプロダクトライン展開を実現する再利用を行う開発プロセス、コンポーネントの組み合わせモデルを的確に表現する表記法、コンポーネント単位での実装を実現する標準化された様々なオブジェクト指向アーキテクチャ技術を体系的にまとめた方法論の一種である。上述のネットワーク家電システムのような複雑なソフトウェアを実現するためには、ソフトウェア開発の上流工程において、コンポーネントベース開発方法論を適用すること、および、実装工程において、コンポーネントベース開発を支援する適切なアーキテクチャ/開発支援環境を適用していくことが非常に重要となる。

例えば、図 1 と同様のネットワーク家電システムを、コンポーネントベース開発方法論によって開発する様子を図 2 に示す。図 2 では、従来の方法論が抱える問題を以下のように克服することに成功している。

- 追跡が容易： コンポーネントベース開発方法論では、システム全体を要求分析の工程から徹底的に分割統治を行い、続いて部分単位での設計/実装を行う。その結果、元々の要求や各工程における成果物間の対応関係が明瞭となり、以下に述べる開発上/保守上の効果が得られる。
- 最適なアーキテクチャを選択可能： 設計/実装工程において、部分ごとに該当部分について最適なアーキテクチャを採用できる。
- 再利用が可能： システム内外における機能の再利用が可能となる。図 2 の例では、会員サービスやスーパーマーケット/レシピ配信サービスに共通な会員認証機能をコンポーネントとして分析/設計/実装し再利用すれば、システム全体を効率よく迅速に開発することができる。
- 変更が容易： 同様に、システム中の構成要素（コンポーネント）間の関係が疎結合かつシンプルとなり、ネットワーク規格や特定個所の仕様変更の影響を該当個所に局所化することができる。また、各コンポーネント単位による徹底的なテスト/検証の実施により高い信頼性を保証すれば、コンポーネントを組み合わせ得られるシステム全体の信頼性を飛躍的に向上させることができる。

本講座では、開発の上流工程における分析/設計の重要性を把握し、コンポーネントベース開発方法論を構成する開発手法、開発環境、開発資産、実行環境、実行基盤（図 3）のうちで、特に、開発手法および開発環境について重点的に学ぶ。実際にインターネット上に配備可能なネットワーク家電システムのソフトウェア開発を通して、採用したアーキテクチャ上で部分を再利用可能かつ全体が変更強いシステムを開発するためのソフトウェア

分析/設計の難しさを体感した上で、早期の分割統治、インタフェース中心の分析/設計、およびプロダクトライン開発を系統立てて実施するためのコンポーネントベース開発プロセスを体得する。代表的な 2 (3) つのコンポーネントベース開発方法論 (Catalysis, UML Components, Kobra) それぞれが提供する分析/設計プロセスを、ネットワーク家電システムのソフトウェア開発に適用した結果について比較評価を行い、分析/設計/再利用プロセスの有用性と適用性を議論する。議論を通してコンポーネントベース開発方法論の理解を深め、分析/設計プロセスの実用的ノウハウを体得する。

