

## プロダクトラインのスコープ決定手法について

岸知二<sup>†,††</sup> 野田夏子<sup>††</sup> 片山卓也<sup>†</sup>

プロダクトラインの開発においては、プロダクトラインに含まれるプロダクト群が共有するプロダクトラインアーキテクチャと、それを踏まえた再利用資産に基づいて、効率的な開発をすることが望まれる。そうしたプロダクトライン開発においては、開発するプロダクト全体の中で、どのプロダクト群をプロダクトラインとして捉えるかという、スコープの決定が重要性を持つ。このスコープ決定においては、プロダクトライン全体での開発効率化といった全体最適の視点と、個々のプロダクトにとって望ましいアーキテクチャを設計するといった個別最適の視点との両方の視点が必要となる。本稿では、この両方の視点に注目し、さらに意思決定の手法を適用したスコープ決定手法を提案する。

### A Method for Product-line Scoping

TOMOJI KISHI,<sup>†,††</sup> NATSUKO NODA<sup>††</sup> and TAKUYA KATAYAMA<sup>†</sup>

In order to develop a product-line effectively, it is desired to design the product-line architecture for products in the product-line, and develop reusable assets based on the architecture. In such a product-line development, it is important to determine the scope of product-line; namely determine the membership of the product-line. In product-line scoping, we have to examine the scope from two aspects: One is from the whole optimal such as total development cost and reuse ratio. The other is from the individual optimal such as quality attributes of each product. In this paper, we propose a method for product-line scoping, in which we utilize a decision-making method, and examine the scope from above mentioned two aspects.

#### 1. はじめに

プロダクトライン<sup>1)10)</sup>とは、共通の特徴を持ち、特定のマーケットやミッションのために、共通の再利用資産に基づいて作られるプロダクトの集合を意味する。プロダクトラインを効果的に開発するためには、プロダクト群に共通する適切なプロダクトラインアーキテクチャを設計し、それに基づいて各種の再利用資産を整備するだけでなく、それをとりまくツール、環境、管理手法、さらに組織などを体系的に整備する必要がある。そういう意味でプロダクトラインは、高度に組織化された戦略的再利用のベストプラクティスをめざすものとも考えることができる。

プロダクトライン開発においては、そのスコープ、すなわち共通の再利用資産から作られるプロダクト群の範囲、を決定することが出発点となる<sup>2)</sup>。共通性を持ったプロダクトを異なったプロダクトラインと捉えることは開発上の効率を損なうおそれがあり、逆に共

通性の持たないプロダクトを同一のプロダクトラインとして捉えると、効果的な再利用が阻害される。したがってプロダクトラインのスコープ決定は、プロダクトライン開発において重要なステップとなる。

一般にプロダクトラインのスコープ決定は、様々な視点からなされる。例えばプロダクトのビジネス展開に関わる視点、プロダクトの対象ドメインの特性に関わる視点、再利用など開発技術に関わる視点などである。本稿ではこのうち、技術に関わる問題にフォーカスして決定手法を検討する。すなわち、ここではプロダクトラインのスコープ決定を、複数のスコープの選択肢の中から、技術的な観点からもっとも適切なスコープを選択する問題であると捉える。

プロダクトラインのスコープの適切性は二つの観点から検討されなければならない。そのひとつはプロダクトライン全体での開発コストの低減や、再利用率の向上といった全体最適の視点である。複数のプロダクトが共有するアーキテクチャを設計し、それらに基づく再利用資産を整備するには大きなコストがかかる。したがって、全体最適性をあげることにより、プロダクトライン全体で、どれだけのコスト削減が見込めるのかという観点での検討が重要となる。

<sup>†</sup> 北陸先端科学技術大学院大学  
Japan Advanced Institute for Science and Technologies  
<sup>††</sup> NEC エレクトロニクスデバイス S1 事業開発本部  
NEC Device S1 Business Development Division

