

3K-7

ソフトウェア生産性向上を目的とした 仮想化／抽象化技術の体系化について

土居公司 飯田豊男 福留五郎

オムロン(株)コントローラ研究所第1技術開発課

1. 背景

ソフトウェアの生産性の革新的な向上には3つのアプローチがある。

- ・ プロセスの革新 : プロセス成熟度モデルなどに基づく組織の整備
- ・ マテリアルの革新 : プラットフォーム、フレームワークなどインフラの整備
- ・ メソッドの革新 : 再利用を容易とするモジュール設計技法などソフトウェア生産技術の整備

我々は今回メソッドの革新として「仮想化／抽象化技術」に着目し、その定義、実現レベルの設定、現状の実現レベルの評価を実施した。本論文では、その結果および今後の課題を報告する。

2. 定義

ソフトウェアの共通性を高め生産性向上を図るためには、次の2つの考え方が重要である。

a. 仮想化

論理的な特性が同一と見なされる資源ごとに統一的な論理仕様を定義し、その仕様をソフトウェア設計の対象とする。複数の動作環境の上で1つのソフトウェアを動かすとき、OSやドライバを設計するときに有効な考え方である。

b. 抽象化

枝葉末節のことには目をつぶり、その時点で考えている問題の本質的な性質、特徴のみをソフトウェア設計の対象とする。再利用可能なソフトウェア部品を設計するときに有効である。

