Vol.2017-SE-197 No.10
2017/11/10

ソフトウェアモデルにおける有用性優先についての考察

岸知二[†1]　野田夏子[†2]

概要：ソフトウェアモデルは大規模・複雑化しており，限られた時間でより効果的なモデリングをすることが重要である．我々は利用目的に照らして重要な部分の定義や洗練を優先する有用性優先のモデリング手法を検討している．本稿ではフィーチャモデルを可変性解決に利用する状況を想定し，有用性優先のモデリング手法について考察する．

On “Utility-First-ness” in Software Modeling

TOMOJI KISHI[†1]　NATSUKO NODA[†2]

Abstract: As software model becomes larger and more complicated, it is important to effectively define/refine model in a limited time. We are studying a modeling technique, in which we give priority to define/refine model elements that are important in terms of the usage of the model. In this paper, we examine the technique using an example of feature model that are used for variability resolution.

1. はじめに

ソフトウェアが大規模・複雑化し，変化が常態化することにより，ソフトウェアモデル（以下モデル）自体も同様の傾向を持つようになっている．また人間が読むドキュメントとしてだけでなく,マシン処理によるモデル駆動開発，形式検証などの活用においてはモデルの厳密性や詳細度が増してくる．その結果，モデルの定義や維持のコストも大きくなっている．

こうした背景の中，我々はモデルの妥当な記述とは何かについて検討している．妥当とは，素朴に言えば，過度に書き込みすぎたり，逆に書き込みが不足したりしていないということである．我々は書き込みが過度なのか不足しているかは目的依存であり，そのモデルがどういう目的で使われるのかを明らかにし，その利用目的に対する有用性の観点から判断することが重要と考える．本稿では，こうした有用性を高めることを優先した有用性優先のモデリング手法について検討を行う．

利用目的には，人間が読んで理解するといったものも考えられるが，こうした側面は属人性が高く客観的な議論が困難である．一方モデル駆動開発などのマシン処理による利用を目的として想定すると，その有用性についてある程度属人生を排除した議論が可能である．本稿ではこうした利用を想定し,有用性優先のモデリングについて議論する．具体的にはフィーチャモデル[9]（以下 FM）を SPL 開発[2]における可変性解決に利用する状況を題材に議論する．

我々は[12]において，可変性解決に関わる FM と成果物間のトレーサビリティリンクに基づき，その成果物の可変性解決に関わるモデル部分を識別し，構造および意味の観点から可変性解決におけるフィーチャ構成導出に対する有用性を議論した．本稿では一般に可変性解決が複数の結合タイミングで分散して行われる点に注目し，FM のモデル部分が，それらのうちどれだけの結合タイミングでの可変性解決に関わるかという観点から有用性を議論する．

以下, 2 章では有用性優先のモデリングとは何かを述べ，本稿の着眼点を述べる．3 章では可変性解決と結合タイミングについて説明する．4 章では可変性解決と FM との関わりについて，5 章ではそれに基づく有用性優先のモデリングについて考察する．6 章では関連する議論を行い，7 章で本稿を締めくくる．

2. 有用性優先と着眼点

2.1 有用性優先のモデリング

有用性優先のモデリングとは，利用目的に関わるあるいは影響の大きなモデル部分の定義や洗練を，利用目的に関わらないあるいは影響の少ないモデル部分よりも優先して行うモデリングであるとする．ここで利用目的には可変性解決，モデル変換，テストケース生成など様々なものが含まれる．人間の理解，ヒューマンコミュニケーションといった利用目的も重要であるが，本稿ではマシン処理あるいは手続き的な処理に基づく利用目的に限定する．

有用性優先の考え方は，アジャイル開発などにおいて，

†1 早稲田大学
Waseda University
†2 芝浦工業大学
Shibaura Institute of Technology

タイムボックスの中でバックログに優先度をつける考え方と類似した考え方であり，限られたリソースの中で，過度の完成度を目指さずそれなりに有用（well enough）[5]な記述に抑えるという意味合いがある．

