

---

# 組み込みソフトウェアの部品化再利用プロセス

## Component and Reuse-Oriented Development Process of The Embedded Software

山田 大介\*

Daisuke YAMADA

---

### 要 旨

デジタル化・ネットワーク化により、増大化・複雑化が進んでいる複写機などに搭載するソフトウェアを、部品化、再利用することで、生産性と品質の向上を実現するために、問題領域単位の大規模な部品化、設計の抽象度をあげたモデリング、固定部と変動部のパッケージ分割、フレームワークという形態を主要なコンセプトとするプロセスを作成した。部品開発においては、問題領域単位にソフトウェア部品を構築して企業資産を蓄積し、商品開発においては、ソフトウェア部品をすばやく組み立てて、高品質な製品を提供し続ける仕組みが確立された。また、本プロセスの適用例として、レビューによるチーム開発、モデルパターン体系化と水平展開等の実践を紹介する。

### ABSTRACT

Component and reuse-oriented development process for the embedded software of copier are developed to improvement of the productivity and quality to correspond increasing and complication. The principal concepts are large scale component by a problem domain, modeling which increased a degree of abstraction of the design, packages divided by common parts and variable parts, framework reuse. Enterprise property is accumulated by building a software components in each problem domains by component development group, and product development group assembles a software component quickly, thus the process enables to keep providing high quality products. We describe some examples of best practices such as development of a team by the review, systematization of design knowledge of model pattern.

---

\* 画像システム事業本部 プラットフォーム開発センター ソフトウェア生産技術開発室 CC開発グループ  
Platform Development Center  
Imaging System Business Group



ること、が十分条件となる。これらの条件を満たし、仕様書とソースコードとマニュアルの3点セットが完成した時点で部品として登録することが可能となる。

UMLで記述された温度制御ドメインのモデル例をFig.3に示す。

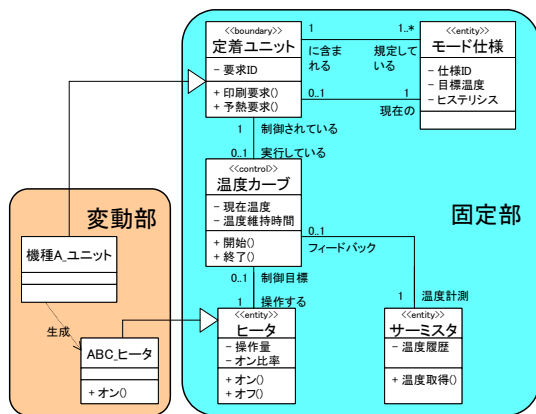


Fig.3 ソフトウェア部品の例

## 2-3 大規模再利用

### 2-3-1 ドメイン単位の再利用

従来の、ソフトウェアの再利用は、演算ライブラリといった機能単位の比較的小規模なブラックボックス再利用が主流であった。しかし、ブラックボックス再利用では、部品の点数が多くなり、少しだけ異なるが似た部分を多く持つ部品も必要となってしまうことから、成功した事例は極めて少ない。

本取り組みでは、モデルを固定部分と変動部分に分離した、いわゆるフレームワークと呼ばれるホワイトボックス再利用を目指している。

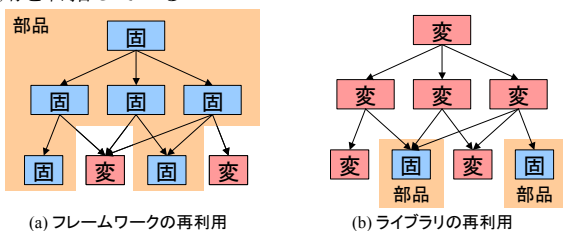


Fig.4 再利用形態の比較

ライブラリの再利用では、変動部から固定部が呼び出されるのに対して、フレームワークの再利用では、固定部から変動部が呼び出されることになる。

このような構成になることにより、ライブラリの再利用で

は、変動部を作ることによる障害の混入可能性が高くなり、かつ、変動部は個人のノウハウに依存してしまう。それに対して、フレームワークの再利用では、変動部を作ることによる固定部分への影響はほとんどなく、かつ、固定部分にドメインのノウハウが詰め込まれているため、変動部設計者はドメイン自体をあまり知らなくても、フレームワークを使うことでドメインを理解し、開発することになる。すなわち、実装の再利用ではなく、設計の再利用が可能となる。

