

# 近似的構成管理について

岸知二<sup>†1</sup> 宮里章太<sup>†1</sup> 野田夏子<sup>†2</sup>

プロダクトライン開発やモデルベース開発などでは複数のモデルを活用するが、モデルの規模が大きくなるとその整合性維持に多くのコストがかかる。それに対応するために著者らは近似的モデリング手法の検討を進めている。本稿では近似的モデリングを利用した、近似的構成管理手法について、その方向性や課題について議論する。

## On Approximate Configuration Management

TOMOJI KISHI<sup>†1</sup> SHOUTA MIYAZATO<sup>†1</sup>  
TOSHIAKI NISHIDA<sup>†2</sup>

Advanced use of software modeling such as those in product-line development, model-based development, and formal verification require large, complicated, precise and consistent models. As a result, the cost of model development and maintenance are becoming too expensive. In order to reduce the cost, we are currently examining an approximate modeling technique. In this paper, we propose an approximate configuration management based on the approximate modeling, and discuss the research direction and issues.

### 1. はじめに

ソフトウェア開発においては相互に関連を持った複数のモデルを活用することがあるが(3)4)16)17)、ソフトウェアの規模の拡大や複雑さの増大に伴い、利用されるモデルやそれらの間の関連付けも大規模複雑化している9)。そうした背景の中、我々は近似的モデリング手法の検討を進めている(10)11)12)13)。この手法は近似化したモデルを作成したり、近似的な整合化をとったりすることによって、モデルの構築、整合化コストを減らすことを狙っている。一方、近似的モデリングはモデルが粗くなったり限定的になったりするというペナルティが発生するため、それを有効に活用するための条件や指針について検討してきた。

本稿では、こうした近似的モデリングを用いた構成管理について、実験を通してその有効性や課題を検討する。一般に構成管理は幅広い活動を含んでいるが、本稿では関係を持った二つのモデルを対象に、ある条件を満たすモデル要素を特定する作業を対象とする。具体的には、プロダクトライン開発における製品導出において、フィーチャモデルと関連付けられたアーキテクチャモデルを対象に、フィーチャに対する一定の条件を与え、それを満たすフィーチャ構成を求め、さらにフィーチャ構成の実現に必要なアーキテクチャ要素群を特定する作業について検討する。実験ではいくつかのモデルや関連付けのパターンに対して、フィーチャモデルを近似化しない場合と近似化した場合のコストやペナルティを比較し、近似的モデリングによる製品導出の有効性や課題について検討する。本実験からは、近

似的構成管理が一定の有効性を持つことが示唆される。

第2章では近似的モデリング手法について、第3章では近似的構成管理について、さらに第4章では、本稿で検討する近似的製品導出について説明する。第5章では近似的製品導出に関する実験とその結果について述べ、第7章ではそれに基づいた議論と考察を行う。

### 2. 近似的モデリング

本章では、我々が過去に検討している近似的モデリングについて、その概要を説明する(10)11)12)13)。

ソフトウェア開発のために何らかの形式性に基いて作られる記述をソフトウェアモデル(以下モデル)、モデルの構築やモデル間の整合化の作業をモデリングと呼ぶ。近似的モデリングは、当初意図したモデリングが高コストになると判断される際に、次善の策として、一定のペナルティを承知でモデリングコスト削減のためにモデルを近似することを意図している。

意図するモデルを近似的に表現したモデルを近似モデル、近似モデルを利用したモデリングを近似的モデリングと呼ぶ。近似方法には記述の抽象度を上げる概略化と、記述する対象範囲を狭める局所化との2種類がある。いずれもモデルの構成要素が減少するので直接的なモデリングコストは減少する。反面、表現されるモデルの詳細度が低下したり対象範囲が狭まったりするため、活用においては後工程で情報を補うなどのペナルティが発生する。

ソフトウェア開発では何らかの目的のために複数のモデル群を活用する。またモデリングに使えるリソースにも限りがある。したがって、その目的達成のために個々のモデルやモデル間の整合にどの程度のコストをかけるべきかを、近似化によるコスト減と、ペナルティによるコスト増

<sup>†1</sup> 早稲田大学  
Waseda university..