もうひとつの視点は、個別最適の視点である。共通の再利用資産に基づいて開発することは、全体としての開発コスト削減にとっては有効であっても、個々のプロダクトにとっては、必ずしも適切とは限らない。例えばプロダクト毎にそれに最適化した設計をした方が、個々のプロダクトにとってはより適切な品質特性が達成できるであろう。一般に、全体最適の視点に基づくスコープの決定と、個別最適の視点に基づく決定とは必ずしも一致しないことが多い。われわれはその両者のトレードオフを考えながら適切なスコープを決定しなければならない。

本稿ではこうした点に考慮したプロダクトラインのスコープ決定手法を提案する<sup>7)</sup>。ここでは、個別最適の視点と全体最適の視点の両者から、個々のスコープの選択肢がどれだけ適切であるかを意思決定<sup>3)</sup>の手法を用いて決定し、それに基づいて最終的なスコープの選択を行う。

2章ではスコープ決定について述べる。3章と4章では、それぞれ個別最適、全体最適の概念的枠組みについて述べる。5章では、プロダクトラインに対する要求について検討する。6章では、スコープの決定手法を提案する。

## 2. プロダクトラインのスコープ決定

### 2.1 スコープ決定とは

プロダクトラインのスコープ決定とは、プロダクトラインを定義すること、すなわちプロダクトラインを構成するプロダクト群を決定することである。一旦プロダクトラインが定義されると、我々はそのプロダクト群が共有するプロダクトラインアーキテクチャを設計し、それに基づいて再利用資産の検討を行う。したがって、スコープを定義する際には、それらのプロダクト群がアーキテクチャを共有することが適切かどうかを判断しなければならない。

アーキテクチャ共有の適切性を判断するためには、プロダクトラインを構成するプロダクト群だけでなく、それらがどのようなアーキテクチャを共有するかを含めて検討することが望ましい。すなわちここでプロダクトラインのスコープ決定とは、与えられたプロダクト群を、ひとつ以上のプロダクトラインに分割し、さらに個々のプロダクトラインに対するアーキテクチャを定義することと考える。

図1は、プロダクトラインのスコープの例である。ここでは、プロダクト群 P1, P2, P3, P4 に対して考えられるいくつかのスコープの例を示している。ここで S1, S2, S3 は考えられるスコープであり、A1, A2

はプロダクトラインアーキテクチャを表している。例えば S1 は全体をひとつのプロダクトラインとして捉えており、そのプロダクトラインはアーキテクチャ A1 を共有している。あるいは S2 は 2 つのプロダクトラインを定義し、ひとつは P1 と P3 から構成され、それはアーキテクチャ A1 を共有し、もうひとつは P2 と P4 から構成され、アーキテクチャ A2 を共有している。

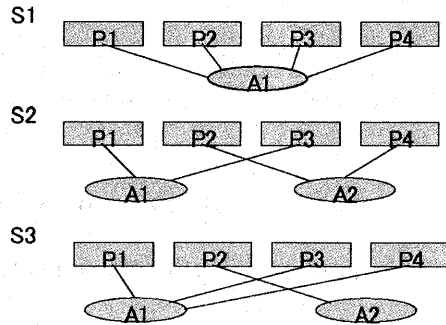


図1 プロダクトラインのスコープの例

以下本稿では、スコープをプロダクトラインの集合として表現し、プロダクトラインをそれを構成するプロダクト群と、共有されるアーキテクチャとの二つ組みとして表現する。表1は、図1に示した S1, S2, S3 の記述である。

表1 プロダクトラインのスコープの記述方法

S1	<P1, P2, P3, P4, A1>
S2	<P1, P3, A1>, <P2, P4, A2>
S3	<P1, P3, P4, A1>, <P2, A2>

なおここでプロダクトラインのスコープの決定においては、いくつかのスコープの選択肢が与えられ、それらの中からもっとも適切な選択肢を選ぶものとする。また、ひとつのプロダクトから構成されるプロダクトラインもスコープとして認めるものとする。

### 2.2 全体最適と個別最適

プロダクトラインのスコープ決定においては、プロダクトライン全体にとっての最適性と、個々のプロダクトにとっての最適性を検討しなければならない。表2はプロダクト P1, P2, P3 のスコープ決定をする場合に考えられる両極端のスコープ例である。

