

複数視点によるプロダクトラインモデル化方法の提案

M2009MM011 近藤 貴大

指導教員 青山 幹雄

1. はじめに

SPL(Software Product Line Engineering)が注目されている[4]が、複雑なプロダクトのモデル化が困難である。本稿では複数視点によるプロダクトライン(PL)のモデル化方法を提案する。ゴール指向に基づきゴールグラフを拡張し、可変性を分析する。複数視点間の関係を定義し、複数モデルの可変性を OVM(Orthogonal Variability Model)で表現する。自動車ソフトウェアへ提案方法を適用し、有効性を示す。

2. 自動車ソフトウェアへの PL の適用課題

(1) 機能要求と非機能要求の混在

フィーチャ図は機能要求と非機能要求が混在しており、多くの情報を同時に表現しているため分析が困難である。

(2) フィーチャの必要理由の表現が困難

フィーチャ図からどのように目標を達成するか、なぜフィーチャが必要かの判断が難しい。

(3) 限定的なプロダクト可変性

UML はシステムを機能、構造、振舞いの 3 つの視点に分けて表現可能である。しかし、PL の共通/可変性を表現するフィーチャは視点が分離されていない。そのため、開発に適用する際に UML との対応づけが困難である。

本稿では PL 要求定義にゴール指向を適用し、機能要求と非機能要求を分離する。モデル間の関係を分析し、複数視点に分離して PL をモデル化する方法を提案する(図 1)。



図 1 PL の視点の分離

3. 関連研究

(1) OVM

開発成果物間の可変性を表現可能なモデリング手法として OVM がある[1]。OVM では可変点と変異体の概念を導入する。可変点は PL の何が変化するか、変異体は可変点がどのように変化するかを表現する。

(2) PL 要求モデルへのゴール指向の適用

PL 要求定義にゴール指向を適用し、ゴールとフィーチャ図の機能、非機能要求を対応づける研究がある[2]が、複数視点間の関係は定義されていない。例えば、品質など非機能要求と、対応する振舞いモデルの関係は未定義である。

4. 複数視点による PL モデル化方法の提案

4.1. プロダクトラインの基礎概念

4.1.1. 複数視点のモデルの表記法

複数視点として Goal/Softgoal, Operational, Functional, Structural, Behavior を導入する。複数視点を表現する表記法をゴールグラフと UML2.0 に基づき分類する(表 1)。

表 1 複数視点を表現する表記法

視点	表記法	表現内容
Goal/Softgoal	ゴールグラフ	システムが達成すべきゴール
Operational	ユースケース図/シナリオ	アクタとシステムの相互作用
Functional	フィーチャ図	フィーチャの集合
Structural	クラス図	システムの論理構造
Behavior	シーケンス図	システムの振舞い

4.1.2. ゴールグラフの拡張

本稿ではゴールを以下の 3 つに分類する。

(G-1) 戦略ゴール：ステークホルダや利害関係者の要求に基づき、ビジネス戦略上何を達成したいか。

(G-2) ソフトゴール：戦略ゴールを達成するために実現したい状態や状況を表す非機能要求の定性的属性。

(G-3) ハードゴール：ソフトゴールを達成するために実現したい状態や状況を表す非機能要求の定量的属性。

さらに、モデル間の関係を厳密にモデル化するため、機能要求を表現するタスクを以下の 3 つに分類する。

(T-1) システムタスク：開発対象となるシステム。

(T-2) 操作タスク：システムタスクの実現手段。

(T-3) ユーザタスク：ユーザが操作するタスク。

4.2. 複数視点からの PL モデル化方法

4.2.1. モデル間の関係の定義

PL を複数視点に分けて図 2 に示すモデルで表現する。複数視点を表現するモデル間には関係がある。例えば、Behavior 視点のシーケンス図は、Operational 視点のシナリオに基づき記述される。モデル間の関係を分析することで、複数視点による PL のモデル化が可能となる。

