

SIにおける中規模システム開発手法

6H-4 岩橋 治、 奥谷 博之、 土井 日輝、 高島 重行、 魚田勝臣
 [(株)三菱電機ビジネスシステム] [(株)ヒロケイ] [三菱電機(株)] [(株)オービック] [専修大学]

1. はじめに

この報告は、昨年本大会で発表されたシステム開発手法のモデルをさらに発展させたものである。今回は、特に上流工程の生産性を向上させるための手法についてより詳細に検討した。

一方、ダウンサイジング・ブームの中で、オフコン（オフィス・コンピュータ）のあり方を再認識するために、現状と将来を分析し、これからのオフコンの位置付けを明らかにした。

2. ダウンサイジング時代のオフコンの位置付け

従来のオフコンは、その高い信頼性、カスタマイズに関する総合的な高生産性を基盤に中堅企業の基幹系業務のホスト・コンピュータとして、システム化の中心にあった。今後も、基幹系業務は膨大なソフトウェア資産の活用、ユーザでの脆弱な運用体制などからオフコンが担当すべき分野である。

一方、情報系の分野では、クライアントはGUIに優れたPCに任せ、オフコンはサーバとして、その基幹系業務で蓄積されたデータベースを集中管理するのが運用面や信頼性において優れている。

企業情報システム内でのダウンサイジングは、オフコンの担当業務の縮小化という短絡的な結果にはならない。図1に示すように、オフコンは、基幹系業務を担当すると同時に新たに基幹系業務から生成されるデータベースのサーバーコンピュータとしての役割を持つことになる。（図1）

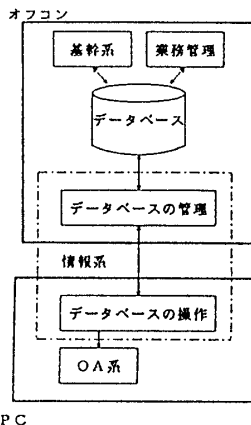


図1 ダウンサイジング時代のオフコンの位置付け

3. SIにおける中規模システム開発モデル

SI（システム・インテグレータ）は生き残りのために、高品質のソフトウェアやサービスを提供できる確固とした開発手法を持つ必要がある。

中規模システムでは工期が短いため、設計やプログラムの製作の工程に重点が置かれる。オフコンCASEをこれらの工程に適用し、生産性が向上している。しかし、従来軽視されがちだった上流工程は、重要性を増すばかりである。

今回提案するのは昨年の本大会で報告した開発モデルの上流工程を深かめ、以下の特徴を持つものである。

（図2）

(1) フォワードエンジニアリングにおける流用

工期の短い中規模システムでは、新規開発といえどもゼロから構築するのではなく、流用できる部分は極力流用して、生産性と信頼性を向上させる必要がある。

しかし、現実には、ソフトウェアの流用は進んでいない。それは、流用できないのではなく、流用できるかどうか分からないところに原因がある。

このモデルはシステム開発情報を事例として蓄積し、流用できるシステムを検索し、そのシステム概要をビジュアルに表現できるようにしたものである。

(2) リエンジニアリング

2代目以降のシステムでは、まったく作り変えるのではなく、元のシステムを基本に再構築することが多い。

しかし、プログラムとドキュメントの保守の同期がとれていないため、再構築を困難にしている。この問題を解決する手法をリエンジニアリングと呼び、リストラクチャリング、リバースエンジニアリングが含まれる。

リストラクチャリングは、障害復旧や仕様変更など度重なる保守のため、不整合になった部分について、ソフトウェアの保守を容易にするために、データ定義やロジックの構造を標準化する工程である。

リバースエンジニアリングは、ソースコード等から自動化ツールにより仕様書を生産する工程である。従来のCASEでは、ソースから画面や帳票レイアウトを生産する等システムの詳細な解析結果を生産することを得意としていたが、今求められているのは、システムの概要を正確かつビジュアルに把握できるドキュメントの出力である。

