RC24118 (W0611-165) November 28, 2006
Computer Science

IBM Research Report

Proceedings of the IBM PhD Student Symposium at ICSOC 2006

Ed. by:
Andreas Hanemann (Leibniz Supercomputing Center, Germany)
Benedikt Kratz (Tilburg University, The Netherlands)
Nirmal Mukhi (IBM Research, USA)
Tudor Dumitras (Carnegie Mellon University, USA)

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Preface

Service-Oriented Computing (SoC) is a dynamic new field of research, creating a paradigm shift in the way software applications are designed and delivered. SoC technologies, through the use of open middleware standards, enable collaboration across organizational boundaries and are transforming the information-technology landscape. SoC builds on ideas and experiences from many different fields to produce the novel research needed to drive this paradigm shift.

The IBM PhD Student Symposium at ICSOC provides a forum where doctoral students conducting research in SoC can present their on-going dissertation work and receive feedback from a group of well-known experts. Each presentation is organized as a mock thesis-defense, with a committee of 4 mentors providing extensive feedback and advice for completing a successful PhD thesis. This format is similar to the one adopted by the doctoral symposia associated with ICSE, OOPSLA, ECOOP, Middleware and ISWC.

The closing session of the symposium is a panel discussion where the roles are reversed, and the mentors answer the students' questions about research careers in industry and academia. The symposium agenda also contains a keynote address on writing a good PhD dissertation, delivered by Dr. Priya Narasimhan, Assistant Professor at Carnegie Mellon University and member of the ACM Doctoral Dissertation Committee.

This year we have received 18 submissions from 9 countries and 4 continents: 8 from North America, 8 from Europe, 1 from Asia and 1 from Australia. The goal of the Symposium is to provide constructive feedback to the authors of both accepted and rejected papers, so each paper has received three reviews from Program Committee members (no external reviewers were used). The submissions were evaluated based on the quality of the research, the knowledge of the area, the relevance to SoC, the quality of the presentation and the maturity of the PhD project. The program committee has finally selected 8 papers of varying maturity levels, guided by the principles of helping students who could benefit most from the feedback and letting their peers who are in an advanced stage of their PhD projects set a good example.

December 2006

Andreas Hanemann
Benedikt Kratz
Nirmal Mukhi
Tudor Dumitraş

Co-Organizers of the IBM PhD Student Symposium

Organization and Committees

The IBM PhD Student Symposium is held in conjunction with the 4th International Conference on Service Oriented Computing (ICSOC 2006). It is organized as a whole-day event on December 4, 2006 in the Hilton Chicago Hotel (Chicago, IL, USA). The homepage of the symposium is at:
`http://infolab.uvt.nl/phd-icsoc06/`.

Mentoring Committee

Rick Buskens (Lockheed Martin Advanced Technology Lab, USA)
Ian Foster (University of Chicago, USA)
Priya Narasimhan (Carnegie Mellon University, USA)
George Spanoudakis (City University, UK)

Program Committee

Claudio Bartolini (HP Labs, USA)
Boualem Benatallah (University of New South Wales, Australia)
Rick Buskens (Lockheed Martin Advanced Technology Lab, USA)
Fabio Casati (University of Trento, Italy)
Vincenzo D'Andrea (University of Trento, Italy)
Elisabetta di Nitto (Politecnico di Milano, Italy)
Rik Eshuis (Technical University of Eindhoven, The Netherlands)
Ian Foster (University of Chicago, USA)
Massimo Mecella (University of Rome, Italy)
Priya Narasimhan (Carnegie Mellon University, USA)
Mike Papazoglou (Tilburg University, The Netherlands)
George Spanoudakis (City University, UK)
Paolo Traverso (ITC/IRST, Italy)

Organizing Committee

Andreas Hanemann (Leibniz Supercomputing Center, Germany)
Benedikt Kratz (Tilburg University, The Netherlands)
Nirmal Mukhi (IBM Research, USA)
Tudor Dumitraş (Carnegie Mellon University, USA)

Sponsoring Institutions

IBM Research, USA

Table of Contents

MoSCoE: A Specification-Driven Framework for Modeling Web Services using Abstraction, Composition and Reformulation

Jyotishman Pathak

Artificial Intelligence Research Lab
Department of Computer Science
Iowa State University
Ames, IA 50011-1040 USA
jpathak@cs.iastate.edu

Abstract. We propose a new framework for modeling Web services based on the techniques of abstraction, composition and reformulation. The approach allows users to specify an *abstract* and possibly incomplete specification of the composite (goal) service. This specification is used to select a set of suitable component services such that their *composition* realizes the desired goal. In the event that such a composition is unrealizable, the cause for the failure of composition is determined and is communicated to the user thereby enabling further *reformulation* of the goal specification. This process can be iterated until a feasible composition is identified or the user decides to abort.

1 Introduction

With the recent advances in networks, information processing and WWW, automatic composition of Web services has emerged as an area of intense research in both academia and industry. The main objective of these approaches is to construct and deploy complex workflows of composite Web services by leveraging autonomously developed software components or services in several application domains including e-Enterprise, e-Business and e-Science. However, despite the recent progress, the current state-of-art in developing composite services has several limitations:

Complexity in Modeling Composite Services: For specifying the functional requirements, the service developer is required to provide a complete specification of the composite (goal) service. Consequently, the developer has to deal with the cognitive burden of handling the entire composition graph, which becomes complex to manage with the increasing complexity of the goal service. Instead, it will be more practical to allow developers to begin with an abstract, and possibly incomplete specification, that can be incrementally modified until a feasible composition is realized.

Analyzing Failure of Composition: The existing techniques for service composition adopt a 'single-step request-response' paradigm for modeling composite services. That is, if the goal specification provided by the service developer cannot be realized by the composition analyzer (using the set of available components), the entire process fails. As opposed to this, there is a requirement for developing approaches that will help identify the cause(s) for failure of composition and guide the developer in applying that information for appropriate reformulation of the goal specification in an iterative manner.

Consideration of Non-Functional Characteristics: Barring a few approaches, most of the techniques for service composition focus only on the functional aspects of the composition. In practice, since there might be multiple component services that can provide

the same functionality, it is of interest to explore the non-functional properties of the components to reduce the search space for determining compositions efficiently.