表2 両極端の例

S1	<P1, P2, P3, A1>
S2	<P1, A1>, <P2, A2>, <P3, A3>

S1においては、すべてのプロダクトがひとつのアーキテクチャを共有している。ここではひとつのアーキテクチャ上ですべてのプロダクトが作られるため、全体の開発コストを減少させることが期待できる。しかしながら、ひとつのアーキテクチャがすべてのプロダクトにとって最適であることは困難であるため、個々のプロダクトの品質特性にとっては、必ずしも適さないことも多い。一方S2では個々のプロダクトはそれぞれ自身のためのアーキテクチャを持っているため、個々のプロダクトの品質特性が最適化されることが期待される。しかしながら個別のアーキテクチャを開発することは全体の開発コストの観点からは不利である。

上記は一般的な傾向を述べているだけではあるが、日常的な我々の体験に整合する観測である。スコープの決定においては、こうした全体最適と個別最適とのトレードオフを検討し、望ましいスコープを選択しなければならない。

### 3. 個別最適性の概念的枠組み

個別最適性とは、個々のプロダクトにとって、どのアーキテクチャを選択することが望ましいか、その望ましさの程度を言う。

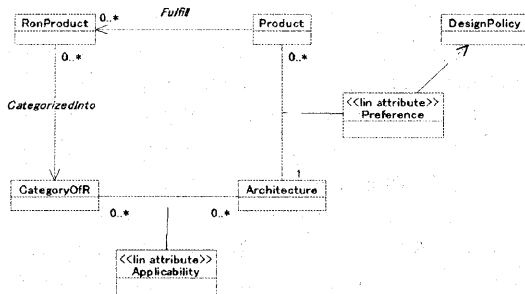


図2 プロダクトとアーキテクチャの概念的な関係

図2は、プロダクトとアーキテクチャとの間の概念的な関係を示している。ここで *RonProduct* はプロダクト (*Product*) への要求を示しており、プロダクトはそれを満たすように構築される<sup>8)</sup>。一方、一般にアーキテクチャ (*Architecture*) は複数のプロダクトの開発のベースとなる基本構造であるから、その上で開発が想定されている複数のプロダクトへの要求群を考慮して、その上に構築されるプロダクトがそれに対する要求を満たすことができるように設計されなければならない。ここで *CategoryOfR* は、要求カテゴリを意味しており、そのアーキテクチャの上で構築されるプロダクト群が達成すべき要求の集合を示している。アー

キテクチャへの要求はこの要求カテゴリに基づいて検討される。

要求カテゴリからアーキテクチャへの要求を導く方法には様々なものがある。例えば機能に対する要求を考えると、それは想定されるプロダクト群の諸機能の共通機能 (ホットスポット) であつたり、あるいはすべての機能の集合であつたりする。一方品質特性面の要求事項とは、例えばプロダクトへの要求の中で最も厳しい要求であつたりする。ここで、アーキテクチャがこれらの要求事項を満たすことができることを適用可能 (*Applicable*) であると言う。スコープの候補を考える際の必要条件として、そのスコープ中で定義されているプロダクトラインに対して対応付けられているアーキテクチャは、そのプロダクトライン中のプロダクト群から導かれる要求項目に対して適用可能であることが求められる。

一般に適用可能なアーキテクチャは複数存在する。しかしながら我々が設計をする際には、適用可能なアーキテクチャの中からもっとも望ましいと思われるアーキテクチャを選択しようとする。例えばできるだけ性能が良い方式を選びたいとか、できるだけメモリ使用量の小さな方式を選びたいとかといった価値判断に基づいて、適用可能なアーキテクチャ候補を順序付ける。図2において、設計方針 (*DesignPolicy*) は、こうした価値基準である。プロダクトと、アーキテクチャ候補、さらに価値基準が与えられることにより、そのアーキテクチャ候補が、そのプロダクトにとってどの程度望ましいかという好ましさ (*Preference*) が定義できる。複数のアーキテクチャ候補毎の好ましさに基づき、それらのアーキテクチャ候補の間に選好順序を決定することができ、それに基づいて最も最適な (好ましい) アーキテクチャ候補を得ることができる。