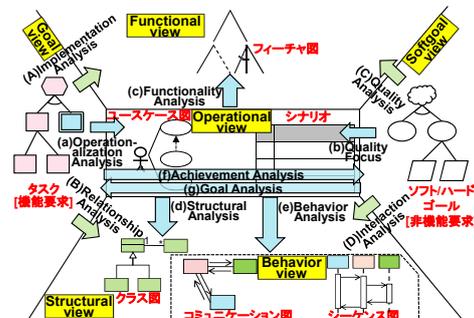


図 2 複数視点による PL のモデル化

モデル間の関係には直接他のモデルと関係する場合と、他のモデルを介して間接的に関係する場合がある。モデル間の関係を以下のように分類する。

(1) 直接的関係

モデルが直接他のモデルに関係する。例えば、Goal Analysisで抽出されたタスクは、ユースケース図のユースケースとアクタに關係する。直接的關係を以下に分類する。

(a) Operationalization Analysis

4.1.2 で定義したタスクよりユースケース図の各構成要素との対応關係を分析する(図 3)。システムタスクは開発対象システムを表すため、ユースケース図の分析対象システムと対応する。操作タスクはシステムタスクの実現手段であるため、ユースケースとアクタに分解される。操作タスクがユーザタスクである場合、アクタはユーザとなる。

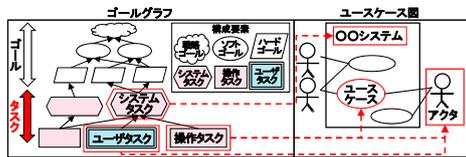


図 3 ゴールグラフとユースケース図の関係

(b) Quality Focus

ソフト/ハードゴールはシステム制約として考慮される。

(c) Functionality Analysis

ユースケース図のユースケースは機能要求であるため、フィーチャ図のレイバリティ層と対応する。アクタはその性質によりフィーチャ図の各層に対応する。例えば、アクタが外部システムならばフィーチャ図のレイバリティ層と対応し、ハードウェアはオペレーティング環境層と対応する。

(d) Structural Analysis

開発対象システムの静的構造を分析する。

(e) Behavior Analysis

ユースケース毎のシナリオからシーケンス図は作成されるため、ユースケース図のユースケースはシーケンス図全体と対応する。ユースケースに関連するアクタは、シーケンス図内のオブジェクトと対応する。また、シナリオのフローはシーケンス図のメッセージと対応する。ユースケース図/シナリオとシーケンス図の対応關係を図 4 に示す。



図 4 ユースケース図/シナリオとシーケンス図の関係

(f) Achievement Analysis

タスクがどのような理由で必要となるかを分析する。ゴール間の關係において、ゴールからみたサブゴールは実現手段、サブゴールからみたゴールは理由を表しており、ゴールを上に出ることでタスクが必要な理由を分析できる。

(g) Goal Analysis

ゴール分析し、どのように目標を達成するかを分析する。

(2) 間接的關係

モデルが別のモデルを介して關係する。例えば、ユースケース図を結合点としてゴールグラフのタスクとフィーチャ図の機能要求は關係する。間接的關係を以下に分類する。

(A) Implementation Analysis

タスクがどのような機能、ハードウェアなどで実現されるかを分析する。

(B) Relationship Analysis

タスクがどのような論理構造で実現されるかを分析する。

(C) Quality Analysis

ゴールで表現される非機能要求が、どのような機能、ハードウェアで実現されるかを分析する。

(D) Interaction Analysis

ゴールで表現される非機能要求がどのような振舞いで実現されるかを分析する。Operationalization AnalysisとQuality Focus, Behavior Analysisを通じて、非機能要求とその振舞いに関して相互作用分析が可能となる(図 5)。

