

# モデルとビューの分離・統合を利用した 要求定義ノウハウの可視化と共有手法

橋本 憲幸<sup>†</sup> 松尾 尚典<sup>†</sup> 位野木 万里<sup>†</sup> 吉田 和樹<sup>†</sup>

東芝ソリューション (株)<sup>‡</sup>

## 1. はじめに

ソフトウェアの品質と開発の生産性を向上させるには、再利用のアプローチが有効であり、下流工程ではソフトウェア部品などの再利用が一般的に行われている<sup>[1]</sup>。上流工程でも同じドメインのシステム開発において、異なる案件の要求仕様を再利用して開発することがあり、同様のアプローチが有効であると考えられる。

再利用の取り組みの一つとして、ソフトウェアの製品系列を対象にドメインモデルを蓄積しておき、MDA (Model Driven Architecture)<sup>[2]</sup>技術に基づいて、製品を開発する研究がなされている<sup>[3]</sup>。しかし、従来の再利用は、UML (Unified Modeling Language) のようなモデリング言語で記述できるものを対象としており、自然言語で書かれた文書を再利用する技術は具体化されていない。

一般的に要求仕様書は、自然言語を中心とした文書として蓄積される。要求仕様書の再利用の課題として、1) 要求仕様の本質と、文書としてまとめるための見目が混在しているため、再利用対象の特定が困難、2) 断片的な再利用を行うと、何を仕様化すべきかが不明確となり、要求仕様の漏れや不整合が発生する、といった課題がある。そこで、要求仕様書をモデルとビューに分離することで、記載された仕様からモデルを抽出して再利用し、要求仕様書の見目の違いにとらわれることなく、要求仕様の構造や確認知識などのノウハウを可視化・共有する手法を提案する。ここで、モデルは要求仕様の本質であり、ビューは様式や記述順序などの文書の見目を定義するものである。

## 2. 要求仕様書の再利用の課題

我々は、要求仕様書を再利用する取り組みにおいて、次の課題に直面した。

### 課題 1: 再利用部分の特定が困難

通常、要求仕様書は文書として蓄積されるが、発注者により様式が異なる。再利用しようとする要求仕様書から、所望の要求を抽出して、新規案件などに活用しようとしても、要求仕様の本質である内容と、文書としてまとめるための見目の部分が混在しており、再利用する部分を特定することが困難である。

### 課題 2: 要求仕様の定義漏れ、不整合が発生

発注者ごとに要求仕様書の記述の仕方が統一されておらず、複数の案件から断片的に要求仕様を再利用すると、

要求仕様としてどのような観点で定義が必要になるかが不明確となり、要求仕様の漏れや不整合が発生する。実際の開発現場では、仕様書の品質は開発者のスキルに依存しており、そのノウハウは人に内在している。

## 3. 提案手法

### 3.1 解決のアプローチ

上記の課題に対し、次のアプローチで解決に取り組んだ。図 1 に要求定義におけるノウハウを示す。

まず、課題 1 に対しては、要求仕様を本質であるモデルと、見目を定義するビューに分離して取り扱う。さらに、別々に記述したモデルとビューを、ツールで統合することで要求仕様書を生成する。

課題 2 に対しては、成功事例の要求仕様書の分析と、ベテラン開発者へのインタビューからノウハウを抽出し、要求仕様の中身として記述すべき項目をメタモデル、発注者にとって分かりやすい文書の見え方をテンプレートとして定義する。また、有識者の確認のノウハウを検証ルールとして設定し、これらに基づいた要求仕様の検証を実施する。

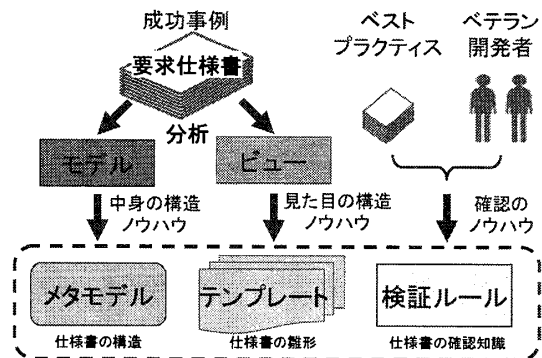


図 1 要求定義におけるノウハウ

### 3.2 メタモデル・テンプレート・検証ルールの例

メタモデル、テンプレート、検証ルールの例を図 2 に示す。メタモデルは、要求仕様の構成要素、構成要素が持つ属性、構成要素間の関連を整理したものである。なお、今回は業務系アプリケーションを対象にメタモデルを定義した。図 2 の例は、メタモデルの概念を示している。この例では、業務分析をビジネス上の目標であるゴール、ゴールを阻む課題、課題に対する解決策の観点で行い、業務を現行業務と、システム導入後の新業務に分け、5W1H の観点で記述することを表している。実際に開発者がモデルを作成する時は、メタモデルに基づいて、表形式で内容を記述する。

テンプレートは、要求仕様書の目次構成、用紙のサイズや向き、箇条書きで記述するか/表で記述するか、などの体裁を定義するものである。例では、目次の構成と、

A Method for Visualizing and Sharing Method of Know-how of Requirements Definition Based on Specification Model and Documentation View

<sup>†</sup> Noriyuki Hashimoto, Matsuo Hisanori, Mari Inoki, Yoshida Kazuki

<sup>‡</sup> Toshiba Solutions Corporation

3.3 節で説明するツールが処理するためのタグを記述している。

検証ルールは、メタモデルの各構成要素の定義漏れや、構成要素間の相互参照の不整合などの検証を行うためのルールである。例えば、メタモデルの「業務」要素に、その業務に関わる人が「関係者」属性として定義されているか、「業務」要素の属性に定義した関係者が、「関係者」要素として定義されているかを検証する。