<sup>†2</sup> 芝浦工業大学  
Shibaura Institute of Technology

とのトレードオフを考えながら検討する必要がある。そこでモデルマップという図法を使って開発に関わるモデル群を俯瞰してモデリングの戦略を検討するための概念的な枠組みを提案した。

### 3. 近似的構成管理

構成管理は、構成要素の識別、変更、文書化、検証などに関する技術面、管理面の手法を広く意味するが6)、本稿ではソフトウェア開発に使われるモデルの構成要素の識別に関する技術的な側面に焦点をあてて議論する。

一般に、関連付けられた複数のモデルがあり、あるモデルに対して、モデル要素に対する何らかの条件が与えられた場合に、そのモデル中で与えられた条件を満たすモデル要素が特定され、それが関連付けられたモデルへの条件となりそのモデルに対するモデル要素が特定される。例えば要求モデルと設計モデル、設計モデルと実装モデルの間にトレーサビリティリンクが定義されて関連付けられている場合、要求中の特定の要求項目（要求モデル中の特定の構成要素）を指定することにより、それに関わる設計モデルや実装モデル中の構成要素が特定される。本稿ではこうした作業について検討する。

構成管理は一般に高コストである。もちろん前述したように、モデル構築やその間の整合化のコストは規模や複雑さの増大に応じて多大となるが、構成管理における構成要素の識別コスト自身が規模や複雑さに応じて大きくなる。例えばプロダクトライン開発において、必要なフィーチャ群を指定し、そのフィーチャ群を含む適切なフィーチャ構成を識別する作業は、フィーチャモデルの規模や複雑さが増すと組み合わせ的に高コストになる。

本研究では近似的モデリングを用いた構成管理を近似的構成管理と呼び、その有効性や課題を検討する。具体的にはプロダクトライン開発における製品導出を取り上げ、実験によってコストの削減やペナルティに関わる指標を測定し、その結果に基づき検討を行う。

## 4. 例題：近似的製品導出

### 4.1 製品導出の方法

本稿で対象とする製品導出が、どのようなモデルを利用してどのように行うことを想定しているかを説明する。

#### 4.1.1 フィーチャモデル

フィーチャモデル(FM)はプロダクトラインが持つ共通あるいは可変なフィーチャ群を階層的に表すモデルである7)。本稿では図 4-1 に示す記法を用いる。図 4-2 はこの記法を用いたフィーチャモデルの記述例である。なおルートから必須の関係だけで迎れるフィーチャを共通フィーチャ、迎れないフィーチャを可変フィーチャと呼び、共通フィーチャにハッチングをつけている。

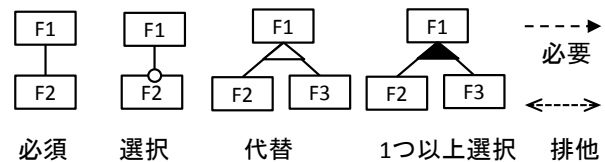


図 4-1 フィーチャモデルの記法

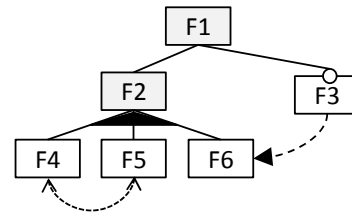


図 4-2 フィーチャモデルの例

#### 4.1.2 アーキテクチャモデル

本稿でのアーキテクチャモデル(AM)は可変性の観点からのアーキテクチャの記述である。可変性を持たせたアーキテクチャ記述に関してはいくつかの提案があるが5)15)17)、ここでは可変性の側面に注目してフィーチャモデルと類似な記法で表す。

図 4-3 は AM の記述例である。角丸四角はコンポーネントを示し、可変点を持つ場合にそこに当てはまり得るバリエーション（製品毎に変わり得る可変なコンポーネント）を選択や代替として示す。またあるコンポーネントを利用する際に必要なコンポーネントや、同時に使えないコンポーネントの関係を必要や排他として示す。また共通コンポーネントにはハッチングをつけている。

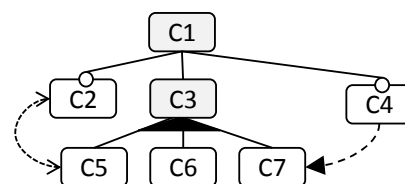


図 4-3 アーキテクチャモデルの例

#### 4.1.3 トレーサビリティリンク

FM 中の可変フィーチャと AM 中のコンポーネント間にトレーサビリティリンク(TL)を定義する。TL はフィーチャの実現に必要なコンポーネント群を示し、フィーチャ単位に対応するコンポーネント群を図 4-4 のように記述する。

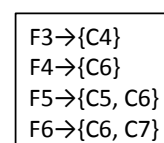


図 4-4 トレーサビリティリンクの例

#### 4.1.4 製品導出

本稿では、フィーチャの集合  $FS_0$  が与えられたとき、それらを持った製品を実現するために必要なコンポーネントの集合を決定する作業を製品導出と呼ぶ。製品導出は以下の手順で行われる。