Handling Differences in Service Semantics: Individual Web services needed for realizing a desired functionality are often developed by autonomous groups or organizations. Consequently, semantic gaps, arising from different choices of vocabulary for specifying the behavior of the services, are inevitable. This requires frameworks for assembling complex Web services from independently developed component services to provide support for bridging the semantic gaps.

Motivated by these concerns, the proposed research develops a new service composition framework, MoSCoE[1] (Modeling Web Service Composition and Execution) to overcome some of the aforementioned limitations. Specifically, our work is aimed at:

- Introducing a new paradigm for modeling Web services based on abstraction, composition, and reformulation. The proposed approach allows users to incrementally develop composite services from their abstract descriptions.
- Developing a sound and complete algorithm for selecting a subset of the available component services that can be assembled into a sequential [1] and parallel [2] composition that realizes the goal service with the user-specified functional and non-functional requirements [3].
- Proposing techniques to identify the cause(s) for failure of composition to assist the user in modifying and reformulating the goal specification in an iterative fashion.
- Building on current approaches for associating semantics to Web services [4, 5] using ontologies and techniques for specifying mappings [6] between ontologies.
- Demonstrating the feasibility of the approach with real-world applications in bioinformatics [6] and electric power systems [7].

The rest of the paper is organized as follows: Section 2 talks about our contributions and results accomplished so far, Section 3 briefly discusses related work in service composition and finally Section 4 ends with conclusions and directions for further research.

2 The MoSCoE Framework

Figure 1 shows the architectural diagram of the MoSCoE framework mentioned above. Developing services in MoSCoE is based on the techniques of *abstraction, composition* and *reformulation*. The system accepts from the user, an *abstract* (high-level and possibly incomplete) specification of the goal service. In our current implementation, the goal service specification takes the form of an UML state machine that provides a formal, yet intuitive specification of the desired goal functionality. This goal service and the available component services (published by multiple service providers) are represented using labeled transition systems augmented with state variables, guards and functions on transitions, namely, Symbolic Transition Systems[2] (STS). In addition, the STSs are semantically annotated using appropriate domain ontologies from a repository by importing OWL ontologies into the UML model [10]. MoSCoE assumes that these ontologies (and mappings between them) are specified by a domain expert using existing tools such as INDUS [6]. The user also provides non-functional requirements (e.g., `cost`, `reliability`) that need to be satisfied by the goal service.

MoSCoE manipulates these input data (user-provided service specification and published component service descriptions) and automatically identifies a composition that

[1] `http://www.moscoe.org`

[2] The STS specifications for component services can be obtained from service descriptions provided in high-level languages such as BPEL or OWL-S by applying translators similar to those proposed in [8, 9].

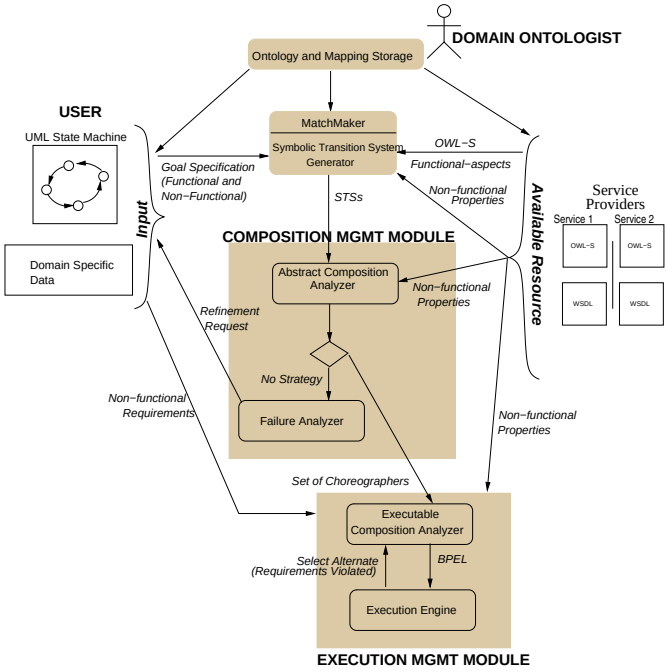


Fig. 1. MoSCoE Architectural Diagram

realizes the goal service. However, in the event that the composition cannot be realized, the system identifies the cause(s) for the failure and provides that information to the developer for appropriate reformulation of the goal specification. The framework consists of two main modules: *composition management module* and *execution management module*. The former identifies feasible compositions (if any) that realize the goal, while the latter deals with the execution of the composite service. We describe these modules in the following.

2.1 Composition Management Module

Given the STS representations of a set of N component services $\{\mathtt{STS}_1, \mathtt{STS}_2, \ldots, \mathtt{STS}_N\}$ and a desired goal $\mathtt{STS}_G$, service composition amounts to identifying a subset of component services, which when composed with a choreographer (to be generated) $\mathtt{STS}_{cr}$, realize the goal service $\mathtt{STS}_G$. The role of the choreographer is to replicate input/output actions of the user as specified by the goal and to act as a message-passing interface between the components and between the component(s) and the client. It is not capable of providing any functionality (e.g., credit card processing) on its own; these are provided only by the components. The algorithm for generating such a choreographer is discussed in [1, 2] and essentially identifies whether $\mathtt{STS}_{cr}$ indeed realizes $\mathtt{STS}_G$ using the notion of *simulation* and *bisimulation* equivalence. Informally, $\mathtt{STS}_1$ is said to be simulation equivalent to $\mathtt{STS}_2$, denoted by $\mathtt{STS}_1 \sim \mathtt{STS}_2$, if every transition sequence from the start state in the former is also present in the latter. A symmetric simulation relation, denoted by $\approx$, leads to a stronger equivalence referred to as bisimulation.

However, the algorithm proposed in [1, 2] suffers from the state-space explosion problem since the number to ways the components can be composed is exponential to the number of component states. This becomes a challenge with the increasing size of the search space of available component services. Hence, to address this limitation, we consider non-functional aspects (e.g., QoS) to winnow components (thereby reducing the search space) and compositions that are functionally equivalent to the goal, but violate the non-functional requirements desired by the user [3]. The non-functional requirements are quantified using *thresholds*, where a composition is said to conform to a non-functional requirement if it is below or above the corresponding threshold, as the case may be. For example, for a non-functional requirement involving the `cost` of a service composition, the threshold may provide an *upper-bound* (maximum allowable `cost`) while for requirements involving `reliability`, the threshold usually describes a *lower-bound* (minimum tolerable `reliability`). If more than one "feasible composition" meets the goal specifications, our algorithm generates all such compositions and ranks them. It is left to the user's discretion to select the best composition according to the requirements.

