戦略的創造研究推進事業
発展研究（SORST）

研究終了報告書

研究課題
「Ｗｅｂサービス・セキュリティ技術」

研究期間：平成１６年１２月　１日～
平成１９年　３月３１日

中島　震
（国立情報学研究所、教授）

１．研究課題名

Web サービス・セキュリティ技術

２．研究実施の概要

近年、インターネットを活用した新しいビジネスソフトウェアの基盤技術として、Web サービスが注目を集めている。顧客の高度な要求に即座に対応するために、既存 Web サービスを組み合わせて付加価値を生む Web サービス連携の技術が重要である。実際、コンソーシアム組織 OASIS では Web サービス連携記述言語 WS-BPEL の標準化が進んでいる。

Web サービスの基盤となるインターネットは開放型の実行環境であるため、安全性ならびにセキュリティの確保が大切である。複数の Web サービスを組み合わせた高度な Web サービス連携は分散協調システムであり複雑な振舞いを示す。そのため、論理的な機能振舞いがデッドロックなどの不具合を起こさない安全性を持つことを確認しなければならない。さらに、機密レベルの高い情報が不用意に漏洩するようなセキュリティ問題も防ぐ必要がある。

本研究課題では、Web サービス連携記述言語 WS-BPEL で書かれたプログラムが、上記のような安全性ならびにセキュリティの観点から不具合を持つか否かを事前に解析する技術を研究した。解析ツールを試作開発することによって、産業界に示せるような具体的な技術として整備することを目標とする。特に、ソフトウェア科学の成果であるモデル検査法の技術を拡張する研究を行った。

安全性検証の問題は WS-BPEL プログラムを並行システム記述と考えて、並行システムを対象とする検証方法であるモデル検査法を適用すればよい。すなわち、検証対象を有限状態遷移システムとして表現することで、既存のモデル検査ツール Promela/SPIN を利用するという「さきがけ研究」で得た基本的な成果を活用する。一方、WS-BPEL は通常のプログラミング言語と同様な記述を可能とするために、多様な言語機能を提供している。そこで、本研究課題では、中間表現の導入、制御変数の抽象化といった新たな方法を導入した。これらの工夫によって、BPEL4WS(v1.1)仕様書記載の例題全てを正しく解析することに成功した。

情報漏洩の有無を調べるセキュリティ問題を取り扱うためには、WS-BPEL に新たな概念を導入する必要がある。本研究課題では、ラティスに基づくアクセス制御 (LBAC)を用いた情報フローの方法を WS-BPEL に導入した。アクセス主体（ビジネスロジックの実行主体）ならびにアクセス対象にセキュリティレベルを割り振り、レベルの低いほうから高いほうへの情報フローだけを許可するという考え方である。しかし、この基本的な考え方だけでは実用的なシステムへの適用が難しい。本研究課題では、一時的なクラス低下(declassification) と呼ぶ方法を採用した。

WS-BPEL にセキュリティレベルの考え方を導入するためには、セキュリティレベルを定義する XML 形式とセキュリティレベルを付与する WS-BPEL の言語要素を決める必要がある。後者について、Web サービス間で交換する機密文書の不用意な漏洩という問題を取り扱うため、外部 Web サービスという比較的大きな粒度の実体をアクセス対象とした。

情報漏洩の有無を調べる情報フローの検査を行う手順は次の２つの要素からなる。第 1 に WS-BPEL プログラムの実行経路を網羅的に探索して外部 Web サービスへのアクセス（読み出し、書き込み）系列を求める。第 2 に、求めたアクセス系列ごとにクラス一時低下を考慮して情報フローの許可・不許可を調べる。前者の方法として、上記の安全性検証で行ったモデル検査法を用いることができる。

モデル検査法は有限状態遷移システムとして表現された並行システムの振る舞いを調べる手段として有効である。状態遷移システムは主として制御の流れを表現することに向いている。しかし、一般のプログラミング言語のようにデータ計算を取り込むことは難しい。一方、情報フロー解析の問題は、アクセス対象間の文書の流れを表したデータフローを、可能な制御の流れの範囲で網羅的に追跡することである。データフローに沿ってセキュリティレベルを受け渡して順序関係を検査する。一時的なクラス低下を考慮する場合には、その時点までに集めた順序関係の制約条件を伝播させるという問題になる。

本研究課題の安全性検証で用いた Promela/SPIN では、プリミティブなデータ型ならびに配列を提供してデータ値に関わる性質を表現することができる。しかし、順序関係に関わる制約条件を記号的に表現できないため、プリミティブなデータ型にコード化しなければならない。直感的でないためコード化の正しさを保障することが難しいという問題に直面した。そこで、新たな研究のアイデアとして、制約オートマトンを着想した。

制約オートマトンは有限状態遷移システムであり、モデル検査を行うことができる。さらに、記号的な制約条件を明示的に取り扱うことができる。システムの実行を表す状態遷移と共に、変数に対する制約条件を追加していく。新たな制約条件の追加時に、既に集めた制約条件を調べて、充足不能であれば不具合があると判定する。一方、制約が充足可能のまま終了状態に至れば正常終了とみなす。

本研究課題では、情報漏洩の解析を行うために次のように制約オートマトンを用いる。すなわち、セキュリティレベル間の順序関係を記号的に制約条件として表現し、新たなアクセス発生に伴う制約追加時に、集めた順序関係が矛盾しないかの制約検査を行えばよい。制約を記号的に表現することが必須であるため、代数仕様言語 Maude とそのモデル検査ツールを用いて、制約オートマトンの考え方を整理した。予備的な実験を行って制約オートマトンを用いる方法の有効性を確認した。

本研究課題では、試作ツールを Eclipse フレームワークのプラグインとして開発した。動作環境は WindowsXP Professional SP2、Java (SDK 1.4.2)、Eclipse SDK Version 3.1.1 である。XML 文書解析、グラフィックフレームワークなどを利用できるため、Eclipse を開発環境として選択した。現状では、モデル検査ツールは Promela/SPIN を用いるように構成している。

一般に新しいソフトウェアに関わる技術を研究する場合、そもそもの問題定式化が重要な役割を果たす場合が多い。昔から知られている未解決問題を解く、いった研究とは全く異なる難しさがある。新たに登場した技術の延長上に、何が問題となって覆いかぶさるかを予測する必要があるからである。

実際、Web サービスを稼動させる技術、Web サービスを連携させて新しいサービスを組み上げる技術、などの新しいソフトウェアが産業界で着実に開発された。一方、利用者が重要と考える Web サービスの信頼性については後手に回っている。

本研究課題では、従来から知られていたソフトウェア科学の成果であるモデル検査法という基礎技術を背景に、Web サービス連携の技術における信頼性に関わる問題点を指摘し、その解決のひとつの方法を示した点に重要性がある。ほぼ同時期に、欧米各国でも、同様な狙いの研究が進行し発表された。初期に発表した論文が先行研究として引用されている事実から、ここに述べた本研究課題の意義を裏付けることができよう。

最後に、欧州では、EU や国家支援の競争的研究資金の後押しで、Web サービス連携に関する研究活動が活発化している。特に、ソフトウェア科学をベースに、ビジネスプロセス・

モデリングと Web サービス連携技術の研究交流が見られる。アメリカの産業界から登場したサービス指向コンピューティングへの対抗意識もあるのか、将来のビジネスソフトウェア基盤に対する戦略的な技術への研究投資と位置づけていることが読み取れる。

３．研究構想

Web サービスの基盤となるインターネットは開放型の実行環境であるため、安全性ならびにセキュリティの確保が大切である。複数の Web サービスを組み合わせた高度な Web サービス連携は、分散協調システムであり複雑な振舞いを示す。そのため、論理的な機能振舞いがデッドロックなどの不具合を起こさない安全性を持つことを確認しなければならない。さらに、機密レベルの高い情報が不用意に漏洩するようなセキュリティ問題も防ぐ必要がある。