1. フィーチャ構成導出： $FS_0$  を含み、FM の制約を満たしたフィーチャの集合  $FS_1$  を決定する。この手続きには、後述するセレクトビリティ駆動製品導出 (SDPD) を用いることを想定している。 $FS_1$  は複数存在することもあるが、その場合はその中の 1 つを選ぶ。 $FS_1$  が見つからない場合、製品導出は失敗する。
2. TL 追跡： $FS_1$  中の可変フィーチャと TL で結ばれたバリエーションの集合  $VS_0$  を決定する。 $VS_0$  が空の場合、製品導出は失敗する。
3. バリエーション構成導出：アーキテクチャ設計上での判断があれば、製品に含めるバリエーションの集合を  $VS_0$  に付け加える。その上で、 $VS_0$  と AM に対して SDPD を適用し、 $VS_0$  を含み AM の制約を満たしたバリエーションの集合  $VS_1$  を決定する。 $VS_1$  は複数存在することもあるが、その場合はその中の 1 つを選ぶ。 $VS_1$  が見つからない場合、製品導出は失敗する。

例えば、 $FS_0 = \{F3\}$  のとき、FM の制約を満たす集合としては  $\{F3, F6\}$ 、 $\{F3, F4, F6\}$ 、 $\{F3, F5, F6\}$  が考えられる。 $FS_1 = \{F3, F6\}$  を選ぶと、 $VS_0 = \{C4, C6, C7\}$  となる。ここで設計判断として C2 も必要と考えれば  $VS_0 = \{C2, C4, C6, C7\}$  となる。これらを含むコンポーネント構成は  $\{C2, C4, C6, C7\}$  しかないの、これに共通コンポーネントを加えると製品に必要なコンポーネントが特定される。

SDPD は与えられたフィーチャの集合を含み FM の制約を満たすフィーチャの集合をひとつ特定するための手法である [14]。与えられたフィーチャの集合から、フィーチャをひとつ選び (SDPD では意思決定と呼ぶ)、それが必要とするフィーチャ群を特定するとともに、それを選んだことによって排他関係や代替関係のために選べなくなるフィーチャ群を FM から除去するという手順を繰り返して、最終的なフィーチャの集合を求める。こうした意思決定は一般に高コストな作業であるが、SDPD はその回数を大きく減らすことに成功している。本稿ではこの意思決定の回数を導出ステップ数と呼び、モデル活用のコストの指標とする。なお本稿では AM も FM と同じ構造を持っているので、コンポーネント集合を求める際にも SDPD を適用する。

なお、実際の作業では手順 2 や手順 3 で失敗した際には手順 1 に戻って、他のフィーチャ構成候補があればそれを試すが、本稿ではそのための要素的な作業の特性を検討するために、上記の範囲の作業を実験の対象とする。

#### 4.2 フィーチャモデルの近似

本稿では、FM 側の近似化のみに焦点をあてる。また 2 つの近似化のうち、概略化だけを考える。フィーチャモデルの概略化は、直感的にはフィーチャ階層の下部を省略する近似化である。可変フィーチャのリストを与え、そのメンバの子孫フィーチャをすべて除去することで得られる。子孫が省略されたメンバには、ステレオタイプ《概略》をつける。この際、子孫フィーチャが参加する必要、排他といったクロスツリー制約も削除するものとする。図 4-5 は図 4-2 のフィーチャモデルにリスト (F2) を与えて概略化した例である。

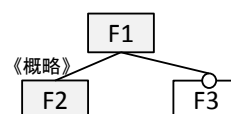


図 4-5 概略化の例

#### 4.3 近似的製品導出の方法

近似化されたフィーチャモデルを用いて行う製品導出を近似的製品導出と呼ぶ。近似的製品導出の手順は 4.1.4 で説明した手順と同様である。ただし近似により省略されたフィーチャが参加するトレーサビリティリンクは同様に省略される。図 4-4 のトレーサビリティリンクの場合、 $F3 \{C4\}$  のみとなる。

例えば、図 4-5 のフィーチャモデルと、このトレーサビリティリンクを用い、 $FS_0 = \{F3\}$  を指定すると、FM の制約を満たす集合は  $\{F3\}$  となり、 $VS_0 = \{C4\}$  となる。先ほどの例と同様に設計判断として C2 も必要と考えれば  $VS_0 = \{C2, C4\}$  となる。これらを含むコンポーネント構成としては、 $\{C2, C4, C7\}$  と  $\{C2, C4, C6, C7\}$  が考えられる。このどちらかを選んで共通コンポーネントを加えたものを近似的なコンポーネント構成 (近似製品) とする。この場合後者を選べば近似化しない場合と同じ製品構成が導出されることになるが、前者を選ぶと異なった製品構成となる。

