

既開発製品群のソフトウェア仕様から 標準モジュールを抽出できる機能構造モデリング

Functional Structure Modeling Method to Extract Standard Modules from Software Specifications in Groups of Existing Products

砂川 英一 SUNAGAWA Eiichi 長野 伸一 NAGANO Shinichi

製造業では、市場セグメントの細分化に伴って、少量多品種の製品開発が一般化してきている。ソフトウェアを含む製品開発の効率化には、再利用性の高い機能を標準モジュールとして整備することが望ましい。しかし、複雑化した機能間の関係や仕様の前提不一致が原因となり、既開発製品群から標準モジュールを抽出するのは容易ではなかった。

そこで、東芝グループは、ある製品群が持つ機能の再利用性を分析するために機能間の関係を表現する機能構造モデルを定義し、オントロジーを用いて機能を抽出・体系化して標準モジュールを特定できる機能構造モデリングを開発した。この手法を小売分野の既開発製品群に試験適用し、標準モジュールが抽出できることを確認した。

Accompanying the progress of market segmentation, manufacturing industries have been increasingly moving toward high-mix, small-lot production in recent years. In order to improve the efficiency of development of products including software, there is a need to extract and prepare highly reusable software functions in existing products as standard modules. Due to the complicated relationships among these functions and the differences in prerequisites for specifications, however, it has been difficult to extract such standard modules from groups of existing products.

To solve this issue, the Toshiba Group has defined a functional structure model that represents the relationships among software functions to analyze the reusability of each function in groups of existing products, and systematized an ontology-based functional structure modeling method for the extraction of standard modules from specifications of software in groups of existing products. We have confirmed the potential of this method by means of feasibility studies using software specifications of existing products in the retail field.

1. まえがき

近年、市場セグメントの細分化により、少量多品種の製品展開が求められている。これに伴い、特にソフトウェアシステムを含む製品では、機能の多様化と開発リードタイムの短縮の両立が必要となった。そこで、東芝グループは、共通の顧客要求を満たす“機能”の標準モジュールを定め、個別の要求から定まる機能と組み合わせることで、多様な製品を効率的に提供する開発技術の確立を目指している。

ここでは、機能間の関係を表現する型となるモデルを用いて、既開発製品群の仕様から機能の標準モジュールを抽出する手順と、これを小売分野の製品に試験適用した結果について述べる。

2. 派生開発のプロダクトライン開発化

2.1 派生開発の課題

多様な顧客要求に合わせたカスタム開発が頻繁に生じる分野では、新たな製品のために、開発済み製品の仕様を部分的に変更する開発、いわゆる派生開発が行われることが多い。ここで、派生開発で生じる二つの問題に注目する。

一つ目の問題は、共通部分と可変部分の区分や根拠の

消失である。派生開発の多くは、ほかで開発した製品の仕様に対し、必要な部分を修正・追加して進められる。このとき、現在の製品への要求と過去の製品への要求、そしてそれを達成する既存モジュールの関係を適切に把握できていないと、再利用時に切り出す範囲が不適切になる。また、参照した派生元製品と開発対象の派生先製品との差分は関心事となるが、派生関係を更に遡って得られる製品との差分まで含めて考えることは少ない。そのため、全体の製品間で同じ部分（共通部分）と各製品に特有の部分（可変部分）に関する対応情報が失われてしまう。これらの結果、製品群全体の仕様の管理が複雑化し、重複開発による無駄や品質担保のための追加コストが発生してしまう。

二つ目の問題は、前提知識の不一致である。多くの顧客や開発チームが製品に関わるようになると、それぞれが持つ事業分野や製品についての前提の違いが、製品間の仕様の違いに現れるようになる。用語の違いであれば、対応関係を整理して解決できる。しかし、仕様上は同じ単語を用いているが当てはまる実例に差があるといった、内容や意味が関係する場合、解決するのは容易ではない。また、前提知識の差は、その存在に気付きにくく、何らかの稼働上の問題が認識されるまで放置されることがある。