2004 年時点でまとめられた IT 業界調査会社の報告概要によると、Web サービスは、ビジネスの立ち上げ段階にある技術であるとされていた。特に、既存 Web サービスを組み合わせて新たなサービスを作り出す連携の仕組みが求められていた。これを受けて、IBM とマイクロソフトが中心になり、Web サービス連携記述言語 WS-BPEL を共同提案した。その後、関連企業の賛同を得て、Web サービス関連技術のコンソーシアム OASIS にて標準化することが検討されている。新しいビジネス基盤ソフトウェアとして大きく期待されている。しかし、OASIS での議論は各社が持つ関連知的所有権のしがらみなど、ビジネス上の駆け引きが中心にならざるをえない。安全性ならびにセキュリティといった本当に重要な側面に対する議論が遅れている。

本研究課題では、OASIS コンソーシアムで提案されている Web サービス連携記述言語 WS-BPEL で書かれたプログラムが、上記のような安全性ならびにセキュリティの観点から不具合を持つか否かを事前に解析する技術を研究する。解析ツールを試作開発することによって、産業界に示せるような具体的な技術として整備することを目標とする。

WS-BPEL プログラムを対象として、セキュリティの観点からの解析を行い、情報漏洩があるかないかを解析するツールを、ソフトウェア科学の成果であるモデル検査の技術を拡張して実現する。具体的に以下の研究項目がある。

(a) 抽象化の方式 : 一般に、WS-BPEL プログラムでは制御フローに影響を与える変数が存在する。モデル検査の際には、具体的な値が確定しないため、制御経路を非決定的に選択する抽象化を行う。しかし、抽象化によって本来は存在しない実行経路が発生することがある。この結果として生じる見かけの不具合(false alarm)を極力減らす抽象化方式を検討する。

(b) WS-BPEL 仕様の拡張 : 情報漏洩に関する解析を行うためには、個々のデータに機密レベルを付与し、高いレベルから低いレベルへのデータ流入がないことを確認する。この考え方を導入するために、セキュリティレベルを定義する等の新たな言語要素を WS-BPEL に追加する。

(c) モデル検査法の応用 : 情報漏洩の解析を行うためのセキュリティレベルに関する制約処理と制御フロー解析を中心とするモデル検査の考え方を統合し、モデル検査ツール Promela/SPIN で実現する方式を検討する。特に、制約処理の正しさを確認できる方法を工夫する。

(d) GUI ツールの試作 : 考案した方式を具体的に示すこと、最終的なユーザの利用イメー

ジ・手順を検討することを目的として、GUI ベースの支援ツールを試作開発する。これによって、産業界に示せるような具体的な技術として整備することを意図する。外部ソフトウェアハウスに本試作ツールの開発を依頼する。

本研究課題は個人研究であるため研究計画を柔軟に設定することが可能である。また、もっとも手間のかかるツール試作の負荷を適切に分散させるために、大きく分けて 2 段階の開発を行う。すなわち、項目(a)を先行させ項目(d)の試作第 1 版をまず完成させ、項目(b)ならびに(c)の検討を行って項目(d)第 2 版の試作開発を行う。

項目(a) については、WS-BPEL から Promela/SPIN への変換を見通しよく系統的に行うために、中間的な表現形式として拡張有限オートマトン(EFA)を導入するアイデアを得た。両者の中間に位置するような情報を表現できる体系であり、EFA を用いることで変換が正しいことの確認が容易になる。また、公開されている BPEL4WS(v1.1)記載の 4 つの例題が処理できるか否かによって検討方式の良し悪しを確認することにした。

項目(b)については、新規の提案であり、セキュリティレベルを表現する有限ラティス構造を XML 文書の形式で定義する。

項目(c)については、Promela/SPIN を用いる方法を検討し、試作することでツールとして実現できるか否かを確認する。さらに、実現したツール機能に誤りがないことを系統的に保障する方式を検討する。

他方、項目(c)について、実際に研究を進める過程で、実現した方式の正しさを保障することが難しいことが判明し、全く異なる新たな方法を検討する必要が生じた。新しい方式である制約オートマトンは、Promela/SPIN で実現することは難しいが、他のツール(Maude)を用いることで実現可能である。これが系統的な解決アプローチであることを確認するために、制約オートマトンに関する基礎研究を行うという新たな目標が生まれた。

最後に、本研究課題における研究内容と先行して実施した「さきがけ研究」との関係について整理する。さきがけ研究では、Web サービス連携の分野で安全性と情報漏洩に関する問題が未解決であることを指摘し、モデル検査法を用いる方法を提案した。WS-BPEL の基になった WSFL を対象としてモデル検査法の適用効果を実証したこと、WS-BPEL を対象としてセキュリティレベルに基づく情報漏洩解析の方式を提案したこと、が具体的な成果である。本研究課題では、WSFL に対する研究成果を基に WS-BPEL に対する安全性解析を実現し、また、情報漏洩解析の方法を具体化することが主な狙いである。そのために、解析ツールの試作を通して、技術を具体化することに主眼を置くこととした。

４．研究実施内容

(1)実施の内容

【Web サービス連携】　近年、インターネット上の Web 技術が新しいビジネスソフトウェアの基盤として注目を集め、サービス指向コンピューティングと呼ぶ概念が提案された。専門性の高いビジネスパートナーと連携することで顧客の高度な要求に即座に対応するビジネスプロセス連携を実現するためのソフトウェア基盤である。このようなソフトウェアを構築するためには多くの要素技術をベンダ間で共通化し相互接続性を高める必要がある。そこで、情報閲覧の仕組みとして考案された Web 技術を拡張する Web サービスと呼ぶ技術体系が提案された。

Web サービスの技術では、通信の仕組みだけではなく、交換する情報あるいはビジネス文書の形式を共通化して XML で表現する。共通化することによって、サービス提供者が送った文書を解読むことが容易になる。ところが、有用な新しいサービスを提供するためには、１対１での情報交換だけでは不十分である。複数のサービス提供者と協調して情報交換することで高度なサービスを実現する必要が出てくる。すなわち、Web サービス連携の仕組みが必要になる。

Web サービス連携の中心となる技術は複数サービス提供者の分散協調動作記述である。各サービス提供者はネットワーク上に分散して存在し自律動作する。図１にその様子を模式的に示した。中央の Web サービスは複数の外部 Web サービスを連携させ新たな付加価値をうむ。外部 Web サーバは各々が独自に自律実行するため、必要に応じて、同時に処理依頼することにより作業効率を向上させることができる。図１では、同時依頼する並行処理の様子を示している。

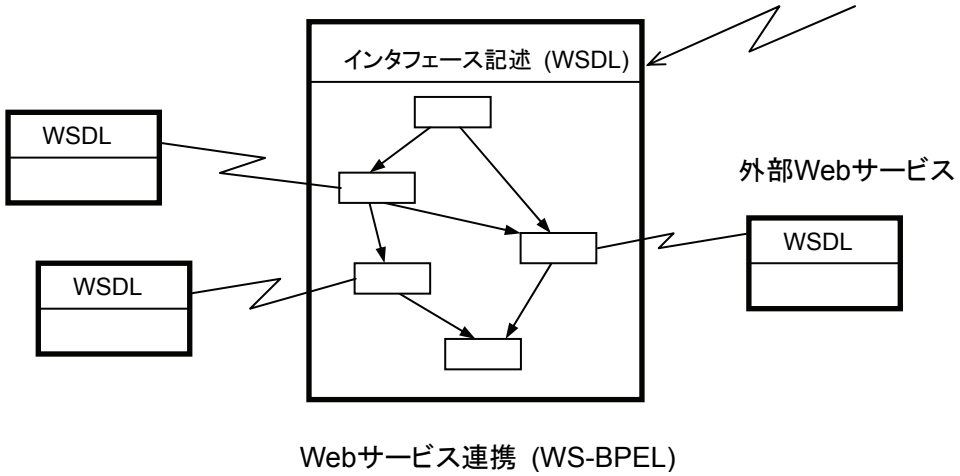


図１　Web サービス連携

Web サービス連携の代表として「旅行代理店」の例が説明されることが多い。今、図１の中央に位置する Web サービスが旅行代理店とする。外部の顧客から旅行計画と必要な予約を依頼された場合を考える。旅行代理店は、航空会社、ホテル、レンタカー会社等のビジネスパートナーに問い合わせを行い、顧客の希望に合った旅行計画を作成する。複数のホテルに空き室状況を同時に問い合わせる等の並行的な処理を行って効率よく作業を進める。