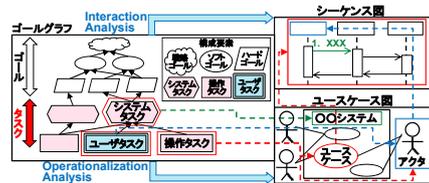


図 5 非機能要求と対応する振舞いの關係

4.2.2. ゴールグラフによる可変性抽出

ゴールグラフの構造に着目することで、要求の可変性を抽出できる。あるノードのゴールの親ゴールが複数存在する場合、そのゴールは複数の親ゴールの共通ゴールとなる。一方、親ゴールが一つの場合、そのゴールは固有のゴールであり可変ゴールとなる。タスクもゴールと同じ木構造であるため、同様に可変性を抽出できる。複数のシステムタスクに対して詳細化を行い、構造の共通性/可変性を分析することで機能要求の可変性を抽出できる。

4.2.3. OVM の導出

以下に従いゴールグラフから OVM を導出する(図 6)。

(1) 可変点と変異体の決定

4.2.2 に基づきゴールとサブゴールの關係に着目する。ゴールを可変点とし、そのゴールのサブゴールのうち可変ゴールを変異体とする。サブゴール以下の關係も同様に分析し、可変点と変異体を決定する。

(2) 可變性依存關係の決定

ゴールとサブゴールの關係が AND 分解の場合、ゴールの実現に全てのサブゴールが達成される必要がある。そのため、可變性依存關係は必須を表す mandatory とする。一方、OR 分解はゴールの実現にいずれかのサブゴールが達成されれば良いため、alternative もしくは optional とする。

(3) 制約依存關係の決定

ゴールが可變点かつ変異体の場合、変異体選択時に可變点を分析対象とするため、可變点への require とする。

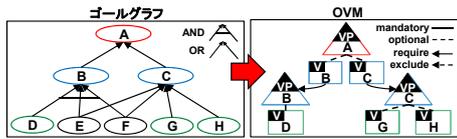


図 6 ゴールグラフからの OVM 導出

4.2.4. 複数視点による分析プロセス

モデルの変更は影響する他モデルに反映されなければならぬ。例えば、シナリオの変更はシーケンス図に反映される必要がある。図 7 に示す複数視点による分析プロセスを定義することで、モデル間の一貫性を保証できる。

非機能要求はシステム制約を表すため振舞いと関わりが強い。そこで、本稿では(1), (2), (3), (6), (7)の分析を行う。

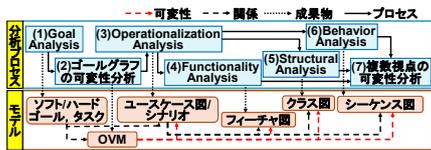


図 7 複数視点による分析プロセス

(1) Goal Analysis

戦略ゴールからソフトゴール、ハードゴール、タスクと段階的に詳細化し、4.1.2 に基づきタスクを分類する。

(2) ゴールグラフの可変性分析

ゴールグラフの構造を分析し、可変性を分析する。ユースケース図の可変性を表現するため、タスクを分析する。

(3) Operationalization Analysis

タスクに基づきユースケース図を記述し、ユースケース毎にシナリオを作成する。

(6) Behavior Analysis

システムの相互作用を分析しシーケンス図を作成する。

(7) 複数視点の可変性分析

OVM を用いて複数視点の可変性を表現する。

5. 提案モデル化方法の適用

提案モデル化方法を自動車ソフトウェアへ適用する。

5.1. Goal Analysis

「安全性向上」を戦略ゴールとし、手段分解の観点からゴール分析する。作成したゴールグラフを図 8 に示す。

実際の自動車の ACC(Adaptive Cruise Control System)は、定速走行システムに先行車追従システムの機能を拡張したシステムであるため、この 2 システムに着目し分析する。

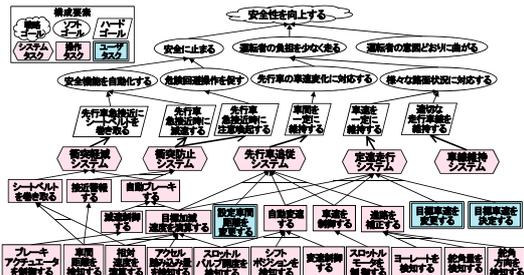


図 8 安全性向上に対するゴールグラフ

5.2. ゴールグラフの可変性分析

(1) 可変点と変異体の決定

可変点を「安全性向上」、変異体を先行車追従システムと定速走行システムに決定した。さらに、「設定車間距離変更」と「進路補正」、「目標車速変更」と「目標車速決定」は両システムを可変点とする可変ゴールであり、変異体と決定した。

