

ソフトウェア意図伝達支援ツールCOMICS (1)  
——理論的背景——

2M-1

○岸本美江\* 西田正吾\* 後藤卯一郎\*\*  
三菱電機(株) \*中央研究所 \*\*制御製作所

## 1. はじめに

ソフトウェア開発の生産性が低い原因の一つに、ソフトウェアの理論的表現の難しさ、他人のソフトウェアを理解することの難しさ、があげられる。そのためプロジェクトで開発する場合のコミュニケーションについて、仕様書やレビューなどが研究されている<sup>1)</sup>。従来の研究は、表記を標準化して個人差やあいまいさを排除することによって、この問題を解決しようとしてきた。しかし我々は、ソフトウェアの内容は完全に標準化しきれものではなく、あいまいな部分をカバーする方法が他に必要であると考えている。そこで実際のソフトウェア開発プロジェクトを調査してコミュニケーションの問題をとらえなおし、ソフトウェアの意図伝達を支援するモデルを構築した。今回、そのモデルを実現するツールを開発したので報告する<sup>2)</sup>。本件では、取材結果から得られた問題の提起と、それを支援するための理論モデルについて発表する。

## 2. 開発の目的

共同作業において、メンバー間の十分なコミュニケーションが重要である。効果的なコミュニケーションは、作業を円滑にするだけでなく、メンバーのモチベーションを向上させ、創造的活動の刺激にもなりうる。特にソフトウェアのように目に見えない対象を扱う場合には、コミュニケーションに注意しなければ、誤解や意図のズレが生産性や品質の低下に影響しやすい。

我々はソフトウェアの意図伝達を支援することによってコミュニケーションの活性化を図り、共同作業の支援を目指している。

## 3. プロジェクトの取材

調査では、プロジェクトメンバーへのインタビューとミーティングの参加を実施した<sup>3)</sup>。結果の中から、コミュニケーションの問題と思われるものを挙げる。

## ①プロジェクトにおけるコミュニケーション

メンバーが別々の組織から派遣されている場合、考え方や行動パターンが異なり、コミュニケーションがうまくいかない場合がある。

## ②開発プロセス中のコミュニケーション

プロジェクト開始時にシステムの説明が不徹底なままプロジェクトが進行してしまう場合がある。

また、開発途中での仕様変更、応援チームの参加、

開発スケジュールの変更など、プロジェクト全体の意図を修正しなければならないような場合がある。

## ③各メンバーにおけるソフトウェアの理解

仕様書をベースにしたシステム説明では不十分だったり、メンバーの能力不足のためにシステムの理解が十分でない場合がある。

また、ソフトウェアは進捗が定量的につかみにくく、適切な評価ができないという問題もある。

これらの問題の原因として、ソフトウェアの意図が表現しにくいこと、プロジェクトの環境がコミュニケーションに影響していること、などが考えられる。これをコミュニケーションの要素(送り手・受け手・送りたいメッセージ・メディア)について見直してみると、

- ・送り手と受け手の問題……視点の差、能力の差により、メッセージが理解できなかつたり、送り手の意図とは違う理解の仕方をしたりする。
- ・メッセージとメディアの問題……ソフトウェアの表現が難しい。自分のイメージをうまく表現する手段(表現方法、伝達メディア)がない。

という2点にしばることができる。もし、ソフトウェアが具体的なイメージとして表現しやすければ、頭の中のイメージをそのまま人に伝達でき、誤解や喰い違いは減少するであろう。

## 4. ソフトウェアの意図の表現

では、なぜソフトウェアのイメージや意図が伝わりにくいのか。新人プログラマーは、「説明を聞いても一つ一つは理解できるが全体がもやもやとしてつながらない感じ……自分で作業を始めてから次第に形が見えてきた」と述べている。確かに仕様書は構造や機能別に細かく分けて作成され、相互の有機的な関係は見えてこない。

また、メンバーは自分の担当のところは詳しく理解しているが、全体がどのように動くかという具体的なイメージは持っていなかった。明確なイメージを抱えているのはシステム設計者だけである。

さらに、メンバーのシステム理解はその立場によって大きく異なる。管理者の見方、プログラマーの見方で重点の置き方も問題の解釈も異なってくる。

COMICS (1)

Computer-based Media for Intention Communication in Software

--- Proposal of Theater Model ---

Mie KISHIMOTO, Shogo NISHIDA, Uichirou GOTOU  
Mitsubishi Electric Corp.

