

解説



ソフトウェアプロセス

10. プロセス指向のソフトウェア設計教育†

望月 純 夫††

1. まえがき

ソフトウェア開発部門にとって、部門の標準的な技術を確立し、それを若年技術者に伝承することは最も重要な課題である。ソフトウェア開発技術、その中でも特に重要な根幹技術であるソフトウェア設計技術は属人的性格が強いため、これまでの設計技術教育は主としてOJT (On the job training)に頼らざるを得なかった。OJTは実業務と教育を同時に行うものであり理想的とも考えられるが、教育という面から見ると様々な問題を含んでいる。たとえば、いつでも教材として適切な業務や講師が準備できるとは限らず、またどうしても実務を優先させる結果となる。さらに実際の設計業務は教材としては冗長性が高く適切といえない場合が多い。ここでOJTに頼らず、実業務と独立した設計教育環境が望まれる。そのためには、まずソフトウェア設計技術を明確に記述することが基本となる。我々はこれまでにプロセス・モデル HFSP¹⁾で提唱されているプロセスの定義やプロセスの構造に基づいて、現場における設計技術の記述^{2),3)}に取り組んできた。この研究の成果を応用して若手技術者を対象とした設計教育環境を開発し、社内教育を実施してほぼ満足な結果を得た。

2. 設計技術の記述

これまでの研究^{3),5)}で、設計プロセスの記述にはオブジェクト中心型モデルとフェイズ中心型モデルがあることを発見した。まず熟練技術者の実際の設計プロセスとオブジェクト(設計情報、設計データ、仕様、ドキュメント等の総称)をプロ

セスの実行順序に忠実に記述した。これをフェイズ中心型プロセスと呼ぶ。このプロセスを他のシステムの設計に再利用して評価した。その結果このプロセスは実際の設計手順を忠実に表現しているが、プロセスの繰返しが多いため他の技術者に分かり難い表現となっていることが判明した。そこで再度プロセスの記述方式を検討し直し、オブジェクト中心型プロセスを得た。このオブジェクト中心型プロセスは同じ分野のシステムの設計を一つの共通な分かりやすい形式で表現することができる^{3),5)}。

2.1 オブジェクト中心型プロセス

オブジェクト中心型プロセスは、オブジェクト間の静的な依存関係を重視したものである。一つのシステムの設計で作成されるオブジェクトを収集して、これらをその内容によってグループに分類する。各グループを作成するプロセスを第1レベルのプロセスとする。次に各プロセスを詳細化することにより、第2レベルのプロセスを得る。この全体がオブジェクト中心型プロセスである。このプロセスは、オブジェクト間の静的な依存関係を簡潔に分かりやすく表現しており、同じ分野のシステムの設計を一つの共通なプロセスで表現することができる³⁾。

2.2 フェイズ中心型プロセス

オブジェクトの依存関係を重視したオブジェクト中心型プロセスに対して、フェイズ中心型プロセスは実際の設計作業をその実行順序に忠実に記述したものである。一つのシステム分野に属するシステムの設計プロセスは、共通な一つのオブジェクト中心型プロセスで表現できるが、実際の設計ではこれを異なった形のフェイズ中心型プロセスに変換して利用している。このフェイズ中心型プロセスは、設計対象システムの規模によって異なっている。小規模なシステムを設計する場合

† An Experience of Process-Directed Software Design Education by Sumio MOCHIZUKI (Shukutoku Junior College).