2.2 プロダクトライン開発とフィーチャーモデル

派生開発で生じる課題に対し、プロダクトライン開発と呼ばれる製品開発の手法が提案されている。プロダクトライン開発とは、特定分野の製品開発で繰り返し利用できる開発資産（標準モジュール）を事前に整備し、最大限に活用しながら多様な製品を効率的に生み出す手法である⁽¹⁾。

プロダクトライン開発を進める上で、製品群の開発資産を分析し、共通部分と可変部分に切り分けることは重要な課題である⁽²⁾。この分析で用いられる代表的なモデルに、フィーチャーモデルがある^{(3), (4)}。フィーチャーモデルは、製品が持つ特徴（feature）と、特徴を実現するための選択肢の関係である必須（mandatory）、選択（optional）、代替（alternative）から構成される。図1に示すフィーチャーモデルの例では、「商品登録」に対して「個数入力」は必須ではない「optional」の関係にあり、製品共通ではない可変的な要素であることが読み取れる。

2.3 標準モジュール抽出の課題

プロダクトライン開発は、多くの場合、既に派生開発を繰り返しながら複数の製品を上市した結果、類似の派生開発で無駄なコストが増加しているなどの問題が顕在化した状況で検討される⁽⁵⁾。既開発製品群から標準モジュールを決定する場合、開発者は任意の視点で発想を広げるのではなく、既存の製品それぞれの視点や前提を含んで残されている開発文書を基に行わなければならない。

2.1節で述べた共通部分と可変部分の区別が消失する問題は、“機能”に注目した製品全体における標準モジュールを用意して解決できる。しかし、フィーチャーモデルは、機能に限定せず製品の様々な特徴を対象としているため、既存の開発文書から機能に関する情報をまとめて比較しやすい形式にはなっていない。

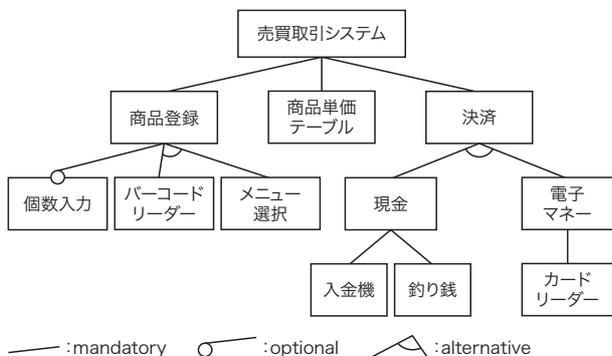


図1. プロダクトライン開発で用いられるフィーチャーモデルの例
製品が持つ特徴（feature）と、特徴を実現するための選択肢の関係（mandatory, optional, alternative）から構成される。

Example of feature model for software product line engineering

また、前提知識が一致しない問題について、フィーチャーモデルは、様々な製品の特徴について知識を集約し、比較しやすくする効果が期待できる。しかし、得られた特徴が製品を横断して通用するかは保証されていない。したがって、得られた特徴のうち、開発者が違和感を覚える部分で詳細情報を確認する必要があるとされており、コストと品質の問題は解決されていない。

3. 機能構造モデリング

東芝グループは、既開発製品群が持つ機能の再利用性を分析するため、機能間の関係を表現する型となるモデル（機能構造モデル）を定義した。そして、オントロジーを用いて表現されたルールに沿って、既開発製品群が持つ機能を抽出・体系化して、標準モジュールを決定する手順を開発し、“機能構造モデリング”と名付けた。

3.1 機能構造モデル

機能に視点を絞ったモデルに、機能分解木がある^{(6), (7)}。一般に、機能はその達成に必要な部分機能の組に分解できる。機能分解木は、この関係を木構造で表したモデルである。分解元の目的機能を根の側、分解先の部分機能を葉の側に配置して階層的に機能を整理するために用いられる。

機能構造モデルでは、機能分解木では扱えない関係、非機能要素、及びオントロジーを参照して記述されるモデリングルールを導入した。以下に、これらの三つの特徴を説明する。

