Formal Methods for Web Services: a Taxonomic Approach

K.S. May Chan

University of Pretoria
Department of Computer Science
Pretoria 0002, South Africa

ksmchan@cs.up.ac.za

ABSTRACT

Formal methods can be used to verify different perspective of a Web service. An ensemble of specific techniques is not supported by a general approach to the problem. To understand which formal method should be combined and used is a challenge. This paper outlines our approach to address this problem.

Categories and Subject Descriptors

A.1 [**Introductory and Survey**], F.4.3 [**Formal Languages**]

General Terms

Design, Reliability, Theory, Verification.

Keywords

Web application, Web service, Service verification, Formal methods, Service development life-cycle

1. INTRODUCTION

Distributed applications have gained a substantial growth in the past decades. Among these applications, the most commonly known one is Web-based applications. Many companies and government departments are using Web applications to communicate with their customers. Legacy information and database systems are being migrated to Web environments, in order to deploy their functionality on the Web.

There are many types of Web applications; they range from simple applications built and run on a single Web server to modern sophisticated enterprise applications running on distributed application servers. Different categories of Web applications are grouped together according to their data and control complexity [7], as shown in Figure 1.

1. *Brochure Web Applications* are the first generation of Web applications. They are composed of static Web, there is no/little control or data flow in these types of applications.
2. *Service-Oriented Applications* offer services to Web users. These applications are oriented towards programming logic which is needed to provide the service online, but with little data flow. The layout of the data is often a secondary concern.
3. *Catalogue Web Applications* provide an interface to browse and query large quantities of data, but with minimal amount of programming logic or control in them. Catalogue Web applications are closely tied to their database.
4. *Information System Applications* are a combination of service-oriented applications and catalogue Web applications. These applications concern with both data flows (for browsing and retrieving data) and control flows (for the different phases involved in manipulating the data). These types of applications can be further divided into two sub-groups: i) *transaction-based* and ii) *process-based.* Transaction-based applications allow a user to register and manipulate transactions. Process-based applications are used to guild the user through a series of steps the user has to perform.

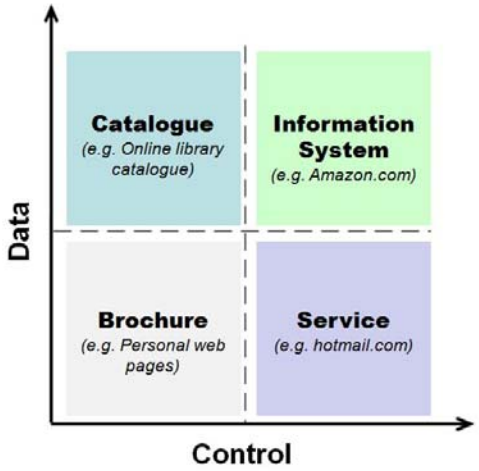

Figure 1. Taxonomy of Web Applications

For this research, we will focus on the *transaction-based information system applications*.

Web services composition is a way of developing Web application [11]. As organizations and businesses are extending their services to be Web-based applications, these applications are considered to be mission-critical or business-critical systems to the organizations or business. That means they are required to function at high levels of reliability and security, any failures could result in high economical losses. Formal methods have been advocated as means for developing critical systems to increase their reliability [3]. They provide rigorous approaches which enable users to analyse and verify any part of a software during its development life-cycle [14].

ICSE '10, May 2-8, 2010, Cape Town, South Africa

2. RELATED WORK

Ensuring the correctness of a system plays a vital part during its development, especially in an open-world environment [12]. There are many similarities between software components and Web services [8], but the differences suggest the need to address new challenges, such as long-run interactions. Bianculli et al. [2] stress the importance of applying verification throughout the development of a service. They argue that verification should be embedded as part of the service development, such that the same properties verified based on services specifications will also be checked after a service becomes an executable artefact and as part of the larger service-oriented system.

A significant number of papers have already been published in applying formal methods to address the challenges in modelling and verifying service-oriented systems [13]. The applications of formal methods in Web services include validation and verification of services, compositions and choreographies, transactions, interoperability and workflow analysis. Most of the researches are concerned in providing an illustration on how formal methods are applied in solving a specific problem in service development.

Morimoto [9] and ter Beek et al. [15] conducted a survey to provide an overview on the application of formal methods in service composition. The formal methods surveyed are compared based on a set of criteria defined by the authors. From the survey we see that formal methods are roughly used in the following manner.

Automata and Labelled Transition Systems (LTS) are used to verify the system behaviour.

Petri Nets are used to capture process control flow for correctness analyses and verification.

