研究課題別評価

1　研究課題名：オブジェクト指向分析モデルの形式的構築法と検証法

2　研究者氏名：青木　利晃

3　研究の狙い：

現在、IT の進歩により、ソフトウェアの規模、及び、新規ソフトウェアの需要が共に急激に増加し続けている。しかしながら、それらのうち誤りを含むものは少なくない。今後、ますますソフトウェアの規模が大きくなり、社会システムの様々な部分が電子化されてくるであろう。これらのことを考えると、ソフトウェアの正しさを保証する手法の確立が急務である。本研究では、ソフトウェアの信頼性を向上させるために、その検証法について研究を行った。狙いは以下の 2 つである。

(1)オブジェクト指向分析・設計モデルの検証

ソフトウェアの規模が大きくなると、ソフトウェアを実現しているプログラムではなく、その設計書の役割が重要になってくる。また、ソフトウェアの複雑さや規模の管理も重要であり、それらに対処できるオブジェクト指向開発法が注目されている。オブジェクト指向開発法では、設計書を、UML などの記法を用いて分析モデル、または、設計モデルとして記述する。そこで、本研究では、そのようなオブジェクト指向分析・設計モデルに焦点を当てた。

(2)定理証明手法とモデル検査手法の適用

ソフトウェアの検証のためのアプローチとしては、定理証明手法とモデル検査手法がある。定理証明手法では、ソフトウェアの性質を推論規則などを用いて対話的に証明することにより、正しさを保証する。モデル検査手法では、ソフトウェアを有限状態で表現し、網羅的に探索して自動的に正しさを保証する。これらの手法には長所と短所がある。前者は、高い記述能力と検証能力を持っているが、対話的に証明を行うためコストが高くなる。一方、後者は、記述能力が低くソフトウェアを有限状態に抽象化する必要があるが、自動的に検証を行うことができる。

本研究では、まず、UML で記述された分析・設計モデルを、定理証明手法を適用により検証する手法を提案した。しかしながら、この手法では、対話的に証明を行うため、検証コストが高くなってしまう。そこで、コストを低くするための手段として、検証を再利用する手法を提案した。また、モデル検査技術により分析・設計モデルを検証する手法についても研究を行った。

4　研究成果：

主な研究成果として以下の 4 つがある。

(1)オブジェクト指向分析・設計モデルの定理証明システムによる検証法の提案。
(2)検証結果の再利用法の提案。
(3)計算機支援環境 F-Developer のバージョンアップ。
(4)モデル検査ツールによる分析・設計モデルの検証法の提案。

4．1　オブジェクト指向分析・設計モデルの定理証明システムによる検証法の提案

これまでに、UML のクラスダイアグラムとステートチャートダイアグラムに基づいた検証法は提案していたが、記述能力に問題があった。具体的には、クラスが持つメソッドを、オブジェクト指向的では無い、副作用無しの関数として記述していた。属性が持つ値は原始的な値のみであり、オブジェクトへの参照を持つことができなかった。また、表明などの性質も、述語論理で記述している。本来、これらの記述は、オブジェクト指向の観点から記述すべきものである。このようなメソッド定義や性質をオブジェクト指向的に記述するための言語として、様々なアクション言語と制約言語が

提案されている。そこで、それらを用いて記述されたオブジェクト指向分析・設計モデルを定理証明システムを用いて検証する手法について提案した。

この手法では、オブジェクト指向の概念を定理証明システムで扱うために、それらを実現する仕組みを作り込む必要があり、余分な証明ステップがかかる可能性がある。そこで、記述能力の他に、定理証明システムで取り扱う際に、オーバヘッドがかからないよう工夫した。評価のためにいくらかのデザインパターンを記述し、それらで成立すべき性質について検証を行った。それらの 1 つである ObserverPattern では、定理証明システム HOL で約 200 行で記述することができた。また、状態が常に反映されるという性質の証明では 4 つの補題を証明することになるが、それぞれの補題はすべて 1 ステップで証明できた。これらのことは、提案した手法が、十分な表現能力を持っていること、および、変換された記述で証明を行う場合のオーバヘッドが小さいことを意味している。