†† 淑徳短期大学 (元三菱スペース・ソフトウェア(株))

は、オブジェクト中心型プロセスをほぼそのまま利用することができる。しかし、実用システムのように規模が大きく複雑なシステムを設計する場合には、全設計工程をいくつかのフェイズに分けて各フェイズの中でオブジェクト中心型プロセスを逐次的に繰り返しつつ設計を進める。このように設計作業の実行順序を重視してプロセスを記述したものがフェイズ中心型プロセスである^{3),5)}。

2.3 設計経験の蓄積とシステム設計への適用

熟練技術者は、自己の豊富な経験から一つの分野のシステムに共通なオブジェクト中心型プロセスを抽出し蓄積している。実際にシステムを設計する場合、これを次のような手順で対象システムに適したフェイズ中心型プロセスに変換して適用している。

(1) 各フェイズの設計作業で作成するオブジェクトの遷移を決定する。

(2) オブジェクトの遷移を実現するようにオブジェクト中心型プロセスの一部を各フェイズに割り当てる。

(3) プロセスの繰返しおよび先読みを定める。

大規模システムの設計では、作成されるオブジェクトの量が多く、これを系統的に作成するために設計作業をフェイズに区切り、一つのフェイズの中でプロセスを繰り返すことによってオブジェクトを詳細化する。さらに、後のフェイズで実行すべきプロセスを前もって実行してあらかじめオ

ブジェクト間の調整を行う、いわゆるプロセスの先読みを行う。この先読みによって大きな手戻りを避けている^{3),5)}。

3. 設計教育

本稿の設計教育の概念を図-1に示す。一つの分野のシステムの設計に共通な普遍的モデルともいべきオブジェクト中心型プロセスがある。これを調節することにより異なる規模のシステムを設計するためのフェイズ中心型プロセスが得られる。このモデルの本格的な変換は、短時間の実習では困難であり、本講座では、課題システムの設計を通してモデルの簡単な変換を体験する。課題システムは、実用システム設計の本質的な部分を残すように次のような仕様とした。

〔課題システムの仕様〕

4つの計測点および制御点を持つ計測制御システムであり、各点の計測値があらかじめ定められた時間的变化を示すように制御点のON/OFF制御を行う実時間処理システムである。

