公開

（事後評価・研究者用）

# 研　究　報　告　書

## 「整合性を保持する形式仕様の自動抽象化システム「ソフトウェア顕微鏡」の開発」

研究期間：　平成 29 年 10 月～平成 31 年 3 月
研究者番号：　50139
研　究　者：　小林　努

1．　研究のねらい

近年、高信頼を求められる複雑なソフトウェアシステムが増加している。例えば、IoT をはじめとした物理空間・情報空間をつなぐシステムは環境との相互作用を考えて構築する必要があるなど複雑であり、しかも高度な信頼性が求められる。
このようなシステムの構築のために、システムの仕様を形式的な言語で記述し、その上で定理証明などで検証を行う手法(形式仕様記述手法)が注目され、欧州を中心に産業界でも大規模なプロジェクトで適用され有効性が認知されつつある。形式仕様記述手法の主な利点として、厳密な仕様と証明を、人が読んで理解することができるという点がある。

しかし、現実に大規模なシステムの形式仕様を記述すると、仕様は多くの変数が出現する長大なものとなり、理解が困難である。さらに、その仕様に関する整合性証明は、仕様と同様に複雑であるだけでなく、その複雑さゆえに制約ソルバなどを用いて自動的に証明が行われることが多い。自動証明では、多くの場合仕様の内容の寄せ集めを大量の仮定リストとして与え、ソルバが結論を証明できたかどうかだけが情報としてユーザに提示されるため、証明木が見られない。
そのため、現実にはユーザが仕様や証明を読んで理解することは困難である。

そこで、本研究では、与えられた仕様と整合性を持ち、しかも一部の変数のみを用いて表現した抽象的な仕様・証明を構築する「ソフトウェア顕微鏡」を開発した(のちに「ソフトウェア展望台」に改名)。
ここで、ナイーブな手法として、仕様と証明のうち特定の変数のみを用いて記述されている部分を切り出して(スライシングして)くるという手法が考えられるが、この手法だと、スライシングによって仮定となる式が欠落するため、既存の仕様では証明できていた整合性が満たされなくなってしまう。本研究では、入力の仕様の整合性証明木の中から、特定の変数のみを用いて表せるような「補題」を獲得して、それを用いて抽象的な仕様を「補修」することを目指した。

2．　研究成果

（1）概要

本研究では、完成した（構築・証明がなされた）形式仕様 s を入力として、ユーザが入力仕様の変数のうちの一部 V を指定すると、V を用いて記述され、s と整合性のある新しい形式仕様 s'（抽象仕様）を自動で出力することを目指した。

この目標を達成するにあたり、主に以下の 3 つの研究テーマに取り組むことを目指した。
テーマA：抽象仕様と入力仕様との整合性確保手法の構築　テーマB：抽象仕様の簡潔な整合性証明木の構築　テーマC：手法の自動ツールとしての統合
結果として、テーマAとテーマCのうちテーマA関連の部分を完遂し、テーマBについてもまとめ上げられつつある。また、テーマAに取り組む中で、最終的な目標である「形式仕様の理解容易性向上」を考慮した際に、研究開始時には考慮していなかった「どのような変数を指定して抽象化を行うと効果的かの分析」（テーマD）が重要であることが判明したため、テーマDについても取り組み成果を上げた。
本研究では近年産業界でも注目されている形式仕様記述手法 Event-B を主な対象として手法の整備を行い、Event-B の開発環境 Rodin ツールのプラグインとして提案手法の実装を行った。プラグインは、抽象化により全体像を見やすくすることを目的としていることから「ソフトウェア展望台」と呼び Web で一般に公開している。

（2）詳細

**研究テーマA：抽象仕様と入力仕様との整合性確保手法の構築**

本研究では、仕様の対象システムの性質を厳密に理解することを補助するため、構築する簡潔な抽象仕様が厳密に入力仕様と整合性を持つことを保証することが必要である。
プログラムコードの理解補助の分野では、特定の変数に影響のある命令や特定の変数が現れる命令を抜き出すアプローチが研究されてきたが、本研究が目的とする、記述に用いられる変数を制限した上で入力仕様と厳密な整合性を持つ手法は提案されていない。
本研究では、入力仕様で既に行われている整合性証明を拠り所とし、入力仕様の整合性のうちで抽象仕様の変数に関係する部分が満たされるように、抽象仕様に含める必要のある式を獲得することで両仕様間の整合性を確保する手法を提案した。
例えば、本研究で対象とした仕様記述手法 Event-B における整合性は、主に 2 種類に分けられる。第一に、仕様には主に対象システムの満たすはずの条件（不変条件）と振舞いとが記述されるが、振舞いが本当に不変条件を常に満たすことを確かめる必要がある。第二に、ある仕様と別の仕様が抽象･具体の関係にあるためには、両者の振舞いをオートマトンで表現した際に、それらが模倣関係にあることを確かめる必要がある。
提案手法では、これらの確かめる必要のある条件（証明責務）の式を拠り所として、整合性のために抽象仕様に含める必要のある式を獲得する。この際、論理学の定理である Craig の補間定理を利用する。Craig の補間定理は、P ⇒ Q なる論理式 P と論理式 Q があり、P と Q に共通する記号が使われているならば、P と Q の両方に使われている記号のみで記述された論理式 X が存在し、P ⇒ X ⇒ Q が成り立つというものであり、補間獲得のためのアルゴリズムが提案されている。仕様の証明責務は「(不変条件) ∧ (イベント発火条件) ∧ (状態遷移の満たす条件) ⇒ (状態遷移後の不変条件)」のように、「前提 ⇒ ゴール」の形をしている。
提案の基本的なアイデアはこの式の補間として抽象仕様に含める式を獲得するというものであるが、前提とゴールの両方に使われている記号（すなわち、補間の式に使われる記号）は必ずしもユーザが抽象仕様に含めたいと指定した記号だけとは限らず、そのまま証明責