#### 4.4 メリットとペナルティの指標

実際のソフトウェア開発で、意思決定を含んだ製品導出を行うためのコストや、近似製品を検討し必要に応じて求める製品へと修正するペナルティは状況によって大きく変わり得る。本稿ではそれらの議論のベースとするために、コストやペナルティに関する指標を設定する。

製品導出作業のコストの指標としては、前述した SDPD の導出ステップ数を用いる。4.1.4 で述べたように、製品導出が手順 2 や手順 3 で失敗した場合には、手順 1 に戻って他のフィーチャ構成を試すため、実際のステップ数はその試行錯誤によって大きくなるが、本稿では基本的な特性の検討として、手順 1 から手順 3 までを一度だけ行い、失敗

した場合には導出不可とする。導出不可の場合は実際にはその後試行錯誤が入るので大きなコストが発生することを意味する。

近似的製品導出によって得られる製品は必ずしも本来の意図を反映したものにはならない。上述の例では、本来は  $FS_0 = \{F3\}$  なので F3 が要求している F6 も選ばれるため、TL により C6 と C7 が最終的な製品に含まれる。一方、近似製品は 2 つの候補があり、C6 を含まないコンポーネント構成が導出される。従って、近似的製品導出のペナルティは、近似的製品が妥当かどうかを検討し、必要に応じてそれを修正する作業が発生することである。本稿では、その作業コストの指標として、近似製品が含むコンポーネントの数と、本来の製品中のコンポーネントの数との比(製品サイズ比)を用いる。例えば上述の例では本来の製品は 4 つだが、前者の近似製品はコンポーネントを 3 つ含むため 0.75、後者は 4 つ含むので 1 となる。もちろんこの値が 1 であっても、異なったコンポーネントを含んでいれば作業が発生するため、あくまでひとつの指標としての位置づけである。もしも同一の製品が得られたときには修正コストはかからないためペナルティ最小と考えられる。従って、以下の実験では同一かどうかの判断を合わせて行う。

なお本稿では、モデルを構築したり整合化をとったりするためのコストは考慮しない。本来近似的モデリングはそれらのコストの削減を狙ったものだが、本稿はあくまで製品導出作業にのみ焦点をあてる。従って、意図された本来のモデルは存在せずに近似モデルのみを作るのか、意図したモデルを作った後にそれを近似化して利用するのか、といった利用のコンテキストは本稿では議論しない。二種類のモデルを用いた製品導出作業の特性を調べることを目的とする。

## 5. 実験：近似的製品導出の特性

本章では、近似的構成管理のメリットとデメリットを検討するために行った実験の内容と結果について述べる。

### 5.1 目的

本来の意図したモデルを使った製品導出（以下、標準製品導出と呼ぶ）と、そのモデルを近似化したモデルを利用した近似的製品導出とを行い、その特性を調べる。具体的には FM、AM、TL、近似を行う際の指定フィーチャなどを変えた様々な状況を設定し、標準製品導出と、近似的製品導出を行って、4.4 で述べた指標を測る。

### 5.2 実験方法

今回の実験では、FM を固定して行った。図 5-1 が用いた FM である。この FM は FM 生成ツール BeTTY2)を用いて無機的に生成したもので、20 個のフィーチャを含んでいる。ハッチングのかかった部分が共通フィーチャ、かかっていない部分が可変フィーチャである。標準製品導出ではこの FM をそのまま用いた。一方近似的製品導出では、F2

を概略化した場合と、F8 を概略化した場合の二つのパターンについて実験した。前者を概略化するとフィーチャは 8 個（root を含む）、後者を概略化すると 16 個となり、それぞれ概略化の度合いが異なる状況に対応する。

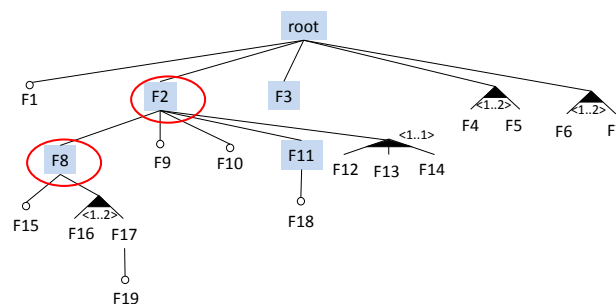


図 5-1 実験に用いた FM

AM は、メタモデルは FM と同じなので、同様に BeTTY を用いて生成した。AM は、FM との要素数のバランスを変えるために、コンポーネント数が 10, 20, 30 の 3 つのパターンについて生成した。以下それらをサイズ 10、サイズ 20、サイズ 30 と呼んで区別する。図 5-2 は本実験で用いたサイズ 10 の AM である。