2.2 着眼点

モデル中のモデル部分（モデル要素あるいはその集合）すべてが同じように利用目的に貢献するわけではない．利用目的に大きく影響するモデル部分もあれば，相対的に影響が少ないモデル部分，あるいはまったく使われないモデル部分などもありうる．そうした利用目的に対するモデル部分の影響度を理解できれば，影響度の大きなモデル部分の定義や洗練を優先することが有用性を高めることになる．特にモデルがソフトウェア開発のある時点だけで利用されるのでなく，時間軸に沿った複数の時点で利用される場合には，モデルの静的な構造だけでなく，利用の順序性による影響を考慮する必要がある．こうした影響度に関する何らかの指標を求めることができれば，優先度優先のモデリングに資することができる．

3. 可変性解決と結合タイミング

3.1 フィーチャモデルと可変性解決

本稿では FM を対象に，SPL 開発などで行われる可変性解決を利用目的とした場合の有用性優先について検討する．

可変性とは製品系列中で製品によって異なり得る機能的・非機能的な特徴をいう[6]．例えば携帯電話で，製品によって画面解像度が高画質，標準，低画質と異なりうる場合，画面解像度は可変性として捉えられる．可変性の表現やモデル化には様々な方法があるが，本稿では FM を利用するものとする．図 1 に FM の記述例を示す．

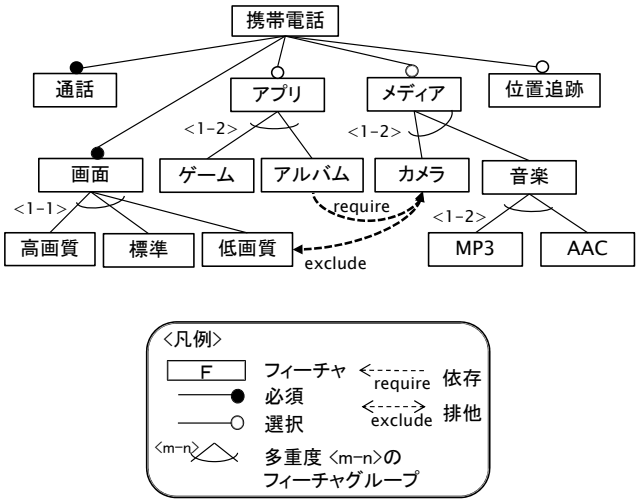

図 1　FM の記述例

この FM では画面解像度，アプリ種類，メディア種類，音楽フォーマット，位置追跡有無に関わる可変性が表現されている．図 2 は可変性と対応するモデル部分（フィーチャ群）を示したものである．なおモデル部分には，フィーチャ以外にもフィーチャ間の関係を示すモデル要素も含まれるが表では省略している．

図 2　可変性とフィーチャ（群）との対応

可変性	対応するフィーチャ群
画面解像度	画面，高画質，標準，低画質
アプリ種類	アプリ，ゲーム，アルバム
メディア種類	メディア，カメラ，音楽
音楽フォーマット	音楽，MP3，AAC
位置追跡有無	位置追跡

一方，成果物中で可変性に対応する部分を可変点と呼び，可変性のバリエーションに応じた可変点の実現（成果物要素など）をバリアントと呼ぶ．例えばコード中でコンパイルスイッチによって高画質，標準，低画質に応じたコードを切り替えるようになっている場合，この個所を可変点，それぞれのコードをバリアントと呼ぶ．図 3 は可変点を含んだソースコードの例であり，画面解像度のバリエーションに応じた 3 つのバリアントが定義されている．