3.1.1 機能分解木では扱えない関係

機能分解木と同様に、機能構造モデルも機能間の目的手段関係を幹として構築される。しかし、機能構造モデルでは、既存製品の機能を体系化し、プロダクトライン開発で利用されるモジュールを抽出する目的に合わせ、合計6種類の関係を用いる。

バリエーション関係は、入出力物がより狭い範囲に限定されていることを意味する関係である。図2に示した例では、「決済」と、「現金決済」や「電子決済」の間にある破線で示される関係が該当する。この関係は、厳密な意味で機能が“分解”されていないため、機能分解木では扱われない。一方、機能構造モデルでは、製品内で認定されている機能の単位を既存の開発資産とひも付けるため、バリエーションも扱う。

このほか、機能を実行可能な状態にするプレ関係、出力物を適正な状態にするポスト関係、目的手段関係になく実行タイミングを合わせる同期関係、実行に順序制約があることを意味する前後関係を定義した。

目的手段、プレ、ポストは、相対的に機能間の依存関係

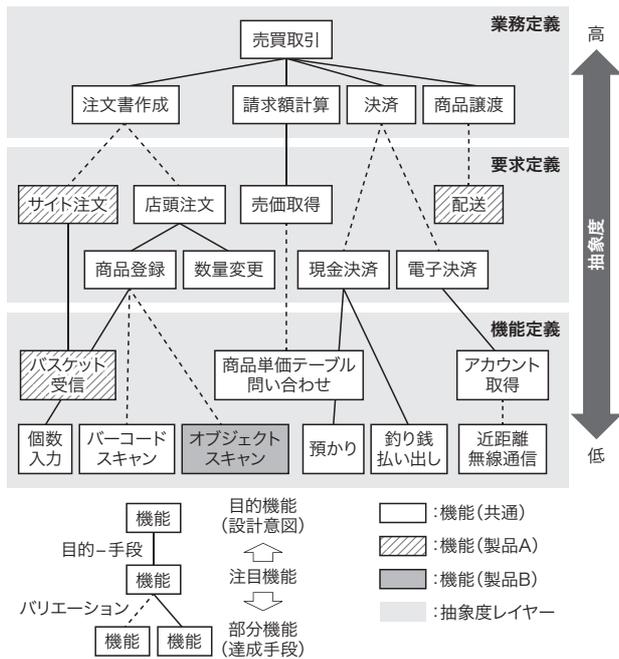


図2. 機能構造モデルの例

機能間の関係を、関係の種類などの制約を追加した木構造で表している。

Example of functional structure model

が強い関係と考えてよく、これらの関係でつながった範囲はモジュール化の検討候補になる。

3.1.2 非機能要素

機能構造モデルでは、非機能の要素を、関連する機能の属性として扱う。

機能構造モデルは、機能を優先的な関心事として、木構造中でノード化する。そのため、機能構造モデルのサブツリーは、そのまま機能に関する論理モジュールとなっていることが保証される。例えば、図2のように、機能を製品と重ねて評価することで、製品における共通性を考慮した標準モジュールの候補を得ることができる。図2は、機能構造モデルの例であり、製品共通の機能と特定の製品に依存した機能を、表示のパターンで区別している。例えば、「決済」に関する機能は全体的に共通であることや、「サイト注文」、「配送」などのネットスーパーに関する機能が製品Aに偏っていることが示されている。

機能構造モデルでは、製品や、入力、出力、実行主体を、それぞれ機能の属性として保持することを定めている。機能の再利用性には、機能間の再利用性（ほかの様々な機能にとっての共通性）と、それ以外の再利用性（例えば、製品に対する共通性やハードウェアでの共通性）がある。機能構造モデルを用い、まず木構造から標準モジュールの候補を得て、次に、異なる視点での共通性を評価できる。