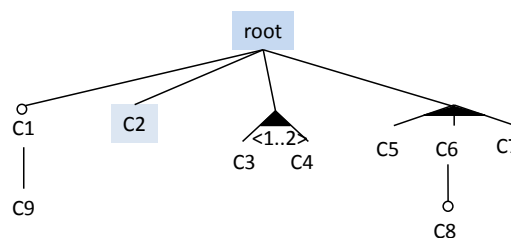


図 5-2 実験に用いた AM(サイズ 10)

TL は、FM と AM の間に、プログラムを用いてランダムに定義した。ただし全可変フィーチャ（コンポーネント）中のどれだけの可変フィーチャ（コンポーネント）に対してリンクを定義するかを比率（パーセンテージ）で指定できるようにした。実験では、フィーチャ側の比率と、コンポーネント側の比率を、100-50, 100-30, 50-100, 30-100 の 4 種類に設定した。サイズ 10 の AM に対して 100-50 の場合の TL の例を図 4-1 に示す。

F1→{C3}	F4→{C7}	F5→{C1}
F6→{C3}	F7→{C1}	F9→{C7}
F10→{C1}	F12→{C3}	F13→{C7}
F14→{C5}	F15→{C5}	F16→{C5}
F17→{C5}	F18→{C5}	F19→{C1}

図 5-3 実験に用いた TL(AM サイズ 10, 100-50)

SDPD に基づいて製品導出を行うツールを Java で実装し、

4.1.4 で説明した手順で製品導出を行った。導出においては、F1 を指定した導出、F4 を指定した導出、というように全可変フィーチャを一個ずつ指定して製品導出を繰り返した。ただし近似 FM 中で存在しないフィーチャは比較できないため、それは除外した。つまり、F2 を概略化した近似モデルでの近似的製品導出と比較する場合は F1, F4, F5, F6, F7 の 5 通り、F8 を概略化した近似モデルでの近似的製品導出と比較する場合は、F1, F4, F5, F6, F7, F9, F10, F12, F13, F14, F18 の 11 通りについてそれぞれ製品導出を行った。

それぞれのパターンについて、標準製品導出の場合と、近似的製品導出の場合について、4.4 で説明した導出ステップ数、導出不可数、製品サイズ比、同一製品数を調べた。

### 5.3 導出例

AM サイズ 10, TL 率 100-50 の場合に、F8 を近似した結果を表 5-1 に示す。

表 5-1 AM サイズ 10, 100-50, F8 近似の場合

指定フィーチャ	標準ステップ	近似ステップ	標準サイズ	近似サイズ	サイズ比	同一
F1	18	9	5	5	1	YES
F4	不可	不可	-	-	-	
F5	17	8	5	5	1	YES
F6	19	9	5	5	1	YES
F7	17	8	5	5	1	YES
F9	不可	不可	-	-	-	
F10	17	8	5	5	1	YES
F12	17	9	5	5	1	
F13	不可	8	-	5	-	
F14	17	8	5	5	1	YES
F18	17	8	5	5	1	YES

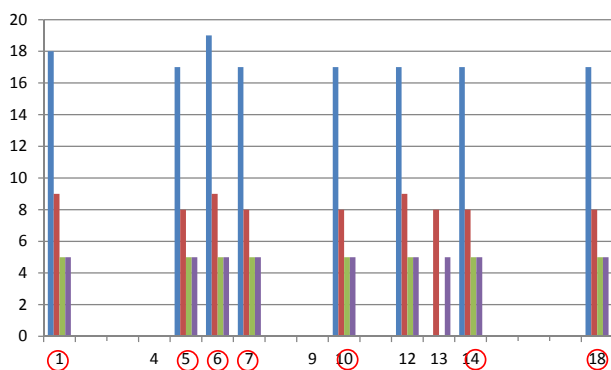


図 5-4 AM サイズ 10, 100-50, F8 近似の場合

表の列は左から、近似において指定したフィーチャ（指定フィーチャ）標準製品導出における導出ステップ数（標準ステップ）、近似的製品導出における導出ステップ数（近似ステップ）、標準製品導出における製品サイズ（標準サイズ）、近似的製品導出における製品サイズ（近似サイズ）、製品サイズの比（サイズ比）ならびにふたつの製品が同一