### 2-3-1 商品開発と部品開発の連携

商品開発と部品開発は、非同期で実施することが可能であり、Fig.5に示すような連携をしている。

商品開発においては、既存のソフトウェア資産を活用し、機種依存の変動部を組み込むことによりプロダクトに搭載する。プロダクトから、ユーザ要求や企画要求を取り入れ、ドメイン単位に、それらの要求を組込んだソフトウェア部品を開発する。そして、その部品を次のプロダクトに展開していく。

このサイクルを繰り返すことにより、使われれば使われるほど部品は進化し、ドメイン単位にソフトウェア資産を蓄積することが可能となる。

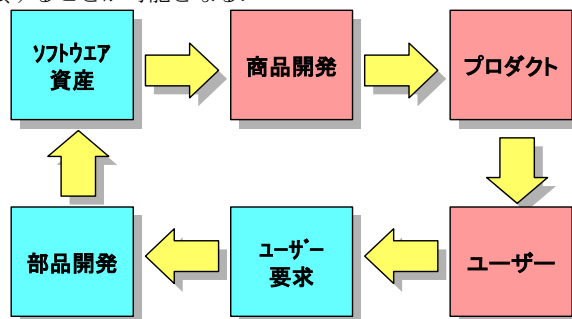


Fig.5 ドメインの再利用サイクル

## 2-3 開発プロセス

### 2-3-1 部品化再利用開発プロセス

部品化再利用の開発プロセスをFig.6に示す。

ソフトウェア部品を中心にして、商品開発と部品開発がそれぞれ平行に進行することを示している。

商品開発においては、どの部品を使うかなどを決定するシステム設計と部品の変動部を作るカスタム設計が特徴であ

り、その他は、既存のプロセスと同等となる。

部品開発においては、ドメイン分析、フレームワーク開発、アーキテクチャモデル開発、の3つの工程が並行して行われる。

商品ごとのプラットフォームへの搭載のためのプラットフォームモデル開発は、商品開発側で実施しても良い。

フレームワークが完成後は、維持管理という工程となる。

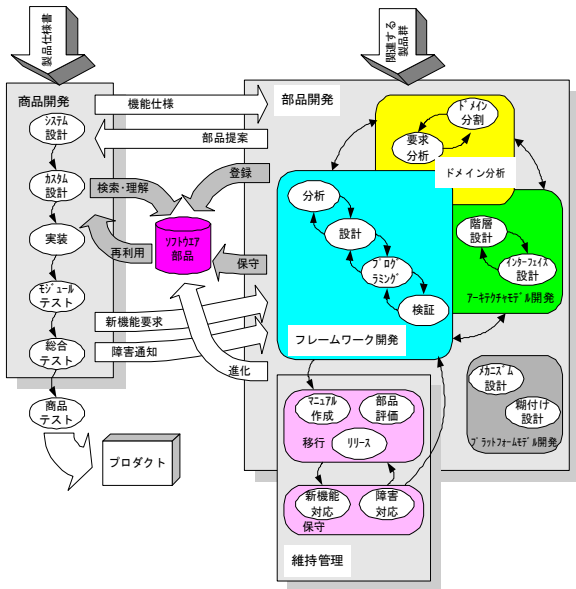


Fig.6 部品化再利用プロセス

### 2-2-3 反復プロセス管理

要求仕様をすべて明確にした上で、基本設計、詳細設計、プログラミング、テスト、と1回のパスで開発を行うウォーターフォール型プロセスではなく、徐々に品質と機能を積み上げる、反復型開発プロセスとなる。

ここでの重要なポイントは、各反復でのマイルストーンを明確にして、かつリスクをできるだけ早い時期に解決しておくことである。

また、開発の前半で、モデリングの指針やドメインの本質機能のモデル化を実施することが非常に重要である。すなわち、開発の前半で品質を作り上げて、後半でその品質を維持したまま、機能を追加していくことが必須であり、これができていないと、各反復が洗練化ではなく、やり直しになってしまう、いつまでたっても開発が終了しないことになりかねない。