Process Algebras are used for automatic verification of functional as well as non-functional properties. These properties are verified in order to determine whether the new process, after recovery, is equivalent to the original process.

3. RESEARCH OBJECTIVES

The main research question we are trying to answer is: Can a meta-model be established to guide the integration and application of formal methods to achieve a specific objective throughout the development cycle of a service? In answering this research question, a number of research methods will be applied.

Literature study The research first looked at the literature on existing Web service research and general formal methods application in this area. The work was examined though a sub-research question:

- **RQ1:** What does the existing research say about the application of formal methods in the context of Web service development?

Survey Following from the literature study, a questionnaire was prepared, distributed and answered by researchers working in the field of Web services, but not necessarily applying formal methods in their research. The aim of the survey was to investigate:

- **RQ2**: How formal methods are used in the context of Web services?
- **RQ3**: Based on what aspects are the formal methods being chosen and how they are used?

Case study The results from the survey illustrate there is a need for integration of formal methods. This further leads to the research question:

- **RQ4**: How the current methods and tool support could be extended or improved?

In answering this question, we will conduct a real case study based on the Amazon ECS [1]. The service will be modelled, focusing on the data and control flows. The steps involved in modelling are:

- Stepwise requirements modelling of the system.
- Formal specification of complex data structure and data flow and their stepwise refinement.
- Formal specification of control flow and their multi-level stepwise refinement.
- Formal verification of data and control flows as far as possible with the support of formal tools (e.g. model checkers).

The results from the above mentioned steps will bring us to our final question:

- **RQ5**: How should formal methods be combined such that they work well together?

Proofs will be used to validate the integration of formal methods derived from the RQ5.

The expected contributions of this research will: (a) provide a taxonomy that forms the theoretical basis in understanding which formal method/approach should be combined and used in a service development life-cycle; (b) provide a meta-model for formal methods integration.

4. PRELIMINARY RESULTS

In this section, the results of our survey are presented. The survey was aimed to answer RQ2 and RQ3, and is only the first step into this PhD project. A more detailed version of the report can be found in [4].

The population for the survey is defined as researchers, mainly academics, conducting research in the areas of Web services, but may be not interested in applying formal methods in their research. Using a structured questionnaire, data was collected between 01 and 30 September 2009.

A total number of 34 responses, out of the population of 52, were received around the world which gives us a total response rate of 65.38%. The response came, in descending order, from Europe, Asia, Africa and Australia. This may be biased due to the initial pool of population was selected from Europe. Nevertheless, the uniform way in which the information is gathered does give some insight on the use of formal methods in the context of Web services.

4.1 Hypotheses

In view of the goal of our survey, we have formulated a list of hypotheses:

Hypothesis 1: Researchers who do not have prior knowledge of formal methods often have to go through a steep learning curve.

Hypothesis 2: Researchers choose to use a formal method that they are familiar with, instead of the most appropriate method with respect to the type of application.

Hypothesis 3: The reason why researchers choose not to use formal methods is because of the time and effort required to learn is too high.

Hypothesis 4: Current formal methods and tools are not sufficient to address issues in the context of Web services.

Hypothesis 5: A formal method can only address certain aspect of a problem in a development life-cycle.

4.2 Data Collected and Observations

Figure 2 presents the research areas in Web services. Based on [10] we have categorised the research areas into three phases of the service development life-cycle, namely, Design and Analysis, Construction and Execution. The figure also presents participants' response to which their projects are related to. From this we see that much effort is spent on the construction of a composite service. The execution phase is comprised of service management, which includes the self-* properties, is also an important research area. During the execution phases it takes into account of all aspects defined in the analysis and design phase as well as the construction phase of the service development life-cycle.

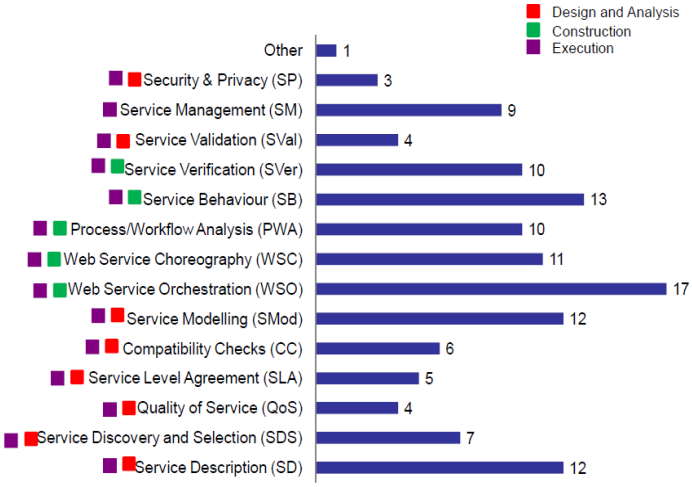

Figure 2. Research Areas

Figure 3 presents the research methods used in the projects. When asked to indicate whether formal methods are being utilised in their research, we found that 22% of the responses does not utilise formal methods. For empirical and theoretical research, as well as empirical and practical implementation research no formal methods are used. One out of seven responses indicates that while doing theoretical and practical implementation, they did not make use of formal methods. Another similar case is for the practical implementation research, in this case the ratio is 1:4.