本講座は、オブジェクト中心型プロセスを十分理解すると共に、前述の課題システムの設計により比較的簡単なモデルの変換を体験することを中心としている。

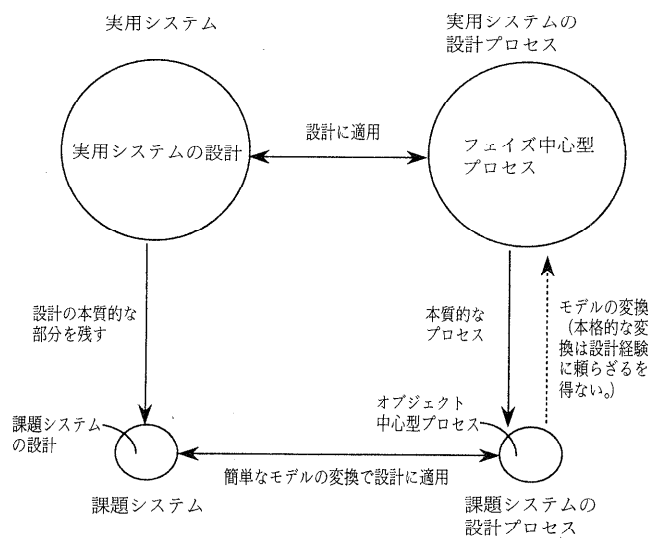


図-1 実用システムと課題システムの設計および各設計プロセスの関係

4. システム設計教育環境

オブジェクト中心型プロセスの習得と課題システムの設計を中心とした設計教育環境を開発し、

社内教育に適用した⁴⁾。その具体的な内容は次のとおりである。

(1) 設計教育環境の構成

設計教育環境の構成を図-2に示す。設計教育

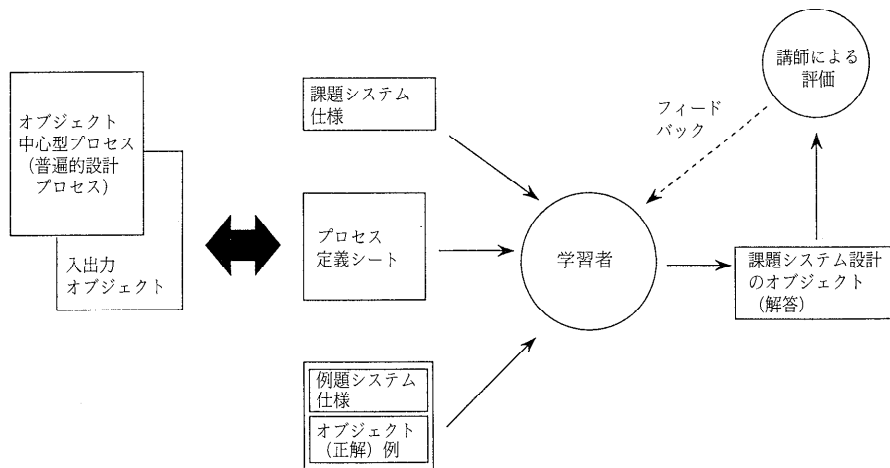


図-2 設計教育環境構成図

プロセス名	機能検討プロセス		前プロセス名	作成者名	竹村
プロセス概要	本プロセスは前プロセス (ユーザ要求の分析と整理, 運用検討) のオブジェクトを利用して必要なソフトウェアの機能の内容と構成を明らかにする。		<ul style="list-style-type: none"> ユーザ要求の分析と整理 運用検討 	作成日付	1991.9.3
			後プロセス名	改訂	1991.9.5 改訂A 1991.9.25改訂B
			<ul style="list-style-type: none"> ソフトウェア仕様のフィージビリティ・スタディ 		
入力オブジェクト	プロセスの機能		出力オブジェクト	チェック項目	
No. b 機能要求	No. 1	運用シーケンスに従い, 機能のブロック化を行い, 機能ブロック間の関連付けを行う。	No. j 機能ブロック図	No. 1	各機能間に矛盾はないか。
f 運用シーケンス			k 機能間インターフェイス	2	ユーザの要求事項をすべて満たしているか。
g モード遷移図	2	各機能の内容を定義する。	l 機能説明表	3	エラーの処理を考慮しているか。
h 操作画面	3	システムへの入力データ, マン・マシン・インターフェイスによる運用者入力データが経路する機能および, その処理内容を定め, 図示する。	m データ処理方法		
I マン・マシン・インターフェイス仕様					
関連情報, 技術, 参考文献等	関連プロセス		関連オブジェクト (あらかじめ統一をとる必要がある他のオブジェクト等)		
<ul style="list-style-type: none"> 「システム設計の構造化手法」 「リアルタイム・システムの構造化分析」 機能定義技術 	<ul style="list-style-type: none"> ユーザ要求の分析と整理 運用検討 システム仕様のフィージビリティ・スタディ 		J 機能ブロック図	k 機能間インターフェイス	l 機能説明表
注意事項 (設計上のコツ, 注意すべき点等)				ユーザ確認項目	
<ol style="list-style-type: none"> 規模が大きいシステムの場合は要求される機能をブロック (グルーピング) 化して考えること。 機能分類は時間をキーとして実施することが重要。常に, 他処理の割り込みを意識すること。 システム全体の中での各機能のバランスに注意すること。 ユーザの要求はシステムの目的に対して本当に必要なものか, 今後の設計に対する影響も含めて考慮すること。問題になりそうな機能はあらかじめユーザに確認をとりつつ先へ進むこと。 検討結果はユーザに確認をとり, かつ, 以後の作業が円滑に進むようなまとめ方をすること。 エラー時, および障害時の機能が考慮されていること。ただし, システムの目的に対して妥当な処理になっているか確認すること。 				j	機能ブロック図
				l	機能説明表

