
Editorial

Yuhong Yan

Institute of Information Technology,
Canada National Research Council,
46 Dineen Dr., Fredericton,
New Brunswick, Canada E3B 9W4
E-mail: Yuhong.Yan@nrc-cnrc.gc.ca

Biographical notes: Yuhong Yan received her PhD from Tsinghua University in 1999 in a joint programme between Tsinghua and University of Leipzig in Germany. She worked in the University of Toronto and the University of Paris as a post-doc from 1999 to 2003. Between the two post-doc terms, she worked as a Software Engineer in Westgroup at Rochester, USA, in 2001. She joined the Canada National Research Council in 2003 as a Research Officer. She was appointed as Adjunct Professor in Computer Science at the University of New Brunswick in 2004. She is an active researcher in web services, service computing, data mining and artificial intelligence. She has published about 40 journal papers and conference papers.

The theme for this Special Issue is 'Service-oriented techniques'. As service is becoming a growing point of economics, using computer techniques to support automated services is attracting attentions from both industries and academics. Service-oriented techniques are the IT techniques around web services and semantic web for various solutions in business process integration and management.

Among the seven papers in this issue, six are selected from the 2006 Workshop on Service Oriented Techniques (SOT06) colocated with the 2006 International Conference on E-Commerce (ICEC06) in Fredericton, New Brunswick, Canada, in August 2006. SOT06 covers topics such as formal methods for service process modelling, service process monitoring, diagnosis and management, service composition and innovative applications. The six selected papers were extended from the workshop version to include the latest research results from the authors. The paper by Wittenburg, Matthes, Fischer and Hallermeier is selected from regular paper submissions.

The seven papers cover theoretical oriented research, best practices and innovative applications. We can see from this issue the frontiers of academic research and industrial trends. The hard problems for academic researchers to deal with are the symbolic representations of service processes and the various reasoning and computation tasks for composing and managing the processes, and verifying the properties of the processes. In industrial applications, web services are used to integrate all kinds of distributed applications, such as simulating and analysing critical infrastructure behaviour, and sharing privacy sensitive data about the collaboration environment, as two examples in this issue. The following is a brief introduction to the seven papers.

The paper by Baldoni, Baroglio, Martelli, Patti and Schifanella presents a method to reason about the capability requirements at choreograph level. Choreographies are public knowledge among the partners.

The partners involved in the interaction need to decide if the capability requirements can be satisfied by their internal capability. A partner that desires to act a role defined by a choreography language can get a policy skeleton that is automatically produced from the choreography specification. Then, it can apply a reasoning process aimed at deciding if it has the right capabilities. In this paper, the reasoning process uses a declarative language that translates capabilities and capability requirements as actions. The reasoning process begins with the substitution of capability requirements by capabilities. The substitution is a kind of match-making process. The reasoning process further proves if a goal can be reached after this substitution. Depending on the situation, the substitution allows the mapping of a subset of capability requirements to a subset of capabilities. This paper presents an extension of WS-CDL by adding capability requirements into it.

The paper by Zhenbang Chen, Ji Wang, Wei Dong and Zhichang Qi is about the formalism for web service interfaces with transactions and its application to formalisation of orchestrations specified with BPEL4WS. The main idea of this paper is to extend the web service interface theory of Beyer, Chakrabati and Henzinger (referenced in the paper) by adding fault handling, compensation handling and nested transaction features. The resulting interface theory provides a semantic foundation for the service interfaces specified by BPEL4WS. This paper describes an automated translation of BPEL4WS processes into this formalism. The purpose of this formalisation is to be able to perform model checking of the BPEL4WS specification in order to check properties represented in temporal logic. This paper has two examples to show the power of the presented methods.

The paper by Wilson and Heywood presents a business logic engine using genetic programming. RuleML and web services are used as the interface to other systems. The business logic engine produces rules best fit the conducted

business processes. The genetic programming is used to compose the rules and evaluate the rules against the conducted business processes.

Service Oriented Architecture (SOA) is regarded a style of information system architecture that enables the creation of applications that are built by combining loosely coupled and interoperable services. The following papers present best practices in real world projects.

Peyton and Hu use SOA for a framework to manage sensitive data and privacy in a collaborative environment for medical consultation. Sensitive data is an important issue in medical consultation systems. When sensitive data is used in collaborative systems, the situation is complex. Multiple users hold multiple roles in a collaborative system. Their interactions are rich and unpredictable. Data in such an environment is from different data sources and has different degrees of sensitivity. The collaborators are allowed to see sensitive data only when necessary and only in such amounts as are necessary to provide the services. The advantage of using SOA is, it can provide a flexible, secure and well-managed mechanism to integrate different collaborative components that share a rich variety of data from different data sources. This paper presents a thorough treatment of issues in managing sensitive data, such as delegation of authority and audit trail. This paper also provides a brief review of the state of the art of collaborative environments in medical care and the standards and techniques of privacy protection. The presented framework leverages state of the art techniques. The performance test shows the response latency is good enough for real-world usage. The problems addressed in this paper can be found in a large number of applications. The method developed here is inspiring.

The paper by Philipps, Aubanel, Healy and Gerber uses web services as middleware to bridge simulation tools and actual physical processes. The simulation software is running on a Sun V60 Cluster, Chorus, and the Sun's Grid Engine scheduler controls the Chorus. There is no API allowing an application to communicate directly with Grid Engine without the shell. An application server is built to accept requests from clients via web service protocol. The application server itself using ssh to connect to Grid Engine and communicate with the Grid through shell commands. This system also wraps other functions, such as process control and repository query, as services. This paper demonstrates the common usage of web services for application integration.

Bagheri, Baghi, Ghorbani and Yari use agent-based simulation to study the interdependencies of critical infrastructure systems. Critical infrastructure systems, for example, power plants or telecommunication networks, are complex networks of adaptive social technical systems. A failure in one of such system can affect the lives of millions of people. Their interdependency is even more threatening since a failure in one can ripple down to others and cause a catastrophic effect. Some events in recent years, especially the California power crisis in 2000 and the 9/11 terrorist attacks, raise the need to understand the behaviour of infrastructure systems. The studies in this domain can be classified into two categories. The first category addresses the issues from a purely mathematical perspective and models the behaviour of infrastructure systems through the employment of differential and/or algebraic-differential equations. The second category of research focuses on modelling infrastructure systems and their interdependencies through the exploitation of intelligent software agents. This paper belongs to the second category. The authors have developed an architecture called Agent-based Infrastructure Modelling and Simulation (AIMS) to study the interdependencies among critical infrastructure systems. In their study, each infrastructure system is modelled as an agent built on top of a component template, which is used to define the internal operations of an agent. The interoperations of the components are defined by the messages and contract agreements between them. The scenarios define the events during simulation and the consequences of the events. The AIMS system also has mechanisms for visualisation and user interface. AIMS system also has mechanisms for visualisation and a graphical user interface. Though the method presented in this paper can reveal the interdependency relations between infrastructure systems, the revelation of the degree of the interdependencies should be achieved through the analyses performed on the results of the simulations.

The paper by Wittenburg, Matthes, Fischer and Hallermeier is selected from regular paper submissions. Their study presents a novel approach for enterprise architecture management and IT governance. We do not observe service-oriented techniques used in their system. However, as many systems are interacting in their framework, we suppose that service-oriented techniques are necessary for integration and interoperation.

In summary, this Special Issue collects a set of papers that study the theoretical foundation of SOA or present best practices for using service-oriented techniques in real world projects.