4．2 検証結果の再利用法の提案。

提案した検証手法では、検証対象の分析・設計モデルが複雑なデータ型や動作を含んでいる場合、特に、証明過程で帰納法や場合分けが必要な場合、証明ステップが多くなったり、適用する推論規則の選択が難しくなる。そこで、検証コストを下げるために、証明の再利用して正しいモデルの段階的構築する手法を提案した。この手法を、表明主導手法(Assertion Driven Approach)と呼ぶ。証明の再利用では、重複した証明を再度行う必要が無いのでコストを下げることができる。また、段階的構築では、早期に誤りが発見できるため、手戻りコストを下げることができる。

図 1 に表明主導手法の概要を示す。この手法は 2 つのフェーズに分かれている。1 つ目は、個々のソフトウェア開発ではなく、ソフトウェアの集合を考えるフェーズ、すなわち、領域分析である。表明主導手法では、領域分析のフェーズで、領域に現れる共通な動作を抽出し、それにまつわる性質を証明して領域ライブラリに蓄積する。この動作と性質は、すでに検証済みのモデルからも抽出することができる。2 つ目のフェーズは、個々のソフトウェアを作成するフェーズ、すなわち、通常のソフトウェア開発である。このソフトウェア開発では、領域ライブラリから適切な動作を探し、アレンジする。そして、モデルに段階的に追加していく。このアレンジと、動作の追加では、動作やモデルの正しさが崩れないようにする。本研究では、このような領域ライブラリと、それを用いた正しいモデルの段階的構築のための基礎理論を提案した。

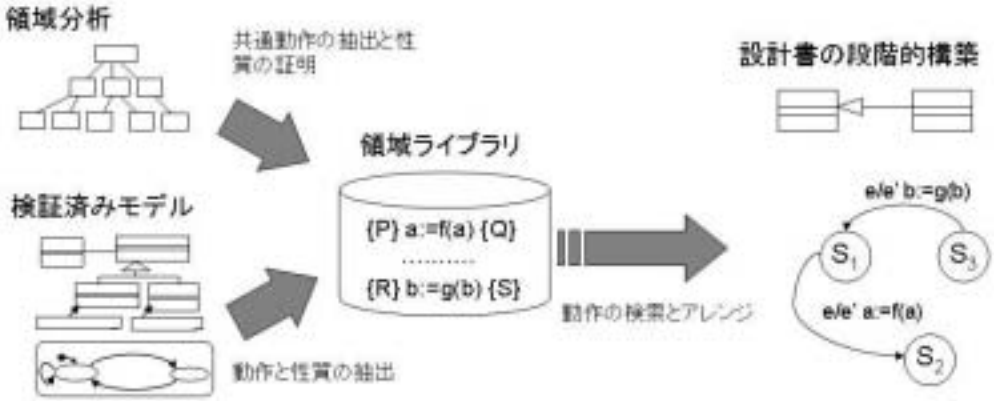

図 1 表明主導手法の概要

4．3 計算機支援環境 F-Developer のバージョンアップ。

本研究では、以上で提案した手法を支援するツール F-Developer を開発している。このツールは、2 つのサブシステム F-Prototyper と F-Verifier から構成されている。F-Prototyper では、UML で記述されたモデルをプロトタイプ実行できる。F-Verifier は、そのようなモデルを定理証明システム HOL で取り扱い可能な形式に自動的に変換して証明を支援する。さきがけ研究をはじめる段

階ではバージョンが 0.12 で非常に限られた機能しかなかったが、現在では、通常のモデル化作業に耐えられるくらいの機能が実装されている。現在は、F-Developer ver.0.2(2003 Dec.版) for Windows を最新版として公開している。