In the event that a composition as outlined above cannot be realized using the available component services, the composition management module provides feedback to the user regarding the cause of the failure. The feedback may contain information about the function names and/or pre-/post-conditions required by the desired service that are not supplied by any of the component services. Such information can help to identify specific states in the state machine description of the goal service. In essence, the module identifies all un-matched transitions along with the corresponding goal STS states. Additionally, the failure of composition could be also due to non-compliance of non-functional requirements specified by the user. When such a situation arises, the system identifies those requirements that cannot be satisfied using the available components, and provides this information to the service developer for appropriate reformulation of the goal specification. This process can be iterated until a realizable composition is obtained or the developer decides to abort.

2.2 Execution Management Module

The result from the composition management module is a set of feasible compositions each defining a choreographer that will enable interaction between the client and the component services. The execution management module considers non-functional requirements (e.g., `performance, cost`) of the goal (provided by the user) and analyzes each feasible composition. It selects a composition that meets all the non-functional requirements of the goal, generates executable BPEL code, and invokes the MoSCoE execution engine. This engine is also responsible for monitoring the execution, recording violation of any requirement of the goal service at runtime. In the event a violation occurs, the engine tries to select an alternate feasible composition. Furthermore, during execution, the engine refers to the pre-defined set of inter-ontology mappings to carry out various data and control flow transformations [5, 6].

3 Related Work

Several efforts including those based on AI planning techniques, logic programming and automata-theory, have led to the development of platforms to support composition and deployment of services. Berardi et al. [11] proposed the Colombo framework for representing services as labeled transition systems and defined composition semantics via message passing. Pistore et al. [9, 12] developed an approach for translation of BPEL and OWL-S code into transition systems and applied planning via symbolic model checking technique to do composition. Similarly, KarmaSIM [13] translates

OWL-S descriptions into petri nets to automate tasks such as simulation, validation, composition and performance analysis. Sycara et al. [14] proposed an approach for automatic discovery, interaction and composition of semantic Web services using hierarchical task network planning. More recently, the authors in [15] developed an architecture for integrating semantic Web services by automatically generating a BPEL workflow using planning-based techniques.

Several techniques have also been developed which consider non-functional requirements for service composition. Cardoso et al. [16] describe a model that allows prediction of quality of service for workflows based on individual QoS attributes for the component services. Their technique allows compensation of composition deficiency if many services with compatible functions exist. Zheng and Benatallah [17] consider service selection task as a global optimization problem and apply linear programming to find solution that represents service composition optimizing a target function, where the function is defined as a combination of multiple non-functional parameters. Yu and Lin [18] modeled the service selection as a complex multi-choice multi-dimension 0-1 knapsack problem, which takes into consideration difference in QoS parameters offered by multiple services by assigning weights.

MoSCoE builds on the approaches mentioned above and aims to provide a model-driven framework for (semi-)automatically composing Web services. However, there are certain aspects of MoSCoE which are inherently different from others. Firstly, our approach has the ability to work with abstract (and possibly incomplete) goal service specifications for realizing composite services. These specifications can be iteratively reformulated until the required functionality is achieved. Such a technique reduces the cognitive burden on the developer to begin with complete (and perhaps complex) composition model. Secondly, in the event of failure of composition, MoSCoE can identify the cause(s) for the failure and provide such information to the developer, which can be used for appropriate refinement of the goal specification. Thirdly, MoSCoE allows users to specify the non-functional aspects of the composite service which are used to select and compose individual components that realize the goal. Furthermore, these requirements are monitored during execution of the composite service to ensure compliance. Finally, MoSCoE supports specification of semantic correspondences between multiple Web service ontologies—an essential element for integration of autonomous and heterogeneous software entities.

4 Conclusion and Further Work

In this paper we introduce MoSCoE, a novel approach for automatically developing composite services by the applying the techniques of abstraction, composition and reformulation in an incremental fashion. The framework provides a goal-directed approach to service composition and adopts a symbolic transition system-based approach for computing feasible compositions. Our formalism allows us to identify and validate all possible compositions that meet the user-specified functional and non-functional requirements. However, in the event that a composition cannot be realized using the existing set of candidate services, the technique determines the cause(s) for the failure (due to violation of functional and/or non-functional requirements), and assists the user in reformulation of those requirements in the goal specification. Furthermore, by applying ontologies and inter-ontology mappings to ground service descriptions, MoSCoE provides a mechanism for bridging semantic gaps between independently developed components. MoSCoE can be used to generate potential workflows by reusing existing Web services in various domains such as power systems [7].

Our on-going work is aimed at developing heuristics for hierarchically arranging failure-causes to reduce the number of refinement steps typically performed by the user to realize a feasible composition. We also plan to explore approaches to reducing the number of candidate compositions that need to be examined e.g., by exploiting domain specific information to impose a partial order over the available services. Additionally, our framework for modeling service composition and execution has focused on services which demonstrate a deterministic behavior without loops. Handling nondeterministic behavior that often characterizes real-world services is an important area of ongoing research. Furthermore, we plan to do a systematic evaluation of scalability and efficiency of our approach on a broad class of benchmark service composition problems.

Acknowledgment. The author would like to express his sincere gratitude to Vasant Honavar (thesis advisor), Samik Basu and Robyn Lutz for their guidance and collaboration in this research. This work has been supported in part by NSF grant 0219699 and ISU Center for CILD (`http://www.cild.iastate.edu`).

