高信頼Webサービス

（研究課題名：高信頼性Webサービス）

「機能と構成」領域　　中島　　震

要　　旨

複数のサービスを組み合わせて利用する高度Webサービスが注目を集めています。このようなシステムでは複数のWebサービスを協調させて作動させることが重要です。このため、WS-BPELと呼ばれる言語が世界標準として提案されています。ところが、組み合わせが複雑になると、不用意な停止や情報漏洩といった不具合がないことを事前に確認することが困難です。ここでは、並行システムのモデル検査技術を応用して、WS-BPELで記述したプログラムに誤りがないことを検証する方法を提案します。また、「社外秘」などの機密レベルを導入し情報漏えいがあるかないかを検査することも可能にしました。

1．研究のねらい

インターネットの発展と共に、Webサービスの技術が、新しい業務サービスの基盤として登場しました。当初は、ひとつの機能を提供するWebサービスが主流でしたが、最近では、複数のWebサービスを統合して新しいサービスを提供する複合サービスの技術に注目が集まっています。複合化によって多数のビジネスパートナーと連携したサービスを提供できるからです。

Webサービスの世界では、異なるベンダが開発したソフトウェア基盤がネットワーク上で情報交換できることが必須です。そのため、W3C（World Wide Web Consortium）やOASIS（Organization for the Advancement of Structured Information Standards）といった中立な組織を中心とする技術標準化の活動が大切です。実際、図1に示すように、数多くの要素技術が体系化され、標準化の議論が進んでいます。Webサービス複合化の技術も、ベンダ提案段階からOASISでの標準化活動に進み、現在、WS-BPEL（Web Service Business Process Execution Language）と呼ぶ一種の分散協調システム記述言語が提案されています。すなわち、数多くの独立性の高いWebサービスを連携させる複合サービスをWS-BPELのプログラムとして表現しようという考え方です。

複合サービスを表現する連携のことをオーケストレーションと呼ぶことがあります。多数の演奏者であるWebサービスが協調して作動するような全体調整を行うことが分散協調システムの特徴だからです。ところが、分散協調システムは動作振る舞いが複雑なため、処理が進行しないデッドロックによるシステム停止などの不具合を除去することが困難です。ある複合サービスが実行中に停止すると、自身にとって困ると同時に、関連するWebサービス提供者にも影響を与えます。これを、安全性の問題と言います。WS-BPELの記述を対象として安全性の観点からの問題がないことを確認する必要があります。

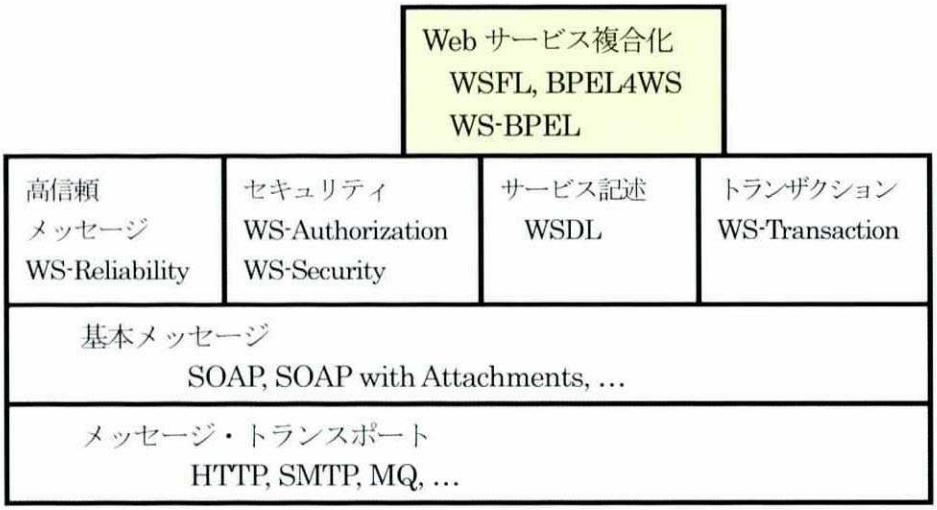

図1　Webサービス技術体系の標準化

次にセキュリティに関する問題を考えます。今後、企業が自身の活動の効率化を目的として、外部のコンサルタント会社等に、従来は社内で行っていた業務を外部委託することが増加すると考えられます。コンサルタント会社は顧客会社のいろいろな情報を入手してはじめて有用なアドバイスができます。将来の一形態として、Webサービス技術を用いたコンサルタント業務を想定できるでしょう。顧客企業は必要な情報をWebサービスとして提供し、コンサルタント会社はノウハウを組み込んだ業務ロジックをWS-BPELのプログラムとして表現します。こうすることで、顧客企業の情報や市場動向調査レポートなどの一般情報を全てWebサービスの技術を用いて収集することが可能になります。