なお一般にこうした意思決定においては、判断基準となるいくつかの観点を明確にする必要がある。我々はアーキテクチャ設計においては、注目する品質特性をその判断基準とすることが望ましいと考えている。すなわち、プロダクトとアーキテクチャ候補が与えられることにより、それが個々の品質特性にとってどの程度適しているかをまず検討し、次にそれを総合して全体としての好ましさを決定する。設計方針は、複数存在する品質特性のうち、どの品質特性をどの程度重視するかという価値観を与えるものと考えることができる。

### 4. 全体最適性の概念的枠組み

個別最適性が個々のプロダクトとアーキテクチャの

間に定義されるのに対し、全体最適性はプロダクト群とスコープの間に定義される。

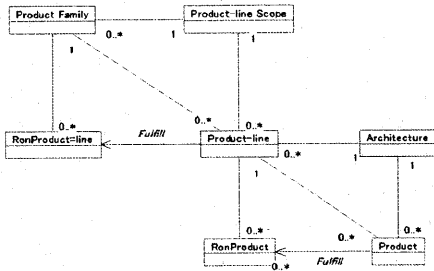


図3 プロダクトラインとスコープの概念的な関係

図3は、プロダクトラインとスコープの間の概念的な関係を示したものである。プロダクトラインのスコープ (Product-line Scope) は、与えられたプロダクト群 (Product Family) 中に複数のプロダクトライン (Product-line) を定義するものである。個々のプロダクト (Product) はいずれかのプロダクトラインに含まれる。ここで RonProduct-line はプロダクトライン (群) への要求を示している。例えば、プロダクト A とプロダクト B を同時に効率的に開発したい、というのは要求の例である。スコープの選択によって、これらの二つのプロダクトが同じプロダクトラインに含まれたり、異なるプロダクトラインに含まれたりするが、それによって例えば開発コストや再利用率などといった、プロダクト群全体の持つ特性が異なってくる。すなわち、プロダクトに対する要求とプロダクトに対して定義できる品質特性によってプロダクトに対するアーキテクチャ候補の選好順序を定義できたように、プロダクトラインに対する要求と、与えられたプロダクト群全体に対して定義できる特性 (全体の開発コストや再利用率) によって、スコープ候補の選好順序を定義することができる。

### 5. プロダクトラインへの要求

プロダクトに対する要求やプロダクトに対して定義できる品質特性という概念はきわめて一般的であるが<sup>1)</sup>、プロダクトラインに対する要求やプロダクト群全体に対して定義できる特性という概念は、十分に議論されていない。以下、プロダクト群全体に対して定義できる特性を、全体特性と呼び、これについて検討する。

一般的には様々な全体特性が定義可能と考えられるが、本稿で関心のある全体特性は、プロダクトラインのスコープによって変化する全体特性である。また特

に戦略的な再利用によって生産効率を上げリソースの有効利用を狙おうとするプロダクトライン開発のコンテキストからすると、まず重要な全体特性として生産性などの開発時特性を考える必要がある。そうした開発時特性を網羅的、体系的に議論するには今後さらに検討が必要であるが、本稿では経験的に重要と考えられるいくつかの観点からの検討を行う。

一般にプロダクトラインはアーキテクチャの共有をすることにより開発効率を改善しようとするものである。従って基本的には共有率を高めるほうが望ましい。しかし2.2で議論したように、実際には個々のプロダクトへの最適性を考えなければならないため、様々なスコープの候補について検討する必要がある。ここで注目するのは、プロダクトが異なったプロダクトラインに含まれることによるデメリットである。すなわち、プロダクトライン開発において以下の状況があるときには、関わるプロダクトがアーキテクチャを共有しない (すなわち異なるプロダクトラインに含まれる) ことのデメリットを注意深く検討する必要があると考える。

- 連続開発：ふたつのプロダクトが連続して開発される場合。
- 同時開発：ふたつのプロダクトが同時に開発される場合。

上記は、関わるふたつのプロダクトがアーキテクチャを共有するか否かによって開発効率に変化しうる重要な状況である。あるスコープの候補が複数のプロダクトラインを持つということは、必ず異なるアーキテクチャ上に開発されるプロダクトが存在するわけであるから、それが上記の状況にどのような影響を持つか検討し、個々のスコープの候補がどのようなデメリットを与えるか判断することによって、個々のスコープの全体にとっての望ましさを決定しようとするものである。

