

リポジトリを活用した要求定義プロセスの事例解析と考察

3K-2

奥 悦史
(株)日立製作所尾畑雄一
東邦ガス情報システム(株)水野元博
日立中部ソフトウェア(株)

1 はじめに

昨今、規制緩和、吸収合併などにもない、企業における業務アプリケーションのシステム開発の需要は大きい。この種の開発は成熟分野であるが、組織による設計作業は一定の困難さとリスクを伴う。そのため、従来からソフトウェア工学の手法やCASE製品などが適用されてきた。しかし、近年は現場サイドでの技術革新が見られないと考える。

本発表では、「工事受発注システム」の開発プロジェクトを事例として、要求定義プロセスとその実施結果を報告し、考察と課題を述べる。本発表では、特に昨今の現場サイドの実情として、次の2点に着眼する。1) データモデルなど共有される設計情報を予め確定させることが困難でも設計を進めることがある。2) 設計者の作業環境は、事実上PCとOAソフト(ワープロ、スプレッドシート、簡易データベース)である。多くの現場では高度な設計ツールを導入できない。

2 事例プロジェクト

(1) ターゲットシステム: 「工事受発注システム」は、営業部門(受注、見積、請求など)および工事管理部門(発注、検収、進捗管理など)の業務を支援する、計18機能で構成される。

(2) 全体工程: 構想、要求定義、システム設計、製造、および、移行のフェーズで構成される。期間は、要件定義から初期稼働までおおむね2年を要する計画である。

(3) 開発方法論: データ中心アプローチをベースに作業を標準化する。適用の目的は、開発の効率化と保守の容易化を狙いとした、プログラムの自動生成とソフトウェア構造の部品化である。そのために、要求定義で整合性の確保された概念モデルを作成する必要がある。

3 要求定義プロセス

要求定義工程のプロセス的な特徴を説明する。

3-1 タスク

このプロセスのタスクは、システムのユーザとの間で仕様を確定することである。仕様は、システム化要件の実現手段に依存しない概念モデルで表現する。概念モデルは、次項で説明する設計情報で構成される。このタスクは、業務設計、データ管理、資源管理のサブタスクで構成される。業務設計とデータ管理は、設計情報を定義する。資源管理は、設計情報を資源として管理・保全する。

3-2 設計情報の構成

プロセスのアウトプットである設計情報の構成を図1に示す。設計情報AとBの1対多の関係は、同時に、AがBから共有的に参照されることを意味する。例えば、エンティティのデータ項目は、情報要求から共有的に参照される。(ドキュメントのイメージでいうと、情報要求定義書に、参照先の“エンティティのデータ項目”を記述する箇所がある。)

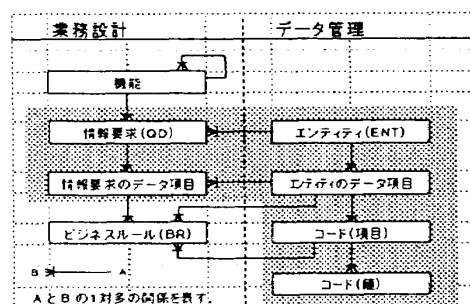


図1 設計情報の構成(メタモデル)

3-3 資源管理の方式

このプロセスの資源管理の方式を図2に示す。

(1) リポジトリ: 図1に示した設計情報を管理するデータベースである。リポジトリには、設計情報の登録と整合チェックのツールが付属する。(整合チェックは形式チェックであり意味論的な機能ではない。)

(2) 運用サイクル: 1) リリースされたデータモデル(エンティティ(ENT)とそのデータ項目)を参照して定義した情報要求(QD)を登録する。2) 登録

直後に整合チェックし、不整合箇所を業務設計に伝達する。3) 不整合箇所の修正は、設計者間で調整しながら、その後の任意のタイミングで実施する。

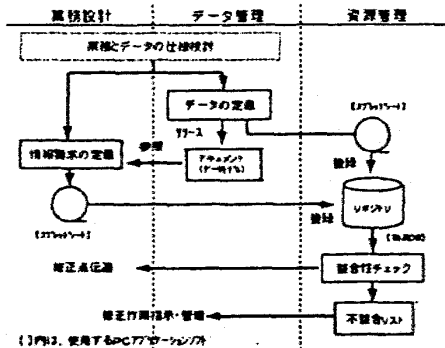


図2 資源管理の方式

この方式では、整合チェックの時点で、QDの定義時点からデータモデルが変更されていることを許す。また、その後のQDの修正時にも、一方でデータモデルを変更することを許す。整合確保は後追いになるため、遅延整合方式とよぶことにする。一般のリポジトリは、チェックイン/チェックアウト方式で設計情報を管理する。この方式は、ある設計者(参照者)が設計情報を参照しているとき、参照者以外の設計者が当該の設計情報を変更することを許さない。従って、常に整合性を崩さないことを保証できる。しかし、場合により設計者間で作業がロック状態になる。重ねて、特定の設計情報を多数の作業者が同時に参照する場合、設計を進行させることが困難になる。要求定義は、そのような特性を持つ設計作業である。要求定義では、データモデルなど共有的に参照される設計情報が存在する。そこで、本プロジェクトでは、予めデータモデルが確定していない条件下で、概念モデルを進化させ、かつ、最終的に設計情報間の整合性を確保された概念モデルを完成させるために、遅延整合方式を採用した。

設計情報(リポジトリ)		件数
機能		18
情報要求(QD)		229
情報要求のデータ項目		11,385
エンティティ(ENT)		203
エンティティのデータ項目		1,410
コード	リソース系	481
	イベント系	924
	項目	178
コード値		777
ビジネスルール(BR)		17,000

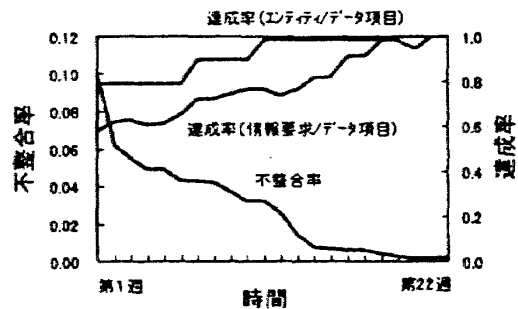
図3 設計情報のボリューム(終了時)

4 実施結果

工程終了時の実績値を図3に示す。設計情報間の共有度の指標として、参照される側とする側の設計情報の比率をみる。エンティティのデータ項目に対し情報要求のデータ項目数は、8.1倍である。

5 考察

図4に、要求定義終了時までの22週間について、不整合の推移を示す。図のとおり不整合率は堅調に減少させることができた。工程途中で仕様変更やリソース系の洗練など全体に影響する変更が発生した。それにも関わらず、不整合率を堅調に減少させることができた。本事例の条件下で、遅延整合方式で、比較的確実にQDとENTのデータ項目の整合性を確保することが可能といえる。



$$\text{不整合率} = \text{不整合箇所数} / \text{QDのデータ項目数}$$

図4 不整合の推移

6 課題

概念モデル全体を考えた場合、図3に示したように、BRのボリュームが最大である。BRもQDと同様、ENTのデータ項目を参照する。しかし、今回、BRについては、目視のウォークスルーのみ実施し、データ項目との間に遅延整合方式による整合検証を実現していない。従って、不良箇所の潜在している危険性がQDとENTの部分より高い。

7 まとめ

実プロジェクトで遅延整合方式による要求定義を実施し、プロセスの定量値を取得し、方式の妥当性を検証した。同じ方式で整合性を確保する範囲を概念モデル全体に拡張することが今後の課題となる。