かどうか（同一）をそれぞれ示している。製品導出が失敗した場合はステップ数の欄には“不可”と記している。

このパターンにおいては近似によって導出ステップが約半分になっている。製品サイズは変わらず、また 7 つの場合で同一の製品が導出されている。また標準製品導出では導出が失敗しているのに、近似することで近似製品の導出に成功している場合（F13）もある。図 5-3 は、標準製品導出と近似的製品導出におけるステップ数と製品サイズをグラフで示したものである。横軸の数字は指定フィーチャに相当し、数字毎に示される 4 つの線は、左から表の標準ステップ、近似ステップ、標準サイズ、近似サイズを示している。数字に丸がついたものは同一製品であることを、グラフが無い部分は導出不可を示している。

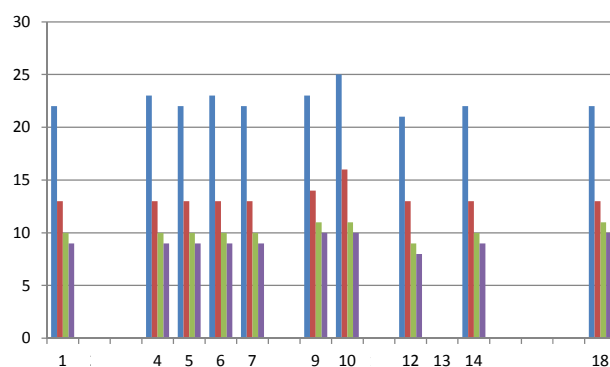


図 5-5 AM サイズ 20, 50-100, F8 近似の場合

図 5-5 は AM サイズ 20, TL 率 50-100 の場合に、F8 を近似した結果である。このパターンでは導出ステップ数が約 6 割になっている。製品サイズはやや小さくなっており、同一の製品はひとつも導出されていない。なおこのパターンにおいて実際に導出された近似製品は、標準製品に対してひとつだけコンポーネントが少ないものであった。また図 5-6 は AM サイズ 30, TL 率 30-100 の場合に F2 を近似した結果である。このパターンでも導出ステップ数は約 6 割であり、5 つとも同一製品だった。

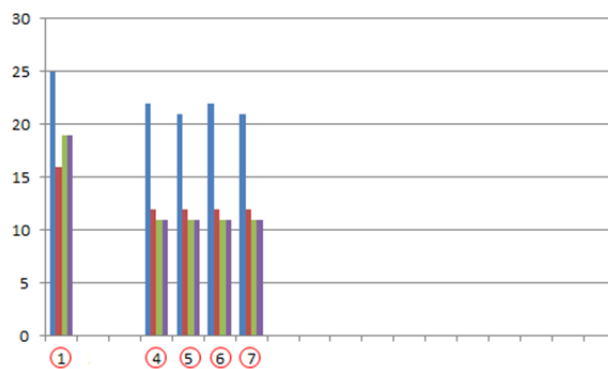


図 5-6 AM サイズ 30, 30-100, F2 近似の場合

## 5.4 実験結果

すべてのパターンにおける実験結果の概要を表 5-1 にまとめる。

表 5-2 全パターンの実験結果の概要

AM サイズ	リンク率	指定フィーチャ	不可	同一	ステップ比	サイズ比
10	100-50	F8	3-2	7	0.48	1
		F2	1-0	0	0.48	1
	100-30	F8	0-0	11	0.47	1
		F2	0-0	5	0.46	1
	50-100	F8	2-2	0	0.44	0.98
		F2	1-0	0	0.43	0.81
30-100	F8	0-0	11	0.46	1	
	F2	0-1	0	0.41	1	
20	100-50	F8	3-3	1	0.59	0.91
		F2	0-0	0	0.54	0.89
	100-30	F8	0-0	2	0.64	0.91
		F2	0-0	0	0.62	0.89
	50-100	F8	1-1	0	0.6	0.9
		F2	0-0	0	0.58	0.8
30-100	F8	2-1	0	0.56	0.78	
	F2	0-0	0	0.56	0.62	
30	100-50	F8	2-2	9	0.59	1
		F2	0-0	4	0.58	0.99
	100-30	F8	0-0	0	0.61	0.95
		F2	0-0	0	0.6	0.89
	50-100	F8	4-4	7	0.58	1
		F2	1-1	4	0.57	1
30-100	F8	3-3	8	0.58	1	
	F2	0-0	5	0.58	1	