すなわちプロダクト群の開発状況から、そこに存在する連続開発と同時開発の要求を明らかにし、スコープの候補がそれらに対してどの程度デメリットを与えるかを判断し、望ましいスコープを選択する。なおすべての連続開発や同時開発が、同じ重要性を持つわけではなく、連続するプロダクトをできるだけ短期間に投入することが強く求められる場合と、それほど戦略上大きな意味をもたない場合などがある。したがって、プロダクトの選好順序を決定するときに品質特性間の重みを設計方針として与えたように、これらの重要性を重みとして与えることで、判断に反映させる必要がある。

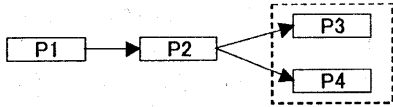


図4 連続開発と同時開発の例

例えば、図4は、4つのプロダクトの開発状況を示したものである。ここで矢印は関わるプロダクトが矢印の示す順序で連続開発されることを示し、破線の箱はそこに含まれるプロダクトが同時開発されることを示している。ここで示される連続開発と同時開発のうち、P2からP3への連続開発とP2からP4への連続開発の二つを最重視するものとする。そういう状況では  $\{ \langle \{P1, P2, P3, P4\}, A1 \rangle \}$  や  $\{ \langle \{P1\}, A1 \rangle, \langle \{P2, P3, P4\}, A2 \rangle \}$  といったスコープは、P2、P3、P4が同じアーキテクチャを共有しているので望ましいと考えられるが、 $\{ \langle \{P1, P2\}, A1 \rangle, \langle \{P3, P4\}, A2 \rangle \}$  といったスコープは、P3やP4が、P2とアーキテクチャを共有しないため望ましくないと判断される。

## 6. 決定手法

前章までに述べた基本的な考え方に基づいた、プロダクトラインのスコープ決定手法を例題を追って以下に説明する。

手法の概要は以下のとおりである。

- まず個々のプロダクトに対する要求を分析し、それぞれのプロダクトに対するアーキテクチャの候補をリストアップする。次に各プロダクトに対してこれらのアーキテクチャ候補の適用可能性を判断し、さらに適用可能なアーキテクチャ候補の中で相対的な選好順序を決定する。この作業は、個別最適の観点からスコープを検討する際のベースとなる。またスコープの候補をリストアップする際に、適用不可能なアーキテクチャをスコープ中に含まないようにするためにも必要である。
- 上記に基づいて、プロダクトラインのスコープの候補をリストアップする。プロダクトラインに対する要求を評価し、スコープの候補の妥当性を検討する。この作業は、全体最適の観点からスコープを検討する作業に相当する。
- 最後に、個々のプロダクトにとってのアーキテクチャの望ましき(個別最適性)と、与えられたプロダクト群全体にとってのスコープの望ましき(全体最適性)の両者を総合し、最も適切なスコープを選択する。ここでは、一般に個別最適性と全体最適性の間のトレードオフに対する考慮が必要と

なる。

- 上記において、選好順序の決定には、意思決定の手法を用いる。本稿では意思決定手法のひとつであるAHP (Analytic Hierarchy Process)<sup>9)</sup>を利用する。

例題では、表3に示すプロダクトを開発するものとする。

表3 開発するプロダクト

P1	日本市場向けの低価格モデル
P2	日本市場向けの標準モデル
P3	日本市場向けの高級モデル
P4	米国市場向けの高級モデル

以下に、スコープ決定の手順を示す。

- 要求を明確にする：個々のプロダクトと、プロダクトラインに対する要求を明確にする。個々のプロダクトに対する要求は品質特性に照らして定義される。プロダクトラインに対する要求は連続開発と同時開発の観点に照らして定義される。