When ask to indicate the reason for not utilising formal methods, the majority of the respondents indicated that formal methods are too theoretical. One of the respondents explicitly stated:

"It is too theoretical. It could be a useful tool, but too far from actual implementation, lack of time to learn."

Other reasons for not using formal methods include: (1) they are too rigorous and are not applicable in the rapidly changing market of Web services; (2) they are used for verification and validation of system architecture (e.g. Web services orchestration and Web services choreography), but not for practical implementation; (3) they deals only with discrete variables.

Based on these observations, we can say the answers of the survey participants supported our hypothesis 3.

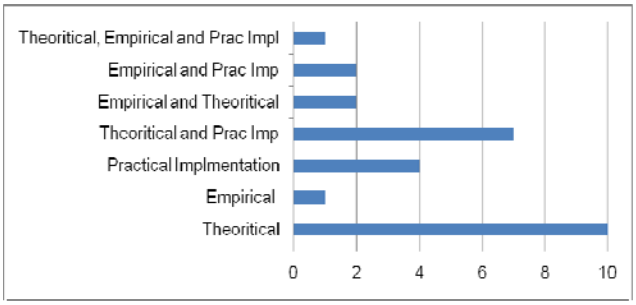

Figure 3. Research Methods

In the survey we also gathered that the most used formal method in this context is LTS, followed by Pi-calculus and CCS. Other formal methods only have one or two responses. The responses are dominated by formal methods in the group of process algebra. The reason could be because process algebra has the ability in modelling concurrent systems and provide the ability to describe interactions between services in a synchronised manner. LTS provides the ability to specify different state of a Web service it's currently in.

When asked to indicate how a formal method is being selected, the overall responses tend to be based on the practitioners' prior knowledge of the formal methods, indicated in Figure 4. Along the line of previous expertise in the formal methods, out of 76% of the responses, which uses formal methods, indicated they have prior knowledge. This includes all those who have "*considerable previous experience*" and those who have "*some previous experience*". The remaining 24%, who does not have prior knowledge, have to go through some learning. We observed researchers choose a formal method they have previous expertise in, or by recommendation. Seldom was a method chosen based on the application needs. Those answers provided support for hypothesis 2.

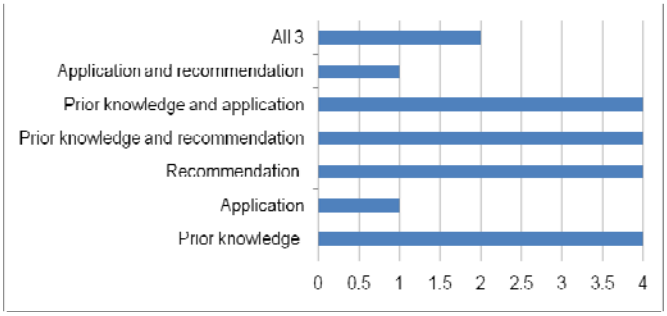

Figure 4. Formal Methods Selection Criteria

The average difficulties experienced during learning is 5.6 with the scale of 0 to 10, where 0 is the easiest and 10 is most difficult. Thus, hypothesis 1 was not supported by the answers we received in this survey.

The respondents who use formal methods in their research indicated that current formal methods lack the ability in:

- handling data which provides in-depth data description;
- handling concurrency;

- handling transaction mechanisms;
- handling real-time issues;
- providing a reasonable and abstract representation of behavioural aspects.

Similarly, the available tools are perceived as not user-friendly; most of the tools do not meet what the users require it to do. Much time is wasted on trying to understand the nuances of the chosen tool. The respondents indicated current tools lack the ability in:

- handling data;
- handling concurrency;
- providing a platform for development.

Accessibility and reliability of a tool are also of users' concerns. We noticed, like other software, formal tools become more reliable as it matures. Due to the lack of accessibility of tools, researchers often have to develop their own prototyping tools and they are consider to be "*To some extend reliable*" if not at all "*not reliable*". This shows that our hypothesis 4 holds.

