

ソフトウェア開発の分析段階における

4 J-3

異なる複数視点の統合機構

木南 毅 河田 恭郎 前川 守

電気通信大学大学院 情報システム学研究科

1 はじめに

一般に大規模システムの開発の分析段階において、対象システムの全容を開発者が把握することは困難である。このことは開発者だけでなく、開発を依頼した側においても同様であり、システムの一部しか把握していないことが常である。したがって、開発を依頼した側から、対象システムを構築する上で代表的な立場にある人間を募り、それぞれの立場から観察した対象システムを分析・モデル化し、その結果得られた複数のモデルを統合して、全体のモデルを得るというアプローチが有効とされる。このアプローチを本稿では複数視点の統合と呼ぶ。

本稿では、それぞれの立場からの対象システムのモデル化に、オブジェクト指向的手法を用いることを前提とした上で、開発の分析段階における複数視点の統合のための機構について説明する。

2 Viewpoint, View, 複数視点の統合

具体的な説明の前に以下の用語を定める。

- Viewpoint

対象システムのモデル化を、どのような見方に立ちおこなったかを表す。

- View

ある Viewpoint において得られた対象システムの分析結果（モデル）。標準的なオブジェクト指向の手法を用いて作成される。

- 複数視点の統合

同一対象に対する複数の View を結合し、単一の対象システムのモデルを得ること。

図1は、開発の対象となるシステム *System* の Viewpoint VP_A , VP_B , VP_C によるそれぞれの View V_A , V_B , V_C を表す。「複数視点の統合」とは、この V_A , V_B , V_C を結合して、対象のモデルを得ることである。

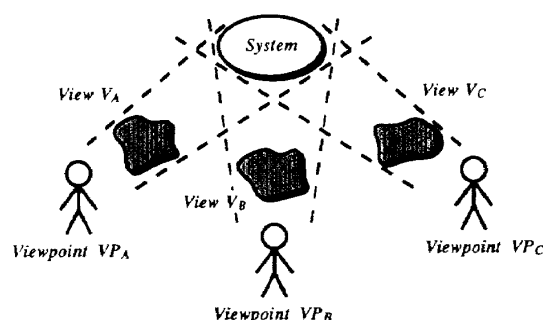


図 1: Viewpoint と View

3 複数視点の統合

本研究における「複数視点の統合」では、各 View 間で機能的に同一視できるオブジェクトを統合オブジェクトとしてまとめ、それを元に各 View を結合する。

統合オブジェクトは、View とは異なる作業領域である統合作業領域に作成され、各 View に存在する統合元のオブジェクトと対応づけられる。統合の様子を図2に示す。

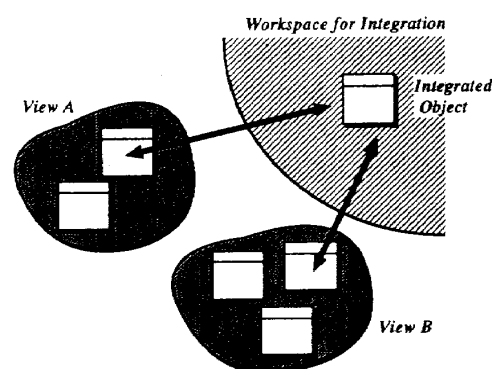


図 2: View の統合

基本的に統合は、統合オブジェクトによって管理されることとなる。このため、統合前後で View 自体に大幅な変更が加わることはなく、View を統合後においても統合前の状態と同じように利用し、更新していくことができる。ただし、View の更新がおこなわれた場合は、統合操作を再度おこなう必要がある。