表4は個々のプロダクトに対する要求を示している。この例題はデータ検索の機能を持ったプロダクトを想定しており、ここではその要求は、扱える検索対象のデータ項目数、検索の応答時間、ならびに必要とされるメモリサイズに照らして定義されている。例えばP1は低価格モデルのため、扱えるデータ項目数も、装備しているメモリ量も小さくなっている。一方、P3とP4は高級モデルであり大量のデータを検索でき、メモリに関する制約も緩くなっている。

表4 プロダクトへの要求

	項目数	応答時間	メモリ量
P1	100	< 2-3 sec	small
P2	1,000	< 2-3 sec	
P3	100,000	< 2-3 sec	
P4	100,000	< 2-3 sec	

一方プロダクト群は、5章で示したで示した図4のように開発される。すなわちまずP1を、次にP2が開発され、最後に、P3とP4が同時に開発されるものとする。一方表5に、プロダクトラインへの要求を示す。すなわち図4に示された連続開発と同時開発の中で、P3とP4の同時開発に対する要求が最も重要な要求であるとする。

- 設計方針を定義する：設計方針として、要求事

表 5 プロダクトラインへの要求

連続開発	P1, P2	
	P2, P3	P1 から P2 への連続開発よりも重要
	P2, P4	P1 から P2 への連続開発よりも重要
同時開発	P3, P4	最も重要な要求

項の間の優先度を定義する。本例題における設計方針は以下のとおりである。

- 低コストモデルについては、メモリサイズの削減が最も重要である。
- 高級モデルについては、応答時間が最も重要である。
- プロダクトラインへの要求を満たすことができる限りにおいて、個々のプロダクトの品質特性をできるだけよくしたい。

- (3) アーキテクチャの候補をリストアップする：個々のプロダクトに対してアーキテクチャの候補をリストアップする。

例題において、もしも品質特性に関する要求が厳しくなれば、アーキテクチャ上の工夫を特段にしなくても要求を達成することができる。そうした場合には、ドメインの意味構造を反映した分析モデルの構造を踏まえたソフトウェア構造を実現することが、拡張性などの観点からも有利であると考えられる。

しかしながら、もしも品質特性に関する要求が厳しい場合には、それに配慮した何らかのアーキテクチャ上の工夫が要求される。表 6 は、この例題におけるアーキテクチャ候補の例である。A0 は品質特性に対する特段の考慮をしない構造、A1 はキャッシュにより応答速度を改善した構造、A2 はキャッシュに加えて初期化のタイミングでインデックスをメモリ上に読み込むことによってさらに応答速度を改善した構造である。

表 6 アーキテクチャの候補

A0	分析モデルの構造を反映した構造
A1	キャッシュを備えた構造
A2	キャッシュに加え、初期化時にインデックスをメモリに読み込む構造

- (4) アーキテクチャ候補間での選好順序を決定する：個々のアーキテクチャ候補について検討し、適用可能なアーキテクチャ候補の中での相対的な好ましさを決定する。ここで好ましさの決定には AHP を適用した<sup>5)</sup>。AHP は意思決定手法のひとつで、評価観点に

照らして最も好ましいものを選択肢の中から選択するものである。この例題では、上述したアーキテクチャ候補が選択肢であり、応答速度やメモリ量といった品質特性が評価観点となる。AHP は評価観点間や選択肢間での単純な一対比較に基づいている。最初に一対比較により、評価観点間の重要性を決定し、次に個々の評価観点毎に、選択肢間での一対比較を行う。これらに基づいて、選択肢の好ましさを決定する。表 7 は、AHP によって求めたアーキテクチャ候補間での好ましさを示している。数値の大きいものがより好ましいことを示し、その順序が選好順序を与えている。