ところが、顧客会社からみると、コンサルタントに必要とはいえ、社外秘情報が外部に出るため、その取り扱いには慎重さを要求します。すなわち、コンサルタント業務を表現するWS-BPELのプログラムが何らかの誤りによって、大切な情報を流出することがないという証拠を、コンサルタント会社に求めるでしょう。WS-BPELの記述を対象としてセキュリティの観点からの問題がないことを確認することも大切です。

安心して使えるWebサービスを実現するためには、WS-BPELを代表とする現在の技術体系（図１参照）に加えて、デッドロック等の不具合がないこと、大切な情報の漏えいがないことを、何らかの方法で確認する手段が必要となります。すなわち、安全性とセキュリティの面からの解析が必要になるわけです。

本研究課題では、WS-BPELのプログラムを解析し、安全性ならびにセキュリティという上記の２つの観点からの不具合がないことを、インターネット上で実行する前に確認する技術の研究を進めました。本研究課題の成果によって、WS-BPELのプログラムを実行してよいかどうかを事前に判断することができ、複合Webサービスを安心して使うことが可能になります。

２．研究方法と成果

２．１　安全性の解析

本研究課題の開始時点では、WSFL（Web Service Flow Language）と呼ぶWebサービス連携記述言語が提案された段階で、まだWS-BPELは、その基になったBPEL4WS（Business Process Execution Language for Web Service）も含めて提案されていませんでした。したがって、当初、研究の目的を、WSFLのプログラムを対象としてデッドロック等の不具合があるかないかを解析する技術を確立することとしました。

WSFLの言語仕様を調査すると、業務のいろいろな作業連鎖を記述するワークフローの考え方をWebサービスに転用したものであることがわかりました。そこで、1994年頃に研究していたワークフロー記述を対象とするモデル検査の方法を応用することを着想しました。この方法は、ワークフローの中心となるタスクを並行システムの構成要素であるプロセスに対応させ、タスク間の情報フローをプロセス間の通信として実現することで、ワークフローを表現することを基本とします。このように対応させることで、並行プロセスのモデル検査という技術を用いることを可能にした点が主要な成果でした。モデル検査の方法は、単純化すると、可能な制御の流れを網羅的に探索してデッドロックのような不具合のある状態を探す技術です。網羅的な経路探索を行うことが特徴です。

WSFLは外部Webサービス起動をノードとし、この起動順序を指定する制御フローならびに外部Webサービスと交換するデータのやりとりをデータフローで表現する一種のネットワーク指向並行記述言語です。すなわち、ワークフローのタスクを外部Webサービス起動に対応させるということで、従来の研究で行った知見を用いることを着想しました。

具体的には、WSFLプログラムの構成要素を、一種の並行システム記述言語であるPromelaに対応させる方法を考案しました。これによって、Promelaを入力とするモデル検査ツールSPINを用いて、デッドロックの有無などを自動解析する方法を実現できることになります。いくつかの実験を行った結果、WSFLの言語仕様に不備があり、デッドロックを除去することが難しいことを指摘しました。同時に、デッドロックを除去する方法を提案し、上記と同様な変換手法によって、提案方式が問題を解決していることを、具体的な実験を行うことで確認しました。

この研究成果はWebサービス連携記述の解析にモデル検査の方法を適用した研究として一定の評価を頂き、その後、同じ研究分野の諸外国の研究者から、論文を引用される機会を得ました。一方、WSFLは同時期に提案されたXLANGと統合され、BPEL4WSとして主要ベンダが共同提案しました。さらに、BPEL4WSはWS-BPELと名前を変えてOASISで標準化検討されています。そのため、本研究の成果が、直接、Webサービスの技術分野に貢献するには至りませんでした。しかし、WS-BPELはWSFLの中心的な言語仕様であるネットワーク指向並行処理記述の側面を部分言語として持ちます。したがって、WSFLで提案した方式を、WS-BPELの安全性解析を行う方法として利用することができます。

2.2 セキュリティの解析

冒頭に述べましたように、Webサービスは開放型ネットワークであるインターネットを土台としているため、標準化でもセキュリティに関する議論が活発です。図1にあるWS-Securityと呼ぶ一群の技術がセキュリティの側面を扱います。現状では、Webサービスの基本通信メッセージであるSOAPを暗号化し通信路で情報が漏えいしないことを保障する技術、および、指定Webサービスへのアクセスを許可するか否かをあらわすアクセス制御ポリシー表現、ならびにアクセス制御実現のためのプロトコルなどがWS-Authorizationとして提案されています。