(2) 可変性依存関係の決定

「安全性向上」からみた先行車追従システムと定速走行システムは OR 分解であるため、可変性依存関係を optional とした。また、両システムのサブゴールは共に AND 分解であるため、可変性依存関係を mandatory と決定した。

(3) 制約依存関係の決定

先行車追従システムは可変点かつ変異体であるため、変異体「先行車追従システム」から可変点「先行車追従システム」への require とした。定速走行システムも同様である。

また、定速走行システムは先行車を認識する必要がないため、「車間距離検知」と「相対速度演算」は定速走行システム選択時に除外される可変性として表現した。

ゴールグラフから作成した OVM を図 9 に示す。

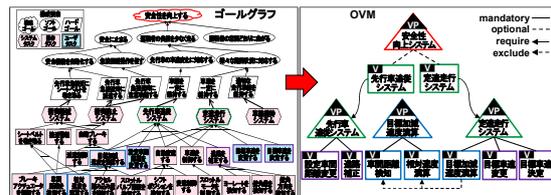


図 9 ゴールグラフから作成した OVM

5.3. Operationalization Analysis

先行車追従システムの「目標加減速度演算」、「自動変速」、「車速制御」、「進路補正」はユースケースと対応する。また、「設定車間距離変更」と対応するアクタはユーザとなる。「車間距離検知」などハードウェアに割り当て可能なタスクはアクタと対応する。定速走行システムも同様に対応づけを行う。ゴールグラフから作成した先行車追従システムと定速走行システムのユースケース図を図 10 に示す。

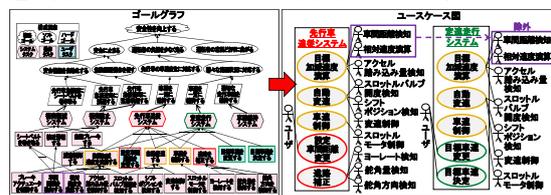


図 10 ゴールグラフから作成したユースケース図

5.4. Behavior Analysis

シナリオに基づき、目標加減速度シーケンス図を作成した。目標加減速度演算ユースケースは目標加減速度演算シーケンス図全体と対応する。また、仕様書[5]より「車間距離検知」と「相対速度演算」はミリ波レーダセンサで実現されると判明した。また、シナリオの各フローはシーケンス図のメッセージと対応する。モデル間の関係からユースケース図/シナリオとシーケンス図を対応づけた(図 11)。

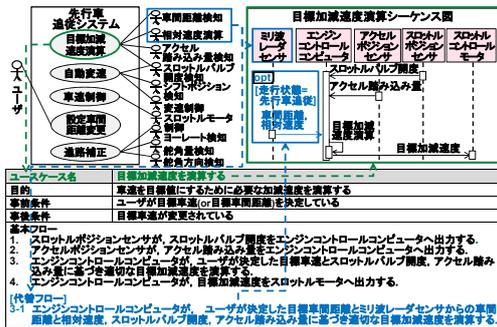


図 11 ユースケース図/シナリオとシーケンス図の対応づけ

5.5. 複数視点の可変性分析

ゴール実現に向け PL 開発を行い、先行車追従システムと定速走行システムのうち、優先度の高い定速走行システムを開発する状況を想定する。両システムの共通ゴールに加え、「目標車速変更」と「目標車速決定」の必要性を認識した。また、「目標車間距離検知」と「相対速度演算」はユースケース図と影響するモデルから除外される(図 12)。