個々の航空会社やホテルを独立した外部 Web サービス提供者とすると、旅行代理店は Web サービス連携によって自身の付加価値を高める Web サービスであることがわかる。

【WS-BPEL】　Web サービス体系は多種多様な記述体系を採用している。第 1 に、外部に対して提供するサービスの種類を公開するためのインタフェース記述を行う必要がある。そのために、WSDL (Web Service Description Language) が標準化された。利用者はサービス提供者の WSDL 記述を調べることで、当該 Web サービス提供者との情報交換のやり方を知ることができる。図 1 に示したように Web サービスは WSDL 記述を持つ。

第 2 に、図 1 に示したような Web サービス連携の記述言語が必要になる。現在、WS-BPEL (Web Service Business Process Execution Language) が提案されて標準化するための検討が進行中である。WSDL がサービス提供者のインタフェースを記述する静的なデータ型定義言語であるのに対して、WS-BPEL は協調動作を表現するためのプログラムである。両者とも、XML 文書の形式を採用することで、ベンダ間で共通に使えることを目的としている。

WS-BPEL ではプログラムを構成する要素をアクティビティ(activity)と呼び、外部 Web サービスと情報交換を行うアクティビティ、並行動作、逐次実行、条件分岐などを行う制御アクティビティ、普通のプログラミング言語の変数に相当する基本要素などを持つ。WS-BPEL は多様な機能を提供するため仕様の大きな言語である。後に、本研究課題で取り扱う WS-BPEL サブセットの範囲を示す。

【Web サービス連携の問題点】 WS-BPEL を用いれば、複数のサービス提供者を連携させるプログラムを作成することが可能になる。しかし、信頼できる Web サービス連携の技術という観点からみると、WS-BPEL プログラムには 2 つの大きな問題点があることがわかった。第 1 に、分散協調動作記述であるためにプログラム作成者が見落としているような複雑な振る舞いを示すことがある。考え落ちによって処理が進行しないデッドロックと呼ぶ不具合に陥るかもしれない。そのため、論理的な機能振る舞いを対象とする安全性の問題を考えなければならない。第 2 に、オープンなネットワークを介して情報をやりとりするために不用意な情報漏洩というセキュリティ問題に対処しなければならない。

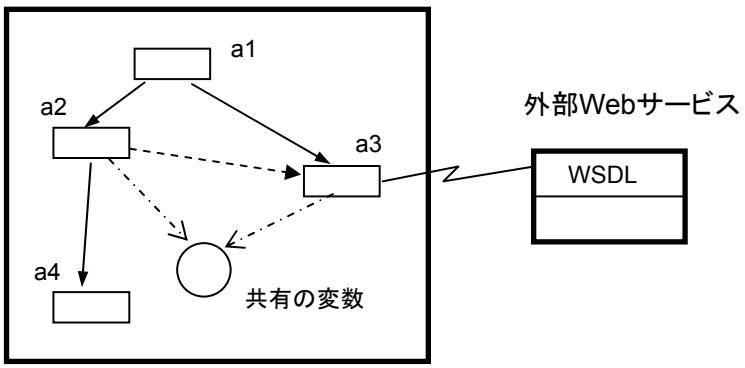


図２　デッドロックの不具合が発生する例

デッドロックを起こして処理が進行しない簡単な例を図 2 に示す。アクティビティ a1 の実行後、2 つのアクティビティ a2 と a3 とが並行実行する状況を示す。また、アクティビテ

ィ a2 から a3 に制御の先行後続関係を示すリンクが設定され、さらに、両者は同じ変数を共有する場合を考える。このとき、たまたま、a2 が a3 よりも先に実行されると問題なく処理が進行する。一方、逆に、a3 が先に実行されて共有変数を獲得するとデッドロックに陥る。すなわち、a3 は a2 が作動してリンク情報が設定されることを待つ、一方、a2 は共有変数が解放されて利用可能になるのを待つ。両者とも相手の処理完了を待つというデッドロックの状況が発生する。

図 2 の例は 2 つの問題があることを示している。第 1 に、WS-BPEL を用いる場合、上例のようなデッドロックの可能性を持つプログラムの作成が可能である。第 2 に、実際のインターネット環境で当該の WS-BPEL プログラムを実行すると、まわりの状況によって、うまく作動する場合もあればデッドロックになる場合もある。一般に、インターネットを利用するシステムは多様で複雑な要素が絡むため、不具合原因の究明が難しいという問題がある。そのため、事前に、WS-BPEL プログラムを解析してデッドロックのような不具合があるか否かを調べておく必要がある。

2 番目のセキュリティ問題を説明するために図 3 を用いる。一般にビジネス文書には、誰でも読むことが可能な公開情報のほかに、社外秘や部外秘といった特定の人だけが参照できる機密レベルの考え方がある。そのような文書を参照する WS-BPEL プログラムは機密性の高い文書が公開サーバに漏洩しないことを保障しなければならない。図 3 で、業務ロジック中のアクティビティ b2 が機密文書を読み込み、b4 は公開情報を参照する。このとき、前者はアクティビティ b3 によって安心なサーバに移送されてもよいが、b5 によって公開サーバに渡されてはならない。一方、b4 が読み込んだ情報は公開サーバに移されても良い。

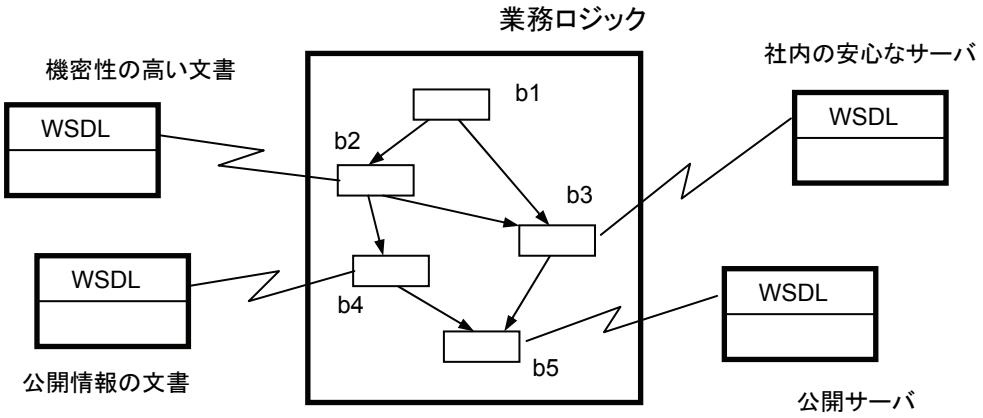


図 3　情報漏洩が問題となる例

上記の情報漏洩は通信の暗号技術だけでは解決できない問題である。業務ロジックを実現する WS-BPEL プログラムは b2 で取得した文書の中身を参照して次の動作を行う。文書の内容を参照するためには暗号化された文書を復号化しなければならない。実際、Web サービスの技術体系では、オープンなネットワークを介して情報をやりとりする際に通信過程での情報漏洩を防ぐ手段として、暗号技術を用いた WS-Security がある。一方、本研究課題では、暗号技術だけでは達成困難な情報フロー解析と呼ぶ問題を WS-BPEL プログラムに対して考える。