表の左から 3 つの列は実験のパターンを示しており、左から、用いた AM のサイズ (AM サイズ)、TL のリンク率 (リンク率)、近似において指定したフィーチャ (指定フィーチャ) を表している。左から 4 つ目以降の列が、各パターンにおける実験結果であり、指定フィーチャが F8 の場合は前述した 11 通り、F4 の場合は 5 通りについて製品導出を試行した結果が示されている。4 列目 (不可) は、導出が不可能だった回数を示しており、標準製品導出での導出不可数をハイフンの左に、近似製品導出での導出不可数を右に示している。5 列目 (同一) は同一製品が導出された回数である。6 列目 (ステップ比) は、導出ステップ数が、近似化によってどれだけの比率になったかの平均、7 列目 (サイズ比) は、導出製品サイズが近似化によってどれだけの比率になったかの平均を示している。

今回の実験では、近似化により導出ステップ数は約半分になっている。一方製品サイズは約 9 割程度であり、また全導出試行数 192 回のうち、74 回で同一製品が導出された。また標準製品導出では導出不可だったが、近似的製品導出によって導出が可能となった場合が 4 回あったが、逆に標

準製品導出では導出できたのに、近似的製品導出では導出できなくなった場合も 1 回あった。

一方、様々なパターンについて製品導出を試行したが、パターンの違いによる傾向は十分には読み取れなかった。あえていえば同一製品が導出されるのは、FM と AM のサイズが違う場合に多く現れているようには見受けられる。

## 6. 議論

### 6.1 実験から得られる示唆

今回の実験結果を見る限り、近似的製品導出は一定の有効性を持つことが示唆されていると考える。

導出ステップ数は半減しており活用コストは大きく減少している。一方、約 1/3 において標準製品導出と同じ製品を導出できており、ペナルティがの少ない状況が一定割合で発生している。またすべての場合の詳細な確認はしていないが、前述した AM サイズ 20、リンク率 50-100、指定フィーチャ F8 のパターンのように、同一製品は生成されていないが、完全にサブセットで 1 つだけサイズの小さい製品が生成されており、この場合には同一ではないが修正ペナルティが比較的少ない場合と考えられる。

もちろん中にはサブセットでなくサイズの差以上に差分の大きな製品が導出されている場合もある。また近似によって製品導出ができなくなっているケースも一回だけはあるが発生しており、このように近似化がうまく機能していない場合もある。

今回の実験では、ステップ数の削減効果はパターンによってそれほど大きく変わっていない。一方、上記のペナルティが、どういう場合に大きくどういう場合に少ないのかという傾向ははっきりと表れていない。我々は過去の研究で、近似的モデリングが有効と考えられるいくつかの状況を定性的に推定し、本実験ではそれを踏まえて複数のパターンで製品導出を行ったが (8)13)、その傾向は捉えられていない。現実を利用するためには、近似的製品導出がどういう場合に有効かという指針が必要となるため、さらに調査が必要であると考えられる。

### 6.2 技術的考察

近似化しているのに約 1/3 において同一製品が導出できた理由については、さらに調査をしなければ正確な言及はできないが、ひとつの推定は、FM や AM は、そこから導出できるフィーチャ構成やコンポーネント構成に制約を持っているため、FM が近似化されて指定されるフィーチャや得られるフィーチャ構成が多少粗くなっても、結果的に同じ製品構成にいきついてしまうという理由が考えられる。

図 6-1 は、単純な状況を模式的に表したものである。FM 中で F1 を指定すると、フィーチャ構成 {F1, F2} が得られ、その結果製品 {C1, C2} が導出される。仮に FM が近似化されて F2 が削除されたとすると、F1 を指定するとフィーチャ構成 {F1} が得られ、TL を辿り {C1} が特定されるが、AM

側に制約があるため、その場合も同一の製品{C1, C2}が導出される。このように制約を持つモデル間の関係付けにおいては、若干の情報の欠落が補正される場合がある。こうした状況はFMとAMの間だけでなく、制約の種類は違っても様々なモデルにおいて発生し得る。こうした制約との関係を捉えることで効果的な近似的構成管理の活用ができる可能性が示唆されている。

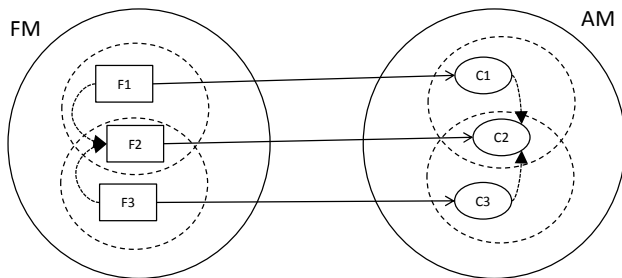


図 6-1 近似化の影響の考察

### 6.3 妥当性への脅威

本稿の実験のはまだ限定的なものである。今回いくつかのパターンについて導出を試行したが、FM、AM、TL などに関する変動因子や各因子の水準は多様であり、今回のパターンはそれらの一部しかカバーしていない。もちろんこれらを網羅的に考慮することはほとんど不可能であるため、方策が必要である。