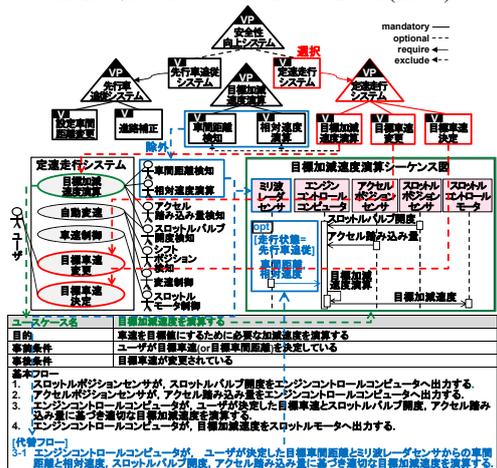


図 12 複数視点の可変性表現

6. 提案するモデル化方法の評価

ゴールグラフ、フィーチャ図、提案方法によるモデル化を以下に示す観点から比較、評価した(表 2)。

表 2 自動車ソフトウェアによるモデル化の評価

評価の観点	評価項目	ゴールグラフによるモデル化	フィーチャ図によるモデル化	提案方法によるモデル化
要求仕様の品質特性	非あいまい性	要素間の依存関係の特定が困難	要素間の依存関係の特定が困難	OMV による依存関係の特定が容易
	追跡可能性	ゴールグラフに限定的	フィーチャの必要判断が困難	複数モデルの後方追跡が可能
モデルの表現能力	機能要求/非機能要求	分割方法が不統一	機能/非機能が混在し分析が困難	段階的詳細化による体系的分離が可能
	可変性	可変性を扱わない	フィーチャと可変性が混在	可変性を分離して表現可能

6.1. 要求仕様の品質特性

IEEE830-1998[3]の品質特性に着目し評価する。

(1) 非あいまい性

大規模なゴールグラフやフィーチャ図は依存関係が複雑

で特定が困難である。例えば、「車間距離検知」と「相対速度演算」は先行車に対応する場合のみ必要だが、ゴールグラフによる特定は困難である。提案方法では OVM により定速走行システム選択時の「車間距離検知」と「相対速度演算」除外を表現でき、要素間の依存関係の特定が容易となる。

(2) 追跡可能性

ゴールグラフの後方追跡はそのゴールグラフに限定的である。また、フィーチャの必要判断は難しい。提案方法ではモデルとその構成要素に対応するタスクを認識し、ゴールグラフを上に出ることで各機能要求に対する非機能要求を表すソフト/ハードゴールの参照が可能となる。

6.2. モデルの表現能力

(1) 機能要求/非機能要求

フィーチャ図は機能/非機能要求が混在しており、分析が困難である。ゴールグラフは機能/非機能要求を分離可能であるが、分割方法は統一されていない。提案方法ではゴールグラフをソフトゴール、ハードゴール、タスクと段階的に詳細化することで機能/非機能要求を明確に分離できる。

(2) 可変性

PL の共通/可変性は一般にフィーチャ図で表現されるが、視点が分離されておらず UML との対応づけが困難である。また、ゴールグラフは可変性を扱わない。提案方法ではゴールグラフの構造から共通/可変性を認識し、モデル間の関係の定義により可変性を分離して表現できる。

7. 今後の課題

多様な製品の可変性を全て表現するために、提案モデル化方法をより詳細化する必要がある。クラス図など構造の視点との関係も明確に定義し、自動車ソフトウェアへの適用による評価を行う必要がある。

8. まとめ

PL 要求モデルを拡張し、複数視点による PL モデル化方法を提案した。ゴールグラフの構造から OVM を作成し、複数視点の可変性を表現した。提案方法を自動車ソフトウェアへ適用し、有効性を示した。

参考文献

- [1] S. Böhne, et al., Why is it not Sufficient to Model Requirements Variability with Features Models?, Proc. AURE04, 2004, pp. 5-12.
- [2] B. González-Baixauli, et al., Product Line Requirements Based on Goals, Features and Use Cases, Proc. IWREQFAM, 2004, pp. 4-7.
- [3] IEEE Std. 830-1998, IEEE Recommended Practice for Software Requirements Specifications, IEEE, 1998.
- [4] K. Pohl, et al., Software Product Line Engineering, Springer, 2009.
- [5] トヨタ自動車, CROWN MAJESTA 新型車解説書 URS206/UZS207, 2009.