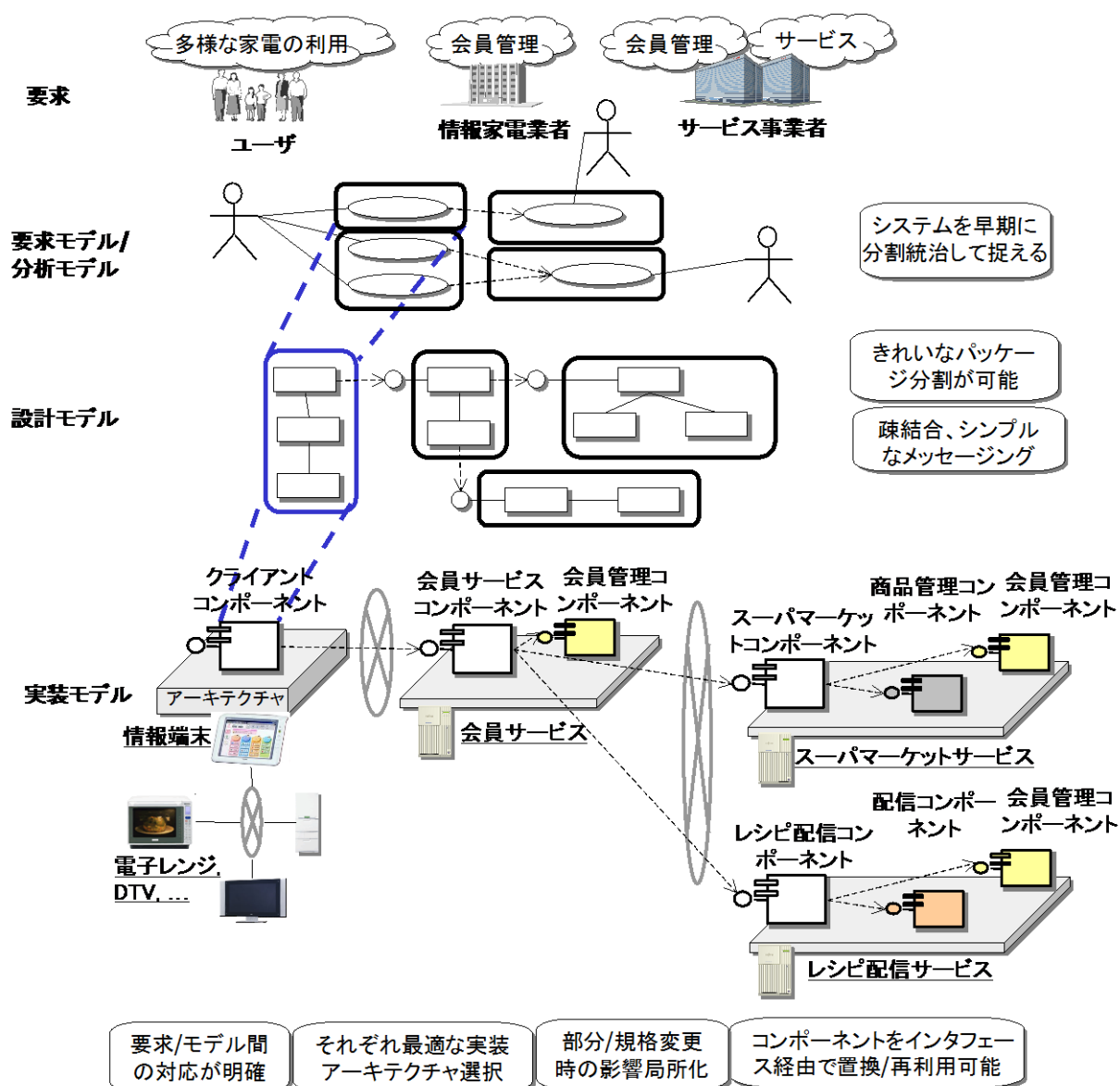


図 2: コンポーネントベース開発によって実現されたシステムの例

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
- UML

8. 講義計画

- ・ 概要（予定、変更の可能性あり）

第1回 コンポーネントベース開発とは

第2回 オブジェクト指向分析

第3回 オブジェクト指向設計と問題

第4回 オブジェクト指向からコンポーネントベースへ

第5回 プロダクトラインとフィーチャモデリング

第6回 インタフェース定義と OCL

第7回 Catalysis の枠組み

第8回 UML Components の枠組み

第9回 UML Components による分析設計

第10回 UML Components による開発とプロダクトライン

第11回 Kobra の枠組み

第12回 Kobra による分析設計

第13回 Kobra によるプロダクトライン

第14回 コンポーネントベース開発事例と効果

第15回 手法比較と全体のまとめ

- ・ 詳細（予定、変更の可能性あり）

第1回 コンポーネントベース開発とは

情報システム

情報システムの開発問題

コンポーネントベース開発による解決

コンポーネントベース開発の位置づけ

第2回 オブジェクト指向分析

プログラミングからモデリングへ

オブジェクト指向開発方法論

ICONIX

ドメイン分析

演習：ホテル予約システムのドメイン分析

第3回 オブジェクト指向設計

ロバストネス分析（システム分析）

演習：ホテル予約システムのシステム分析

設計

演習：ホテル予約システムの設計

第4回 オブジェクト指向からコンポーネントベースへ

ICONIX 演習の振り返り

ICONIX の評価

オブジェクト指向からコンポーネントベースへ

コンポーネント実装技術

コンポーネントベース開発の効果

第5回 プロダクトラインとフィーチャモデリング

再利用

プロダクトラインエンジニアリング

フィーチャモデリング

演習：ホテル予約システムのフィーチャモデリング

第6回 インタフェース定義と OCL

OCL の基礎

OCL の応用

演習：事前/事後条件の OCL 記述

第7回 Catalysis の枠組み

Catalysis の概要

Catalysis によるコンポーネントベース開発の枠組み

Catalysis の開発プロセス

第8回 UML Components の枠組み

UML Components の概要

UML Components が導く実装アーキテクチャ

UML Components の開発プロセス

第9回 UML Components による分析設計

要求

仕様：コンポーネントの識別、相互作用、仕様

演習：UML Components によるコンポーネントベース開発

第10回 UML Components による開発とプロダクトライン

提供と組み立て

演習：UML Components におけるプロダクトライン開発

第 11 回 Kobra の枠組み

Kobra の概要

Kobra の枠組み

Kobra の開発プロセス

第 12 回 Kobra による分析設計

要求

コンテキスト実現

コンポーネント仕様

コンポーネント実現

演習：Kobra によるコンポーネントベース開発

第 13 回 Kobra によるプロダクトライン

Kobra による分析失敗例

演習：Kobra によるプロダクトライン開発

第 14 回 コンポーネントベース開発事例と効果

コンポーネントベース開発事例

コンポーネントベース開発の効果の再考

第 15 回 手法比較と全体のまとめ

コンポーネントベース開発方法論の比較

コンポーネントベース開発方法論と周辺技術

9. 教育効果

