様式 C-19

科学研究費補助金研究成果報告書

平成　21 年　 6 月　12 日現在

研究種目：若手研究（B）
研究期間：2007～2008
課題番号：19700030
研究課題名（和文）　有界モデル検査による仕様検証手法を利用した
ソフトウェア開発コストの削減
研究課題名（英文）　Cost reduction for software development
by design verification using bounded model checking

研究代表者
横川　智教（YOKOGAWA　TOMOYUKI）
岡山県立大学・情報工学部・助教
研究者番号：50382362

研究成果の概要：本研究課題では，動的性質の検証手法であるモデル検査を用いた UML 設計群の検証手法について研究開発を行った．本研究では特に，ソフトウェアの各モジュールの動作とモジュール間のメッセージ通信に関する設計の整合性に注目しており，各モジュールが与えられたメッセージ通信仕様を満たすか有界モデル検査により検証するものである．また，UML 設計群を一つのモデルに統合することによるモデルの巨大化に伴い，検証コストが指数的に増大することが予想されるため，有界モデル検査の適用とその効率化についても研究を行っている．

この目的を達成するため，まず，モジュールの動作並びにモジュール間のメッセージ通信を記述する UML 設計群から有界モデル検査のためのモデルを抽出する手法について研究開発を行った．各 UML 設計で記述された動的振る舞いを論理式表現し，それらを組み合わせることで有界モデル検査による検証を実現する．次に，有界モデル検査を UML 設計群の検証に最適化するため，状態空間の探索アルゴリズムの比較実験を行った．また，適用実験の一環として，Web ページのナビゲーション構造の検証についても研究開発を行い，UML の状態マシン図によりモデル化されたナビゲーション構造をモデル検査により検証する手法について研究開発を行った． 最後に，提案する検証系を計算機上に実装するため，与えられた UML 設計群から論理式表現を自動生成するツールを開発した．

その成果として，UML 設計群として与えられたモジュールの動作並びにモジュール間のメッセージ通信が互いに矛盾なく記述されているかを有界モデル検査により検証する枠組みを実現するとともに，自動検証ツールについて実装の一部を完了した．また，適用実験として，Web ナビゲーション構造の設計に対して提案法による検証を行い，有効性を確認した．

交付額

（金額単位：円）

	直接経費	間接経費	合　計
２００７年度	1,800,000 円	0 円	1,800,000 円
２００８年度	1,400,000 円	420,000 円	1,820,000 円
年度			
年度			
年度			
総　計	3,200,000 円	420,000 円	3,620,000 円

研究分野：総合領域
科研費の分科・細目：情報学・ソフトウェア
キーワード：ソフトウェア

１．研究開始当初の背景

大規模なソフトウェア開発においては，要求分析や設計といった上流工程における誤りの混入が，実装やテストなど下流工程における手戻りを発生させてしまい，開発プロセス全体のコストを増大させてしまう要因となる．仕様の曖昧性やドキュメント間の矛盾といった上流工程での誤りを早期に発見し，回避するため，開発現場では形式的な仕様記述手法が用いられつつある．形式仕様記述ではあらかじめ定められた記法に従って仕様を記述することで曖昧性を排除し，一貫性をもったシステムの開発が可能となる．形式仕様記述の一種である UML は，ソフトウェア開発を始めとして，組み込みシステムや WEB アプリケーションの開発など多くの分野で用いられている．形式仕様記述を用いることで，設計段階で上記のような誤りは発見することができる．しかしながら，システムが設計者の意図した通りの動作を行うか否か，といったシステム自体の正しさについては保証することができない．

形式的に記述されたシステムの正しさを検証する手法として，近年注目されているのがモデル検査である．モデル検査とは，状態機械としてモデル化されたシステムが与えられた特性を満たすか否かを全ての状態を網羅的に探索することで検証する手法である．モデル検査は完全に自動化された検証であり，人手による検証では発見が困難な多くの誤りを発見することができる．しかしながら，対象となるシステムが複雑になるとモデルの状態数が爆発的に増加し，検証が不可能となってしまうという状態爆発の問題がある．モデル検査の一種である有界モデル検査は，探索の対象とする状態空間を一部に限定することでこの問題を回避し，大規模なシステムに対する検証を可能とする手法である．

２．研究の目的

本研究課題では，UML によって記述された仕様の検証に有界モデル検査を用いることにより，複雑な仕様記述を自動検証する枠組みを実現することを目的としている．これにより，複雑な仕様記述に対しても多くの誤りを発見することが可能となる．その結果，上流工程での誤りの混入を防ぐことが可能となり，システム開発コストの大幅な削減が期待される．