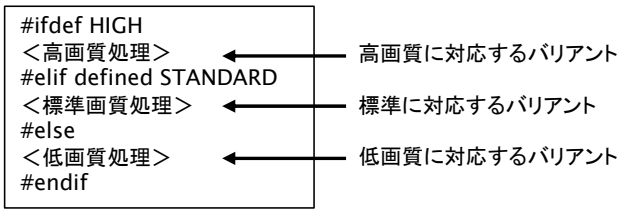

図 3　可変点を含んだソースコードの例

可変性解決とは，特定の製品を開発する際に，成果物中の可変点にどのバリアントを適用するかを決定する作業である．上記の例では，コンパイル時にマクロを定義して特定のコードをコンパイル対象に選択する作業が対応する．

3.2 結合タイミング

開発において可変性解決が行われるタイミングを結合タイミングと呼ぶ．表 1 はいくつかの結合タイミングやその実現手段のいくつかの例を示したものである．

表 1　結合タイミングと実現手段の例

結合タイミング	実現手段
設計時	クラスの継承
製造時（コンパイル時）	コンパイルスイッチ
出荷時	パッケージング
起動時	設定ファイル，起動オプション
利用時	モード設定

一般に製品系列は複数の可変性を持つ．開発の特定のタイミング（例えば製造時）にそれらすべての可変性解決がなされることもありうるが，いくつかの開発上のタイミングに分散して行われることも多い．これは技術的な理由によることもあるが，ビジネス的な視点から結合タイミングを決めることが重要だからである[10].

例えば製造時と起動時の二つの結合タイミングを考える．製造時にコンパイルスイッチを利用すると製品毎に異なったバイナリができ管理は煩雑だが，起動時には単にそのバイナリを起動すればよい．一方起動時に設定ファイルを利用する際には，バイナリはひとつで済むがフィールドサポートが適切に設定ファイルの記述を行う必要がある．そのどちらが適切かは， 想定される製品種類数，開発者・管理者・フィールドサポートの負担や技術レベル，あるいは販売や支援の体制などによって変わる．従ってビジネス形態や状況に応じて，適切な結合タイミングを決めることが重要となる．

表 2 は, 図 1 の FM で示される可変性に関する可変性解決の方針例である．また各行の右にはその方針に該当する可変性を記述している．

表 2　可変性解決の方針例

	方針	対応する可変性
製造時	搭載するハードウェアに関係するものは製造時に決める	画面解像度 メディア種類
出荷時	販促キャンペーンなどで変更しうるものは出荷時に決める	アプリ種類
利用時	ユーザの嗜好やポリシーに依存するものは利用時に決める	音楽フォーマット 位置追跡有無

結合タイミングに時間的な順序性があるとすると，上記の方針付けは，可変性解決の間に図 4 に示すような制約を導入することになる．

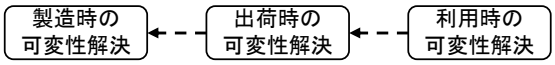

図 4　可変性解決間の依存性

例えば出荷時にアプリ種類の可変性を解決する際には，製造時の画面解像度とメディア種類の可変性解決は終わっているので，その結果の影響を受けることになる．製造時にカメラが選択されていなければ，出荷時にアルバムを選択することはできない．つまり FM 中の制約と可変性解決の順序性との二つの観点から FM の構造などについて検討することが重要となる．

4. 可変性解決と FM との関わり

4.1 可変性解決と FM

可変性解決においては，FM 上で対応するフィーチャの選択・非選択がなされ，それに応じて可変点でのバリアントが決定される．FM 上では必須，選択，あるいは依存，排他などのフィーチャ間の制約が定義されているため，これらの選択・非選択はそれらの制約を満たしながら行われなければならない．こうしたフィーチャの構成（フィーチャへの選択・非選択の割り当て）を決める作業をフィーチャ構成導出と呼ぶ．

特定の製品を開発する際にすべてのフィーチャに関わる選択・非選択の決定を一度に行い，それに基づいて各結合タイミングで可変性解決を行うということも行われ得る．一方，各結合タイミングではそこでの可変性解決に関わるフィーチャに対する選択・非選択を決め，その後の可変性解決に関わるフィーチャ群の選択・非選択の決定を保留することもある．上記の例では，出荷時に販促キャンペーンなどで搭載するアプリを変更する余地を残すならば，製造時にはそれらの判断を保留する必要があるし，ユーザの嗜好やポリシーに関わる決定は利用時までできない．このように，フィーチャの選択・非選択の決定はインクリメンタルに進められうる．