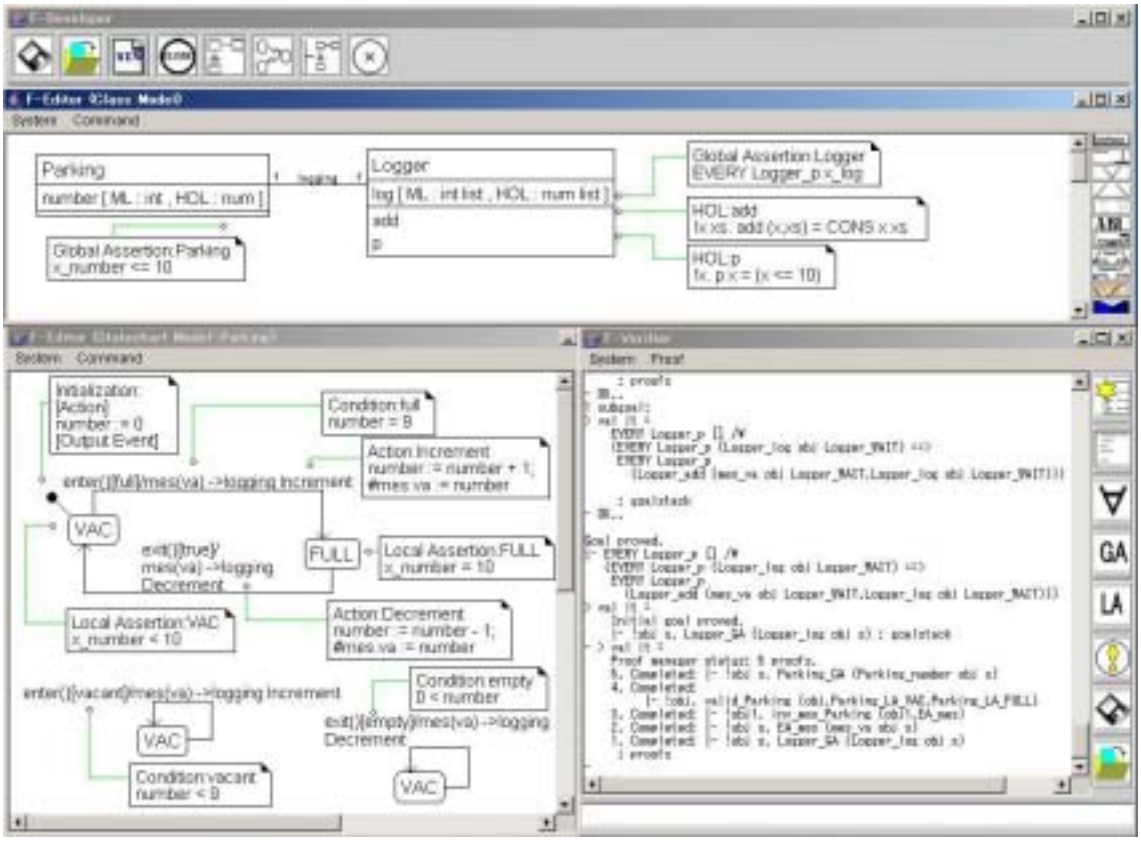

図 2 計算機支援環境の画面スナップショット

4．4 モデル検査ツールによる分析･設計モデルの検証法の提案。

モデル検査技術では、並行動作する複数の有限状態モデルの動作を網羅的に検査することにより、到達性やデッドロックなどの性質を調べる。最近では、モデル検査ツールの実装が進み、マニュアルなども整備されてきた。これにより、様々な対象に適用されるようになり、プログラムやその開発工程で作成されるモデルを対象とした検証法も提案されている。

モデル検査ツールを用いて設計モデルの検査を行う場合、調べたい性質毎に記述や検査の手法が異なる。そこで、設計モデルにおけるデータフロー解析、及び、センサーの取り扱い方に関する研究を行った。

4．4．1 データフロー解析

組み込みシステム開発、特に制御系のシステムでは、データの変換的側面、すなわち、データフローのモデル化を行う。近年、組み込みシステムの規模の劇的な増大、及び、多様化により、オブジェクト指向アプローチが取られるようになってきた。データフローの視点は、構造化分析設計アプローチで中心的な役割を担って来たが、オブジェクト指向アプローチにおいても、特に、組み込みオブジェクト指向開発では重要である。そこで、オブジェクト指向設計モデルにおけるデータフローの意味論を定義し、それを解析する手法を提案した。