3.1.3 オントロジーによるモデリングルール

モデル構築の省力化と品質向上のため、抽出・体系化の指針となるルールを定めた。ルールは、オントロジーが提供する概念を引用して抽象的に表現される。

オントロジーとは、システムの対象世界に登場する物事の識別単位となる概念とそれらの関係を扱うモデルである。

図3の右側はオントロジーの例を示しており、概念をノードで、概念間の関係を実線で表している。例えば、「現金」、「金券」、「電子マネー」の三つの概念が、より抽象度の高い概念である「対価」に分類されている。要素抽出ルールは、開発文書を基にモデルを構築する際、手掛かりとなる表現パターンと得られる要素との対応を定めたものである。ここで、抽出ルールの例として、“「機能」のために「機能」という表現パターンを考える。「機能」はオントロジー上の概念の一つを表しており、二つの「機能」の間に目的手段関係があることを意味する。このルールを適用すると、文書から表現パターンにマッチするテキスト表現を検索して、「機能」に該当する二つの要素を抽出し、それらの間に目的手段関係があると特定する。例えば、「商品コードを読み込むためにバーコードをスキャンします」という記述から、「商品コードを読み込む」と「バーコードをスキャン」の二つの機能、及び両者の間の目的手段関係が得られる。

また、体系化ルールは、モデル要素間の関係が満たすべき条件を定めたものである。例えば、図2で示した機能構造モデルには、「決済」のバリエーションとして「現金決済」と「電子決済」がある。これに入力の属性を加えたものを図3の左側に示す。ここで、体系化ルールの例として、“バリエーション関係にある二つの「機能」は、その入力の間にはis-a関係がある”を考える。is-a関係とは、ある概念が

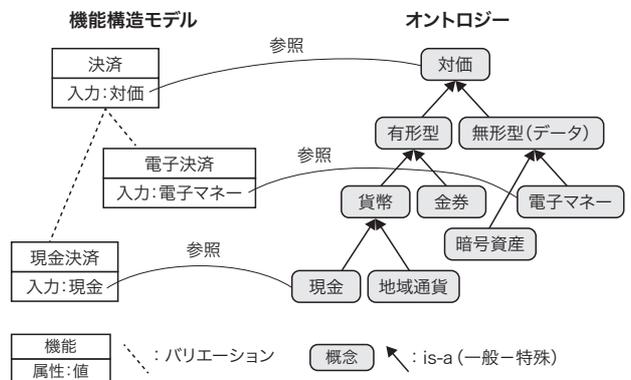


図3. オントロジーと機能構造モデルの関係

機能構造モデルの要素を構築する際に、オントロジーを参照して機能及び属性を記述する。

Relationships between ontology and functional structure model

もう一つの概念に分類されることを表す関係である。「機能」は、オントロジー上の概念の一つを表す。このルールは、バリエーション関係は該当する二つの機能の間に is-a 関係があれば、各機能の入力属性の間にも is-a 関係が成立することを意味する。図2の「決済」と「現金決済」に着目すると、二つの機能間にバリエーション関係があり、かつ両機能の入力属性である「対価」と「現金」との間に is-a 関係があるため、このルールは成立すると分かる。同様に、「決済」と「電子決済」についてもこのルールが成立していると分かる。

ルールを用いることで、迅速な判断や自動処理による省力化、矛盾の確認や一貫性の維持による品質向上を期待できる。

3.2 機能構造モデリングの実行手順

機能構造モデリングは、以下に示す(1)から(4)までの4ステップで実行する

- (1) 対象製品の開発文書から事業や業務レベルの機能を抽出し、機能構造モデルの最上位層を記述する。
- (2) (1)で得られた上位の機能構造を、開発単位となっている機能のレベルへ拡張し、対象機能が達成されるプロセスを機能構造モデルとして記述する。
- (3) (2)で得られたモデルを、体系化のルールによって評価し、拡張と修正を繰り返す。
- (4) (3)で得られたモデルから、ほかの部分と関係が少なく独立した部分を探し、ほかの属性の視点も考慮して標準モジュール化すべきかを検討する。