図-3 プロセス定義シート

表-1 設計教育講座の構成と各項目の時間数

設計教育講座の構成		時間数	備考		
ステップ1	講義 ガイダンス	3.0 (Hrs.)	第1日目		
ステップ2	講義 ソフトウェア・プロセス全体の解説	5.0 (Hrs.)			
ステップ3	演習	設計課題の説明	3.0 (Hrs.)	第2日目	
		ユーザ要求の分析と整理	2.0 (Hrs.)		
		運用検討 (その1)	3.0 (Hrs.)	第3日目	
		運用検討 (その2)	4.0 (Hrs.)		
		機能検討 (その1)	4.0 (Hrs.)		
		機能検討 (その2)	2.0 (Hrs.)		
			システム仕様の フィージビリティ・スタディ	1.0 (Hrs.)	第4日目
			データおよび処理検討	5.0 (Hrs.)	
			制御およびタイミング検討	7.5 (Hrs.)	第5日目
			ソフトウェア設計の フィージビリティ・スタディ	0.5 (Hrs.)	
ステップ4	まとめ	0.5 (Hrs.)			
合 計		40.5 (Hrs.)	5日間		

環境の各構成要素は次のとおりである。

- ①オブジェクト中心型プロセス(第1レベルおよび第2レベル)
- ②プロセス定義シート
第1レベルの各プロセスに関する技術情報を各プロセスに対して1枚ずつ用意した。その例を図-3に示す。
- ③設計課題システムの仕様
- ④システム設計のオブジェクト例
人工衛星の運用管制システムを単純化したシステム設計のオブジェクトを正解例として参照させた。

(2) 受講者の資格

講座の受講者は入社2~3年程度の次のようなレベルの技術者を対象とする。コンピュータおよびリアルタイムOS等に対する基礎知識があり、簡単なリアルタイム処理プログラム開発の経験がある。

(3) 設計教育講座の構成

設計教育講座は、設計プロセスを中心とした講義と課題システムの設計演習の二つから構成される。具体的な構成を表-1に示す。

5. 設計教育の実施結果

この講座に要した時間を表-1に示す。この設計教育講座の結果は次のとおりである。

(1) 受講者は全員がシステム設計の未経験者であったが、ほぼ満足なオブジェクトを作成することができた。

(2) 本講座は、ソフトウェア全体の基本設計を短時間のうちに疑似経験できるものであり、OJTに代わる設計教育として技術の伝承に効果的である。

(3) プロセスおよびオブジェクトによって設計が誘導されるので初心者でも取り組みやすい。また、設計作業を誘導するものとしては、プロセスよりもオブジェクトの方が具体性があり、効果的である。

(4) 教育を目的としたソフトウェア・プロセスの詳細度は、あまり細かくするとかえって理解し難くなる。逆に粗すぎると自由度が多過ぎて不適切となる。この意味でも前述の二つのレベルのプロセスは適切であった。

(5) プロセス定義シートは、プロセスを補うものとして有効である。

(6) オブジェクトの評価に長時間を必要とするため、本講座の受講者数は1回に4~5人程度が限度である。

6. 設計教育講座の評価

6.1 本講座の設計教育効果

本講座の設計教育に関する効果は次のとおりである。

(1) 標準的技術としてのオブジェクト中心型プロセスを十分に理解させることにより短時間で効率的な技術の伝承が可能となる。

(2) 設計演習を通して、オブジェクト中心型プロセスをフェイズ中心型プロセスへ変換する小規模な体験ができる。

(3) 本講座の設計は実用システムの設計とは比較できないほど小規模なものである。この差を埋めるには、実際にシステム設計の体験を重ねる以外にない。

(4) 本講座のプロセスとオブジェクトによる設計教育を通して設計技術を顕在的に表現することの有効性と重要性を理解させることができる。

6.2 講座全体の印象

受講者に対して、本講座全体の印象に関するアンケート調査を行った。その結果を図-4に示す。評価項目としては、参考文献2)のCAI評価の21項目に、図中の*印の3項目を加えた24項目で行った。アンケートの回答は、各項目ごとに5段階の評価(-2, -1, 0, +1, +2)で記入させた。同図に各項目の平均点を示す。これによれば、本