オブジェクト指向設計モデルでは、オブジェクトは、互いに協調しながら並行動作している。それぞれのオブジェクトにおけるアクションの実行などによりデータフローが生じるが、それらは、オブジェクトの並行性を考慮して解析する必要がある。そこで、並行性解析に用いられるモデル検査ツール Spin により、オブジェクトの並行協調動作の結果生じるデータフローを、それらの可能な振舞いを網羅的に探索することにより解析する手法を提案した。これにより、オブジェクト指向設計モデルとデータフローを明確に対応づけ、自動解析することが可能になった。今後は、オブジェクト指向設計と Simulink などで用いられるブロック線図による設計の整合性検証などへの応用を考えている。

4．4．2 センサーの取り扱い

組込みシステムの分野では、従来から、実時間性、メモリ制約などのハードウェア制約の充足性が主な議論の対象であった。近年、組込みシステムの大規模化、多様化に伴い、組込みシステムの定義や議論の対象が変化してきた。組込みシステムと呼ばれるものは多種多様であり、通常のシステムやソフトウェアとの境界が曖昧になってきたため、それらを明確に分類することが困難なのが現状である。そこで、まず、組込みシステムで重要な役割を担っている部品、センサーに注目することにして、モデル検査を適用する手法について研究を行った。

これまでに、センサーの仕様を SCR(Software Cost Reduction)法のモード遷移表を用いてモデル化する手法、および、それをモデル検査ツール SMV で検証する手法について提案した。モード遷移表には自然数などの変数が記述されており、SMV で検証する前に、それらを抽象化する必要がある。提案した手法では、抽象解釈法の一種であるデータマッピング法で、定理証明システムを用いながら効率的に抽象化を行う手法を提案した。今後は、検査した仕様に基づいて、センサーへのアクセスメカニズム、および、制御のためのスケジューリングを考慮に入れた設計モデルをいかに構成するかについて研究を行う予定である。

5 自己評価:

本研究では，(1)オブジェクト指向分析･設計モデルを検証する手法と，(2)正しいモデルを構築する原理を明確にすることを目指した．(1)に関しては，研究を進める過程で検証対象となるモデルの記述能力と検証能力の低さが明らかになったが，アクション言語と制約言語、および、その取り扱い方を提案することにより解決することができ，目標を達成することができた．

(2)に関しては，正しいモデルを段階的に構築する規則や条件を明確にして，表明主導手法としてまとめた．これにより、基本的な原理を明らかにすることができた。当初，さらに実際のモデル構築で行われるような作業に対応する構築手法の提案を目指していたが，現時点では，明確には対応づけられていない．モデルへの要素の追加の単位がアクションや状態遷移などであり，粒度が細かいためである．今後は，デザインパターンやソフトウェアコンポーネントといった，大きな粒度の要素を扱うことができるようにする予定である．このような大きな粒度の要素の取り扱いは，現在までに提案した小さな粒度の要素に関する原理の積み重ねにより実現できそうな感触は持っている．また，提案した手法を計算機支援環境 F-Developer に実装することも今後の課題である．

主な検証技法としては定理証明技法とモデル検査技法があり，当初は，前者のみに焦点を当てていた．しかしながら，研究を進めるにつれて，オブジェクト指向分析･設計モデルの検証では，双方の使い分けが重要であることを実感した．そこで，途中から，モデル検査技法に関する研究も並行して行うことにした．そして，2 つの検証手法を提案することができた．これは当初の計画には無い成果である．これらの手法も，今後，F-Developer 上に実装する予定である．

また，このように提案した手法を F-Developer 上に実装していくことを考えると，現在のアーキテクチャでは困難であることもわかってきた．現在は，特定の検証手法用に実装されているので，複数の検証手法を柔軟に組み込めるようにはなっていない．そこで，そのようなアーキテクチャの