References

1. Pathak, J., Basu, S., Lutz, R., Honavar, V.: Selecting and Composing Web Services through Iterative Reformulation of Functional Specifications. In: ICTAI. (2006)
2. Pathak, J., Basu, S., Lutz, R., Honavar, V.: Parallel Web Service Composition in MoSCoE: A Choreography-based Approach. In: ECOWS. (2006)
3. Pathak, J., Basu, S., Honavar, V.: Modeling Web Services by Iterative Reformulation of Functional and Non-Functional Requirements. In: ICSOC. (2006)
4. Pathak, J., Koul, N., Caragea, D., Honavar, V.: A Framework for Semantic Web Services Discovery. In: WIDM. (2005)
5. Pathak, J., Caragea, D., Honavar, V.: Ontology-Extended Component-Based Workflows - A Framework for Constructing Complex Workflows from Semantically Heterogeneous Software Components. In: SWDB. (2004)
6. Caragea, D., Pathak, J., Bao, J., Silvescu, A., Andorf, C., Dobbs, D., Honavar, V.: Information Integration and Knowledge Acquisition from Semantically Heterogeneous Biological Data Sources. In: DILS. (2005)
7. Pathak, J., Li, Y., Honavar, V., McCalley, J.: A Service-Oriented Architecture for Electric Power System Asset Management. In: WESOA. (2006)
8. Pistore, M., Traverso, P., Bertoli, P.: Automated Composition of Web Services by Planning in Asynchronous Domains. In: ICAPS. (2005)
9. Traverso, P., Pistore, M.: Automated Composition of Semantic Web Services into Executable Processes. In: ISWC. (2004)
10. Duric, D.: MDA-based Ontology Infrastructure. Computer Science and Information Systems **1**(1) (2004) 91–116
11. Berardi, D., Calvanese, D., Giuseppe, D.G., Hull, R., Mecella, M.: Automatic Composition of Transition-based Semantic Web Services with Messaging. In: VLDB. (2005)
12. Pistore, M., Traverso, P., Bertoli, P., Marconi, A.: Automated Synthesis of Composite BPEL4WS Web Services. In: ICWS. (2005)
13. Narayanan, S., McIlraith, S.: Simulation, Verification and Automated Composition of Web Services. In: WWW. (2002)
14. Sycara, K., Paolucci, M., Ankolekar, A., Srinivasan, N.: Automated Discovery, Interaction and Composition of Semantic Web Services. Journal of Web Semantics **1**(1) (2003) 27–46
15. Agarwal, V., Dasgupta, K., et al.: A Service Creation Environment Based on End to End Composition of Web Services. In: WWW. (2005)
16. Cardoso, J., Sheth, A., Miller, J., et al.: Quality of Service for Workflows and Web Service Processes. Journal of Web Semantics **1**(3) (2004) 281–309
17. Zeng, L., Benatallah, B.: QoS-Aware Middleware for Web Services Composition. IEEE Transactions on Software Engineering **30**(5) (2004) 311–327
18. Yu, T., Lin, K.: Service Selection Algorithms for Composing Complex Services with Multiple QoS Constraints. In: ICSOC. (2005)

REST-Based Service Oriented Architecture for Dynamically Integrated Information Systems

Daniel Szepielak

Deutsches Elektronen-Synchrotron, Notkestrasse 85, 22607 Hamburg, Germany
dszepielak@o2.pl

Abstract. Integration solutions based on Service Oriented Architecture typically follow an operation centric approach which puts emphasis on the diversity of operations. This paper describes a resource centric approach based on REST web services which provides means of representing legacy systems data and processes as resources which are accessible through uniform interfaces and standardized operations. A key component of the proposed integration approach is a web service registry which enables dynamic ontology-based service discovery and execution.

1 Introduction

The complexity of business models and software environments of modern enterprises puts emphasis on the need for cross-application communication. In such complex environments information systems can no longer operate as independent entities without having a negative effect on enterprise efficiency. Integration of information systems is thus a frequent challenge in enterprise environments.

Like any software environment, an integrated software environment changes over time. Evolution of technology and changes in business needs are reasons for upgrading or replacing existing applications to achieve better, faster and up-to-date software environments. Such continuous evolution demands flexible and scalable integration platforms that minimize the amount of human involvement necessary for adaptation and maintenance. The dynamic integration concept addressed by the following research fulfills these requirements by introducing mechanisms which are capable of discovering and accommodating a growing number of systems and services during operation.

2 Background

The concept of integration has been present in the software development domain in various forms for the last two decades. The approach currently considered to be the best practice for addressing the integration problem is the Service Oriented Architecture (SOA) [1]. The main principle of SOA is to deliver software components in form of web services - small, self-contained modular applications that support direct interaction with other software agents through exchange of XML messages in a platform and programming language independent manner.

The concept of SOA and Web Services is typically associated with the Simple Object Access Protocol (SOAP) due to wide adoption and industrial support for SOAP-based web services. SOAP defines a messaging protocol between requestor and provider objects, allowing the requesting objects to perform remote method invocation on the providing objects in an object oriented programming fashion. An alternative to SOAP web services has been proposed in form of the Representational State Transfer (REST) [2]. The motivation for REST was to apply the successful characteristics of the Web to the SOA. REST achieves this by exposing resources in form of XML documents accessible through URIs using a uniform HTTP protocol interface. This approach results in REST straightforwardness and simplicity.

A key component of any SOA environment is the service registry which enables dynamic service discovery based on descriptions of services capabilities. Typical service description and discovery mechanisms depend on WSDL and UDDI standards, however ontology-based methods for semantic service description has also been proposed (e.g. OWL-S) [3]. Ontologies aim at capturing the exact meaning of data by providing explicit formal specifications of concepts used in a particular domain [4], based on semantic relationships observed among these concepts.

3 Integration Approach

3.1 Resource Centric Approach

In most SOA-based integration approaches web services can perform any operations possible within an information system. This freedom leads to highly complex descriptions of web service capabilities. The discovery mechanisms that depend on these descriptions are typically limited to matching of service inputs and outputs [5]. As services from different providers that deliver the same functionality can have completely different combinations of inputs and outputs, except for trivial cases it is often impossible to automatically discover or utilize them without prior adaptation to their interfaces [6].

The presented research describes an approach which does not require complex service description and thus results in easier interoperability among information systems. The interoperability problem is addressed by providing a fixed functional interface that limits the freedom of operation definition, but assures common understanding of defined operations and uniform representation of objects (resources), on which they operate. This approach can be described as the resource centric approach and its successful application can be found for example in the database domain where every table is considered to be a resource and SELECT, UPDATE, INSERT and DELETE operations are used for data manipulation. This pattern is often referred to as CRUD [7] which is an acronym for Create, Retrieve, Update and Delete. While the CRUD pattern is a practical and effective application of the resource centric approach, the operation set does not have to be limited to CRUD and can be adapted if necessary - as long as the meaning of all operations is defined and known to providers and consumers.