【安全性の検証】　安全性検証の問題は WS-BPEL プログラムを並行システム記述と考えて、並行システムを対象とする検証方法であるモデル検査法を適用すればよい。すなわち、検証対象を有限状態遷移システムとして表現することで既存のモデル検査ツール Promela/SPIN を利用することができる。この考え方は、さきがけ研究の際に、当時提案されていた WSFL を対象として行った研究の際に採用したものである。WS-BPEL の並行実行制御（flow アクティビティ）は WSFL の考え方を踏襲しているので、基本的な方法はさきがけ研究のときの成果を用いればよい。特に、DPE(Dead-Path Elimination)と呼ぶ方法は WSFL と WS-BPEL で同じである。しかし、WS-BPEL は通常のプログラミング言語と同様な記述を可能とするために、多種多様な基本アクティビティを導入した。そのために、本研究課題では、中間表現形式の導入、制御変数の抽象化といった新たな方法を採用した。

本研究課題では、WS-BPEL から Promela への変換を見通しよく行うために、中間的な表現形式として拡張有限オートマトン(EFA)を導入した。EFA は有限オートマトンに変数の概念を追加したものである。WS-BPEL の変数を役割に応じて 3 種類に分類し、おのおのを明示的に取り扱えるようにしたことで、制御変数の抽象化方法を導入できるようになった。試作したツールの処理概要を図４に示す。

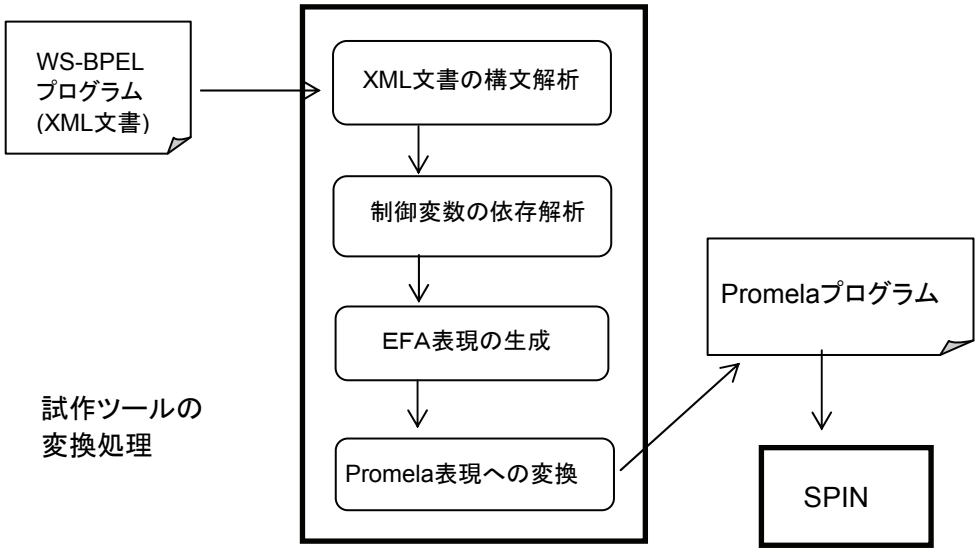


図４　試作ツールの処理概要

図４の処理概要を説明する。第１に、入力 WS-BPEL プログラムである XML 文書の構文解析を行って内部表現の構文木に変換する。このとき、XML 文書としての整形式であるかを検査する。第 2 に、制御変数の依存解析を行って抽象化に必要な関係を求める。第 3 に、EFA に変換し、最後に、EFA 表現から Promela プログラムを変換生成する。Promela プログラムは SPIN の入力形式である。これによって、モデル検査法を用いた解析が可能になる。デッドロックがある場合には不具合の状況を出力するので、これを参考にして元の WS-BPEL プログラムの不具合を修正する。中間表現として EFA を採用することによって、WS-BPEL プログラムと Promela の対応関係を明確化することができた。

アクティビテ	機能の説明
invoke receive reply	外部Webサービスのオペレーションを起動 外部からの要求メッセージ待ち 外部への返答メッセージを生成
assig	変数的な要素に値を代入
sequence switch while flow	逐次実行制御 多重の条件分岐 繰り返し 並行実行制御
scope	共有変数スコープを設定

表１　試作ツールの対象 WS-BPEL アクティビティ

なお、試作ツール開発に際して、WS-BPEL が提供する基本アクティビティをサブセットに限定した。表１に示すように、デッドロック等の並行システム特有の問題点を解析するために必要な機能は網羅している。扱わなかった機能で重要なものとしては、例外ハンドラ等がある。これらは WS-BPEL でも明確な定義が与えられておらず、システム依存になっているようである。実際、例外ハンドラの厳密な定義を与えるという研究がいくつかあり、現在も検討中の機能である。

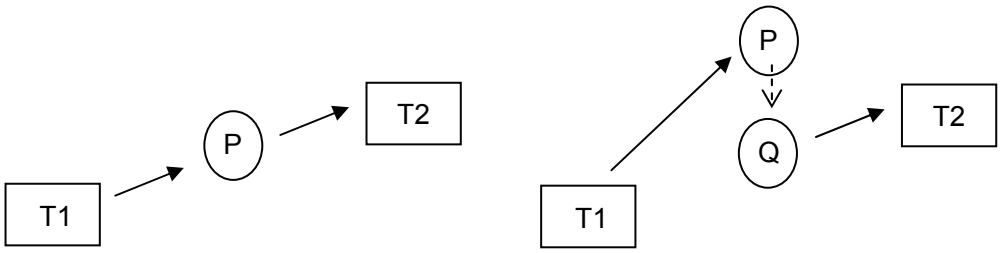
名称	特徴的なWS-BPEL	状態数	必要な手法
Purchase Order	変数、並行実行制御	249	基本的な解析手法のみ
Shipping Service	多重条件分岐、繰り返し	21	制御変数の抽象化
Loan Approval	並行実行制御	3,516	制御変数の抽象化、DPE
Auction Service	複数開始点	57	基本的な解析手法のみ

表２　試作ツールによる例題の解析結果

表２に試作ツールを用いて BPEL4WS (v1.1)標準化文書記載の 4 つの例題を解析した結果を示した。各々特徴的な WS-BPEL 機能を用いており、それに応じて、解析に必要となった手法を併記した。また、状態数の項は Promela/SPIN を用いてモデル検査法によるデッドロックの有無を調べた際の数字である。数が大きいほど複雑な処理を行っている例題と考えることができる。たとえば、３つめの Loan Approval では 5 つの並行アクティビティが作動するので最も複雑さが大きい。なお、さきがけ研究で行った際の WSFL 検証実験では 100 万状態に達するなどスケーラビリティの観点から方式の妥当性が低かった。表２のように状態数を小さくすることができたのは、WS-BPEL が表１のような高水準アクティビティを持つことが大きな理由である。また、試作ツールで用いた抽象化の方法も寄与していると考えている。

【セキュリティ問題の検証】　セキュリティ問題を取り扱うためには WS-BPEL に新たな概念を導入する必要がある。ここでは図２に示したように、不用意な情報漏洩の問題があるか否かを調べることである。本研究課題では、ラティスに基づくアクセス制御の方法(LBAC)を WS-BPEL に導入する。