上記の目的を達成するため，以下の 4 つの課題について研究開発を行った．

(1) UML 設計群からの論理式モデル抽出手法の開発

有界モデル検査では，検証の対象となるシステムを状態遷移グラフとしてモデル化し，論理式を用いて記述する必要がある．そのため，まずは複数の UML 図によって様々な側面から記述されたシステムの仕様記述を統合し，一つの状態遷移グラフを求めるための手法を考案する．その上で，その状態遷移グラフを論理式表現するための手法を開発する．

(2) 有界モデル検査の効率化

本研究課題では，UML によるシステムの設計記述に対してモデル検査を適用することで，システムの正しさを検証する．しかしながら，対象となるシステム全体の規模は，それぞれの設計記述の規模に対して指数的に増加するため，従来の有界モデル検査では現実的な時間での検証が困難となる．

そこで，有界モデル検査のためのアルゴリズムを効率化し，高速化することで，大規模システムに対しても現実的な時間での検証を可能とする．

(3) Web ナビゲーション検証への適用

適用実験の一環として，提案する UML 設計の検証手法を，UML の状態マシン図によって記述された Web ナビゲーションの設計へと適用する．ここで，Web ナビゲーションとは，Web ページ間の移動を制御する仕組みである．

(4) モデル自動生成ツールの開発

(2)のモデル抽出手法によって生成される論理式のサイズは対象とする UML 図のサイズに比例して大きくなるため，人手による生成には誤りの混入，生成時間等の問題がある．従って，提案手法をソフトウェア開発プロセスに対して適用するためには，論理式モデルの自動生成ツールの開発が必要となる．

３．研究の方法

(1) UML 設計群からの論理式モデル抽出手法の開発

UML 設計群として記述されたシステムの仕様記述から統合された一つの状態遷移グラフを求めるため，まず各 UML 図が表す状態遷移システムを論理式表現する手法を考案し，その上でそれらを一つの状態遷移システムへと統合する手法を考案する．

本研究課題では UML 図の中でも動的な性質を記述するものとして広く用いられている，状態マシン図とシーケンス図およびコミ

ュニケーション図を扱うものとする．

(2) 有界モデル検査の効率化

UML 設計検証への適用を前提として有界モデル検査の効率化を行うため，図 1 に示す 2 種類の状態空間の探索手法のいずれが UML 設計の検証に適しているかについて，比較検討を行う．

有界モデル検査では，予め検証する性質を表す目的状態を定めておき，初期状態から目的状態へと到達可能であるか否かについて検証を行う．ここで，初期状態を起点に探索を行う場合（Forward Search，図 1(a)）と目的状態を起点に探索を行う場合（Backward Search，図 1(b)）では到達可能な状態空間の大きさに差が生じるため，検証に要する時間も異なる．ここでは，例題システムに対する適用実験をもとに比較検討を行う．

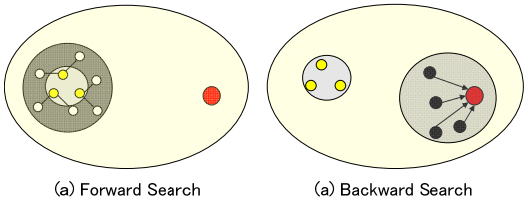


図 1：状態空間の探索手法

(3) Web ナビゲーション検証への適用

提案手法の適用実験の一環として，Web ナビゲーションの構造を記述する状態マシン図群を対象として論理式によるモデルを生成し，検証を行う．

ここでは，図 2 に示す状態マシン図として記述された Web ページのナビゲーション構造を論理式表現へと変換し，モデル検査ツール SMV によって検証する．なお，Web ナビゲーション構造に対して検証すべき性質としては，全ての Web ページへの到達可能性を考えるものとする．

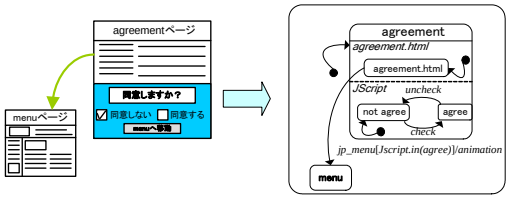


図 2：状態マシン図による
Web ナビゲーションのモデル

(4) モデル自動生成ツールの開発

提案した UML 設計の検証手法を Windows アプリケーションとして実装する．作成したツールの概要を図 3 に示す．このツールは記号モデル検査ツール SMV による検証を想定しており，フロントエンドとして，XML 解析器を，バックエンドとして SMV 入力生成器をもつ．また，UML 図の動的な構造を中間データとして一時保存する機能を持っている．

このツールは，UML 描画ツールで記述したモデルの構造を表す XML ファイルを XML 解析器によって解析して中間データへと保存し， SMV 入力生成器によって中間データから記号モデル検査ツール SMV の入力記述を自動的に生成するものである．