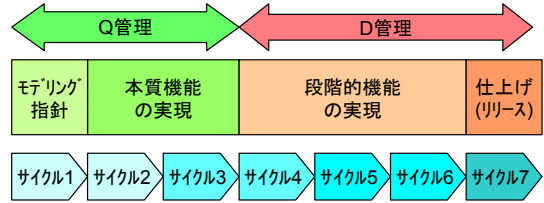


Fig.7 反復プロセス

以上のように、成果物としてモデル、プロセスとして各反復、がそれぞれ可視化されたことにより知識の水平展開が容易となり、ノウハウや失敗事例を知識として体系化し水平展開することが重要である。

## 2-3 知識

### 2-4-1 レビューによる発展型プロセス

反復プロセスを採用しているプロジェクトをうまく管理する秘訣は、設計レビューにある。

モデルが重要な設計成果物であることに伴い、複数人でのレビュー駆動の開発に移行している。

レビューとしては、欠陥を見つけて合否判定を行うレビュー（検証レビュー）に加えて、モデルの品質を検討する比較的インフォーマルなレビュー（検討レビュー）が重要な役割を果たすことになる。

すなわち、開発の成果物に対してレビューを行い、成果物の洗練化とともに、開発プロセス、チェックリスト、モデルパターンとしてフィードバックし、次の反復開発時に利用することが有効である。

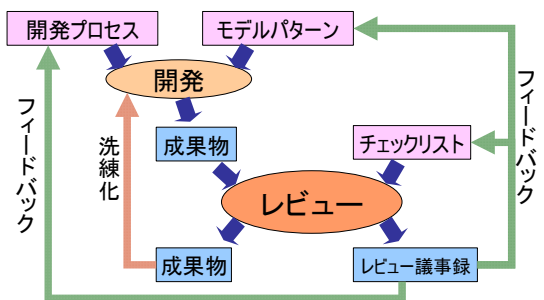


Fig.8 レビューの位置付け

## 2-4-2 チームの習熟と部品カテゴリ

部品開発プロジェクトを成功へと導く為には、成長する可能性のあるチーム構成を作ることが重要であり、それを支援するためのメンタリング及びコンサルティング仕組みが不可欠となる。

オブジェクト指向技術の習得曲線は、リニアではなく、デジタル的なステップアップであるといわれており、気づくタイミングで適切なアドバイスを与えることや、少人数での検討レビューによる相乗効果が、チームのスキル向上には効果的である。

また、部品開発チームの経験やスキルに応じて、目標とする部品カテゴリを設定する必要がある。

C言語による開発の経験が豊富な設計者ほど、手続き指向なモデルになる傾向がある。この場合も、最初から抽象化モデルを目指さずに、まずは責務モデルでのカテゴリ2を目指し、次段階で抽象化モデル（カテゴリ3）とすることを推奨している。

部品カテゴリ	モデル要件	メリット
<b>カテゴリ3</b>	抽象化モデル	再利用効果が高い →生産性の向上
<b>カテゴリ2</b>	責務モデル	保守性が高い(変更に強い) →品質の安定
<b>カテゴリ1</b>	責務と手続きのハイブリッドモデル	理解性が高い →属人性低く、チーム開発
<b>カテゴリ0</b>	手続きモデル	従来のプログラミングと同等

Fig.9 部品カテゴリ

## 3. 成果

ソフトウェア開発の生産性と品質の向上を実現するための、ソフトウェア部品の定義と部品化再利用の開発プロセスが定式化された。

ソフトウェア部品を、登録、保守、検索、利用、するためのインフラも整備されており、部品リポジトリには、すでに数十個の部品が登録され、再利用されている。



Fig.10 ソフトウェア部品リポジトリ

## 4. 今後の展開

今後、部品点数を増やし、部品カテゴリの向上を計画的に実施していく。それとともに、モデリングや反復プロセス管理の知識を体系化して水平展開を計る。

また、モデルからソースコードを完全自動生成するMDA(Model Driven Architecture)や、ライトウエイトなプロセスとして注目を浴びているAgileプロセスに関して、調査・検討を行ない、それらの長所を取り入れることにより、部品化再利用の枠組みを発展させていく。