3.2 Ontology

Integrating information systems using the resource centric approach requires defining various resource types which are present in a company's software environment. In the described solution this task is realized with the help of an ontology. This ontology should reflect only business meaningful concepts shared among participants of the integrated environment. Defining too fine grained resources can significantly increase the number of necessary services.

A small fragment of an ontology that has been built for the purpose of this research is shown in Fig.1. The general ontology concepts are mapped to individual concepts in specific information systems.

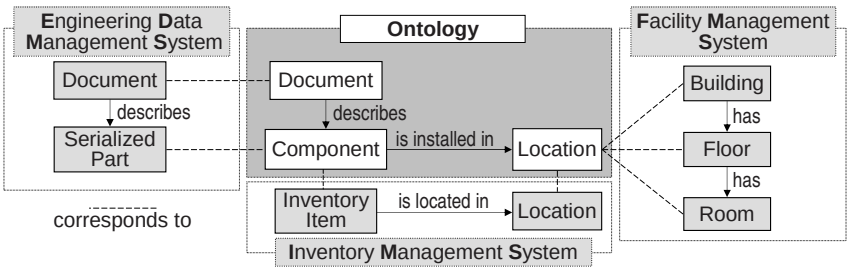


Fig. 1. Mapping of *Ontology* concepts to systems specific concepts.

The example ontology addresses the documentation of technical infrastructure in general: it states that *Documents* describe *Components* which are installed in *Locations*. It was created for an environment containing three information systems: EDMS, IMS, FMS each offering partial implementations of the ontology concepts - each from its own perspective. The ontology provides a combined view at the resources of the underlying systems' e.g. the *Component* class offers a unified view at the EDMS *SerializedPart* and the IMS *InventoryItem*.

The example ontology shows resources that can be considered as data objects, however, a business model of a typical company consist not only of data but also of processes. In the proposed approach the concept of process is not explicitly defined, but processes can be encapsulated and exposed as resources. Operations on resources can be used for controlling processes e.g. creating a resource can initialize a process, retrieving it can be used for checking the process status etc.

3.3 REST Web Services for Resources

The REST architectural style has been chosen for realizing the resource centric approach for the reason of its simplicity and because it can be mapped to it in a natural way. The HTTP methods can be mapped to the Create (POST), Retrieve (GET), Update (PUT) and Delete (DELETE) operations and the XML documents can be used to provide a uniform resource representation.

Building an integrated environment using the proposed approach requires delivering interfaces for accessing the resources which are defined in the ontology. For each resource type, a set of REST web services is required that provides the CRUD operations. The complete set of services for all resource types forms a service pool which provides uniform access to data stored in the underlying information systems (Fig.2).

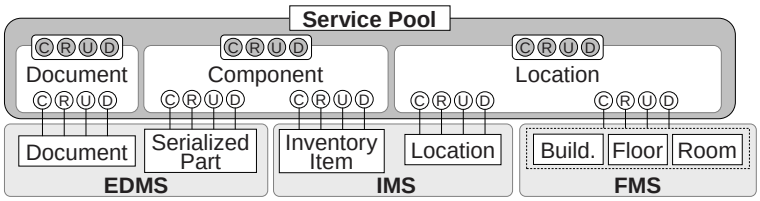


Fig. 2. Set of web services offered by information systems to the *Service Pool*.

The proposed approach requires developing families of web services for all specific ontology classes. For supporting the development efforts, a prototype framework has been built which facilitates reuse when developing families of web services. It provides functional blocks that can be generalized for all web services as well as implementation skeletons for blocks that have to be implemented for every specific information system and every ontology class. The development of web services for the purpose of integration is thus reduced to delivering appropriate system drivers which implement the required behavior on specific resources in the specific information system.

3.4 Registry

The key component of a dynamically integrated environment is a web service registry. The proposed registry utilizes the conceptual model provided in the Ontology to build a Service Directory, which extends the conceptual model representation of the ontology to include information about operations and their providers (Fig.3). The two core components of the registry which coordinate access to these repositories are the *Ontology Engine* and the *Services Directory Engine*. With their help providers can selectively register single operations which act on resources only, in contrast to simply registering any service as done frequently in registries like UDDI.

In addition to acting as a single point of contact for service providers and requestors, like any registry does, the proposed registry follows a novel concept of directly accessing object fragments from different sources and assembling them into high-level objects according to the ontology. This functionality is enabled by the *Request Execution Module* which allows clients to directly operate on resources without any knowledge about their providers, almost as if the registry itself would be providing all the services. Instead of first querying the registry

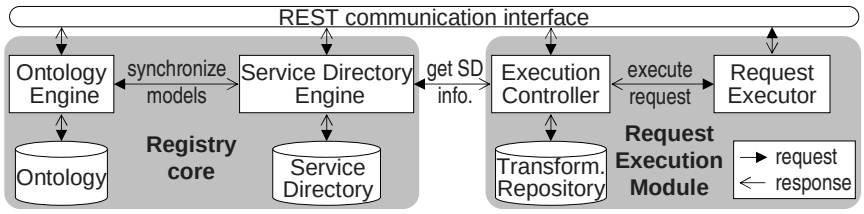


Fig. 3. Registry architecture overview.

for a list of providers and then sending requests to them, the client can simply send the request for required information directly to the *Execution Controller*. The *Execution Controller* automatically retrieves from the *Service Directory* the information which is required for identifying the proper providers, adapts partial requests to provider interfaces using *Transformation Repository* and passes them on to the *Request Executor* threads which invoke individual web services. The partial responses are then combined into a complete response which is returned to the client. In perspective of the earlier example, the client could e.g. ask for the complete information about a *Component* resource and the *Request Execution Module* would retrieve and compose the data from both the EDMS and the IMS. The described method allows clients to ask for information without worrying who will deliver it, what usually corresponds to the client point of view.

3.5 Integration Architecture

The overview of the proposed integration architecture is shown in Fig.4.

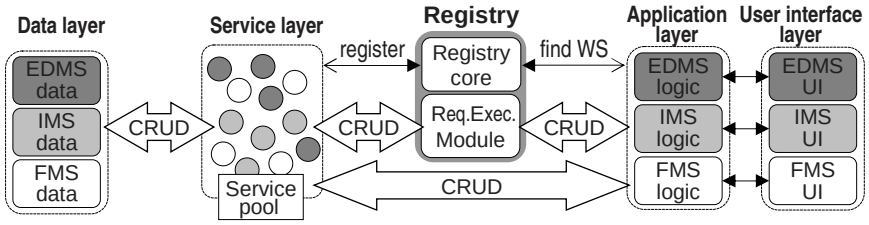


Fig. 4. Overview of intended integration architecture.