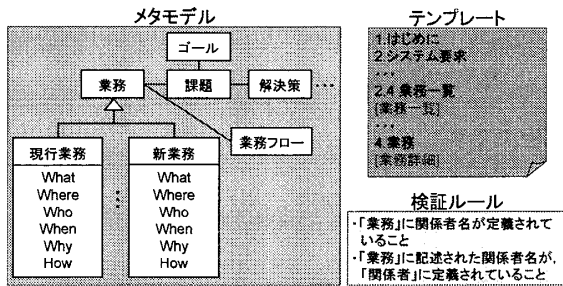


図 2 メタモデル、テンプレート、検証ルール例

### 3.3 仕様書生成・検証支援ツールの仕組み

分離したモデルとビューを統合するために、図 3 に示す支援ツールを開発した。本ツールは、メタモデル、仕様生成・検証エンジン、検証ルール、及び生成ルールで構成する。メタモデルと検証ルールは 3.1 節で説明したものである。仕様生成・検証エンジンは、入力された要求仕様データを検証し、指摘事項を表した検証レポートを生成する。また、要求仕様データとテンプレートを合成して要求仕様書を自動生成する。生成ルールは、モデルとテンプレートを合成する手順を定義したものである。

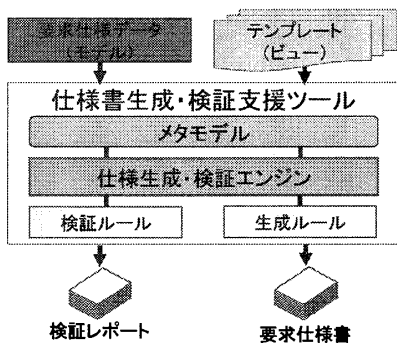


図 3 仕様書生成・検証支援ツールの構成

### 3.4 要求定義ノウハウの共有手法

要求定義ノウハウの共有手法を図 4 に示す。要求定義ノウハウは、メタモデル、テンプレート、検証ルールとして蓄積する。また、要求仕様書は、要求仕様データ（モデル）として蓄積する。要求定義ノウハウとモデルを再利用の対象とする。

本手法では、要求定義を次の手順で行う。開発者は、成功事例やベテラン技術者のノウハウを利用して要求定義を行う。要求定義の成果物は、要求仕様データ（モデル）として蓄積しておき、同じドメインのシステム開発において、蓄積されたモデルを再利用して開発を行う。また、システム開発のプロセスとは別に、既存のノウハウや、メタモデルに基づいて蓄積されたモデルの分析を

行い、メタモデル、テンプレート、検証ルールを改善する。このスパイラルを回していくことで要求定義ノウハウを蓄積する。蓄積したノウハウをツールに組み込んで利用し、組織全体で共有する。

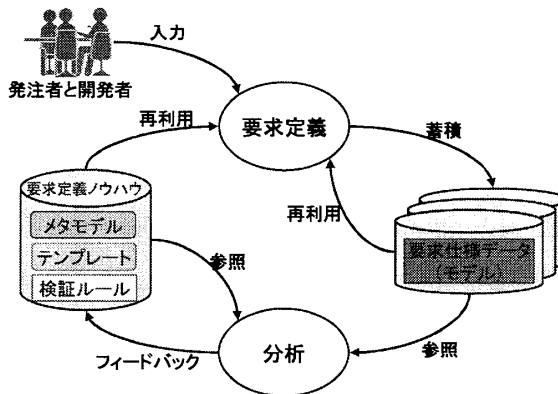


図 4 要求定義ノウハウの共有手法

### 4. 期待効果

本手法に基づく、要求仕様はメタモデルに基づいた均質な形で一元管理されるため、ドキュメントの中に要求仕様埋もれることなく、再利用の対象を特定しやすくなる。また、メタモデルに従って漏れや重複のない要求仕様を定義し、検証ルールに基づいて整合の取れた要求仕様となり、テンプレートと合成することで最終ドキュメントを生成できる。

要求定義ノウハウを統一した形で資産化し、ツールによる共有で知識継承が行える。さらに、組織全体で、メタモデルに基づいた開発を継続的に実施し、メタモデルの改善を行うことで、要求定義ノウハウそのものの蓄積・再利用が可能になる。

### 5. まとめ

要求仕様書をモデルとビューに分離・統合することにより、要求定義ノウハウを可視化・共有する手法を提案し、これらのノウハウを組み込んだツールを開発した。

今後の課題として、メタモデルによって要求仕様の記述レベルは統一化できたが、個々の仕様の詳細度（どのくらい細かく書けばよいか、など）のばらつきを抑える必要がある。このために、様々なドメインのモデルを蓄積していき、参照モデルを開発することを検討している。また、様々な実案件に適用し、メタモデルと検証ルールを充実させる必要がある。

### 参考文献

[1] Mili, H., et al., Reuse-Based Software Engineering, Techniques, Organization and Controls, John Wiley & Sons, 2002.  
 [2] J. Miller and J. Mukerji, MDA Guide Version 1.0.1, omg/2003-06-01, Object Management Group, June 2003.  
 [3] J. Greenfield et al., Software Factories: Assembling Applications with Patterns, Models, Frameworks and Tools, John Wiley & Sons, 2004.