

## UML を用いた制御システムの開発事例

3Q-4

佐藤 洋介 西山 裕二 松永 武 安井 浩之 松山 実

武蔵工業大学

### 1. はじめに

組み込みシステム開発では、互いに開発サイクルが異なるハードウェア、ソフトウェア両方を開発する。通常、ハードウェア開発は機構系設計、回路設計の要求仕様が抽出されるまでに時間がかかる。その影響から、ソフトウェア開発は常にハードウェア開発の後手に回り、ハードウェアの仕様変更に対応できない。したがって、開発工程上流で信頼度の高い規模見積もりや、変更影響度の把握が難しくなっている。また、開発規模の増大や複雑化により、生産性の改善も難しくなっている。

開発現場では構造化開発手法以前の開発プロセスが使用されている。その開発プロセスは、ハードウェア・ソフトウェア両方を含めたシステムレベルでの検討ができないため要求抽出に工数がかかる、分析工程から設計工程への接続性が悪いといった問題を抱えているため、上記の問題を解決するには至っていない。[1]

本開発事例では、現行の組み込みシステム開発の問題点を解決するために、オブジェクト指向開発手法及び UML(Unified Modeling Language)を採用した。オブジェクト指向開発手法は分析設計工程の接続性、再利用を容易にするという点から、UML はビジュアルモデリング言語としての表現力の高さ、全開発工程を通じてのドキュメント記述能力という点から採用した。これらを問題解決技術として利用した開発プロセスを提案し、それを適用した開発事例について紹介する。[2]

### 2. システム概要

本開発事例で紹介するシステムは、PalmOS を搭載した PDA(WorkPad50J)を用いて模型戦車のモータ制御を行う。実装言語には携帯電話や PDA、その他様々な組み込み機器において使用されて

いる J2ME(Java 2 Micro Edition)を使用した。[3]

### 3. 開発プロセスの提案

本プロセスでは、フィンランドの Nokia Research Center で 1993 年より開発されている組み込みオブジェクト指向開発プロセス OCTOPUS[4]をリファレンスモデルとして採用した。(図 1)

開発サイクルの違いには、OCTOPUS の特徴である、サブシステム分割、インクリメンタルアプローチを利用して対処した。ソフトウェアアーキテクチャである Coad のコンポーネント[5]を基にサブシステムへの分割を行うことにより、ハードウェアと直接関係の無いサブシステム開発とハードウェア開発を並行して行うことが可能となる。これにより、ハードウェア設計が終了しないとソフトウェア開発が開始できないという問題の解消を試みた。

開発規模の増大には UML(Unified Modeling Language)を使用したオブジェクト指向によるモデル化により対処した。開発に関わるドキュメントを UML により統一して管理することにより、曖昧性の排除、意思疎通の効率化が期待できる。

#### 3.1 本プロセスの特徴

本プロセスでは Coad のコンポーネントを基にサブシステムへの分割を行う。Coad のコンポーネントを利用した開発手順では、分析工程で問題領域の概念部分を PD(Problem Domain)として定義する。また、設計工程で PD を実行するための設計情報となる HI(Human Interface)や DM(Data Management),SI(System Interaction)を定義する。

本プロセスではアーキテクチャ設計以降の工程で、抽象レベルが一定である各サブシステム群 (HI/PD/SI) を独立して開発する必要がある。したがって、最小の結合度で各サブシステムが結合されることが要求される。そこで、Coad のコンポーネントを Layers パターンによって結合する

A Development case of control system with UML.