The Data Layer corresponds to the resources defined in the ontology. Access to these resources is provided via web services in the Service Layer which offers a uniform interface for manipulating various resources. The business processes are controlled by the Application Layer, which can utilize web services either through the *Request Execution Module* or find them using the *Registry Core* and use them directly. The received responses are processed within the business logic modules and the results are passed on to the User Interface Layer.

3.6 Expected Contributions

This paper presented an overview of the research project which expects the following contributions within the field of Information and Computer Science:

- a resource centric approach for integration of information systems
- an integration architecture for realizing dynamic integration based on resource centric approach
- an ontology based registry for REST web services enabling dynamic service discovery and execution
- a framework for building REST web services for the resource centric integration

3.7 Results and Conclusions

This research project originated from the general research conducted at Deutsches Elektronen-Synchrotron (DESY) in Hamburg. The research is still ongoing, however a number of components have been already developed and put into operation including: an ontology for DESY information systems, a framework for REST web services development achieving an average level of 90-95% code reuse, a prototype registry for REST web services and a set of web services for accessing the core resources within multiple information systems.

The presented approach has proven to be an efficient method for the integration of legacy systems. The ontology based uniform service interfaces ensure common interpretation of all available services and thus allow minimizing necessary maintenance work in case of modifications of the integrated environment. Experience shows that the resource centric approach is particularly suitable for environments focused mainly on data manipulation. While it is possible to model business processes as resources, the proposed approach can be hard to realize in environments with complex and numerous processes.

References

1. Erl, T.: Service-Oriented Architecture (SOA): Concepts, Technology, and Design. Prentice Hall PTR, Upper Saddle River (2005)
2. Fielding, R., Taylor, R.N.: Principled design of the modern Web architecture. ACM Press New York, NY, USA (2000) 407-416
3. Balzer, S., Liebig, T., Wagner, M.: Pitfalls of OWL-S: a practical semantic web use case. ICSOC'04. ACM Press, New York (2004) 289-298
4. Dietz, J.L.G.: Enterprise Ontology: Theory and Methodology. Springer, New York (2006)
5. Grigori, D., Corrales, J.C., Bouzeghoub, M.: Behavioral matchmaking for service retrieval, ICWS'06. IEEE Computer Society Press, Los Alamitos, CA. (2006) 145-152
6. Shirky, C.: Planning for Web Services: Obstacles and Opportunities. O'Reilly Media, Sebastopol, CA. (2002)
7. Wiegers, K.E.: Software Requirements. 2nd edn. Microsoft Press, Redmond, Washington (2003)

Self Checking Protocols: A Step towards Providing Fault Tolerance in Services

Gunjan Khanna

Dependable Computing Systems Lab,
School of Electrical and Computer Engineering,
Purdue University, IN
gkhanna@purdue.edu

Abstract. In this paper we present ongoing work on building a detection and diagnosis system for Services architecture. We propose a hierarchical detection and diagnosis framework instantiated in a system called the Monitor. The Monitor verifies the messages exchanged between services against an anomaly based rule set. The Monitor architecture is application neutral making it generically applicable to a wide variety of web applications composed of services. Monitor treats service entities as black boxes making it non-intrusive. We provide a hierarchical framework which scales with the number of service entities. We extend the Monitor framework to provide diagnosis of failures through causal dependency modeling. The contribution of this work is a step towards building autonomic management system for a large class of web applications that are deployed today.

1 Introduction

With the realization of services as an essential part of the driving force in the economy, several research endeavors have been initiated for understanding and development of services. Services could potentially belong to different management domains or completely different countries. Reliability is an important parameter which must be provided by the conglomeration of services working together to achieve a task. Because of the distributed nature, proprietary control over individual services and only message interactions, fault tolerance is an extremely challenging task. In many situations, it is necessary to not just detect a failure, but also to diagnose the failure, i.e., to identify the source of the failure. Several solutions to diagnosis have been proposed in different contexts including [4], [5], [6]. Diagnosis (or fault localization) is challenging since high throughput applications with frequent interactions between the different services allow fast error propagation. In service oriented applications only descriptors about the services are available, with no access to internal components. Hence, it is desirable to consider services as black-boxes for the diagnostic process. The aim of this thesis is to develop a framework which can provide detection and diagnosis of failures in high throughput applications composed of black box services. It is a step towards developing an autonomic infrastructure for service management.

We propose a hierarchical Monitor architecture for fault detection and diagnosis in services architecture. There are several design motivations for the Monitor system. First, it is desirable that the Monitor system operate asynchronously to the services, so that the application's throughput does not suffer due to the checking overhead. Second, there is a requirement of fast detection and diagnosis, so that substantial damage due to cascaded failures is avoided. It should be able to catch error propagation from one service to another due to message exchanges. Third, the monitor system is not intrusive to the verified application or the group of services. This matches perfectly with the services concept because most of these services could be legacy and argues in favor of having the payload system be viewed as a black-box by the Monitor system. While it is possible to build very optimized and specialized fault tolerance mechanisms for specific applications (e.g., like in [1] with the TRAM protocol), such solutions do not generalize well across applications.

Figure 1 represents how we envision Monitor working in a distributed setting. Monitor snoops over the message interactions between the services. The fault model for services consists of configuration errors, messaging errors, and cascaded failures. It verifies these interactions against a set of rule base to provide detection of failures.

Services Oriented Architecture (SOA) comprises of multiple services working in tandem to support an application typically interact through messages (see Figure 1). Also these services can be provided through multiple vendors and across multiple domains. Message interaction and service discovery is supported by Enterprise Service Bus [9]. Monitor is a natural fit to SOA because of black-box abstraction of services and complete reliance on message exchanges for detection and diagnosis. In SOA, service providers provide services with descriptors which can be used for accessing the service. These service descriptors (for e.g., Web Services Description Language (WSDL)) and Service Level Agreements (SLA's) can be used together to generate the rule base used by the Monitor(s) (see Figure 1).

