# 設計履歴を利用したソフトウェア設計･保守支援方式

7S−5

浜田 雅樹　竹中 豊文
(株) ATR 通信システム研究所

## 1. はじめに

ソフトウェア開発中やメンテナンス時に発生する要求／仕様変更の影響吸収作業を効率化することは、ソフトウェアの生産性向上に大きく貢献する。一般に、要求／仕様の変更やその影響吸収を行う場合、ドキュメントとして残されている情報だけでは不足する(例えば、モジュール構成を採った理由等)。本論文では、上述の問題点を解決する方法として、ソフトウェア設計中に設計者が行った設計対象のモデル化過程(=設計履歴)を記録･利用して、要求／仕様の変更による影響の吸収を支援する方法について提案する。以下、ソフトウェアに対する要求からデータフローダイヤグラム(DFD)を作成する工程を例に、設計履歴モデルおよび修正時における影響範囲の解析手法について述べる。

## 2. 設計作業のモデル

前提とするソフトウェア設計作業モデルを示す。

- ○ 設計者は、まず要求を見て問題の理解を試みる。その結果、設計者は自分なりに問題の構造や性質をモデル化したもの(=問題構造)を視点別(例えば、ハードの制約や機能構成等)に構築する(通常は設計者の頭の中)。　以下に示す作業は順不同である。
- ○ それまでに構築した問題構造や要求を参照しながら問題構造をさらに詳細化する作業を繰り返す。設計作業(=設計プロセス)は、論理的にまとまった作業単位の列から構成される。ここで、ある作業単位で参照した問題構造とその作業単位で生成した問題構造との間に、「設計プロセスにおける因果関係」があると言う。
- ○ 設計プロセスには、問題構造を参照し、以後の設計の進め方を決定する作業単位が存在する。
- ○ ある程度問題構造が詳細化されると、解として設計生産物を生成する。設計生産物は設計法や作業標準等により定められたものである。

## 3. 設計履歴モデル

設計履歴は、上記の設計作業モデルに基づき以下に示す(1)-(3)の情報を記録する設計履歴ネットワークで表現する。Fig.1は、要求からデータフローダイヤグラムを生成する作業の設計履歴ネットワークの一部を示したものである。例題は、酒屋の販売在庫管理である[3]。

(1) 設計生産物、問題構造とその視点

問題構造は、問題ノード(機能、データ、概念等を表す)、関係(問題ノード間の関係を木構造またはノード間のリンクで表す、part-of、is-a等がある)、視点(その問題構造が、どの視点からモデル化されたものかを表す)から成る。設計生産物は、設計法や作業標準で定められた正規生産物を指す。生産物も問題構造と同様に視点、例えばDFD(図中v3, v9)など、を持つ。

(2) 作業単位と扱った問題構造の範囲

作業単位は作業ノードとして記述する(図中○)。各作業ノードは、行われた順番に関する情報(○内数字)、扱った問題構造／生産物の範囲を表すビュー(図中v<数字>)、作成した問題構造/生産物の箇所に関する情報を持つ。

(3) 設計生産物と問題構造内／間に存在する、設計プロセスにおける因果関係

設計プロセスにおける因果関係を、参照関係referを用いて以下のように定義する。

cause(P2,P1) <- edit(W1,P1) ^ has(V, P2) ^
refer(W1, V).

問題構造P2がP1の原因になっている(cause(P2,P1))とは、P1を作成(または編集)した作業単位をW1とし(edit(W1,P1))、ビューVの内容が問題構造P2のとき(has(V, P2))、W1の作業単位においてVを参照した場合成り立つ(refer(W1, V))。問題構造ネットワークでは、因果関係を得るために必要な参照関係(refer)を記録する。

以下、fig.1に記述されている設計履歴の内容について説明する。図中、作成するプログラムをX店在庫管理とする。また、原要求v0は与えられたものと

---

Software Design and Maintenance Support Using Design History
Masaki Hamada, Toyofumi Takenaka
ATR Communication Systems Research Laboratories

する。各問題構造が持つ視点は、それが属するビューの上に付した。作成関係は一部表示を省略した。

(1) 設計者は、まず原要求を読んでこの問題が一般的な在庫管理の問題であると考えた。その結果、"X店在庫管理"と"在庫管理"の"is-a 関係"を、視点として"問題種別"を持つ問題構造として記述した。この問題構造は1つのビューを形成し(v1)、作業ノード1(以下w1のように記す)がこのビューに割り当てられる。更にw1からv0 に参照関係が張られる。

(2) 次に設計者は、"X店在庫管理"は"出庫処理"と"入庫処理"から成り立っていると設計した(v2)。v1の内容を基にw2を行ったという関係が、w2からv1に対する参照関係で表されている。