本講座を受講することにより、実証科学（科学＋工学）としてのコンポーネント開発方法論におけるモデル化/ものづくり/評価の3能力を習得し、実際のソフトウェアの開発に適用できるようになる。具体的には、モデル化能力として、背景にあるオブジェクト指向設計の原則、インタフェース中心分析/設計による抽象化/モジュール化技術を習得できる。ものづくり能力として、実問題を扱った演習による実践的分析/設計能力、具体化された手順に沿った既存コンポーネントベース開発方法論の適用能力を習得できる。評価能力として、導出モデル/システムの評価能力、異なるコンポーネントベース開発方法論の比較検討/改善能力を習得できる。

10. 使用する開発方法論

ICONIX: 標準的な軽量オブジェクト指向開発プロセス

- 選択理由: オブジェクト指向開発プロセスとして軽量かつ標準的
- 必要ノウハウ: 適用時のコンポーネント化の検討指針

Catalysis: 追跡可能性とパターン再利用を重視したコンポーネントベース開発方法論

- 選択理由: 他のコンポーネントベース開発方法論の基礎を成し、成功事例がある
- 必要ノウハウ: 一部表記の UML への準拠

UML Components: データ分割とインタフェース分離設計を重視したコンポーネントベース開発方法論

- 選択理由: 従来のオブジェクト指向プロセスからの移行容易
- 必要ノウハウ: コンポーネント／インタフェース識別手順の具体化

KobrA: トップダウンな機能分割/再利用を重視し、プロダクトラインを加味したコンポーネントベース開発方法論

- 選択理由: (可変性分析以外は) 具体的、プロダクトライン開発の実現
- 必要ノウハウ: 機能分割および可変性分析手順の具体化、コンポーネントの粒度の均整化

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。ネットワーク家電システムを構成するサービス提供ソフトウェアを題材にして、コンポーネントベース開発方法論に基づくソフトウェアの分析/設計を行う。インタフェース中心分析/設計の難しさを体感しながら分析/設計プロセスを体得する。グループ内で、2（3）つのコンポーネントベース開発方法論を適用した結果について比較評価を行い、コンポーネントベース開発プロセスの有用性と適用性を議論する。議論を通してコンポーネントベース開発方法論の理解を深めながら、分析/設計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

- C. Szyperski, “Component Software: Beyond Object-Oriented Programming,” 2nd Edition, Addison-Wesley Professional, 2002.
コンポーネントベース開発方法論を構成する基盤技術としての開発環境、開発資産、実行環境および実行基盤について述べられており、この講義に最適である。
- D.F. D’Souza and A.C. Wills, “Objects, Components, and Frameworks with UML: The Catalysis Approach,” Addison-Wesley Professional, 1998.
- J. Cheesman and J. Daniels, “A Simple Process for Specifying Component-Based Software,” Addison-Wesley Professional, 2000.
- C. Atkinson, et al., “Component-Based Product Line Engineering with UML,” Addison-Wesley Professional, 2001.
それぞれ、コンポーネントベース開発方法論 Catalysis、UML Components、KobrA の入門から実践までの一通りが述べられており、各方法論の講義と演習に最適である。