務の式の補間を獲得して得た式は直接抽象仕様に含めることができない。そこで提案手法では証明責務の式 f を同値な別の式 g に変換するルールで、前提とゴールに共通する記号がユーザの指定した記号の部分集合となるようなものを定義した。これにより、特定の変数で記述された、整合性のために必要な式を獲得する手法を構築した。
さらに、この手法を Event-B に限らず仕様記述と詳細化を扱う理論 Action systems の枠組みで一般化し、研究成果(1)-2 として Formal Aspects of Computing 誌で発表した。

**研究テーマB：抽象仕様の簡潔な整合性証明木の構築**

テーマAの成果により、構築された抽象仕様は入力仕様と整合性を持つことが保証される。しかし、本研究の目的である理解容易性の向上を実用的なツールとして実現することを目指すと、なぜ整合性が満たされるのかという証明を分かりやすい形で示すことが重要である。研究期間終了時現在、本テーマの成果は未公開ではあるものの、テーマA遂行において得られた知見をもとに手法を構築しまとめ上げつつある。

**研究テーマC：手法の自動ツールとしての統合**

本研究テーマの目的は、自動ツールとして提案手法を統合することである。柔軟で厳密な仕様記述言語として注目を集める Event-B の開発環境 Rodin platform はプラグインを開発することによって機能拡張を行うことが可能である。そこで、Rodin のプラグインとしてツールを実装した。
ツールの主要インタフェースは、Event-B 仕様のビューアである。ビューアでは、チェックボックスで入力仕様の中の変数の一部を選択することができ、さらにそれをもとに入力仕様のうちでそのまま抽象仕様に含めることのできる式を自動で選択する機能を有する。これらの機能を用いてユーザの意向を入力すると、ツールは抽象仕様のひな形を構築する。ここで、Event-B 仕様の文法や記述ルールに従った仕様が構築されるように実装を行っている。
さらに、テーマAの手法を実装した。ここでは、証明責務の式をテーマAで提案したルールで変換し、補間計算機能を有する既存の制約ソルバ Z3 を利用して補間を獲得して得た式を抽象仕様のひな形に追加することで整合性が保証されるようにしている。テーマBの手法は現在実装中である。
構築したツールは、抽象化により全体像を見やすくすることを目的としていることからツール名を「ソフトウェア展望台」と変更し、Web で一般に公開している(http://research.nii.ac.jp/slicenmerge/)。

**研究テーマD：どのような変数を指定して抽象化を行うと効果的かの分析**

研究テーマAを遂行する中で、ある 1 つの仕様に対し、ユーザが選択する変数集合の違いによって様々な抽象仕様の可能性があることに着目した。抽象化を通じた理解容易性を考える上で、これらの抽象仕様の可能性の間の違いは重要である。特に、Event-B のように詳細化を扱うことのできる手法では、多段階の抽象化を考えることができる。
そこで、ソフトウェア展望台を利用して様々な抽象化の戦略を比較する実験を行った。具体的には、例えばある構築済みの多段階の仕様 s が変数 a, b, c を持つような時、ソフトウェア

展望台を用いて抽象化を行うことで変数 a, b を持ち s の抽象版である仕様 t1、さらに変数 a を持ち t1 の抽象版である仕様 t2 を構築できる。さらに、変数 b, c を持ち s の抽象版である仕様 u1、変数 c を持ち u1 の抽象版である仕様 u2 も構築できる。こうしてできる 2 つの多段階抽象化戦略「t2 → t1 → s」と「u2 → u1 → s」とでは、仕様のサイズや証明の複雑さにどのような違いが出るのかを調査した。

仕様の抽象化は変数の除去と見ることができるが、ソフトウェア展望台を用いると、複数変数を 1 ステップで除去するような抽象化を複数ステップに分割することや、逆に複数ステップを結合するような操作を行うことができる。また、複数のステップの順序を入れ替える操作も可能である。実験では、このような操作を入力仕様に適用することにより様々な抽象化の戦略に従った仕様を構築した。