4.2 結合タイミングに応じたモデル部分

結合タイミングでどの可変性に関わるフィーチャの選択・非選択を行うかによって，各結合タイミングでの可変性解決で利用される FM のモデル部分は変わる．ここで利用されるモデル部分とは，1)その結合タイミングで解決される可変性を直接的に表現しているモデル要素，および 2)依存，排他などの関係や，それらの関係で接続されているフィーチャなど，可変性を直接的に表現しているモデル要素の選択・非選択に影響を及ぼし得るモデル要素をいう．

図 5, 図 6, 図 7 は, それぞれ表 2 の方針に沿った場合，製造時，出荷時，利用時での可変性解決に関わるモデル部分を示したものである．

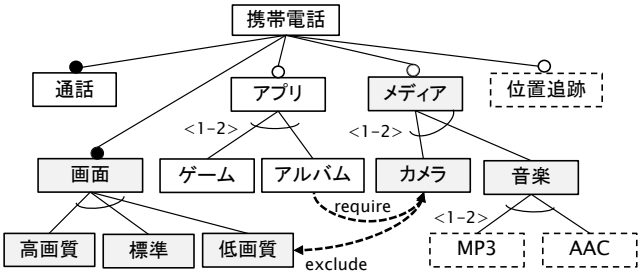

図 5　製造時の可変性解決に関わる FM 部分

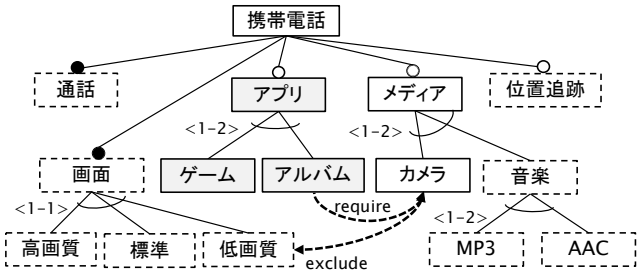

図 6　出荷時の可変性解決に関わる FM 部分

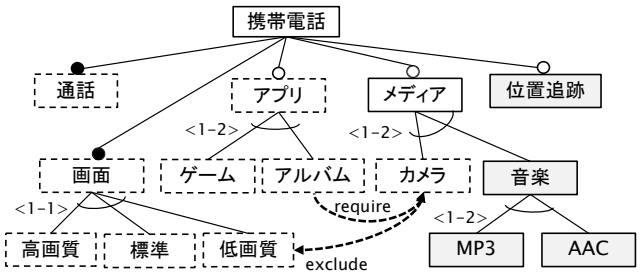

図 7　利用時の可変性解決に関わる FM 部分

ここではその結合タイミングで解決される可変性を直接表現しているフィーチャはグレー，考慮しなければならないフィーチャは白，考慮不要のフィーチャは破線で示している．フィーチャ以外のモデル要素についてはフィーチャから判断できると考え，図では特に区別をしていない．なおそれ以降の結合タイミングでのフィーチャ選択を考慮することもあり得るが，図ではその影響は示していない．

5. 有用性優先モデリングへの活用

5.1 モデル部分の重要度

FM が可変性解決におけるフィーチャ構成導出という利用目的で使われるとし，その利用目的に照らしたモデル部分の重要度を考える指標として，モデル部分を以下の 3 種類に分類する．

- 可変性解決に利用されないモデル部分
- 可変性解決に利用されるが，結合タイミングをまたがった影響を持たないモデル部分
- 可変性解決に利用され，結合タイミングをまたがった影響をもつモデル部分