以上の手順によって、システムの問題点が明確にな

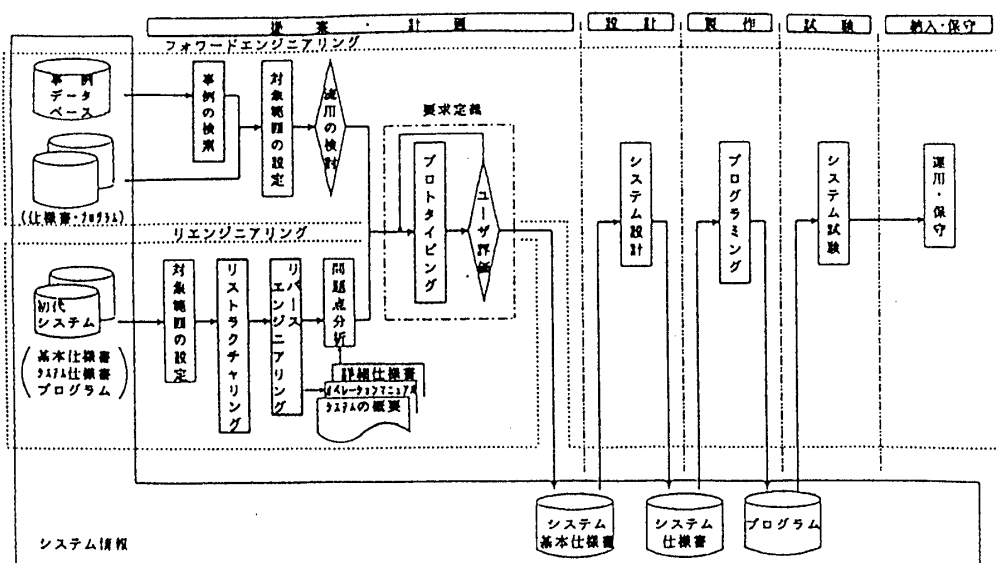


図2 中規模システム開発モデル

り、システムの最適化を可能にする。

リバースエンジニアリングが完了すると、現状のシステムの問題点を抽出し、その原因を追求し、解決策を検討した後は、フォワードエンジニアリングと同様の工程をたどる。以上の全体を通じてリエンジニアリングと呼んでいる。

(3) プロトタイピング

仕様を早期に凍結させるためには、主要な画面や帳票のプロトタイプを作成し、ユーザに提示して、その評価を仕様へ反映させ、ユーザとSIの間の機能やユーザインターフェースの仕様に関するギャップを埋め、手戻りを防止することが重要である。このプロトタイプは、使い捨てではなく、実用システムの設計・開発に活用できる進化型プロトタイプである。システムの再構築の場合は、リエンジニアリングの対象となる旧システム自体が最初のプロトタイプとなる。

既存のソフトウェアを流用するのは難しいが、流用できない場合でも、既存のシステムは使い捨てのプロトタイプになる。しかし、このプロトタイプは、プロトタイピングの最初の段階では、即効的で非常に有効である。

以後、設計、製作、試験、納入の工程をたどるが、この時の成果が、次のシステム開発の入力になる。かくて、過去のシステムが循環工程を経て新しいシステムに生まれ変わっていくのである。

この上流工程の充実したモデルを適用することによ

て、ユーザの満足する高品質のシステムを提供できる。

4. CASEの要件

現在、システム設計・開発作業のかなりの部分では、CASEによる機械化・自動化が浸透しており、システム開発と切り離せなくなっている。

一方、追加すべきCASEの機能として次の項目があげられる。

- (1) ソフトウェア資産を蓄積・検索・流用する機能
- (2) テストデータの自動生成
- (3) オペレーションマニュアルの作成支援
- (4) 性能見積・評価機能
- (5) システム開発プロジェクト管理支援
- (6) GUIを活かした設計支援機能

その他、情報系業務へのデータベース・サーバとしての役割強化策として、データ中心アプローチによる分析・設計に関連した、情報要求分析、データモデルの作成、データの標準化、正規化、保全、データとデータ固有の処理のカプセル化を実現する支援ツールの重要性が増している。

5. むすびに

今回発表したシステム開発工程は、今後のコンピュータ技術の進歩に伴い、プロトタイプがスパイラル型開発モデルを生んだように、システム開発工程自体さえ変えていくであろう。今後も、このモデルを発展させ、CASEツールの構築に応用していきたい。

参考文献

- (1) 富沢研三他：“SIにおける中規模システム開発手法について” 第46回情報処理学会全国大会論文集，1993.3, pp.5-341-342
- (2) 山崎利一他：“SIにおけるシステム開発手順の統一化” 第48回情報処理学会全国大会論文集，1994.3
- (3) 安諸春夫他：“SIにおける（中規模）システム開発手法—開発および保守環境” 第48回情報処理学会全国大会論文集，1994.3
- (4) 出口博章他：“層別化及び可視化による情報処理システムの構築” 第48回情報処理学会全国大会論文集，1994.3