

ビジネス分野向けオブジェクト指向開発手法

4R-6

白谷 勇人*¹ 葉木 洋一*² 大成 宣行*³ 齋藤 浩*³

*¹日立ソフトウェアエンジニアリング(株) *²(株)日立製作所ビジネスシステム開発センタ

*³(株)日立製作所情報システム事業部

1. はじめに

ソフトウェア開発において、オブジェクト指向開発（OOD）手法は高生産性と高信頼性を提供する開発手法と期待されている。しかし、現状の開発ではOOD手法の利点を得られるまでには到っていない。これは、要求仕様の定義が不十分なために、適切なオブジェクトモデル、動的モデル作成がスムーズに行えない。プログラム作成においてスキルが要求される、などの問題のためである。

筆者らは、ビジネス分野のシステム開発において高生産性と高信頼性を得ることのできる実用的OOD手法の研究を行っている。本稿ではオブジェクト指向分析に親和性の高い要求仕様定義プロセスと適切なソースコードを生成するための設計ガイドラインを中心に述べる。

2. オブジェクト指向開発の課題

OODにおいてその利点を得るためには、以下の解決すべき課題がある。なお、1で述べたように2.1、2.2に的を絞って対応策を提案する。

2.1 要求仕様の定義プロセスの詳細化

システムがもたらすメリットと業務を、どのように改善するのかを明らかにした要求仕様の定義プロセスを詳細化する。特に、この後工程で作成するオブジェクトモデル、動的モデルとの関係を明らかとすることにより、定義すべき項目を明確化させる。

2.2 設計ガイドラインの設定

設計工程で使用するガイドラインの設定。特に実装を設計情報からソースコード変換の機械的な作業工程とするために、オブジェクトモデルで記述すべき項目を規定する。

2.3 品質管理法の設定

OODの特性を利用した効率的な品質管理方法の定義。

2.4 プロセス管理法の設定

OODの特性とスパイラル開発を考慮したプロセス管理法の設定。

2.5 見積り法の設定

要求されるシステムの規模の見積り法、開発期間を考慮した価格の見積り法の設定。

3. 要求仕様の定義プロセス

3.1 手順概要

要求仕様の定義の手順概要を図1に示す。本OOD手法では、特に「既存業務の定義」「新業務の定義」の成果物をオブジェクトモデル、動的モデル構築につなげる考慮を行う。

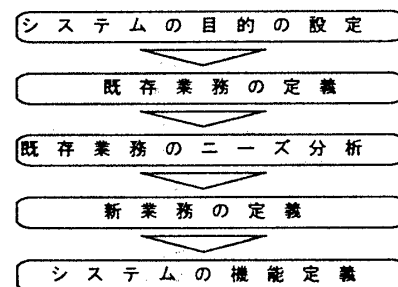


図1 要求仕様の定義の手順概要

Object-Oriented Development Method for Business Application

Hayato SHIRATANI*¹, Youichi HAGI*², Noriyuki ONARI*³, Hiroshi SAITOH*³

*¹Hitachi Software Engineering Co., Ltd. *²Hitachi, Ltd. Institute of Advanced Business System,

*³Hitachi, Ltd. Information Systems Division

3.2 業務の定義・新業務の定義

業務の定義では、業務をどのように行うのかを定義し、システム化する範囲を決定する。

(1) システム化する業務の詳細レベルの定義

システム化する業務の詳細レベルを表1に示す三つのレベルに分けて行う。この工程での詳細度の最終粒度を定義することにより、不必要な詳細化を防止する。

表1 業務の詳細レベル

レベル	定義
Task	1つの業務システムの単位となるようなレベル。複数の業務から構成される。
業務	まとまった役割を持った業務イベントのかたまり。この単位で業務シナリオを定義する。
業務イベント	実務運用上可能な業務レベル。入出力画面(帳票)に対応する業務レベルである。

(2) 業務シナリオの作成

業務レベル単位で業務シナリオを作成する。その例を表2に示す。このシナリオは、イベントトレース図の基になるだけでなく、オブジェクトモデル構築のための実体と入出力情報を提供する。下記表中の「入力データ」「戻り値」には、該当する業務イベントに伴う入力情報、出力情報を記入し、「発生元」には業務イベントのリクエスト元を、「消費先」にはリクエスト先の実体を記入する。

表2 業務シナリオの例

システム名「販売管理システム」					
業務シナリオ名「受注業務(在庫品)」					
業務イベント	説明	入力データ	戻り値	発生元	消費先
納期間い合せ	なし	得意先	営業担当
.....
.....
.....

4. 設計ガイドライン

[OMT91]¹によるオブジェクトモデルの記法ではC++ソースコードをこのモデル情報より生成するには不十分である。また、[BOOCH91]²においては、C++ソースコードを生成するに十分な記法が定義されているが、分析レベルで表現したい概念との関係づけが弱い。

そのため、[OMT91]をベースとして適切なC++ソースコードを生成できるように、関連に関する新たな概念を6つ定義し、集約概念に対しては再定義を行った。

また、この新しい概念を用いた場合の典型的なパターンに対するC++のコードの対応表を作成し、より実用性を高くすることを試みている。

5. まとめ

OOD手法での課題のうち、要求仕様定義プロセスと設計ガイドラインについての提案を行った。現在、この要求仕様定義を支援するツールを作成しその有効性の評価を行っている。また、設計ガイドラインを実プロジェクトへ適用し、評価を行っている。

今後、本要求仕様の定義プロセスの結果である個々の業務仕様を蓄積し、この中から基本的な要求仕様のパターンを抽出すること、設計ガイドラインを多数のオブジェクトの相互作用を含む場合にも適用できるような拡張を検討していきたい。また、課題の残りの項目についても並行して検討し、現在評価中である。

¹ [OMT91] James Rumbaugh, Michael Blaha, William Premerani, Frederick Eddy, William Lorenson. OBJECT-ORIENTED MODELING AND DESIGN. Englewood Cliffs, New Jersey: Prentice-Hall, 1991 邦訳 羽生田 栄一 監訳「オブジェクト指向方法論 OMT モデルと設計」, 株式会社トッパン、1992

² [BOOCH91] Grady Booch. OBJECT-ORIENTED DESIGN WITH APPLICATION. The Benjamin/Cumming Publishing Company, 1991.