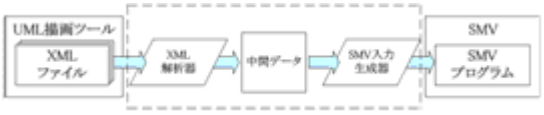


図 3：モデル自動生成ツールの概要

４．研究成果

(1) UML 設計群からの論理式モデル抽出手法の開発

以下に示すように，状態マシン図，シーケンス図およびコミュニケーション図による UML 設計群として記述されたシステムの仕様記述から，統合された一つの状態遷移グラフを求めるための手法を開発した．

まず，状態マシン図については，各遷移一つ一つを一つの論理式として表現し，その上でそれらを論理和結合することで，与えられた状態マシン図で表される状態遷移システムと等価な論理式を求めることができる．さらに，状態マシン図を表す全ての論理式を論理和結合することで，状態マシン図群で表されるシステム全体の遷移関係を表す論理式を求めることができる．

シーケンス図およびコミュニケーション図については，メッセージの送受信処理一つ一つを一つの論理式として表現し，その上でそれらを論理和結合することで，与えられたシーケンス図およびコミュニケーション図で表されるメッセージ処理と等価な論理式を求めることができる．

最後に，状態マシン図群から求めた論理式と，シーケンス図およびコミュニケーション図から求めた論理式を論理積結合することにより，与えられた UML 設計群全てを表す論理式を求めることができる．

(2) 有界モデル検査の効率化

比較実験の結果を表 1 と表 2 に示す．表 1 に示すように，この問題においては到達可能な状態数は Forward Search に比べ，Backward Search の場合が遙かに大きかった．また，有界モデル検査では初期状態からの遷移回数（ステップ数）によって探索する空間を限定し検証を行うが，このステップ数の増加に応じたそれぞれの探索手法におけ

る検証時間の変化について示したのが表 2 である．ここでは目的状態へと到達するステップ数 25 までについて結果を示している．表に示すとおり，Forward Search が検証速度においても大きく優れているという結果が得られた． このような結果となったのは，UML 設計の検証問題においては初期状態が一つに限定されているのに対し，目的状態の数は遙かに大きいため，それにより Backward Search の方が到達可能な状態数が大きくなるためだと考えられる．

表 1：到達可能な状態数の差

	Forward	Backward
状態数	5400	64684

表 2：ステップ数に応じた検証時間(秒)

ステップ数	Forward	Backward
1	0.015	0.016
2	0.016	0.016
…	…	…
22	0.906	6.030
23	2.708	8.813
24	2.327	7.860
25	2.218	14.000

(3) Web ナビゲーション検証への適用

図 2 に示す状態マシン図で表された Web ナビゲーションを論理式表現し，記号モデル検査ツール SMV によって検証する．ここでは，図 2 の状態 agreement. html から状態 menu への遷移を論理式表現する手続きについてのみ述べる．

まず，この遷移につけられたラベル jp_menu [JScript.in (agree)] / animation は，それぞれがイベント[ガード条件]/アクションを表しており，イベントは遷移が起こるきっかけ，ガード条件はその遷移が行われる条件，アクションはその遷移後に行われる処理を意味するものである．この遷移が起こるのは，システムの状態が agreement であり，かつ agreement のサブ状態が agreement.html であるときに，イベント jp_menu が発生したときである．また，この遷移が起こる条件はガード条件 Jscript.in(agree)が満たされたとき，すなわちサブ状態 Jscript の状態が agree であるときである．そして遷移が起こった後にはシステムの状態は menu となり，イベント jp_menu が破棄され，アクション animation が実行される．

従って，この遷移を論理式で表すと以下のようになる．

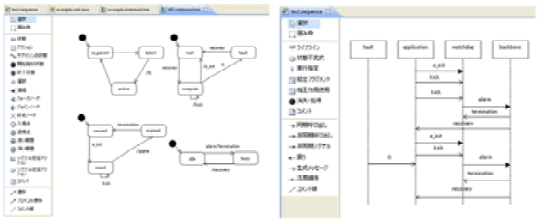
S = agreement ∧ jp_menu
∧ $S_{agreement.html}$ = agreement.html
∧ $S_{JScript}$ = agree
∧ S' = menu
∧ ¬jp_menu'
∧ animation'

ここではイベントの発生とアクションの実行をそれぞれ二値変数として表している．また，ある変数 variable の遷移後の値を variable'として表している．この遷移が起こるのは，この論理式が真であるときかつそのときのみであり，同様にして全ての遷移の論理式表現を求めることができる．

SMV による検証の結果，この状態マシン図で表される Web ナビゲーションが正常に行われることが確認できた．また，検証に要した時間は 1 秒以下であった．

(4) モデル自動生成ツールの開発

ツールの実行例として，入力された UML の状態マシン図とシーケンス図，そして生成された SMV プログラムを図 4 に示す．この SMV プログラムを記号モデル検査ツール SMV で実行することにより，整合性の検証が可能となる．本研究課題で検証に用いる有界モデル検査は記号モデル検査の一種であるため，本ツールは有界モデル検査へと容易に拡張可能である．