以下に、これを踏まえたうえでの「複数視点の統合機構」を用いた開発スタイルを示す。

Step 1. View の構築

Step 2. View の統合

Step 3. View の更新

Step 4. 新しい Viewpoint (新しい View) の追加

Step 5. View の統合

Step 6. Step 3 ~ 5 を適当な回数だけ繰り返す

3.1 統合オブジェクトの生成

以下に、簡単な場合として、1つの統合オブジェクトに対応づけられる各 View 内のオブジェクトが1つ以内存在する場合の、統合オブジェクトの生成手順を示す。

統合オブジェクトの生成手順

Step 1. オブジェクトレベルの統合操作

- (a) 各 View 間で、機能的に重複しているオブジェクト群 (統合対象となるオブジェクト群) を検出する。すべての View に重複している必要はない。
- (b) Step 1.(a) で検出したオブジェクト群を統合するための名前だけからなる統合オブジェクトを作成する。名前のつけ方は任意。

Step 2. スロット*レベルの統合操作

- (a) Step 1.(a) で検出したオブジェクト間において、機能的に重複しているスロットを検出する。
- (b) Step 1.(b) で作成した統合オブジェクトの要素として、Step 2.(a) で検出したスロットを統合するための名前だけからなるスロットを作成する。名前のつけ方は任意。
- (c) Step 2.(b) で作成したスロットの値を、統合元のスロットの値から意味的に適切なものを選択し、すべての統合元のスロットで共通の値として設定する。
- (d) Step 2.(c) で設定した統合オブジェクトのスロットの値を、統合元のオブジェクトの対応するスロットの値として反映させる。

以下にこの手順にもとづいた統合の例を示す (図3)。

4 まとめ

以上に挙げた機構によって、1つの対象に対する分散したモデルである View 群を、統合オブジェクトによって結合し、全体として対象のモデルとして捉えることが可能となる。

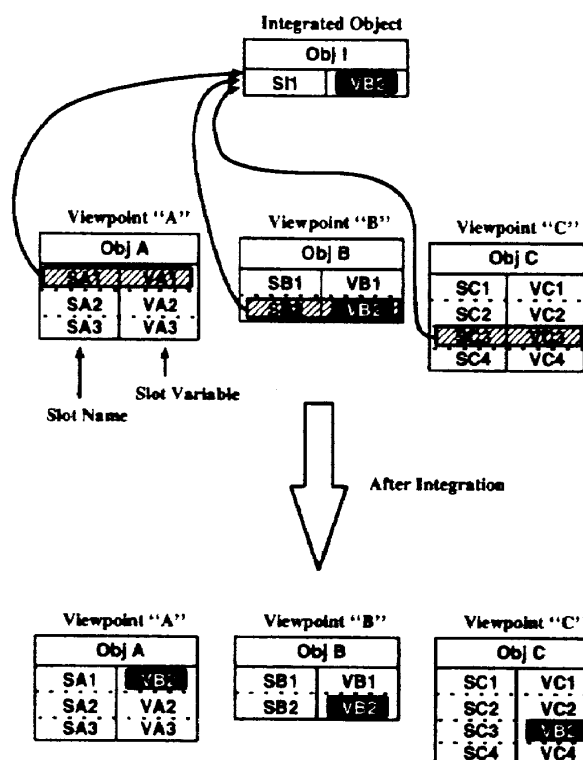


図3: 統合の例

また、View 開発者に対しては、View という局所的な作業領域における開発が保証されるとともに、開発全体で共有している概念に対する共通した見方が提供される。

5 今後の課題

現段階では、1つの統合オブジェクトに対応する統合元のオブジェクトが、それぞれ各 View において1つ以内の場合に限った統合となっている。今後はこれを拡張し、複数の統合オブジェクトの組が、各 View 内の複数のオブジェクトの組に対応するような機構が求められる。

参考文献

- [1] A. Finkelstein, J. Kramer, B. Nuseibeh, L. Finkelstein, and M. Goedick. Viewpoints: A framework for integrating multiple perspectives in system development. *International Journal of Software Engineering and Knowledge Engineering*, 1992.

*ここでは、変数と手続きを区別せずスロットとして扱う