様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

令和　　元 年　　６ 月 １９ 日現在

機関番号：６２６１５
研究種目：研究活動スタート支援
研究期間：2017～2018
課題番号：１７Ｈ０７３２３

研究課題名（和文）段階的詳細化の柔軟な変更および設計指針の確立

研究課題名（英文）Flexible Refactoring and Effective Guiding of Stepwise Refinement Design

研究代表者
小林　努（Kobayashi, Tsutomu）

国立情報学研究所・アーキテクチャ科学研究系・特任研究員

研究者番号：１０８０３４０５

交付決定額（研究期間全体）：（直接経費）　　2,300,000円

研究成果の概要（和文）：高信頼ソフトウェアシステムの構築のための形式仕様を用いた開発において、厳密な詳細化を通じて導入される対象システムの構成要素の導入順を整合性を保ったまま変更することがその変更容易性向上に有効である。
本研究では、既存の導入順の変更手法の改良のため、専門家の間で有効と知られている仕様のパターンを分析し、パターン由来の要素を用いた柔軟な要素導入の変更手法を提案し、さらに導入順をどのように変更すると有効かについて分析を行い知見を獲得した。

研究成果の学術的意義や社会的意義
本研究は、ソフトウェア工学における最も重要な概念であるモデルの抽象化・詳細化に、厳密な方法と新しい切り口で貢献するものである。
さらに、本研究の成果は形式仕様を用いた開発のコストを下げ、積極的な再利用を促進する。そのため、本研究は高信頼システムの開発プロセスの改良につながり、より多くのソフトウェアシステムを安全にすることに貢献するものと期待される。

研究成果の概要（英文）：Using formal specifications is known to be effective for constructing high-assurance software systems. Stepwise refinement mechanism of some formal specification methods enables developers to gradually introduce elements of a target system to specifications. We found that changing the order of elements' introduction while preserving the consistency improves maintainability and reusability of the specification.
In this research project, we aimed at improving the flexibility of existing methods for changing elements' introduction. We defined patterns of refinement from existing formal specifications. We also constructed a method for changing elements' introduction with new elements derived from the patterns. In addition, we analyzed introduction orders and found the best practices for designing refinements.

研究分野：ソフトウェア

キーワード：ソフトウェア　段階的詳細化　形式手法　ソフトウェアモデリング　ソフトウエア開発効率化・安定化　Event-B　プロブレムフレーム

様　式　C－１９，F－１９－１，Z－１９，CK－１９（共通）

１．研究開始当初の背景

現代のソフトウェアシステムの不具合は莫大な損害を引き起こし得る．そこで，形式モデルの構築とその正しさの証明が重要視されている．
近年はそのような手法の中でも「段階的詳細化」の機構をサポートするもの(Event-B[1]など)が注目されており，交通や宇宙関係などの大規模な高信頼システムの構築に使われている．
段階的詳細化を用いた形式仕様記述では，開発者はまず対象システムの一部の要素だけに関する抽象モデルを作り，後から別の要素を導入した具体モデルを作り，それらの間の整合性を検証する．
この仕組みにより，プログラム外の環境を含む，システムの多数の構成要素を厳密に扱う際の複雑さを複数段階に分散することができる．

しかし，厳密な検証を要する形式モデルの開発コストは大きく，適用は容易でない．
特に，１度開発したモデルの変更・改善・再利用のコストは大きく，対策が必要である[2]．
開発者は既存の厳密で複雑なモデルを理解し，適切に変更し，その整合性を再度証明する必要がある．
段階的詳細化を用いた手法では，対象システムの要素の導入順を選べる．
我々は，この導入順によって構築される仕様の見通しの良さや再利用性が異なることに着目した．

はじめから導入順を変更・改善・再利用に適するように設計することが望ましいが，再利用の可能性を予測して設計を行うことは難しい．
そこで，我々の先行研究では，構築済みのモデルの要素の導入順を，元のモデルとの整合性を保ちつつ組み替える手動の手法を提案した[3]．
導入順を組み替えると各段階で記述に使える要素の集合が変わるため，元のモデルと整合性のある新しい記述を一部の要素のみを用いて構成する必要がある．
そこでこの先行研究では，既存モデルの証明の分析により手動で整合性のために必要な記述を構成することで組み替え後のモデルに整合性を持たせる手法を提案し，さらにケーススタディとして組み替えで既存モデルの要素の導入の分割や汎用的な記述の抽出を行いモデル変更や再利用を容易化できた．
これにより，既存のモデルの変更・再利用を促進する基礎を固めた．

[1] J. R. Abrial, “Modeling in Event-B: System and Software Engineering,” Cambridge University Press, 2010.

[2] G. Klein, “Proof Engineering Considered Essential,” FM'14, pp. 16--21.