我々は図1に示すようなソフトウェアモデルを定義し、環境の仮想化、対象の抽象化という観点から仮想化／抽象化の実現レベルを評価した。

なお、環境の仮想化と対象の抽象化という観点の違いはあるが、両者ともソフトウェアの共通化を目的とした考え方であり、本質的には同一の考え方といえる。

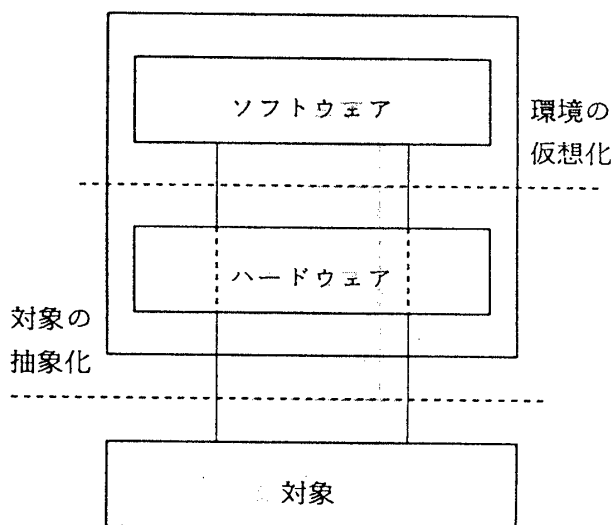


図1 仮想化と抽象化

3. 仮想化／抽象化の実現レベル

仮想化／抽象化の実施状況を評価するために、レベル1からレベル4までの実現レベルを設定する。

レベル1 : 環境／対象を直接操作

レベル2 : 共通インタフェースを定義し、共通ライブラリを使用して環境／対象を間接的に操作

レベル3 : 環境／対象をモデル化し、モデルを操作することで環境／対象を間接的に操作

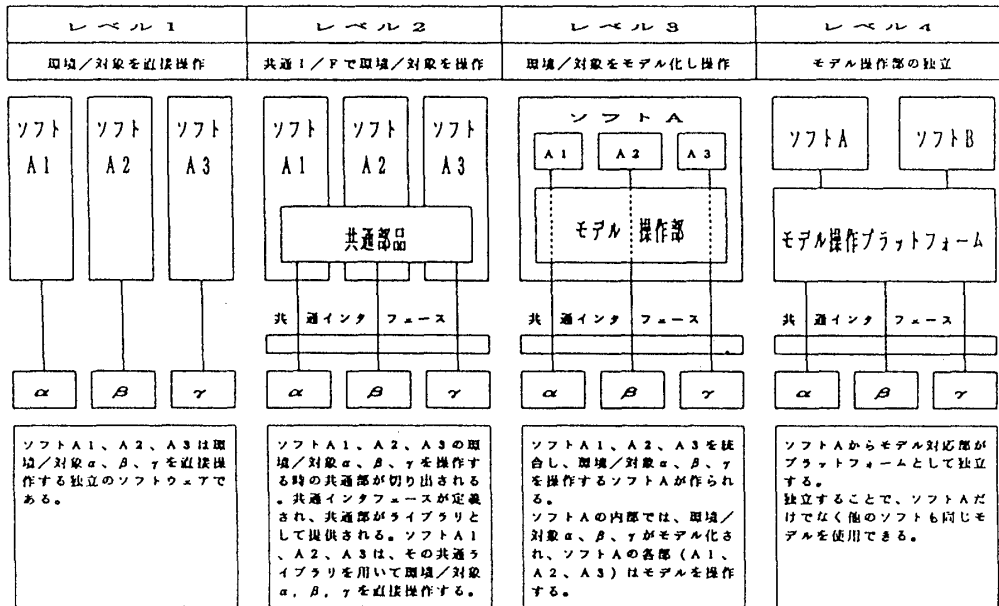
レベル4 : モデル操作部をプラットフォームとして独立

Techniques of virtualization and abstraction for increasing software productivities.

Koji Doi, Toyoo Iida, Goro Fukutome

OMRON Co. Intelligent Controller Lab.

表1 仮想化/抽象化の実現レベル



4. 仮想化/抽象化技術マップ

(1) 縦軸に仮想化、横軸に抽象化を設定した2次元のマトリクスを使用して、分野ごとのソフトウェア開発について技術の成熟度を評価した。

(2) 10分野についてマトリクスを作成したところ、次の2タイプに分類できることが判明した。

Aタイプ: 抽象化の実施が先行し、仮想化の実施が追隨するタイプ

先にモデルを定義し、そのモデルを操作するソフトウェアを開発するという観点から、ソフトウェア工学的には理想的なタイプといえる。通信関連のソフトウェアや業務パッケージなどがAタイプに分類できる。

Bタイプ: 仮想化の実施が先行し、抽象化の実施が追隨するタイプ

複数の環境の上で動作することを重要視する場合におこりやすいタイプであり、現実的なタイプといえる。基本ソフト、ミドルソフトなどがBタイプに分類できる。

(3) 一般的にはレベル1よりレベル2、レベル2よりレベル3の方が実現レベルが高く、レベル4が技術的には最も優れているといえる。分野によっては、必要メモリ容量や要求性能、コストなどの動作環境の制約を受け、レベル2やレベル3が最適なケースも起こり得る。

コストや速度要件の厳しい製品に組み込むソフトウェアを開発する場合に、この傾向が顕著となる。

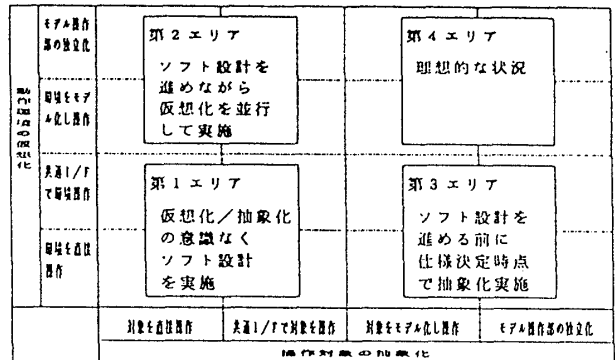


図2 仮想化/抽象化技術マップ

5. 今後の予定

(1) 各分野において物理的な制約(コストなど)を加味した上で、マトリクス上の最適な位置を見究める。

(2) 各分野においてマトリクス上でより高位な位置へ上がるために習得すべき要素技術を抽出する。

6. 参考文献

- (1) 片岡雅憲: 「ソフトウェア再利用技術」 pp67-78、日科技連
- (2) 相磯秀夫ほか編: 「コンピュータの事典」第2版 pp28-39, pp478-494、朝倉書店
- (3) E. ヨードン: 「ソフトウェア管理の落とし穴」、トッパン