表 7 アーキテクチャ候補の好ましさ

	P1	P2	P3	P4
A0	0.367	0.26	0.193	0.193
A1	0.406	0.357	0.307	0.307
A2	0.228	0.364	0.501	0.501

- (5) アーキテクチャ候補が個々のプロダクトに対して適用可能であるかどうかを検討する：ここでは個々のアーキテクチャ候補が要求を達成できるかどうかを検討する<sup>4)5)6)</sup>。最初に個々の品質特性に照らして、アーキテクチャ候補を評価し、同一のカテゴリに含まれる要求が同一のアーキテクチャ候補の集合によって達成されるように、要求をカテゴリ分けする。例えば、応答速度について考えると、データの項目数が 100 から 1,000 の間であれば、どのアーキテクチャ候補によってもそれを達成することはできるが、項目数が 100,000 になると、A1 もしくは A2 でないとそれを達成することができないと判断されたとする。この場合、要求は二つのカテゴリに分けられることになる。同様に、メモリ量の観点からも要求は二つのカテゴリに分けることができる。表 8 と表 9 に、それぞれの観点からの要求カテゴリの検討結果を示す。ここで\*\*は、特段の要求がないことを示す。

表 8 要求カテゴリ (応答速度)

	項目数	応答速度	アーキテクチャ候補
CP1	100-1,000	< 2-3 sec	A0, A1, A2
CP2	100,000	< 2-3 sec	A1, A2

次に個々のプロダクトへの要求を検討し、アーキテクチャ候補の適用可能性を判断する。例えば応答速度の観点からは、A0 は P1 と P2 に対

表 9 要求カテゴリ (メモリ量)

	メモリ量	アーキテクチャ候補
CS1	small	A0, A1
CS2	*	A0, A1, A2

して適用することができる。一方メモリ量の観点からは、A0 はどのプロダクトに対しても適用することができる。従って、A0 は応答速度とメモリ量の両方の要求を満たすため、P1 と P2 に対して適用可能であると判断される。表 10 に、その判断を示している。マージ結果の欄に示されているプロダクトが、該当するアーキテクチャを用いることができる。

表 10 適用可能性の検討

	A0	A1	A2
応答速度	P1,P2	P1,P2,P3,P4	P1,P2,P3,P4
メモリ量	P1,P2,P3,P4	P1,P2,P3,P4	P2,P3,P4
マージ結果	P1,P2	P1,P2,P3,P4	P2,P3,P4

表 11 は、表 10 を違ったビューからまとめたおしものである。ここではどのアーキテクチャ候補がどのプロダクトに適用可能であるかをマトリクスとして示してある。'x' は適用可能であることを、空白は適用不可能であることを示している。

表 11 適用可能性マトリクス

	P1	P2	P3	P4
A0	x	x		
A1	x	x	x	x
A2		x	x	x

- (6) プロダクトラインのスコープの候補を検討する：ここでは上記の検討に基づき、適用不可能なアーキテクチャ候補をプロダクトに適用しないように留意してスコープの候補を検討する必要がある。

例題においては、表 12 に示す候補をリストアップした。S1 はアーキテクチャの共有を最大化したスコープ、S2 は低価格モデルのみ他とは異なったアーキテクチャ上で開発するスコープ、S3 は高機能モデルのみ他とは異なったアーキテクチャ上で開発するスコープである。

- (7) プロダクトラインのスコープ候補の間での好ましさを決定する：再度 AHP を使って、スコープの候補間での好ましさを判断する。ここではスコープの候補が選択肢であり、連続開発と同

表 12 プロダクトラインのスコープの候補

S1	< P1, P2, P3, P4, A1 >
S2	< P1, A1 >, < P2, P3, P4, A2 >
S3	< P1, P2, A1 >, < P3, P4, A2 >

時間開発に関する要求が評価基準となる。連続開発に関しては、開発系列の間に一度もアーキテクチャの変化が伴わない S1 が最も望ましい。S2 は S3 よりも望ましい。なぜならば、P2 から P3、および P2 から P4 への連続性は、P1 から P2 への連続性よりも重要だからである (表 5)。同時開発に関しても、やはり S1 が最も望ましい。一方 S2 と S3 については、いずれも P3 と P4 が同一のアーキテクチャを共有しているので、同等の望ましきであると判断される。

表 13 は、こうした判断に基づいて得られた結果である。全体最適は、上記の判断に基づいて AHP を用いて求めた値である。個別最適は、表 7 で求めた値に基づいて得る。例えば、S2 では P1 には A1、P2、P3、P4 には A2 を適用しているので、 $0.406 + 0.364 + 0.501 + 0.501 = 1.772$  となる。この値は個別最適の観点からの好ましさを示す指標となる。