一方、暗号技術とアクセス制御の技術だけでは、情報漏えいの問題を扱うことができません。そのために、従来から情報システムセキュリティの分野で、ラティスに基づく情報フロー制御の方法が提案されていました。本研究課題では、WS-BPELを対象として、情報フロー制御の方法を用いた情報漏えいの有無を解析する方法の適用可能性を検討しました。これは、現在のWebサービス技術体系では未だ取り扱っていない問題です。冒頭のコンサルタント会社のような先進的なWebサービスを実現する上で必須の技術になると考え、先行して研究を進めることにしました。

ラティスに基づく情報フロー制御の方法では、セキュリティからみた情報の重要さを導入し、重要さを表すセキュリティラベルに順序関係を与えます。この順序関係をdominates 関係と呼び、これがラティスとなることが方式の名称の由来です。代表的な例では、「秘密文書」、「公開文書」などを考えることができ、「秘密文書」は「公開文書」よりもdominates関係が大きい、あるいは支配的であるといいます。次に、利用主体と資源の双方にセキュリティラベルを与え、先に与えたdominates関係を満たす方向だけに情報の流れを許可します。すなわち、機密関係の高い方向にだけ情報の流れを許可するので、Flow-Upと呼びます。

ところが、Flow-Upだけでは、高いレベルの利用主体が読んだデータは２度と読み出すことができなくなるという問題があります。すなわち、アクセスの都度、dominates関係の支配的な方向に情報が動き、最終的に、誰も読み出すことができない「ブラックホール」のような超高機密レベルに落ち込みます。この問題を避けるために、クラス低下（Declassification）と呼ぶ方法があります。

今、利用主体P1が資源T1から読み出したデータを資源T3に書き込む場合を扱い、次の関係が満たされているとしましょう。ここで、L（P1）はP1が持つセキュリティラベルのことです。

L（P1）dominates　L（T1）

L（P1）dominates　L（T3）

L（T3）dominates　L（T1）

３つめのdominate関係を考えるかぎり資源T1からT3へのフローは許可される状況です。しかし、利用主体P1がデータをアクセスするために、P1がT1から得たデータのセキュリティラベルが表面上、P1と同一になります。そのため、２番目の関係からP1からT3へのフローが禁止されます。

この問題を解決するために、利用主体P1のラベルを一時的に低下させるのが、declassificationです。つぎのような関係を満たす一時的なラベルDCLが見つかればよいとします。

L（P1）　dominates　L（DCL）

L（DCL）dominates　L（T1）

L（T3）　dominates　L（DCL）

１番目はDCLがP1よりも小さいこと、２番目はDCLによってT1からのフローが許可されること、３番目はDCLからT3へのフローが許可されることを示します。この例では、L（T1）に一致するようにL（DCL）を選べばよいことになります。

ラティスに基づく方法は、上の例を一般化することで次のように整理することができます。

ある実行経路上に現れる全てのアクセスごとにdominates関係を集めてきて、集めたdominates関係すべてを満たすような一時ラベルDCLが存在するかを調べます。すなわち、DCLの値を具体的に求めることができれば、情報フローに誤りがなく、したがって、情報漏えいがないといえます。一方、DCLの値がなければ情報漏えいの問題があると結論するわけです。

上に述べたように、情報漏えいの問題は、流れるデータに付随するセキュリティラベルの値がラティスを形成し、その値を比較検討することで、集めてきたdominates関係が解を持つか否かを判定することです。ところで、この処理は2つの問題に分割することができます。すなわち、実行経路を網羅的に生成し当該経路上のdominates関係を集めてくる1番目の処理、さらに、集めてきたdominates関係の制約を解く2番目の処理です。

ここで提案する手法は、1番目の処理に2.1節で検討したモデル検査の方法を用いるというものです。すなわち、セキュリティラベルを付加できるように拡張したWS-BPELを対象として、2.1節で用いたモデル検査の方法を適用します。2番目の制約計算処理は、大小関係の比較演算で実現することができます。

以上、本研究によって、Webサービス連携の記述を対象とする情報漏えいの解析がはじめて可能となりました。

2.3 解析ツールの方式

研究の次の段階は、今までの考察結果をもとに、WS-BPELのプログラムを対象とする具体的な解析ツールを開発することです。本研究課題では、汎用のモデル検査ツールSPINを用いて、上記のアイデアを実現する方法を考察しました。