表 3 は，各フィーチャが可変性解決に利用されているかどうかを示したものである．ここで◎は可変性を直接表現するフィーチャ，○は考慮が必要かもしれないフィーチャ，無印は利用されないフィーチャである．なお「携帯電話」はすべてに関わるが，これはルートフィーチャであり必ず必要なモデル要素なので除外している．

表 3　可変性解決に利用されるフィーチャ

	製造	出荷	利用	スコア
通話				0
画面	◎	○		2
高画質	◎			1
標準	◎			2
低画質	◎	○		2
アプリ		◎		1
ゲーム		◎		1
アルバム		◎		1
メディア	◎	○	○	3
カメラ	◎	○		2
音楽	◎		◎	2
MP3			◎	1
AAC			◎	1
位置追跡			◎	1

「通話」はどの可変性にも関わっていないため，可変性解決には利用されない．したがって可変性解決という利用目的に対しては重要度が低いことを示す指標を付与する．

その他のフィーチャは可変性解決に利用されているが，「高画質」,「標準」,「アプリ」,「ゲーム」,「アルバム」,「MP3」,「AAC」,「位置追跡」は，ひとつの結合タイミングの可変性解決にのみ参加しており，結合タイミングをまたがった影響を持たない．その影響は相対的に限定的となるため，中程度の重要度を示す指標を付与する．

「画面」,「低画質」,「メディア」,「カメラ」,「音楽」は，結合タイミングをまたがった影響をもつため，重要度が高いという指標を付与する．特にメディア種類に関わる部分はすべての結合タイミングでの可変性解決に影響しており，最も重要度が高い指標を付与する．

こうした指標にどのようなスコアを与えることが妥当かは要検討だが，ここでは表の右の列に，利用される結合タイミングの数をスコアとして示している．

5.2 有用性優先への活用

有用性優先のモデリングは，利用目的に関わるあるいは影響の大きなモデル部分の定義や洗練を優先する考え方であるから，前節での検討を有用性優先モデリングの戦略に活用することができる．活用は開発の状況やコンテキストに強く依存するが，典型的な状況として以下の状況を考える．

- 特定の結合タイミングにおける可変性解決での FM の利用のみを考える状況
- 複数の結合タイミングにおける可変性解決での FM の利用全体を考える状況

前者は，当面の利用を考えて，そのために必要なモデリ

ングを行う状況である．例えばこれから製造時での可変性解決が行われるときに，将来の出荷時や利用時のことはさておき，製造時での可変性解決にむけて関わる部分の定義・洗練を行う場合に相当する．この場合には，その結合タイミングで可変性解決を行う予定の可変性に関わるモデル部分に注力することが有用性優先に沿うことになる．製造時であれば，画面解像度やメディア種類に関わる部分に注力することになる．

後者は，想定される今後の利用を見据えてモデリングを行う状況である．この場合には，すべての可変性を考慮する必要があるが，特に複数の結合タイミングでの可変性解決に関わるモデル部分の定義や洗練に注力することが重要になると考えられる．ここではスコアが 2 以上のフィーチャに相当する．これらのモデル部分は，ある結合タイミングの可変性解決の結果が，後続の結合タイミングの可変性解決に影響を及ぼし得るので注意深くモデリングする必要がある．前述したようにメディア種類部分の定義は，広範な影響を持つため，表 2 で示したような可変性解決の方針に照らして FM が妥当かどうか十分に考えることが必要である．

6. 議論

本稿では，有用性優先モデリングについて，可変性解決における FM の利用を例に．モデル部分に対して利用目的に対する重要度の指標を付与し，それを活用する考え方について検討した．重要度は意味的な観点，主観的な観点など多様な観点から検討する必要があるが，本稿ではモデルの機械的な利用を想定し，主として構造的な観点から重要度を指標化する考え方を示した．特に本稿では，モデルが利用される時間軸を考慮し，時間の前後関係によって生じる影響を踏まえた検討を行った．なお目的に関わるモデル部分の特定には，近似的な方法[11]などを用いることも有用と考える．