LBAC は、アクセス主体（ビジネスロジックの実行主体）ならびにアクセス対象にセキュリティレベルを割り振り、レベルの低いほうから高いほうへの情報フローだけを許可するという考え方である。たとえば、図５(a)では、アクセス主体 P のレベルが情報供給者である Web サービス T1 よりも高く、さらに、情報格納先の T2 よりも低い状況を示す。この時、レベル T1 の情報は読み出すことが許可されて、意図通りの格納先に移すことができる。なお、セキュリティレベルは全順序関係を満たすとしても良いし、半順序関係の場合であっても良い。セキュリティレベルの全体は有限ラティスを構成する。



(a) 許可される情報フロー　　(b) 一時的な低下

図５　情報フロー

上記の基本的な方法だけでは実用的なシステムに LBAC を適用できないことが多い。図５(b)はそのような状況の例を示す。情報提供元の T1 と格納先の T2 のレベル比較では情報の移送が許可されるべきである。一方、T1 から情報を読み出したアクセス主体 P のレベルが T2 よりも高い場合、P から T2 への情報フローが許可されない。不用意な情報漏洩を排除するという観点からはこの制限は正しい。しかし、アクセス主体が信頼できる場合、レベル P の値が高すぎるために本来許可したい情報フローを排除してしまうという問題がある。アクセス主体が信頼できるという条件の下で何らかの対応策を考えるべきである。

上記の問題を解決するために、アクセス主体 P のレベルを一時的に下げるクラス低下（Declassification）と呼ぶ方法を採用する。丁度 T1 と T2 の中間に位置するようなレベル Q を見つけて、この値をアクセス主体のレベルとすればよい。Q の値は予め決まっているわけではない。たとえば、T1 よりは高い等のアクセス主体に予め付与された条件、さらに、アクセス対象のレベルが決まった時に、アクセス可能とするための条件などから計算する。すなわち、集めた順序関係に矛盾しないような値として Q を計算すればよい。集めた順序関係に矛盾があれば一時低下が不可能であり、その結果、情報フローが許可されないという結論を導くことができる

【WS-BPEL へのセキュリティレベル導入】　WS-BPEL にセキュリティレベルの考え方を導入するためには、セキュリティレベルを定義する XML 形式とセキュリティレベルを付与する WS-BPEL の言語要素を決める必要がある。前者は新たに定義すればよいが、後者についてはいくつかの方法が考えられる。

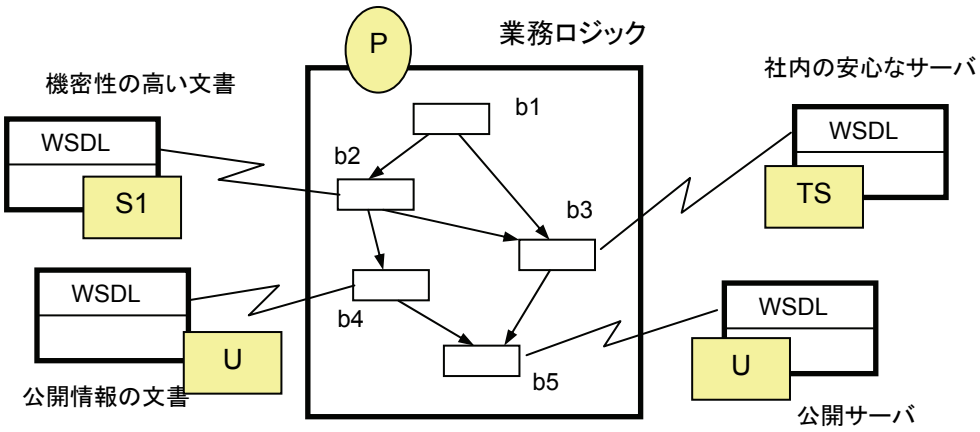


図６　セキュリティレベルの付加

本研究課題では、図６に模式的に示したようにした。すなわち、アクセス主体は Web サービス連携記述である WS-BPEL プログラムの実行主体にする。これは WS-BPEL プログラム上は明示的に現れない。通常のプログラミング言語の用語であれば実行スレッドに対応する仮想的な実体である。また、外部 Web サービスが情報の提供元ならびに情報格納先になるので、これをアクセス対象とした。WS-BPEL プログラムではデータ値を保持する変数があるが、これは当該プログラム実行の終了と共に消滅する一時的な実体であるためアクセス対象にしなかった。すなわち、外部 Web サービスという比較的大きな粒度の実体をアクセス対象として選んだ。

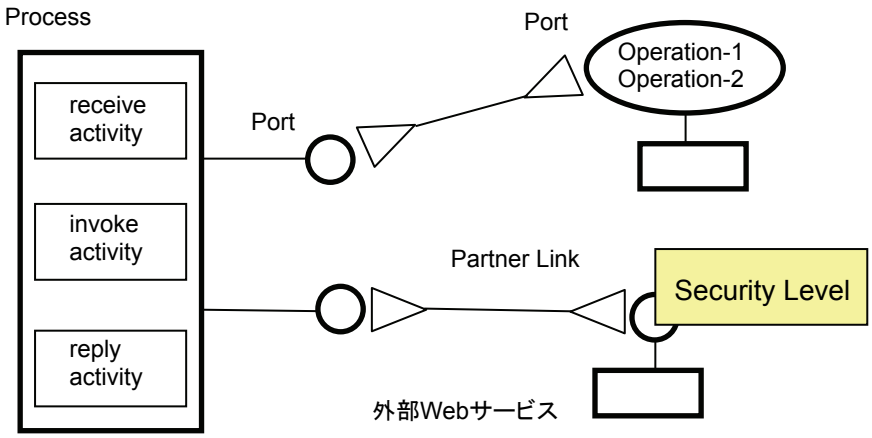


図７　セキュリティレベルつきポート

もう少し詳しく説明する。は図７に示したように、WS-BPEL プログラムからの外部 Web サービスの見え方であるポート(Port)を拡張してセキュリティレベルを付加するようにした。Web サービスの技術体系では柔軟性に力点を置くため実際の外部 Web サービスは実行時になってはじめて決まる。WS-BPEL プログラムからわかるのは、どのようなポートを持つかという情報だけである。そこで、ポートにセキュリティレベルを追加することとした。

【モデル検査法に基づく情報フロー検査】　情報フローの検査を行う手順を整理すると次

の２つの要素からなることがわかる。第１に WS-BPEL プログラムの実行経路を網羅的に探索して外部 Web サービスへのアクセス（読み出し、書き込み）系列を求める。第 2 に、求めた系列ごとにクラス一時低下を考慮して情報フローの許可・不許可を調べる。

第 1 の点については、与えられた WS-BPEL プログラムの制御フローを追跡して可能性のある実行経路を調べる問題であるため、安全性の検証で採用したモデル検査法を用いればよい。第 2 の点を統合するためには、モデル検査法に基づく経路の網羅的な探索過程で、外部 Web サーバへのアクセスごとに、セキュリティレベル間の順序関係を集める方法を採用する。新たな順序関係が追加された際に、集めた順序関係が矛盾するか否かに関する制約検査を実施する。制約が満たされなければ情報フロー違反として直ちに検査を終えればよい。一方、すべての経路の探索ならびに制約検査に成功すれば、当該 WS-BPEL プログラムに情報漏洩の問題がなかったと結論つけることができる。

上記の方法をモデル検査法の観点から整理する。モデル検査法は有限状態遷移システムとして表現された並行システムの振る舞いを調べる手段として有効である。状態遷移システムは主として制御の流れを表現するのに向いている。本研究課題での安全性検証で行ったように制御変数を導入し、さらに、適切な抽象化方法を採用することによって、変数値に依存した制御フローを取り扱うように拡張することは可能である。しかし、一般のプログラミング言語が行うようなデータ計算を取り込むことは難しい。