(a)

```
MODULE main
VAR
 S_fault:{latent,no_present,active}; S_application:{wait,compute,fault}; S_watchdog:{unused,count,expired};
 p1_1:boolean; p2_1:boolean; p2_2:boolean; p2_3:boolean; p2_4:boolean; p2_5:boolean; p2_6:boolean; p2_7:
INIT
 S_fault = no_present & S_application = wait & S_watchdog = unused & S_backbone = idle & !ft & !kick
 & !p1_1 & !p2_1 & !p2_2 & !p2_3 & !p2_4 & !p2_5 & !p2_6 & !p2_7 & !p2_8 & !p3_1 & !p3_2 & !p3_3 & !p3_4
TRANS
( S_fault = active & next(S_fault) = no_present & next(S_application) = S_application & next(S_watchdog
 & next(ft) = ft & next(kick) = kick & next(e_init) = e_init & next(alarm) = alarm & next(recovery) = reco
|
...
```

(b)

図 4：ツールの入出力例
(a) UML 図　(b) SMV プログラム(一部)

５．主な発表論文等

（研究代表者、研究分担者及び連携研究者には下線）

〔学会発表〕（計 11 件）

[1] S. HARADA, T. YOKOGAWA, H. MIYAZAKI, Y. SATO, and M. HAYASE, "A Tool Support for Verifying Consistency between UML Diagrams by SMV," In The 24th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC2009), 2009 (採録済).
[2] H. MIYAZAKI, T. YOKOGAWA, K. SEKO, Y. SATO, and M. HAYASE, "Formal Verification of Web Navigation by Symbolic Model Checking," In The 23rd International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC2008), pp. 397-340, 2008.
[3] 原田慎士, 横川智教, 宮崎仁, 佐藤洋一郎, 早瀬道芳, "SMVを用いたUML設計の整合性検証ツールの作成," 電子情報通信学会2009年総合大会ISS特別企画学生ポスターセッション, ISS-P-136, 2009年.
[4] 宮崎仁, 横川智教, 佐藤洋一郎, 早瀬道芳, "記号モデル検査によるUML設計間の整合性検証," ソフトウェア信頼性研究会第 5 回ワークショップ(FORCE2009)論文集, pp. 27-34, 2009 年.
[5] 瀬古剛一, 横川智教, 宮崎仁, 佐藤洋一郎, 早瀬道芳, "状態マシン図によるWebサイトのナビゲーション構造のモデル化," ソフトウェア信頼性研究会第 5 回ワークショップ(FORCE2009)論文集, pp. 67-73, 2009 年.
[6] 宮崎仁, 横川智教, 佐藤洋一郎, 早瀬道芳, "記号モデル検査を用いた複数種のUML図設計間の整合性の検証," ウインターワークショップ 2009・イン・宮崎論文集, 2009(3), pp. 79-80, 2009 年.
[7] 瀬古剛一, 横川智教, 宮崎仁, 佐藤洋一郎, 早瀬道芳, "状態マシン図を用いた動的なWebナビゲーションのモデル化," ウインターワークショップ 2009・イン・宮崎論文集, 2009(3), 2009 年.
[8] 宮崎仁, 横川智教, 瀬古剛一, 佐藤洋一郎, 早瀬道芳, "記号モデル検査を用いた状態マシン図とシーケンス図の無矛盾性の検証," 情報処理学会研究報告, 2008-SE-16, pp. 41-47, 2008 年.
[9] 宮崎仁, 横川智教, 佐藤貞仁, 佐藤洋一郎, 早瀬道芳, "有界モデル検査を用いた複数UML図の形式的検証," 電子情報通信学会技術研究報告, 107(505, SS2007-59), pp. 13-18, 2008 年.
[10] 瀬古剛一, 横川智教, 佐藤洋一郎, 早瀬道芳, "記号モデル検査を用いたWebナビゲーションの形式的検証," 電子情報通信学会2008年総合大会ISS特別企画学生ポスターセッション, ISS-P-145, 2008年.
[11] 佐藤貞仁, 宮崎仁, 横川智教, 佐藤洋一郎, 早瀬道芳, "複数のUML図を対象とした記号モデル検査による形式的検証手法の提案," ソフトウェア信頼性研究会第 4 回ワークショップ(FORCE2007)論文集, pp. 69-77, 2007 年.

６．研究組織

(1)研究代表者

横川　智教（YOKOGAWA　TOMOYUKI）
岡山県立大学・情報工学部・助教
研究者番号：50382362

(2)研究分担者

なし

(3)連携研究者

なし