複数の結合タイミングでの可変性解決に関わるモデル部分はより重要であるという考え方は，ソフトウェアアーキテクチャ評価手法である ATAM[7]でのセンシティビティポイントやトレードオフポイントと類似した考え方である．前者は重要な品質への影響を持つソフトウェア構造上の決定（個所），後者は複数の品質への相反した影響を持つアーキテクチャ上の決定をいう．ATAM ではソフトウェア構造中でこれらの個所はアーキテクチャ上の重要性を持つ部分でありビジネス目的に大きな影響をもつため特に慎重な検討が必要といわれている．FM においても複数の利用目的へ影響を及ぼす部分はより優先して定義・洗練を行うべきと考える．別の言い方をすると，これらは FM 中のアーキテクチャ上の重要性を持つ個所といえる．

フィーチャ構成導出をインクリメンタルに行う方法としては，段階的構成[3]やマルチプロダクトライン[4]などがある．段階的構成はフィーチャ構成導出の手法を提案するもので FM のモデリングに言及するものではない．一方マルチプロダクトラインは当初より FM を分割して構築するといった方法がとられることもある．一方本稿では，必ずしも当初よりそうした分割を行っていることを想定しない．FM を作り始めた時点で可変性解決の方針は決まっていないこともあるし，製品によって可変性解決のタイミングが異なることもある．結合タイミングをまたがった影響を持つモデル部分を洗練するにあたっては，FM のリファクタリングをし，結合タイミングをまたがった影響を理解しやすくすることなども必要になると考える．

FM のモジュール化議論もされているが[1][8]，その際に FM の構造を静的にみるだけでなく，結合タイミングなど時間軸にそった観点からみることで，その利用コンテキストに応じた適切なモジュール化が行える可能性がある．

7. おわりに

本稿での議論は有用性優先モデリングに関する初期の検討結果のひとつである．有用性優先モデリングに向けてはこれ以外にも様々な観点からの考察が必要であり，今後さらに検討を進めたい．

参考文献

[1] M. Boskovic, G. Mussbacher, E. Bagheri, D. Amyot, D. Gasevic and M. Hatala: Aspect-oriented Feature Models, In Porc. MODELS'10, pp.110-124, 2010.
[2] P. Clements, and L. Northrop: Software Product Lines: Practices and Patterns. Addison-Wesley, 2001.
[3] K. Czarnecki, S. Helsen and U.W. Eisenecker: Staged configuration through specialization and multilevel configuration of feature models. Software Process: Improvement and Practice, 10(2), pp.143–169, 2005.
[4] D. Dhungana, D. Seichter, G. Goetterweck, R. Rabiser, P. Grünbacher, D. Benavides and J. Galindo: Configuration of multi product lines by bridging heterogeneous variability modeling approaches. Proceedings of SPLC 2011, pp.120-129, 2011.
[5] D. Garlan: Software Engineering in an Uncertain World, FoSER'10, pp.125-128, 2010.
[6] ISO/IEC, ISO/IEC 26550, Reference Model for Product Line Engineering and Management 2015.
[7] R.Kazman, M. Klein and P. Clements: ATAM: Method for Architecture Evaluation, CMU/SEI-2000-TR-004, 2000.
[8] C. Kaestner, S. Apel and K. Ostermann: The Road to Feature Modularity?, in Proc. SPLC 2011, volume 2, article No.5, 2011.
[9] K. Kang, S. Cohen, J. Hess, A.W. Novak and S. Peterson: Feature-oriented domain analysis (FODA) feasibility study. CMU/SEI-90-TR-21, 1990.
[10] K.Kang, J. Lee and P. Donohoe, Feature-Oriented Product Line Engineering, IEEE Software, Juley/August 2002.
[11] 岸知二, 野田夏子: 近似化によるフィーチャモデルからの製品導出法, KBSE2016-41, pp.13-18, 2016.
[12] 岸知二, 野田夏子: フィーチャモデルの記述の妥当性に関する考察, FOSE 2017, (to appear), 2017.