Yosuke SATO, Yuji NISHIYAMA, Takeshi MATSUNAGA,  
Hiroyuki YASUI, Minoru MATSUYAMA  
Musashi Institute of Technology

方針を採っている。(図2)

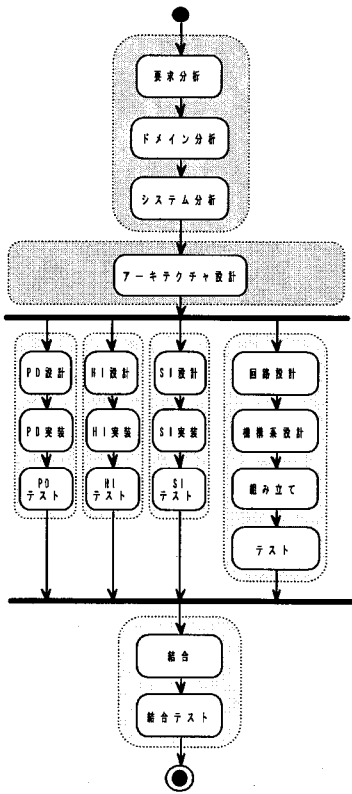


図1 開発プロセス

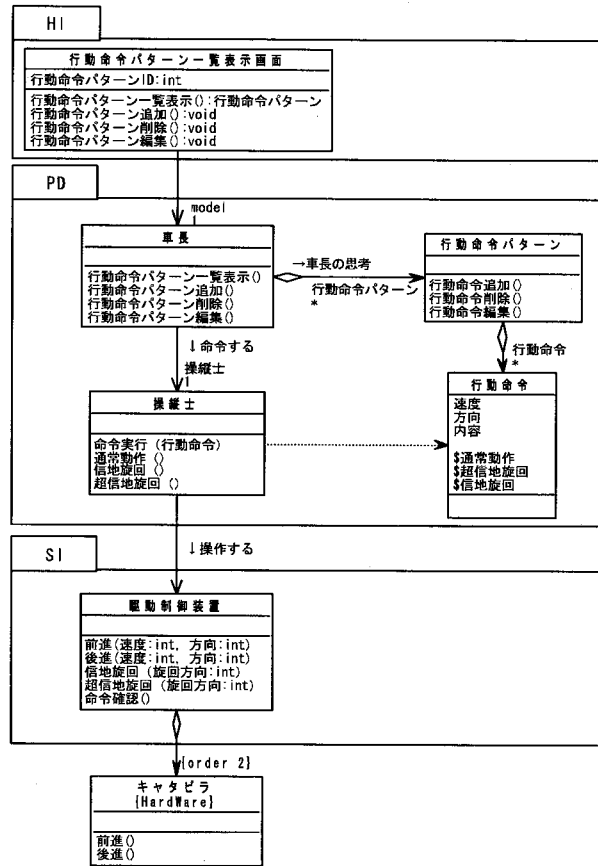


図2 アーキテクチャ設計クラス図

#### 4. まとめ

オブジェクト指向を開発プロセスに適用することにより、大部分の組み込みシステム開発では軽視されるドメイン分析を十分に行うことができた。これにより、ソフトウェア、ハードウェア両方を含めたシステムレベルでの検討が可能となった。

また、ソフトウェア、ハードウェア両方を含めた統合プロセスによって、サブシステムを非同期に同時開発することが可能となった。開発現場で頻繁に見られる、「一番遅いサブシステムの進捗に左右され、他のサブシステムの完成が遅れる」という事態は起こらなかった。無駄の無い開発によって計画の遅延を大きく低減することに成功したと言える。

#### 参考文献

- [1] 杉浦 英樹：オブジェクト指向の目指すもの、Interface 3月号，CQ 出版(2001)
- [2] 佐藤 洋介：開発プロセス，XML Press Vol. 4，技術評論社(2001)
- [3] 西山 裕二，他：J2ME を用いた制御システムの開発事例，第 63 回情報処理学会全国大会，2001
- [4] Maher Awad, Juha Kuusela, Jurgen Ziegler, "Object-Oriented Technology for Real-Time Systems, A Practical Approach using OMT and Fusion", PRENTICE HALL, 1996
- [5] 佐藤 啓太，他：ソフトウェアパターン，共立出版(1999)