

7R-1

# 設計情報を用いたソフトウェア変更支援方法

清水洋子, 舛雑匠, 小尾俊之, 加地浩一

株式会社 東芝 システム・ソフトウェア技術研究所

## 1 はじめに

ソフトウェアシステムに対する要求は変化していくものであり、そのために既存のソフトウェアを変更することは多く発生する。この場合、(1) 変更による影響範囲を調べるのが困難である、(2) 既存のソフトウェアに対する理解不十分により、変更を重ねるにつれ冗長なソフトウェアになっていく、などの問題が起こる。そこで我々は変更作業を調査、分析し、これをもとにソフトウェア変更の支援方法を考察した。

## 2 変更時の作業

変更作業を分析するため数人の作業者を対象に、作業観察、ヒアリング、アンケートを実施した。この結果を実作業と問題点に分類して整理した。

### 2.1 実作業

変更の実作業をモデル化したものを図1に示す。

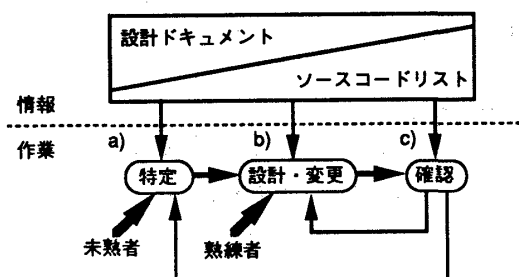


図1 変更作業モデル

変更作業は以下に挙げる3つのステップに分かれる。

- ステップ(a) 変更する部分を特定する。
- ステップ(b) 変更部分の再設計をし、変更を施す。
- ステップ(c) 変更による他への影響を確認する。

作業は基本的にはこのステップを順に進んでいくが、(c)から(b)へ、(c)から(a)へという後戻りも発生する。

また、対象ソフトウェアを十分理解していない未熟者は

(a)の作業から始めるが、熟練者になると経験により容易に変更部分を見当づけられ、(a)の作業はほとんどなく、すぐに(b)の作業から始めることもある。

変更作業は設計ドキュメントやソースコードリストという情報を参照しながら進められる。(a)の作業では、ソフトウェアの全体構成や各サブルーチンの外部仕様、データの意味などの設計情報が主に参照される。(b)の作業では、サブルーチンの内部仕様などの設計情報の他、ソースコードリストから実際の構造なども参照する。(c)の作業で参照するのはほとんどソースコードリストで、サブルーチンのコール関係やグローバルデータアクセスを追っていく。

そしてこれらの作業は、設計ドキュメントやソースコードリストをめぐりながら、(1)次々に関係する部分を追っていく、(2)前に参照したところへ戻る、(3)重要なところに印をつける、(4)気づいた点や疑問点をメモする、といった行動をとりながら進められる。

### 2.2 問題点

上述した変更の実作業のなかで特に問題となっているのは以下の3点である。

- (1) 未熟者の場合、ステップ(a)の作業に多くの時間がかかる。
- (2) 設計ドキュメントを見て処理の内容を理解するときに、なぜここでこのような処理をしているのかという意図がわからないことがある。
- (3) ステップ(c)の作業において、影響範囲を十分に確認するのが難しい。

## 3 変更支援の方法

「現状の実作業」、「現状の問題点」の両方を支援することを目的とした変更支援方法について考察した。簡単のためソフトウェア構成要素(以下エンティティと呼ぶ)として、タスク、サブルーチン、グローバルデータの3つだけを取りあげた。以下支援方法について3つの観点から述べていく。

### 3.1 保持すべき情報

[現状実作業に対して]

設計ドキュメント、ソースコードリストの中から実作業で参照している以下の情報は保持の必要がある。

- ・タスクの概要
- ・サブルーチンの外部仕様・内部仕様
- ・グローバルデータの意味
- ・サブルーチンコール関係
- ・グローバルデータアクセス情報

[問題点1 に対して]