For the purpose of illustration, we have constructed a concept lattice, using *Concept Explorer* [5], to identify the relationships between research areas and formal methods. Such concept lattices are built by applying Formal Concept Analysis (FCA) [6] to the data collected from the survey. For more details on the analysis process refer to [4].

A fragment of the lattice in Figure 5 shows the service description and their behaviour of a Web service choreography is needed for service verification. By looking at the bottom node, we can see that the nine formal methods for doing servicer verification. In addition, we also noticed the following:

- Not a single formal method seems to be able to solve the problem of specifying and verifying a larger Web service application entirely.
- There seems to be no systematic way of combining various formal methods to solve a service development problem.
- There seems to be no well-understood, systematic way to ensure the validity of the integration of formal methods with respect to a specific objective.
- Current approaches to combining formal methods are still ad-hoc and explorative.

This provides support for our hypothesis 5.

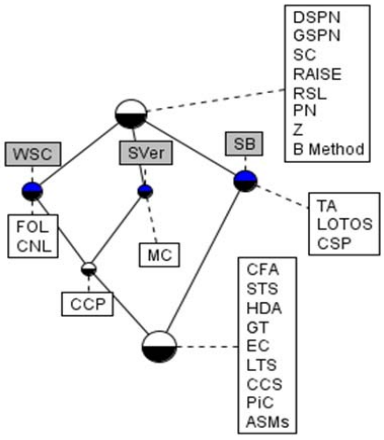

Figure 5. Service Verification and Formal Methods

5. CONCLUSION AND FUTURE WORK

Web services play an important role for the development of Web applications. One of the challenges facing in such development is to ensure its reliability. Formal methods have been used to increase the reliability of an application. In this paper, we show that a taxonomy is needed for guiding the combination and usage of formal methods in a service development. The proposal for deriving such taxonomy is also presented.

One of the forthcoming steps is to conduct a case study, more specifically an online bookstore based on the Amazon ECS [1]; and model it using a set of selected formal methods. In doing so, a list of criteria for guiding the selection process such as the ability to handle asynchronous communication, will be setup.

REFERENCES

[1] Amazon E-Commerce Service (API Version: 2007-07-16). http://developer.amazonWebservices.com/connect/entry.jspa?externalID=886. Last accessed: 13 November 2009.

[2] Bianculli, D., Ghezzi, C. and Pautasso, C. *Embedding Continuous Lifelong Verification in Service Life Cycles*. In Proc. of PESOS 2009, pp. 99-102, 2009.

[3] Bowen, J.P. and Hinchey, M.G. *Ten Commandments of Formal Methods*. Computer, 56-63, 1995.

[4] Chan, K.S.M. *The Usage of Formal Methods in Web services Context: a Reality Check*. Technical Report, 2009. http://www.cs.up.ac.za/cs/ksmchan/reports/FM_WS_Survey_2009.pdf

[5] Concept Explorer 1.3. http://sourceforge.net/projects/conexp/. Last accessed: 13 November 2009.

[6] Ganter, B., Stumme, G. and Wille, R. *Formal Concept Analysis: Foundations and Applications*. Springer-Verlag. 2005.

[7] Hassen, A. E. *Architecture Recovery of Web Applications*. Masters Thesis, University of Waterloo, Canada. 2002.

[8] Iribarne, L. *Web Components: A Comparison between Web Services and Software Components.* Colombian Journal of Computation, 5(1), pp. 47-66, 2004.

[9] Morimoto, S. *A Survey of Formal Verification for Business Process Modeling*. In Proc. of ICCS'08. LNCS 5102, pp. 514-522, 2008.

[10] Papazoglou, M.P. and van den Heuvel, W-J. *Service-Oriented Design and Development Methodology.* International Journal of Web Engineering and Technology, 2(4), pp. 412-442, 2006.

[11] Ren, Z., Jin B. and Li, J. *A New Web Application Development Methodology: Web Service Composition*. In Proc. of WES 2003. LNCS 3095, pp. 134-145, 2004.

[12] Singh, M.P. and Huhns, M.N. *Service-Oriented Computing: Semantics, Processes, Agents*. Wiley, 2005.

[13] Web Services and Formal Methods. LNCS 4184, 4937, 5387. Springer. 2006-2009.

[14] Woodcock, J., Bicarregui, J. and Fitzgerald, J. *Formal Methods: Practice and Experience*. ACM Computing Surveys, 41 (4), pp. 1-36, 2009.

[15] ter Beek, M.H., Bucchiarone, A. and Gnesi, S. *Formal Methods for Service Composition*. Annals of Mathematics, Computing and Teleinformatics, 1(5), pp. 1-10, 2007.