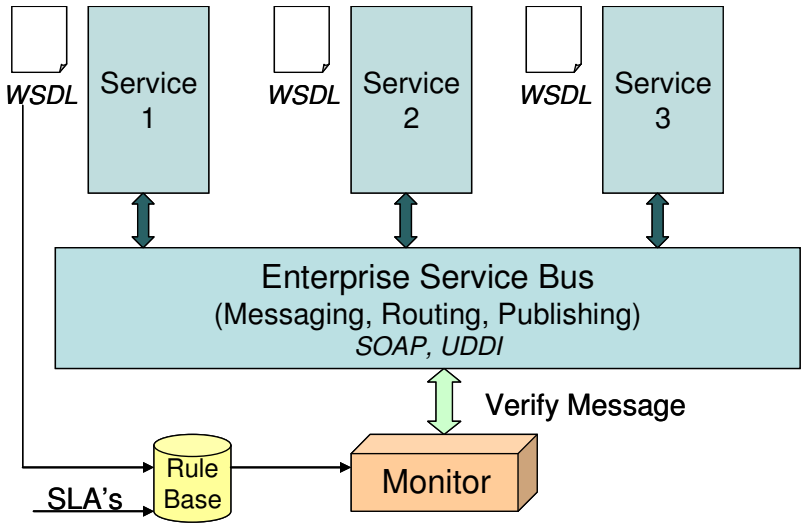


Figure 1: Logical View of the Monitor in a services domain.

The problem of diagnosis in distributed systems has been studied in [10] which have relied on participation by the protocol entities and the classes of faults have also been restricted. Authors in [5] provide a black-box mechanism to find causal dependencies but they do not provide any diagnosis.

The contributions of this work are as following: 1) Asynchronous detection and diagnosis of failures in distributed applications, 2) It is application neutral and treats the entities as black-box with no explicit tests given to the entities, 3) Resilient to failures within the monitoring framework, and 4) Provides scalability through distributed hierarchical monitoring.

2 Monitor Architecture and Detection

The Monitor architecture consists of several modules classified according to their functional roles. These modules include, in order of their invocation, the Data Capturer, the State Maintainer, the Rule Matching Engine, the Decision Maker, the Interaction Component, and the Rule Classifier. Figure 2 gives a pictorial representation of the Monitor components.

The details of the structural and functional roles of the components have been described in [2] and only a basic description is given here. The Data Capturer is responsible for 'capturing' the messages exchanged between the service participants over the network, and passing it on for further processing by the Monitor. Message capturing can be through passive monitoring of traffic or using active forwarding support from the service entities (or Enterprise Service Bus). Monitor may be placed in the same domain (for e.g. LAN) as the service entities or in a completely different domain, with service entities providing active forwarding of messages. Port mirroring on switches and routers can also achieve forwarding of messages to the Monitor without any cooperation from the service.

The State Maintainer contains static information of the reduced state transition diagrams for each verified service and dynamic information of the current state of each. The combination of current state and incoming event determines the set of rules to be matched. The Matching Engine is invoked by the state maintainer when an incoming packet triggers a rule that has to be matched. This component is highly optimized for speed to reduce the detection latency. Once the matching engine finishes its rule matching, the Decision Maker combines the results of rule matching for the different rules in the rule base and raises an appropriate flag in case of error. The Interaction Component deals with communication between Monitors at different levels in the hierarchical approach.

As previously stated, rules can be generated from WSDL descriptions and SLA constraints. Rules are classified into temporal and combinatorial categories[2]. The rule syntax is rich so as to be able to express a large class of interactions. For all seen messages, the Monitor verifies if it adheres to the rule base. We have developed fast matching algorithms to support online detection. In [2] we show detection feature of the Monitor by applying it to a distributed TRAM protocol across Purdue's network. We obtain a high accuracy of failure detection with very low latency. Detection of problems or failures is a first step in fault tolerance. This needs to be followed by a diagnosis mechanism which pinpoints the exact cause of the failure.

3 Diagnosis using Monitor(s)

Providing diagnosis is challenging because the Monitor treats the entities as black-boxes. It is thus unaware of the valid request-response for the service and cannot send any explicit test message to these service entities. Moreover, the service instance may not currently be in the same state as the one in which the fault was triggered. In practical deployments, the monitoring framework may not have perfect observability of the service instance because of congestion or specific placement of the entities and the monitoring framework. A failure manifested at a service entity could be because of a fault which originated at this service entity or because of error propagation through a message which the service entity received. If the error is propagated through a message then it must causally precede the message which resulted in failure detection. All services which interacted with the failed service could be possible causes of failure because of error propagation.

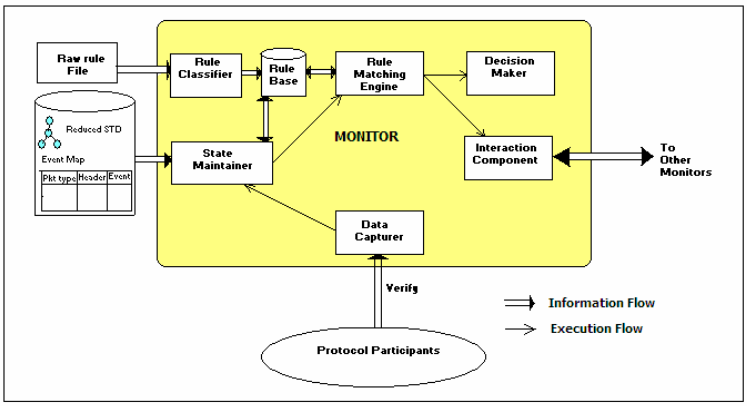


Figure 2: Monitor architecture with process flow and information flow among multiple components

During runtime the Monitor uses the incoming messages to construct a *causal graph* (CG) where a node represents the service and links represent the exchanged messages between different services[7]. On detection of failure, a Diagnostic Tree (DT) is built with the entity at which failure was detected being the root of the tree. Each node in the DT is checked using a set of *causal tests* (like diagnosis tests) which checks the causal behavior of the service. These causal tests are obtained from the actual causality present in the application. These can be easily derived from the service's formal specification like WSDL, or STD.

A causal test has a specific format to verify the causal relationship between the messages and thus to address error propagation. It tests that all the causal messages in the service for a particular state did transpire. These tests are needed only when a service failure happens because overhead of executing all the tests during the entire operation of the application protocol would be prohibitive and unnecessary if no failure occurs.