設計と実装を行いたいと考えている．これは，研究を進めてきて発見した新たな研究課題である．

6 研究総括の見解：

青木氏の研究は，定理証明などによる形式的検証により，正しいソフトウェアの開発を行う手法とその環境に関して研究を行ったものである．従来から形式手法によるソフトウェアの検証は研究されてきたが，検証コストが高いことにより産業界で採用されるには至っていない．青木氏は，この問題を解決するための，オブジェクト指向分析・設計モデルの検証のコストを下げることを目的として，定理証明やモデル検査による検証方法論や検証支援環境，検証結果の再利用方法論などの研究を行った．オブジェクト指向分析設計モデルの検証としては先進的な結果を出し，また，開発したツール(F-Developer)の公開など形式技術のソフトウェア開発への適用に関して優れた研究成果をあげたと評価される．

7 主な論文等：

論文

1. 青木利晃、片山卓也：オブジェクト指向分析モデルの検証支援環境、第 19 回 日本ソフトウェア科学会 全国大会論文集(CD-ROM), 2002.
2. 岡崎光隆、青木利晃、片山卓也：並行オブジェクトモデルから並行スレッドモデルへの変換法、情報処理学会 ソフトウェア工学研究会 研究報告 2003-SE-140，pp.39-46，2003.
3. 立石孝彰、 青木利晃、 片山卓也:振舞い近似手法を用いたステートチャートに対する不変性の検証、情報処理学会論文誌 Vol.44 No.6，pp.1448-1460，2003．
4. 青木利晃、片山卓也：状態遷移図の段階的構築のための論理的基盤、情報処理学会 ソフトウェア工学研究会 研究報告 2003-SE-143，pp.21-28，2003．
5. 青木利晃、岸知二、片山卓也:定理証明システムとモデル検査ツールを併用した設計モデルの検証実験、日本ソフトウェア科学会 第1回ディペンダブルソフトウェア研究会、 pp.49-58, 2004.
6. 青木利晃、片山卓也：オブジェクト指向分析モデルにおけるデータフローの形式化と解析手法、日本ソフトウェア科学会 学会誌 コンピュータソフトウェア、Vol.21, No.4,pp.1-26, July, 2004.
7. 青木利晃、岸知二、片山卓也：センサーのモデル化とモデル検査技術の適用について、組込みソフトウェアシンポジウム 2004, pp.118-125, 2004.
8. 青木利晃，片山卓也：アクション言語と制約言語を用いて記述されたオブジェクト指向設計モデルの検証法，日本ソフトウェア科学会 第2回ディペンダブルソフトウェア研究会, pp.61-70, 2005．
9. 矢竹健朗，青木利晃，片山卓也:コラボレーションに基づくオブジェクト指向モデルの検証，日本ソフトウェア科学会 学会誌 コンピュータソフトウェア，Vol.22, No.1,pp.58-76, 2005.
10. Takaaki Tateishi, Toshiaki Aoki and Takuya Katayama: Successive Behavior Approximation Method for Verifying Distributed Objects, Third International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'02), pp.439-446, 2002.
11. Mitsutaka Okazaki, Toshiaki Aoki and Takuya Katayama: Extracting threads from concurrent objects for the design of embedded systems, Ninth Asia-Pacific Software Engineering Conference 2002, pp.107-116, 2002.
12. Mitsutaka Okazaki, Toshiaki Aoki and Takuya Katayama: Formalizing sequence diagrams and state machines using Concurrent Regular Expression: Proceedings of 2nd International Workshop on Scenarios and State Machines: Models, Algorithms,

and Tools SCESM'03, pp.74-79, 2003.
13. Kenro Yatake, Toshiaki Aoki and Takuya Katayama: Collaboration-based Verification of Object-Oriented models in HOL, Proceedings of the 2nd International Workshop on Verification and Validation of Enterprise Information Systems, VVEIS 2004, pp.78-80, 2004.
14. Toshiaki Aoki and Takuya Katayama: Foundations for Evolutionary Construction of State Transition Models, the Seventh International Workshop on Principles of Software Evolution 2004, pp.143-146, 2004.