表 13 全体最適・個別最適の判断結果

	全体最適	個別最適
S1	0.619	1.377
S2	0.229	1.772
S3	0.153	1.765

- (8) スコープを決定する：ここまでの検討に基づき、望ましいスコープを決定する。既に述べたように、S1 はひとつのアーキテクチャをすべてのプロダクトで共有しているのであるから、全体最適の観点からは S1 が最も望ましいことになる。しかしながら、ひとつのアーキテクチャを共有することは、必ずしも個々のプロダクトの品質特性にとって最適であるとは限らない。

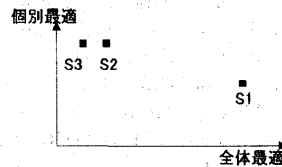


図 5 各スコープの候補のポートフォリオ

図 5 は、全体最適性を横軸、個別最適性を縦軸

にして、スコープの候補をそこにプロットしたものであり、上述した状況をわかりやすく示している。各スコープの候補が、全体最適性と個別最適性に関して異なった特性を持っていることを理解することができる。設計方針において、要求事項を満たす限りにおいて個々のプロダクトの品質特性をよくしたいと決めていたため、ここではS2をスコープとして決定する。

## 7. おわりに

本稿ではプロダクトラインのスコープ決定手法について提案した。

本文中で述べたように、本来スコープ決定は、技術的な側面だけでなく、ビジネス的な側面など様々な側面からの判断に基づいてなされるものであるが、本稿では技術的な側面にフォーカスして議論を行った。

従来スコープ決定は、プロジェクトリーダーやアーキテクトなどが経験に基づいて行うことが多かった。しかしながらこうした作業でも、体系だった手法を用い、その時点でどのような情報と価値判断に基づき、どのような判断を行ったかを明確にすることは重要である。そうすることにより、より客観的な判断ができるだけでなく、開発の進行に伴って新たな情報が得られたときに、その情報が過去の判断にどのような影響を及ぼすかを検討することも可能となる。

本稿では意思決定の手法を用いることにより、こうした判断を支援している。ここで採用したAHPは、判断しやすい小さな決定(一対比較)を行うことにより、全体として自分が何をより好ましく考えているかを明らかにする手法であり、上位の設計判断の支援に有効であると考えている。今後実際のスコープ決定の局面の事例などと比較することにより、意思決定手法のより有効な活用方法について検討していきたい。

## 参 考 文 献

- 1) Bass, L., Clements, P. and Kazman, R.: Software Architecture in Practice, Addison-Wesley, 1998.
- 2) DeBaud, J. and Schmid, K.: A Systematic Approach to Derive Scope of Software Product-Lines, Proc. of ICSE'99, 1999.
- 3) Jozwiak, L. and Ong, S.A.: Quality-Driven Decision Making Methodology for System-Level Design, Proc. of EUROMICO-22, 1996.
- 4) Kishi, T. and Noda, N.: Aspect-Oriented Analysis for Product Line Architecture. 1st Software Product Line Conference (SPLC1), 2000.
- 5) Kishi, T.: On Software Architecture - Architectural Selection based on AHP -, SIGSE, March, 2001. (In Japanese)
- 6) Kishi, T., Noda, N. and Katayama, T.: Architecture Design for Evolution by Analyzing Requirements on Quality Attributes, 8th Asia-Pacific Software Engineering Conference (APSEC 2001), p111-118, 2001.
- 7) Kishi, T., Noda, N. and Katayama, T.: Method for Product-Line Scoping based on Decision-Making Framework, 2nd Software Product Line Conference (SPLC2), 2002 (to be appeared).
- 8) Ran, A.: Architectural Structures and Views. Proceedings of the 3rd International workshop on Software architecture (ISAW-3), p117-120, 1998.
- 9) Saaty, T.L.: The Analytic Hierarchy Process, McGraw-Hill, 1980.
- 10) Software Engineering Institute.: The Product Line Practice (PLP) Initiative. [http://www.sei.cmu.edu/plp/plp\\_init.html](http://www.sei.cmu.edu/plp/plp_init.html).