講座は技術的な深みや魅力が感じられて好ましく、今後の仕事に役立つ、等という好印象を与えるものの、量が多く、難解で取り付きにくいというマイナス要因もあることが分かる。

7. まとめ

これまでソフトウェア開発部門に内在する標準的な設計技術の伝承は、OJTを中心に進められてきた。OJTは実務そのものを利用しての教育であり、設計技術だけでなく業務の実態を把握できることを含めればきわめて大きな効果が期待される。しかし、一人の技術者が全体設計を把握するまでには長い期間を要し、また教育効率が良くないという欠点もある。我々は、部門の標準的な設計プロセスの習得とそれを設計に適用する実習を中心とする設計教育講座を開発した。さらに、これを若手技術者の教育講座として実施して、ほぼ満足な結果を得た。

本講座の目標は、部門の標準的な設計技術をプロセスとオブジェクトにより顕在的に表現し、その技術を伝承させることにある。また、受講者がこれをきっかけとして実務の中で設計経験を積み重ねながら各自の設計プロセスに発展させることを期待している。

参考文献

- 1) Katayama, T. : A Hierarchical and Functional Software Process Description and its Enaction, proc. of the 11 th International Conference on Software Engineering, pp.343-352(1989).
- 2) 城間, 中山他: テレビ電話を利用した個別教育システムの開発と評価, 電子通信学会論文誌, Vol. J 75-A, No. 2, pp. 226-234(1992).
- 3) 望月, 山内他: 人工衛星チェックアウト・システムの基本設計プロセスのプロセス・モデルHFSPによる記述とその評価, 情報処理学会論文誌, Vol. 33, No. 5, pp. 691-706(1992).
- 4) 望月, 山内: 実時間処理システム設計教育環境の開発, 電子情報通信学会研究報告 ET 92-35, pp. 51-58(1992).
- 5) Mochizuki, S., Yamauchi, A. and Katayama, K. : Two Models for Describing Software Design Process: Object-Centered Model and Phase-Centered Model, proc. of the 5 th International Conference on Software Engineering and Knowledge Engineering, pp. 291-295 (June 1993).

(平成6年8月29日受付)

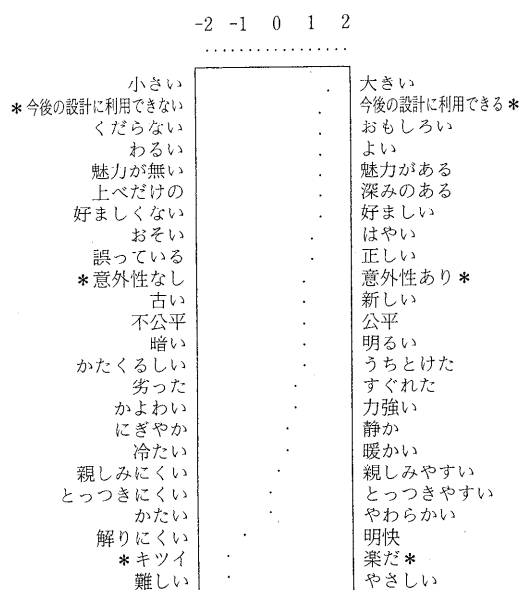


図-4 設計教育講座の印象



望月 純夫(正会員)

昭和14年生。昭和38年東京工業大学工学部電気工学科卒業。昭和40年同大学院理工学研究科修士課程修了。同年三菱電機(株)入社。以来一貫して実時間処理システムの開発に従事。昭和62年三菱スペース・ソフトウェアに移り、宇宙関連ソフトウェア開発部門を担当。平成6年三菱電機(株)を退職し、淑徳短期大学教授、現在に至る。電子情報通信学会、ACM、IEEE COMPUTER SOCIETY 各会員。