情報フロー解析の問題は、換言すると、アクセス対象間のデータフローを可能性のある制御の流れの中で全て追跡することである。データフローが生じるか否かだけではなく、当該データフローに沿ってセキュリティレベルを受け渡し、必要に応じてこれを検査する。特に、一時的なクラス低下を考慮する場合には、具体的なセキュリティレベル値ではなく、その時点までに集めた順序関係の制約条件の集まりを受け渡すという問題である。

安全性の検証でモデル検査ツールとして用いた Promela/SPIN は、bool 型や int 型といったプリミティブなデータ型ならびに配列を提供してデータ値に関わる性質を表現することを可能にしている。したがって、順序関係に関わる制約条件という記号的な表現が相応しい対象を、Promela/SPIN が提供するデータ型にコード化する必要がある。本研究課題の試作では、取り扱うセキュリティレベルの種類が有限であることを利用してコード化を行った。ところが、直感的でないためコード化の正しさを保障することが難しいという問題に直面した。そこで、新たな研究の展開を考える必要が生じた。

【制約オートマトン】　制約オートマトンは有限状態遷移システムであって、通常のモデル検査の方法に加えて、制約条件を明示的に取り扱うことを可能とする。この方法は、システムの進行、すなわち、状態の遷移と共に、変数に対する制約条件を追加していく。新たな制約条件の追加時に、既に集めた制約集合を調べて、充足不可能であることがわかれば不具合があると判定する。一方、充足可能のまま終了状態に至れば、変数に対する制約を解いて変数値を定めることもできる。すなわち、変数の値を個別に扱うのではなく制約条件の集まりとして記号的に扱うことで、データ値の系統的な取り扱いが可能となる。。

本研究課題との関連では、セキュリティレベル間の順序関係を記号的に制約条件として表現し、新たな制約追加時に集めた順序関係が矛盾しないかの制約検査を行うことで情報漏洩の解析を実現できることになる。特に、制約を記号的に表現することが必須であるため、代数仕様言語 Maude とそのモデル検査ツールを用いて、制約オートマトンの考え方を整理し、予備的な実験を行い良好な成果を得ることができた。今後、さらに研究を深める必要があると考えている。

(2)得られた研究成果の状況及び今後期待される効果

本研究課題では試作ツールを外部ソフトウェアハウスに依頼して開発した。図８に試作ツールの画面例を示した。本試作ツールは、図９の構成概要にあるように、Eclipse フレームワークのプラグインとして開発した。動作環境は WindowsXP Professional SP2、Java (SDK 1.4.2)、Eclipse SDK Version 3.1.1 である。XML 文書解析、グラフィックフレームワークなどを利用できるため、Eclipse を開発環境として選択した。現状では、モデル検査ツールとして、Promela/SPIN を用いるように構成している

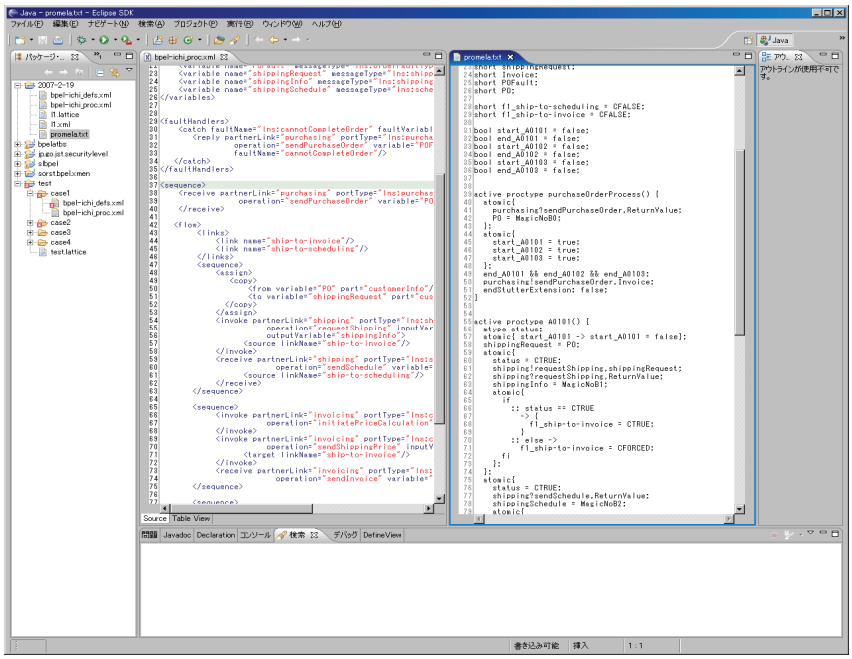


図８　試作ツールの画面例

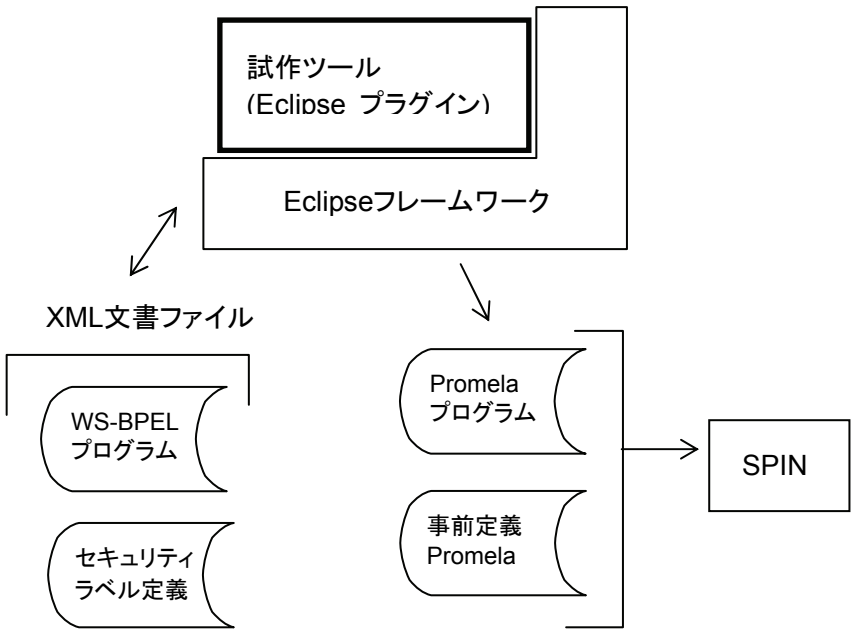


図９　試作ツールの構成

さて、現時点で、Web サービス技術はビジネス面の立ち上がりは遅いように思える。IBM やマイクロソフトといったリーダ的な役割を果たす企業が推進の中心にあるとはいえ、ベンダ間の相互接続性を達成するために多くの課題が待ち構えている。技術的な課題と同時に、標準化の調整にはビジネスの力学が働くため進み方が遅いと考えられる。

純粋に技術的な面から考えると、数多く提案された関連技術（WS-＊と総称する）によって、サービス提供者ならびに利用者が手軽に Web サービスの機能を稼動させることができるようになっている。単独の Web サービスを支える技術は確立し、安定して作動させることが可能になった。一方、本研究課題の問題意識にもあった Web サービス連携の枠組みに関しては、BPEL4WS が 2003 年に提案されて OASIS での標準仕様 WS-BPEL の議論が 4 年にわたって続いている。ようやく 2007 年になって標準化の投票を行うというアナウンスがあった。これによって、既存 Web サービスを連携させて新たなサービスを定義する方法が確立すると期待できる。