このように、従来の説明や仕様書でソフトウェアのイメージが伝わりにくかったのは、部分についての詳細な説明ばかりで、全体の動きについての説明が少ないこと、抽象的な説明が多いので具体的なイメージが抱きにくいこと、などが原因と考えられる。そこで、ソフトウェアのイメージを表現するために、以下の方法を採用する。

- ・システムの輪郭が直感的に把握できるように、ソフトウェアの全体構造を視覚的に表示し、部分と全体の関係、部分と部分の関係、その中での部分の役割を表現する。
- ・システムの動きが頭の中でシミュレーションできるように、プログラムが機能を実現していくプロセスを具体的にダイナミックに提示する。

これらは、新しいシステムを理解するための枠組を提供し、人間の頭の中でのイメージ形成を支援する。明確なイメージを持っていれば、頭の中で動かしてみたり、修正してみたり、別の視点から見てみたり、などがダイナミックにできる<sup>4)</sup>。

#### 5. 劇場モデル

このような表現方法を実現するモデルとして、劇場モデルを提案する。劇場は、抽象的なテーマを観客に伝えるコミュニケーションの手段であり、脚本家が自分の意図を観客に伝える方法が、システム設計者がソフトウェアの意図をプロジェクトメンバーに伝える場合にも有効ではないかと考えられる。

劇場とソフトウェアの説明とを比較すると (Fig.1)。

- ①構造面では、劇場は、脚本家・伝えたいメッセージとそれを演じる俳優・観客から構成される。これに対して、ソフトウェアの説明は、システム設計者・伝えたいシステムイメージとそれを実現するために分割されたモジュール・説明を聞くプログラマーから構成される点で劇場と類似している。
- ②機能面では、劇場では、脚本家の意図は一人一人の俳優にあるのではなく、その演技の連続、ストーリーの中に表現される。同じテーマでも脚本家によってストーリー展開は異なるし、同じストーリーでも俳優の選択によって雰囲気は異なってくる。

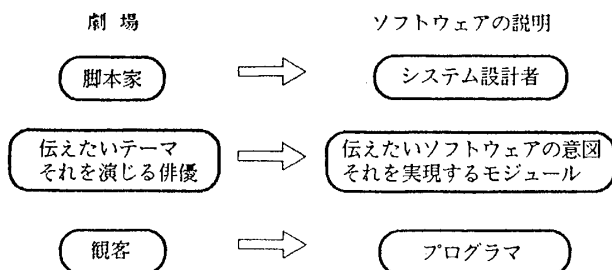


Fig.1 ソフトウェアの説明と劇場メタファー

一方、ソフトウェアは、一つ一つのモジュールを見ても意図を読み取ることはできず、その動きの流れの中に意図がある。また、同じ機能を実現するシステムでも設計者によってモジュール設計が異なったりする点でも劇場と類似している。

ソフトウェアの説明を劇場コミュニケーションとみなせば、脚本家であるシステム設計者がソフトウェアのモジュールたちを効果的に割り振り、それぞれの役を決めて動かすと、その動きはシステムのストーリーとして表される。しかし、劇場では全体の構成が最後までわからない場合もある。そこで、このモデルにさらに全体構造が一目で見渡せること、部分と全体との関係が常に見えていること、を加える。

このモデルでソフトウェアを表現すると、まず、モジュール構造と同時にハードウェア構成やデータ構成などが一覧できる。次に、モジュールやデータはシステム設計者の作ったシナリオに従って舞台上で登場し、自分の役割を遂行する (Fig.2)。モジュールは単独の存在ではなく、システムという組織の中で他のモジュールと相互作用して始めて認識される。このプロセスがシステムの動きである。観客であるプログラマーは、システムを動きのある視覚的なイメージでとらえ、自分の担当モジュールの動きを全体の流れの中で理解することができる。

#### 6. おわりに

ここでは、ソフトウェア開発プロジェクトにおけるコミュニケーションの問題と、ソフトウェアの意図を表現する方法として劇場モデルを提案した。

##### 《参考文献》

- (1)花田：「ソフトウェアの仕様化と設計」日科技連1986
- (2)岸本、他：「ソフトウェア意図伝達支援ツールCOMICS (2) ツールの開発」情報処理学会第37回全国大会1988
- (3)岸本、他：「ソフトウェア生産プロセスにおけるインタラクションの分析」情報処理学会第35回全国大会1987
- (4)佐伯：「イメージ化による知識と学習」東洋館出版社1978

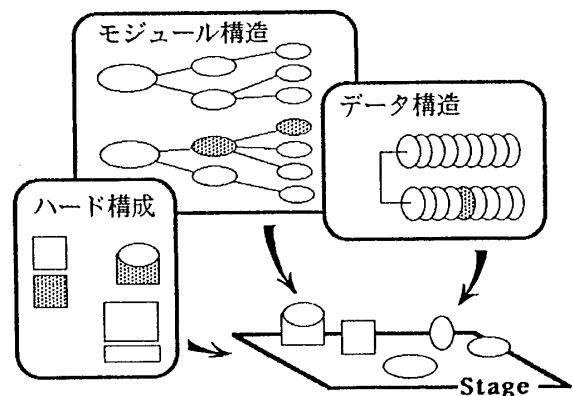


Fig.2 劇場メタファーを用いたソフトウェアの説明