[3] T. Kobayashi, F. Ishikawa, and S. Honiden, “Refactoring Refinement Structures of Event-B Machines,” FM ’16, pp. 444–459.

２．研究の目的

上記の先行研究は詳細化の導入順変更手法の基礎を与えるものとなったが，課題が残っていた．
そこで本研究では以下の目標達成を目指した．

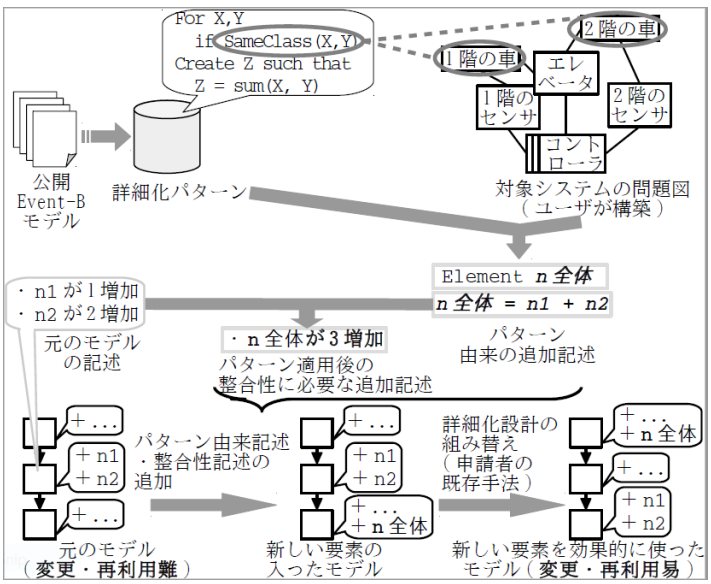

図１：本研究提案の主要部分

目標1: 新しい要素を用いた詳細化の導入手法の構築

先行手法では，元のモデルの要素をそのまま用いて，その導入の順序を変更し組み替える．
よって，先行手法では組み替えの可能性は元のモデルの要素に限定される．
しかし，仕様記述手法の機構上，元のモデルにない要素を導入して新しい詳細化を行うことが有効であることが知られている．
効果的な詳細化の一例に，「同じクラスの複数の対象のまとめ上げ」がある．
例えば抽象モデルでは建物全体の中にいる人の数(例えば，変数 n_bldg で表現)について抽象的・形式的な記述を行い，具体モデルではフロアごとに，あるいは部屋ごとに人数(例えば，変数 n_101，n_102，... ，n_309 で表現)の具体的な記述を行い，両モデルを比較した際に記述が,n_bldg = n_101 + n_102 + ... + n_309 を満たすことを確かめることができる．

ここで，既存のモデルに n_101，...，n_309 はあるがまとめ上げのための要素(n_bldg)がない場合，このモデルには n_bldg に関する整合性ある記述を追加して組み替えを行うと，n_bldg に関する，理解・証明が容易で汎用的な抽象モデルを構築できる(図 1)．

そこで，このような新たな要素を用いた，整合性のある詳細化の導入手法の構築を目指した．
これにより，既存要素に限定されない開発コスト低減に効果的な詳細化を可能にすることを目指した．

目標 2: 新しい要素を用いた詳細化設計の指針に関する研究
目標 1 を達成して新しい要素を用いた設計の組み替えが可能になっても，現在はどのような設計に組み替えるのが適切かの方針に欠け，これは既存研究でも扱われていない．
この方針を得るために，先行研究における整合性ある記述の構成の問題を体系化し，さまざまな新しい要素を用いた詳細化設計の可能性を候補として列挙することを可能にすることを目指した．
さらに，導入順組み替え手法を用いた実証実験を行い，「どのように詳細化設計の組み替えを行うべきか」についての方針を確立することを目指した．

３．研究の方法

はじめに，目標 1 に取り組んだ．

まず，ソフトウェアとやり取りする環境も含めた問題の分析に適しているとされる分析手法 Problem Frames[4]の図式を利用し，形式モデル上の詳細化パターンを実問題の図式に対応させる手法を構築した．
具体的には，図式の各要素が形式モデルのどの要素に対応するかを定義し，Problem Frames において複数の抽象度の図式間の関係を適切に扱うための手法を考案した．
これにより，ユーザが本手法を利用する際に図式を用いた直観的な指示を与えられるようになった．

次に，モデルに新しく記述を導入した際に，既存の記述と整合性を持たせるために必要な補助的な記述を導出する手法を構築した．
例えば，n_101 が 1 増加し，n_309 が 2 増加するような振る舞いについて，n_bldg の振る舞いは，(n_bldg = n_101 + ... + n_309 であることを考えると，)n_bldg が 3 増加するような振る舞いが対応する．
そこで，具体変数の振る舞いの式と，n_bldg = n_101 + ... + n_309 という抽象変数と具体変数を繋ぐ式が与えられた時に，抽象変数の振る舞いの式を獲得する手法を提案した．
これにより，パターン由来の記述が不適切なものでないならば，元のモデルの整合性を崩すことなく記述を導入できるようになった．