以降の説明は省略する。本手法の特徴として、例えば、"出庫処理"の機能構成として、不足する・しないのケースにより分かれない機能構成を採った(v8)のは、「両ケースで共通の処理が存在する」(v4・7)ためであり、それは更に「不足するケースでも可能な数だけ出庫する」ことにしたこと(v5・6)に起因していることが記録できている点が上げられる。

## 4. 影響範囲解析

影響箇所解析は以下の2通りの方法で行う。

r1 問題ノード間に張られた同一等の関係を利用した影響解析。

r2 設計プロセスにおける因果関係を利用した影響解析。

影響範囲解析の例を、fig.1の例題について説明する。[]内は用いた上述の方法を示す。

例題：依頼者の要求に対して不足が生じた場合の処理として、現在採用している「可能な数だけ出庫する」方式から「出庫しない」方式へ変更する。

① ユーザは、v5・6の問題構造におけるalternativeの採用を変更し、影響範囲解析機能を起動する。

② システムは、v5のalternativeの採用理由を記録した問題ノード"設計が簡単"[r1]、v4・7の問題ノード"不足するケース"[r1]、w7で作成された問題ノード"出庫指示部"と関係"common"[r2]をユーザに提示する。

③ ユーザは提示された問題ノードについて影響の吸収を行い(v5・6にある採用理由を修正、v4・7の"common関係"および"出庫指示部"を削除)、再び影響範囲解析機能を起動する。

④ システムは、v4・7の問題ノード"出庫指示部"と同一関係があるv8の問題ノード"出庫指示部"[r1]をユーザに提示する。その後、ユーザの修正を経てシステムはv9の"出庫を指示する"[r1]を提示し、ユーザがこれを修正する。

## 5. 今後の課題

今後はプロトタイプシステムを作成し本手法の評価を行うとともに、設計履歴を用いた、エキスパート設計技術の再利用支援機能について検討する。

【参考文献】

[1] 島 健一, "ソフトウェア設計のための判断履歴の蓄積, 利用について", 情報処理学会 ソフトウェア工学研究会 Vol.89, No.101, 1989

[2] Colin Potts and Glenn Bruns, "Recording the Reasons for Design Decision", Proceedings of 10th ICSE, 1988

[3] 山崎 利治, "共通問題によるプログラム設計技法解説", 情報処理学会誌 Vol.25, No.9, 1984

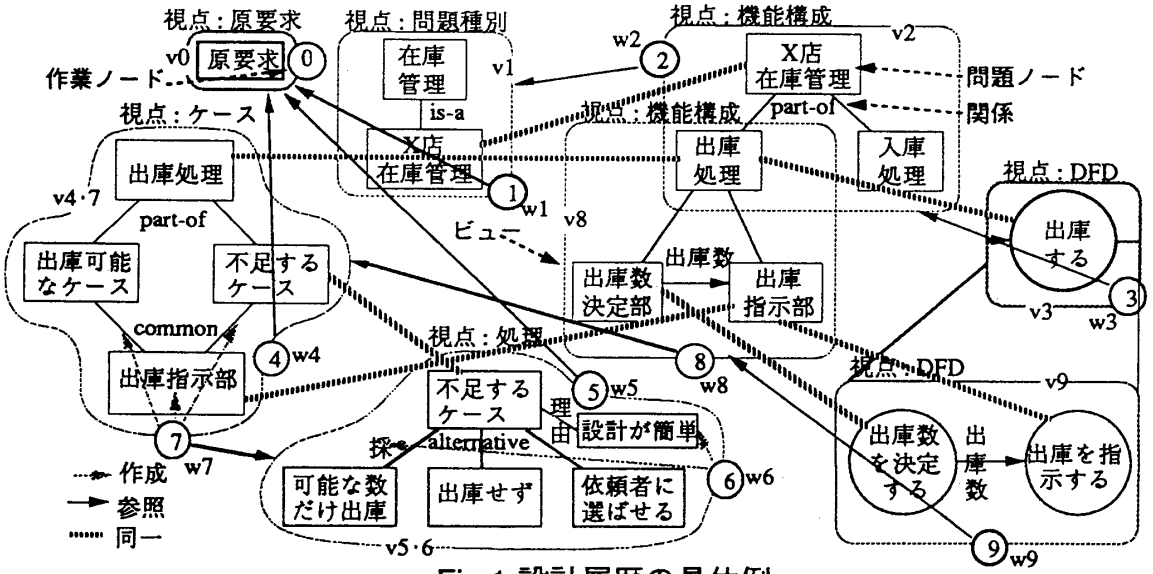


Fig.1 設計履歴の具体例