FMやAMの生成にはBeTTyを用いたが、このツールは指定した条件を満たすモデルを無機的に生成する。そのためクロスツリー制約などの生成をしても、意味のない制約や、きつすぎる制約などが生成され、現実的なモデルとは言いづらい部分もある。したがって実際に使われているFMなどでの評価もあわせて必要かもしれない。

今回はSDPDに基づいて製品導出を行ったが、製品導出の方法は様々であり、SDPDを使うことの妥当性についても議論が必要である。またSDPDは複数のフィーチャ構成が得られるときにランダムにひとつを選ぶことになっており、選び方などツールの実装依存の部分がある。こうした点についての考慮も必要である。

## 7. おわりに

本稿では、我々の提案する近似的モデリングを用いた構成管理について検討した。具体的にはプロダクトライン開発における製品導出をとりあげ、近似化しないモデルを使った製品導出と、近似モデルを使った製品導出を比較し、近似的製品導出の持つ特性について実験を通して検討した。その結果、近似的製品導出の有効性が示唆された。今後さらに特性の調査を進めるとともに、それに基づいた効果的な近似的構成管理の方法を検討していきたい。

## 参考文献

- 1) Sheng Chen, Martin Erwig, “Optimizing the Product Derivation Process”, In Proceedings of 15th International Software Product Line Conference (SPLC2011), pp.35-44, 2011.
- 2) <http://www.isa.us.es/betty/>
- 3) <http://www.omg.org/mda/>.
- 4) <http://www.splc.net/>.
- 5) Hassan Gomaa, Designing Software Product Lines with UML, Addison-Wesley Professional, 2004.
- 6) ISO/IEC/IEEE 24765, System and software engineering – Vocabulary, 2010.
- 7) K. Kang, S. Cohen, J. Hess, W. Novak, A. Peterson, “Feature-Oriented Domain Analysis (FODA) Feasibility Study”, In Proceedings of Technical Report CMU/SEI-90-TR-21, Software Engineering Institute, Carnegie Mellon University, 1990.
- 8) 川嶋優樹, 岸知二: 可変性モデル間の製品パリエーション不均衡に関するメトリクスの提案, 情報処理学会 第76回全国大会, 2014(1). pp.435-436, 2014.
- 9) Tomoji Kishi and Kyo-Chul Kang: Scalable Modeling Techniques for Software Product Lines (SCALE 2009), proceedings of SPLC2009, p.299, 2009.
- 10) 岸知二: ソフトウェアモデル間のスケーラブルな整合化戦略について, ソフトウェア工学の基礎 XVIII 日本ソフトウェア科学会 FOSE 2011, pp.97-102, 2011.
- 11) 岸知二, 近似的モデリング技法についての考察, 情報処理学会 ソフトウェア工学研究会 研究報告, Vol.2012-SE-177, no.34, pp.1-7, 2012.
- 12) 岸知二: 近似的モデリングメカニズムについての考察, 情報処理学会 ソフトウェア工学研究会 研究報告, Vol.2013-SE-181, pp.1-7, 2013.
- 13) 岸知二, 川嶋優樹, 野田夏子: 近似的モデリングアーキテクチャに関する考察, 情報処理学会, ソフトウェア・エンジニアリング・シンポジウム(SSE2014), pp.152-157, 2014.
- 14) 永野寛丸, 岸知二: ソフトウェアプロダクトラインにおける非機能特性を考慮した製品導出支援手法の提案, 情報処理学会 ソフトウェア工学研究会 研究報告, Vol.2013-SE-182, no.13, pp.1-8, 2013.
- 15) Haugen, Øystein, Wąsowski Andrzej and Czarnecki, Krzysztof: CVL: common variability language, Proceedings of the 16th SPLC - Volume 2, pp.266-267, 2012.
- 16) OMG: Unified Modeling Language Specification, Version 2.4, 2010.
- 17) Norbert Siegmund, Martin Kuhlemann, Marko Rosenmüller, Christian Kästner, Gunter Saake, “Integrated Product Line Model for Semi-Automated Product Derivation Using Non-Functional Properties”, In Proceedings of the International Workshop of Variability Modelling of Software-Intensive Systems (VaMoS), pp.25-31, 2008