さらに，パターンに由来するモデルの変更の体系化を行った．
ここでは，新しい変数を用いて新しく作ったモデルが元のモデルと整合性を持つことを保証することが主要な問題であった．
そこで我々は，Event-B 手法における整合性保証の式(証明責務)の式を分析し，パターンに由来する式の追加を行った場合にどのような証明責務を証明する必要があるのかについての分析を行った．
また，証明責務の式に対して，論理学における Craig の補間定理[5]を利用して，証明責務を満たすために必要な式を獲得する手法を提案した．
Craig の補間定理では，P ならば Q という式があった時に，「P ならば X かつ X ならば Q」を満たす X(補間と呼ぶ)で,X の記号が P と Q の両方に現れるようなものが存在することが保証される．
また，このような X を獲得する既存のアルゴリズムが存在する．
我々は，パターンに由来する証明責務の式を変換した後に補間を獲得することで必要な式を獲得する手法を提案している．

また，目標 2 にも取り組んだ.

まず，実モデルの分析による，保守・発展に効果的であることが経験的に知られているパターンの獲得を行った.
ここでは，公開されている Event-B の多段階形式モデルを分析し，特に複数のモデルの要素間の関係について記述した式(例における n_bldg = n_101 + ... + n_309)の形に着目して，パターンの定義を行った.
これにより，モデルの保守・発展に効果的であることが経験的に知られているパターンを獲得した.
例に現れた「まとめ上げ」のパターンの他，抽象モデルでの越権行為の許可や時系列のまとめ上げなど，一般的なパターンを得ることができた.

そして，モデルに様々な変更を施した結果の比較を通じた分析を行った.
具体的には，特定の例題に対してパターンを用いた要素の追加をいろいろな順序で行うことを試し，どのようなパターン由来の詳細化がモデリングと証明の複雑さの軽減に効果的であるかの分析を行った.
結果として，変数の出現頻度の高いものを早い段階で導入すると良いなどの指針の他，複数のモデルの要素間の式に出現する変数が他の要素間の式にどのような現れ方をするかが強く関わっていることが判明し，Problem Frames のような図式上で計画を行うことの有効性が示唆された.

[4] J. Michael. “Problem frames: analysing and structuring software development problems.” Addison-Wesley, 2001.

[5] W. Craig, “Three Uses of the Herbrand-Gentzen Theorem in Relating Model Theory and Proof Theory,” The Journal of Symbolic Logic 22 (1957), no. 3, pp. 269--285.

４．研究成果

以上より，経験的に有効と知られている詳細化のパターンをもとに，図式を用いた直観的な指示で既存のモデルの保守・発展性を高める手法と，どのように詳細化を設計・改良すると有効かの指針を獲得した.
一方，今後の課題として，より具体的で網羅的なパターンの定義やオブジェクト指向分析など既存の手法との関係の分析，また，また生成されるモデルの式の可読性などがある.
今後もさらなる分析やユーザ実験を通じ，拡充の取り組みを続けていく.

５．主な発表論文等

〔雑誌論文〕（計 1 件）

1. Tsutomu Kobayashi, Fuyuki Ishikawa, Shinichi Honiden. Consistency-preserving refactoring of refinement structures in Event-B models. Formal Aspects of Computing (2019) 31: 287.

〔学会発表〕（計 3 件）

1. Tsutomu Kobayashi and Fuyuki Ishikawa. Refactoring Refinement of Event-B Models. Shonan Meeting towards Industrial Application of Advanced Formal Methods for Cyber-Physical System Engineering (2018).
2. Shinnosuke Saruwatari, Fuyuki Ishikawa, Tsutomu Kobayashi and Shinichi Honiden. Extracting Traceability between Predicates in Event-B Refinement. In Proceedings of 24th Asia-Pacific Software Engineering Conference (APSEC 2017).
3. 小林 努. 博士論文紹介：Supporting Planning and Refactoring of Refinement Structure of Event-B Models. 第 197 回ソフトウェア工学研究発表会（招待講演）(2017).

〔図書〕（計 0 件）

〔産業財産権〕
○出願状況（計 0 件）
○取得状況（計 0 件）

〔その他〕

６．研究組織

(1)研究分担者

(2)研究協力者

※科研費による研究は，研究者の自覚と責任において実施するものです．そのため，研究の実施や研究成果の公表等については，国の要請等に基づくものではなく，その研究成果に関する見解や責任は，研究者個人に帰属されます．