一方、利用者の立場から考えると、提供される Web サービスが信頼できるものでなければならない。高い信頼性を達成するためには、非常に多くの技術が絡むことは事実である。本研究課題では、WS-BPEL プログラムを対象として不具合による停止がないこと、ならびに大切な情報が不用意に漏洩しないこと、を検査する方法の研究を行った。真に実用的な高信頼 Web サービスを提供するためには、幅広い技術が必要になるので、本研究課題は、その一部について試作ツールを開発しただけである。しかし、モデル検査法というソフトウェア科学の基礎技術を Web サービス連携の検証に応用することで、技術的な解決アプローチのひとつを示せたと考えている。

関連研究の動向について述べる。不具合による停止があるかないかを調べる安全性の検証については、2003 年以来、欧米でも多くの研究成果が発表されている。基礎とするソフトウェア科学の理論によって幾つかのアプローチがあり、ペトリネット、モデル検査法、プロセス代数、イベント計算、などが用いられている。WS-BPEL に適用する工夫に関する基本的な研究が多い。本研究課題のように、BPEL4WS(v1.1)仕様書記載の 4 つの例題を正しく解析できたことを明確に述べた研究は他にはない。この分野の先行研究として引用されることが多い。

情報漏洩の有無を調べる研究については、プロセス代数を用いた理論研究があるが、先に述べたような他手法で試みる研究は他にはない。2006 年になって IBM 東京基礎研から Web サービスを対象とする情報漏洩の解析を型理論の方法でアプローチする研究が発表された。情報漏洩に関わる検査は重要であるため、今後、研究が増えていくと思われる。

欧州では、Web サービス連携に関する研究活動が全般的に活発化している。2001 年に IBM が提案した WSFL の中心人物 Frank Leymann 博士は、BPEL4WS の活動でも先導的な役割を果たしてきた。現在はドイツのシュツットガルト大学教授として、標準化活動を精力的に行うと同時に、ドイツ政府の資金を得て Correctness and reliability of composed web services modeled in BPEL と呼ぶ研究プロジェクトを続けている。EU の研究資金を背景に、SENSORIA 等の研究プロジェクトにおいて、ソフトウェア科学の成果を Web サービスに応用する研究が進行中である。英国とイタリアが中心であるため、プロセス代数によるアプローチが多い。また、欧州では、ソフトウェア科学の立場をベースに、ビジネスプロセス・モデリングと Web サービス連携の研究交流が増えている。アメリカの産業界から登場したサービス指向コンピューティングと狙いは同じで、将来のビジネスソフトウェア基盤と位置づけていることが読み取れる。

５．類似研究の国内外の研究動向・状況と本研究課題の位置づけ

Web サービス技術に基づくビジネスプロセス記述を形式検証することでデッドロック等の不具合がないことを確認する研究は 2001 年頃にはじまった。2001 年 5 月に開催された国際学会WWWで、Narayanan and McIlraith がセマンティックＷｅｂ技術を用いたＷｅｂサービス連携記述をペトリネットの方法で形式検証する研究を発表したのが最初である。

同じ 2001 年 5 月には、WSFL(IBM)と XLANG(マイクロソフト)の提案書が同時期に公開された。これを受けて、Nakajima は 2002 年 1 月にモデル検査ツール Promela/SPIN を用いた WSFL 記述の形式検証に関する構想を発表した。このワークショップでの発表が標準化を狙って提案された Web サービス連携記述を対象とする形式検証の最初であると思われる。ここでの構想は、さきがけ研究として提案・実施されて、2002 年 11 月(シアトルの OOPSLA ワークショップならびに東京で開催された国際会議 CW2002)に成果の一部が公表された。

一方、IBM とマイクロソフトの両社は標準言語をひとつに統一するために、関連する他会社と協力して BPEL4WS を提案し、2002 年 7 月に第 1 版の資料を公開した。現在、コンソーシアム組織 OASIS で標準化の議論が進行中の WS-BPEL の源流である。OASIS の WS-BPEL は 2003 年 5 月の BPEL4WS (v1.1) を基とし、多くの企業を巻き込んで標準化の活動を行ってきた。当初の予定に比べて大幅に遅れたようであるが、ようやく本年(2007 年)4 月に WS-BPEL2.0 の標準化に関する投票が行われることになった。言語機能の大枠は BPEL4WS(v1.1)から大きな変更はない模様である。

2003 年以降、欧米で本分野の研究発表が増えている。2003 年以降に公表された論文では、WSFL や XLANG ではなく、BPEL4WS を対象とした形式検証の技術を研究している。しかし、Narayanan and McIlraith と並んで Nakajima の論文を先行研究として引用している論文が多数見られる。以下では、BPEL4WS との細かな違いを無視して、包括的に、WS-BPEL 記述の形式検証と呼ぶ。

WS-BPEL 記述の形式検証に関しては、Narayanan and McIlraith が採用したペトリネット、Nakajima が採用した Promela/SPIN によるアプローチ以外に、プロセス代数を用いる方法があり主として欧州で人気のある研究テーマである。Foster et al は FSP と呼ぶ有限プロセス代数と LTSA モデル検査ツールを用いた。Van Breugel and Koshkina は CCS を用いて WS-BPEL 記述を表現し CWB-NC ツールで解析した。Salaun et al も CWB-NC を用いた方法を公表している。なお、プロセス代数による方法は、WS-BPEL と強い関係がある言語 WS-CDL の振る舞い検証にも使われている。代表的なものとして、在英の Yoshida and Honda グループによる理論研究がある。なお、Promela/SPIN を用いる研究としては、Fu et al によるものが、モデル検査ツールが不得意なデータに関する性質を議論しようとする試みであり、プロセス代数が不得意な性質を議論できるので興味深い。

Web サービスは本質的に単独コンピュータに閉じないネットワーク連携であるため、情報漏洩に関わるセキュリティの問題を取り扱わざるを得なくなる。標準化技術として議論されている WS-Authorization は Web サービスへのアクセス制御ポリシーならびにアクセス制御実現のためのプロトコルを検討している。いわば、Web サービス間の関係を議論する。一方、本研究課題では、Web サービス連携を表現した WS-BPEL 記述が情報漏洩の可能性があるか否かを取り扱った。

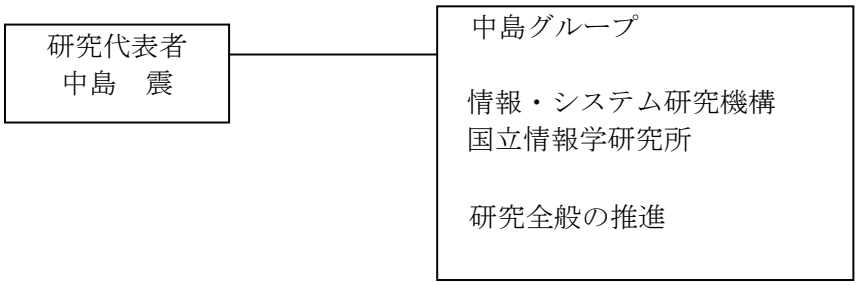
情報漏洩の有無を調べる研究はシステムソフトウェアの分野で 1970 年代に Denning が提案した LBAC に基づく方法が有力である。その後、Java 等のプログラミング言語で書かれたプログラム記述に情報漏洩の可能性があるか否かを調べる情報フロー解析の基本概念として使われてきている。代表的な研究に Myers による Java を拡張した JFlow がある。IBM

東京基礎研では JFlow の考え方を Web サービスに適用する中間報告を公表していた。また、Yoshida and Honda は彼らが以前に提案したπ計算に基づく情報フロー解析の理論的な成果を Web セキュリティに応用する研究を進めている。