4. 小売分野の製品を対象とした試験適用

小売分野における製品開発では、社会的な購買スタイルの変化に応じた拡張から事業者が店内で行う施策まで、規模・内容ともに幅広い要求への対応が求められる。

東芝グループでの小売分野の製品開発を調査したところ、決済などの小売全体に関わる機能から、注文管理や販売促進などの業種や顧客に依存する機能まで、幅広く開発していることが確認できた。そして、開発管理の主な単位が部門や製品になっているため、機能の標準モジュール化によって効率向上が期待できることを確認できた。

そこで、3.2節で述べた機能構造モデリングの実行手順に沿って、リテールシステムに関する開発文書、及び事前に整備したオントロジー（約600件の概念を含む）を参照しながら、機能構造モデルを構築し、標準モジュールの検討を試験的に行った。

その結果、生産・製造や、売買取引、物流など、抽象度の高い目的機能と、その要求に応じる各製品の開発機能との関係を明らかにすることができた。その過程で、従来の開

発体制では別に扱われていた機能の幾つかは、抽象化して集約できた。例えば、販買取引の領域については、既存製品の中では延べ約700機能が数えられていたが、約90個の標準モジュールを含む約300機能に集約され、単純な機能数の比較で約60%の重複を削減できる見込みが得られた。

5. あとがき

既開発製品群が持つ機能の共通性を分析するための機能構造モデルと、オントロジーによる抽出・体系化のルールを参照しながら機能構造モデルを構築する手順を確立した。そして、この手順を小売分野の製品群に適用し、将来製品の基礎となる機能を標準モジュールとして抽出できた。

機能構造モデルの構築コスト削減のため、現在、東芝グループでは、分析対象となる製品が大量にある場合に有効な、自然言語処理やオントロジーによるルール処理を応用した機能構造モデルの半自動構築技術を開発中である^{(8), (9)}。引き続き、より広い事業分野への適用に向け、手法と要素技術の改良や洗練化を進めていく。

文献

- (1) 岸 知二. プロダクトライン開発の全体像と要求工学. 情報処理. 2009, **50**, 4, p.268-273.
- (2) 野田夏子. プロダクトラインの可変性管理 ～可変性のモデル化とアーキテクチャ設計. 情報処理. 2009, **50**, 4, p.274-279.
- (3) Kang, K. C. et al. Feature-Oriented Domain Analysis (FODA) Feasibility Study, Software Engineering Institute, Carnegie Mellon University, 1990, CMU/SEI-90-TR-21 ESD-90-TR-222, 155p.
- (4) 野田夏子, 岸 知二. プロダクトライン開発における可変性のモデル化手法. コンピュータソフトウェア. 2014, **31**, 4, p.66-76.
- (5) 長野伸一, 砂川英一. 派生開発製品の機能構成を図解するセマンティック技術. 人工知能. 2020, **35**, 2, p.163-169.
- (6) Pahl, G. et al. Engineering Design - A Systematic Approach. 3rd ed., Springer-Verlag, 2006, 617p.
- (7) 來村徳信, 溝口理一郎. オントロジー工学に基づく機能的知識体系化の枠組み. 人工知能学会論文誌. 2002, **17**, 1, p.61-72.
- (8) 砂川英一, 長野伸一. 既存システムの設計文書群からの機能モデル抽出の試み. 人工知能学会研究会資料. 2018, **44**, 1, p.1-10.
- (9) Sunagawa, E.; Nagano, S. "Extraction of Functional Structure Graph from System Design Documents". Vol-2293 Workshop and Poster Proceedings of JIST2018. Awaji, 2018-11, JIST Steer. Comm., 2018, p.122-125.



砂川 英一 SUNAGAWA Eiichi, Ph.D.
研究開発センター 知能化システム研究所
システム AI ラボラトリー 博士 (工学)
人工知能学会会員
System AI Lab.



長野 伸一 NAGANO Shinichi, Ph.D.
研究開発センター 知能化システム研究所
システム AI ラボラトリー 博士 (工学)
人工知能学会・電子情報通信学会・情報処理学会会員
System AI Lab.