結果として、基本的に多くのステップに分けて少しずつ変数を除去するような抽象化や、ごく一部の不変条件のみに出現する「珍しい」変数を先に除去するような抽象化が複雑さの低減に効果的であることが判明した。Event-B のように詳細化を用いてモデリングを行う際の open problem として、「詳細化戦略の立案」、つまりどのような順序で変数を導入して記述を進めていくと複雑さを抑えたモデリングと検証ができるかという問題がある。詳細化戦略とは、すなわち本実験で扱った抽象化戦略を逆に見たものである。そのため、本実験で得られた知見は、詳細化戦略の立案問題におけるガイドラインを示唆するものであると言える。

この成果は国際会議 ICFEM で発表し、best paper award を受賞した（成果(1)-1、 成果(3)-受賞 1）。

## 3. 今後の展開

本研究では、入力の形式仕様に対して整合性を持つ「抽象版」の形式仕様とその整合性の証明を構築することを目的としていた。

今後は、整合性証明の構築部分（研究テーマ B）とツール（研究テーマ C）を論文としてまとめて発表することや本格的な評価実験に加え、自動化を活かした各種応用へ展開することを考えている。例えば、ある仕様を抽象化することによって、仕様のうち再利用可能な部分を抽出し、既存の仕様の対象システムと共通要素のある別の対象システムの仕様を構築する基礎を過不足なく獲得することや、さらに発展させて、既存の複数の仕様の共通要素をくくり出すことで、形式仕様の再利用可能なライブラリを構築することも考えられる。

このように、本研究を発展させ、形式仕様を用いた開発に新しいパラダイムを導入するように展開したい。

さらに、同様の手法を Event-B とは異なる仕様記述言語やプログラムコードに適用可能な形に拡張することも考えている。

## 4. 自己評価

本研究では、形式仕様の整合性ある自動的な抽象化を目指し、「2. 研究成果」に記載した 4 つのテーマに沿って研究を遂行した。

テーマ B とテーマ C について当初の目標を完遂することができなかったことは反省点の一つである。特に、時間のかかる実装の進め方などについては改善の余地が大きく、今回の経験を以降の研究で活かしていく所存である。

一方で、テーマAの研究が理論的な一般性や、提案手法の仕様の理解容易性や再利用性向上への貢献が認められ、当該分野の代表的な論文誌 Formal Aspects of Computing に論文が掲載された。

テーマDの研究については、論文が当該分野の国際会議 ICFEM 2018 に採録され best paper award を受賞した。適切な理論的分析に基づいた自動化手法を活かした新しい視点の実験を通じ、検証手法のみに偏りがちな当該分野の他の研究と異なり、「形式的開発手法をいかに利用し、いかに活かすか」という工学的応用についての知見を獲得したことが評価されたためである。

さらに、ツールはオープンソースで公開しており、提案ツールが他の形式的開発手法利用者に貢献できるようにしている。加えて、「3. 今後の展開」で述べたように、本研究は形式的なソフトウェアの開発手法の新しい形を与える可能性を秘めており、ひいては形式的開発手法の裾野を広げ、世のソフトウェアを安全にすることに寄与できるものと期待している。

以上より、学術・産業・社会の各分野に貢献できたと考えている。

研究体制として、基本的に単独で全ての作業を担当したが、ACT-I のアドバイザーや他の研究者との議論は研究遂行にあたって大変有意義であった。また、国内・海外の専門家との議論やワークショップなどでの発表の機会を多く持つように心がけたことも有効であった。研究経費は主に実験で多数の仕様を扱い自動証明器を実行するための高性能コンピュータや専門家との議論のために活用した。結果として、効率的に研究を遂行することができた他、新たな視点を得る貴重な機会を多く得ることができた。

## 5. 主な研究成果リスト

### (1) 論文 (原著論文) 発表

1. Tsutomu Kobayashi and Fuyuki Ishikawa. Analysis on Strategies of Superposition Refinement of Event-B Specifications. The 20th International Conference on Formal Engineering Methods (ICFEM 2018). 2018. pp. 357-372. Best paper award.
2. Tsutomu Kobayashi, Fuyuki Ishikawa, and Shinichi Honiden. Consistency-preserving refactoring of refinement structures in Event-B models. Formal Aspects of Computing. 2019. in-press (available online).

### (2) 特許出願 0件

### (3) その他の成果 (主要な学会発表、受賞、著作物、プレスリリース等)

受賞 1. Tsutomu Kobayashi and Fuyuki Ishikawa. Analysis on Strategies of Superposition Refinement of Event-B Specifications. Best paper award at The 20th International Conference on Formal Engineering Methods (ICFEM 2018). 2018.

口頭発表 1. Tsutomu Kobayashi. Analysis on Refinement Strategy of Formal Specifications. Oral presentation at OU-NII-Lero-Khalifa-LYON1 Workshop on Software Engineering for Cyber-Physical-Social Systems (CPSS 2018). 2018.

口頭発表 2. Tsutomu Kobayashi and Fuyuki Ishikawa. Refactoring Refinement of Event-B Models. Oral presentation at Shonan meeting towards industrial application of advanced formal methods for cyber-physical system engineering. 2018.