第1に、安全性の解析ならびにセキュリティ解析の実行経路を集める1番目の処理に関しては、2.1節の方法を基本としたモデル検査の方法で対応することができます。ところが、制約計算で行う大小関係の比較は、モデル検査ツールが得意な処理ではありません。値の計算はモデル検査の枠外なのです。そこで、SPIN4.0が新規に導入した「埋め込みC言語」の手法を応用して、SPINの機能拡張を行う方法を着想しました。別途、dominates関係を計算するC言語の関数を作成しSPINに統合するという方法です。いわば、dominates関係の計算処理を基本機能として持つようにSPINを拡張することに相当し、これによって実行効率の大幅な向上を達成できます。

なお、上記では、2つの処理を独立して行うような説明をしましたが、実際のツール化に際してはさらに工夫が必要です。経路探索を行いながら制約計算により大小関係の整合性を確認していくという統合処理などの最適化方法を考案しました。

3．今後の展望

今後、Webサービスの技術はさらに重要となり、本研究課題で扱ったコンサルタント業務のような高度なサービスが登場すると期待できます。高度なWebサービスを安心して利用するためには、デッドロック等のシステム停止に至るような不具合がないこと、さらに、提供した大切なデータが流出しないこと、などを事前に解析して確認する技術が必須となるでしょう。

本研究課題では、並行システムのモデル検査やラティスに基づく情報フロー制御といった科学的な裏づけのある技術が、ビジネス主導の技術であるWebサービスの世界で重要な役割を果たすことを示すことができました。本当に実用的な解析ツールを開発するためには、まだ、解決すべき課題も多く残っています。また、本研究課題の中で、ラティスに関する制約処理をモデル検査と統合する必要があり、これを一般化することで新しい研究課題をみつけることもできました。今後は、実用的なツールの実現ならびに、新たな課題を解決するための基礎研究を行っていきたいと考えています。

4．成果リスト

論 文 誌

1．中島　震，玉井哲雄: EJBコンポーネントアーキテクチャのSPINによる振舞い解析，コンピュータ・ソフトウェア，Vol.19, No.2, pp.2-18（2002年3月）.
2．中島　震: Webサービスフロー記述のモデル検査検証，情報処理学会論文誌，Vol.44, No.3, pp. 942-952（2003年3月）.
3．中島　震: コンポーネントフレームワーク振舞い解析への多値遷移システムの応用，コンピュータ・ソフトウェア，Vol.21, No.2, pp.32-36（2004年3月）.

国際会議

1．S. Nakajima: Verification of Web Services Flows with Model-Checking Techniques, Cyber World（CW 2002), pp. 378-385（2002年11月）.
2．S. Nakajima: Behavioural Analysis of Component Framework with Multi-Valued Transition Systems, Asia-Pacific Software Engineering Conference（APSEC 2002), pp.217-226（2002年12月）.
3．S. Nakajima: Model-Checking of Safety and Security Aspects in Web Service Flows, International Conference on Web Engineering (ICWE 2004), pp. 488-501（2004年7月）.

口頭発表

1．S. Nakajima: On Verifying WEB Service Flows, Proc. SAINT 2002 Workshop on WebSE 2001（2002年

1月).

2．中島　震: WSFLを用いたWebサービスフロー記述の自動検証技法，情報処理学会研究報告（2002年7月).

3．中島　震: WSFLの記述パターンとデザインチェッカ，情報処理学会研究報告（2002年10月).

4．S. Nakajima: Model-Checking Verification for Reliable Web Services, OOPSLA 2002 Workshop on Object-Oriented Web Services（2002年11月).

5．中島　震，玉井哲雄: セキュリティポリシー変更に関するデザイン解析，情報処理学会研究報告（2003年7月).

6．中島　震，玉井哲雄: 高レベルセキュリティポリシーのデザイン検証，第20回日本ソフトウェア科学会大会（2003年9月).

7．中島　震: 組み込みソフトウェアへのモデル検査検証技術の応用（チュートリアル)，情報処理学会組込みソフトウェアシンポジウム（ESS 2003)（2003年10月).

8．中島　震: Webサービスにおける安全性とセキュリティの解析，電子情報通信学会信学技報（2003年11月)

9．中島　震: モデル検査検証のソフトウェア開発への応用（チュートリアル)，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ワークショップ（DSW 2004)（2004年2月).

依頼原稿

1．中島　震: 書評 - G.J.Holzmann著 The SPIN Model Checker，コンピュータ・ソフトウェア，Vol.21, No.2, pp.61-69（2004年3月).

2．中島　震: 組込みソフトウェアへのモデル検査の応用，情報処理，Vol.45, No.7, pp.690-693（2004年7月).

表彰など

1．2003年度日本ソフトウェア科学会論文賞受賞（2004年6月).

謝　辞

さきがけ研究という恵まれた環境を与えて頂いたことに対し、片山卓也研究総括、領域アドバイザの諸先生、ならびに、科学技術振興機構の皆様に感謝いたします。