本研究課題では、先行のさきがけ研究と同様に、Promela/SPIN を用いたモデル検査法によって WS-BPEL 記述の検証を行った。特に、BPEL4WS(v1.1)仕様書記載の 4 つの例題を全て正しく解析できることを示した。さらに、情報漏洩の有無を調べるために、この方法を拡張する研究を行った。当初は、Promela/SPIN を用いる検討を進めたが、取り扱う問題の本質的な側面を解決する新たな考え方として、制約オートマトンのモデル検査法を用いる方法を考案した。制約オートマトンは従来のモデル検査法が不得意な構造データに対しても有効な検証方法の基礎となる。

６．研究実施体制

(1)体制



(2)メンバー表

①　中島グループ

氏　名	所　属	役　職	研究項目	参加時期
中島　震	国立情報学研究所	教授	研究全般の推進	平成１６年１２月～ 平成１９年３月

７．研究期間中の主な活動

(1)ワークショップ・シンポジウム等

年月日	名称	場所	参加人数	概要
平成 18 年 12 月 1 日	形式手法ワークショップ	国立情報学研究所	約 80 名	情処学会 SIGSE 組込み WG、IPA/SEC、NII アーキテクチャ科学研究系の協力により、産業界に向けた啓発活動のためのチュートリアル、パネル討論を実施。

(2)招聘した研究者等

（該当なし）

８．発展研究による主な研究成果

(1)論文発表（英文論文　２件　　邦文論文　２件）

○　S. Nakajima、Lightweight Formal Analysis of Web Service Flows、Progress in Informatics (No.2、pp.57-76、2005).

中島 震、UML ステートダイアグラムの亜種を用いた組み込みソフトウェア振舞い解析、情報処理学会論文誌 (Vol.46、No. 11、pp.2643-2653、2005).

中島 震、 モデル検査法のソフトウェアデザイン検証への応用、コンピュータソフトウェア(Vol.23、No.2、pp.72-86、2006).

S. Nakajima、Model-Checking Behavioral Specifications of BPEL Applications、Electric Notes in Theoretical Computer Science (No.151-2、pp.89-105、2006).

(2)口頭発表

①学会

国内　９件,　　海外　２件

②その他

国内　３件,　　海外　１件

(3)特許出願（本研究に係わり、JST から出願したものとで研究機関から出願したもの）

（該当なし）

(4)その他特記事項

①　モデル検査法を含む形式手法の技術動向について、日経 BP 社より受けた取材内容が同社の雑誌（日経エレクトロニクス、日経コンピュータ）に掲載。

②　モデル検査法を含む形式手法の技術を鳥瞰する解説記事「形式手法の実像を知る」

が「日経エレクトロニクス」（平成 18 年 8 月 28 日号）に掲載。

③「ソフトウェアの信頼性と形式手法」に関する署名記事が「日刊工業新聞」（平成 19 年 3 月 1 日）に掲載。

９．結び

[研究の実施について] 本研究課題に先行して実施したさきがけ研究では、Web サービス連携の分野で安全性と情報漏洩に関する問題が未解決であることを指摘し、モデル検査法を用いる方法を提案した。特に、WS-BPEL の基になった WSFL を対象としてモデル検査法の適用効果を実証したこと、WS-BPEL を対象としてセキュリティレベルに基づく情報漏洩解析の方式を提案したこと、が具体的な成果であった。本研究課題では、WSFL に対する研究成果を基に WS-BPEL に対する安全性解析を実現し、また、情報漏洩解析の方法を具体化することを目標とした。特に、解析ツールの試作を通して技術を具体化することに主眼を置いた。

試作ツール開発は外部のソフトウェアハウスにお願いした。先方の開発要員配置の関係から進め方を工夫する必要があり、要員確保できる期間に合わせて開発内容を計画した。一般に、試作開発相当の「軽量な」開発量を請け負う会社を見つけることが難しい。この点、幸運であったと考えている。

「全体計画書」（平成 16 年 10 月）との比較では、研究面での修正を行う必要があった。当初は、系統的詳細化の手法を用いて情報漏洩検査を行う制約検査プログラムの正しさを確認しようと計画した。しかし、B メソッドを用いる予備検討の段階で、この方法が難しいことがわかった。一方、制約を記号的に表現することで正しさの確認が容易になることに気がつき、新しい方法である制約オートマトンに行き着いた。このことは予定になかった成果である。制約オートマトンは従来のモデル検査法が不得意な構造データに対しても有効な検証方法の基礎となる面白い研究対象であると考えている。

一般に新しいソフトウェアに関わる技術を研究する場合、そもそもの問題定式化が重要な役割を果たすと考えている。これは、昔から知られている未解決問題を解く、いった研究とは全く異なる難しさがある。登場した新しい技術の延長上に何が問題となって覆いかぶさるかを予測する必要があるからである。

多くの場合、新しい機能を持つソフトウェアは、それ自身が興味深いので、ビジネスのたねになることが多い。そのため、産業界では、まずは「動く」プログラムを仕上げようとする。Web サービスを稼動させる技術、Web サービスを連携させて新しいサービスを組み上げる技術、などが興味の中心となる。信頼性といった側面は後の課題としてとり置かれるか一時的に忘れられる。ところが、利用者の立場からは、提供される Web サービスが信頼できるものでなければならない。

本研究課題では、従来から知られていたソフトウェア科学の成果であるモデル検査法という基礎技術を背景に、Web サービス連携の技術における信頼性に関わる問題点を指摘し、その解決のひとつの方法を示した点に重要性がある。ほぼ同時期に、欧米各国でも、同様な狙いの研究が進行し発表された。初期に発表した論文が先行研究として引用されている事実から、ここに述べた本研究課題の意義を裏付けることができよう。

なお､Web サービスを対象とする形式手法に関する国際ワークショップ（WS-FM）が 2004 年からイタリアの研究者を中心に始まった。第 1 回から継続して PC 委員を依頼されており日本から唯一の委員である。また、Van Breugel はサーベイ論文で、本研究課題の成果

をまとめて紹介している。Web サービス連携の形式検証に関する研究は、ニッチな研究分野ではあることも事実である。しかし、当該分野において国際的に一定の評価を得られたと考えている。

さきがけ研究から本研究課題の終了に至る期間は、国内でモデル検査法への関心が盛り上がった時期である。モデル検査法について早くから携わっていたこともあり、チュートリアル講演・セミナーや解説論文執筆の機会を多く頂いた。初期にはソフトウェア研究者に対して、また、後には産業界の方を中心とする啓発活動であった。また、2006 年 12 月 1 日に行った「形式手法ワークショップ」は、そのような啓発活動のひとつであり、約 80 名という参加者を集めることができた。直接的な研究活動ではないが、本研究課題に関わる技術の裾野を産業界に広めるという意味で、重要な活動であったと考えている。

[戦略的創造研究推進事業に対する意見]　さきがけ研究が競争的研究資金であった上に、本事業は、さきがけ終了者が応募するという意味で二重に競争的な面を持つ。したがって、採択されたことを非常に誇らしく感じることができた。実際、長く企業に勤めていたため、自身が研究者として一人前に活動していくことについて、甚だ心もとなかったことは事実である。その点、本事業に採択されたことで、研究者として認知されたことを実感できた。

研究遂行の上では、「発展・継続」第二研究事務所の支援が大きかった。なによりも購入に関する問い合わせへの回答が迅速であり安心して作業を進めることができた。また、本事業の特徴として、予算執行管理を所属組織（報告者の場合は国立情報学研究所）から独立に行う事務所執行分がある。これは 4 月の年度開始直後から執行できるので非常に重宝であった。なお、「発展・継続」の書類・帳票類は、CREST 等のチーム研究を想定したものであり、さきがけの個人研究の性格に合致しない「重厚さ」があるように感じられた。

最後に、研究総括の三谷先生、三上技術参事をはじめとする「発展・継続」第二研究事務所および科学技術振興機構の皆様、ならびに「さきがけ・機能と構成」の領域総括であった片山先生に感謝いたします。