まずその原因を次のように分析した。

一般に仕様と設計は1対1に対応しないため、ある仕様に絡むソフトウェア部分を特定することは難しい。設計書はトプダウンに読んでいくと分かりやすいようになっていることが多く、知りたいところへダイレクトにはいけない。そのために変更する部分を特定していくのが困難になっている。

そこで我々はエンティティの名称や属性に対しキーワード検索を行い、知りたいエンティティを絞り込んでいく方法を考えた。これはエンティティ名称を規則性をもってつけたり、エンティティの属性に標準化した業務用語を使うことでより有効な情報となる。

[問題点2 に対して]

現状どのドキュメントにも残されていない情報を必要としているので、新たにこれを意図情報としてとり入れる。具体的には、サブルーチンが他のサブルーチンをコールする意図(目的)、サブルーチンがグローバルデータをアクセスする意図(理由)である。

[問題点3 に対して]

影響範囲を調べやすい情報を提供する。例えば、サブルーチンがグローバルデータを介して関係するサブルーチン、グローバルデータの範囲といった情報である。

以上の4点を加味し、変更支援に必要な情報の概念スキーマをまとめた。そのER図を図2に示す。

### 3.2 情報の提供方法

図2に示した情報から次のような提示法が可能である。

- ・エンティティ一覧を表示する。
- ・エンティティ名指定によりその属性を表示する。
- ・エンティティと指定された関係にあるエンティティを抽出する。また、その関係の意図を提示する。
- ・指定キーワードを属性に含むエンティティを抽出する。

### 3.3 情報検索作業支援

実作業での行動を支援するため次の機能が必要である。

[履歴機能] 検索履歴を保持しておく。

[マーク機能] 提示されている情報全体に印をつけておく。

[メモ機能] エンティティごとにメモ書きをつけられる。

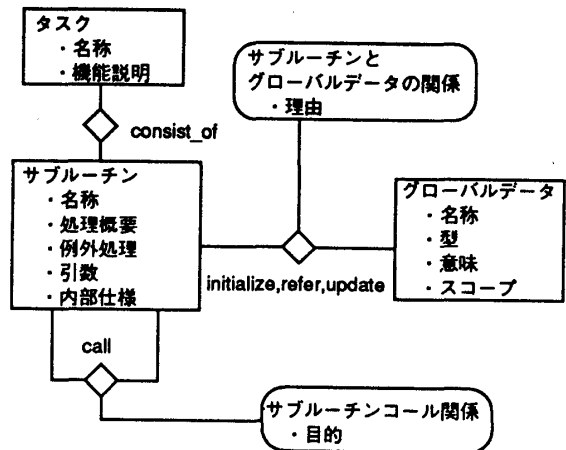


図2 概念スキーマ

## 4 期待される効果

今まで述べた方法に基づいたシステムで変更作業を支援することにより、以下の効果が期待できると考える。

- 設計ドキュメントをコンピュータ支援により検索できることで、変更作業が効率化する。
- 設計情報にエンティティ間の意図を隔に記述するため、ソフトウェアをより深く理解することができる。
- 設計ドキュメントを保守などでのソフトウェア変更時に積極的に活用することにより、設計情報のドキュメント化に対し強い動機づけができる。

## 5 おわりに

ソフトウェアの変更作業を調査、分析し、これをもとに支援方法を提案した。現在我々は、本稿で述べた方法に基づくシステムをJ3100上に試作し、評価中である。今後はこの方法の有効性を検証していく。

なお本システムの開発は、当社の、ソフトウェア生産の工業化を目指すIMAP(Integrated software Management and Production support system)開発の一環として行っているものである。

### 【参考文献】

- James S. Collofello & Mikael Orn  
"A Practical Software Maintenance Environment"  
Conference on Software Maintenance-1988 (1988)
- 斎藤明紀他 "ソースコード検索閲覧システム"  
情報処理学会研究報告 90-SE-71 (1990)
- 松村一夫他 "ソフトウェア設計記述法"  
東芝レビュー Vol.41 No.8 pp6-10 (1986)