The service entities may be spanning several networks and even organizational boundaries and be verified by different individual monitoring modules each of which constructs a part of the CG obtained from its local information. This entails the requirement of distributed diagnosis. Complete transfer of the local CGs to construct global information at some central monitoring authority is not scalable. Instead, we design a distributed protocol under which each Monitor performs local computations and then it is aggregated at a single higher level Monitor which pinpoints the faulty service.

3.1 Further Challenges

A monitoring framework as proposed in this paper is likely to be affected by failures. It is quite likely that an arbitrary failure within the Monitor can lead to potential missed alarms or false alarms. Besides, there can be challenges related to rule base mis-configurations, load balancing, guarantee order amongst Monitor(s) etc. Due to space constraints we do not discuss these issues but would like to refer the reader to [2], [3], [7] on how to address these challenges.

4 Future Direction & Conclusions

4.1 State Space Reduction

Distributed applications in use today have a large state space due to the complexity of the individual services and the large scale of the application. This leads to the well known problem of *state space explosion* commonly occurring during the process of formal verification of hardware design or distributed protocols. This problem persists in a service oriented domain as well because the state of the entire system (S_s) is a tuple consisting of state of each service i.e., $S_s = <s_1, s_2 \ldots . s_i \ldots>$. State space explosion hampers the role of the fault tolerance or management system that reasons about the state of the system, especially if the system especially if the system consists of services which are real time.

The fundamental premise is that not all states are equally valuable from the point of view of fault tolerance services. The less valuable states can be ignored and the more valuable ones tracked reducing the total number of states that needs to be tracked. Often it is desired by the system administrator to narrow down on few suspecting services. Verification of the entire state space in such a scenario is futile and expensive.

Monitoring systems typically have a set of rules against which the monitored service is verified. Additionally, physical placement of these monitoring systems may cause inability to see all the external activities of the service entity, making some of the external transitions unobservable. For monitoring systems, we identify two sources for doing state space reduction; 1) Removing internal or unobservable states or transitions, and 2) Removing the states which have no rules for verification.

The former is important for correctness as well as performance since an internal transition cannot be tracked accurately by a monitoring system and will therefore have to be estimated, perhaps inaccurately. The latter is important because it lets the system administrator lead the detection and diagnosis system to focus on behavior of importance to the system. However, any reduction step that is designed should be agnostic to the monitoring system's performance. We are currently exploring mechanisms which provide state space reduction without any effect to the semantics of the service as perceived by the monitoring system.

4.2 Diagnosis in High throughput streams

State space reduction helps reducing the latency of detection but in a high throughput scenario, it is unlikely to verify of all the messages which are exchanges between the service modules. Algorithms have been proposed in theoretical computer science to obtain φ-quantile measure of stream including [8], which shows how to perform detection using optimized bloom filters. In spite of these approaches there is a lack of research, in performing on-line diagnosis. Typically in existing systems, after online detection, systems perform off-line diagnosis for high message stream applications. We are focusing on developing an approach to perform accurate, real-time diagnosis using an approximate estimate of the incoming stream.

References

[1] G. Khanna, J. Rogers, and S. Bagchi, "Failure Handling in a Reliable Multicast Protocol for Improving Buffer Utilization and Accommodating Heterogeneous Receivers," In the 10th IEEE Pacific Rim Dependable Computing Conference (PRDC), pp. 15-24, 2004.

[2] G. Khanna, P. Varadharajan, and S. Bagchi, "Self Checking Network Protocols: A Monitor Based Approach," In the 23rd IEEE Symp. on Reliable Distributed Systems (SRDS), pp.18-30, 2004.

[3] G. Khanna**,** P. Varadarajan, and S. Bagchi, "Automated Online Monitoring of Distributed Applications Through External Monitors**,**" in IEEE Transactions on Dependable and Secure Computing (TDSC), 2006.

[4] A. Brown, G. Kar, and A. Keller, "An Active Approach to Characterizing Dynamic Dependencies for Problem Determination in a Distributed Environment," In Intl. Symp. on Integrated Network Management (IM), 2001.

[5] M. K. Aguilera, J. C. Mogul, J. L. Wiener, P. Reynolds, and A. Muthitacharoen, "Performance Debugging for Distributed Systems of Black Boxes," In the ACM Symp. on Operating Systems Principles (SOSP), 2003.

[6] R. Buskens and R. Bianchini Jr., "Distributed On-line Diagnosis in the Presence of Arbitrary Faults," In 23rd Intl. Symp. on Fault-Tolerant Computing (FTCS), 1993.

[7] G. Khanna, Y. Cheng, S. Bagchi, M. Correia, and P. Verissimo, "Probabilistic Diagnosis through Non-Intrusive Monitoring in Distributed Applications," Purdue ECE Technical Report 05-19, December 2005.

[8] G. Cormode, S. Muthukrishnan, "What's New: Finding Significant Differences in Network Data Streams", IEEE INFOCOM (2004).

[9] IBM: Service Oriented Architecture, Online: www.ibm.com/soa.

[10] I. Katzela and M. Schwartz, "Schemes for Fault Identification in Communication Networks," In IEEE/ACM Trans. On Networking, vol. 3, no. 6, pp. 753-764, Dec 1995.

Internet-Scale Web Services-based Event Notification Systems

Yi Huang

Department of Computer Science, Indiana University
150 S. Woodlawn Ave, Bloomington, IN, 47405, USA
yihuan@cs.indiana.edu

Abstract. Internet-scale Web services-based event notification services face several new challenges that have not been addressed in traditional event notification systems. In this paper, an Open Publish/Subscribe (OPS) model is proposed to address these new challenges. The openness of OPS model enables collaboration among different event notification systems from different organizations. The layered architecture of this model separate different concerns in event notification systems and can be used to address several challenges in Web services-based event notification systems. Two issues we will address in the thesis are supporting the expressive XPath filtering on XML messages without losing scalability and reconciling the competing Web services specifications on such systems.

1. Introduction

Web services-based event notification is an emerging technology that combines the asynchronous publish/subscribe communication feature of the event notification mechanism and the interoperability feature of Web services technologies. Web services-based event notification systems are getting more and more attention in the emerging Service-Oriented Architecture (SOA) and Event-Driven Architecture (EDA) for integrating applications both within an organization and across organization boundaries. WS-Eventing [1] and WS-Notification [2] are two major specifications for such systems. These specifications aim at defining standard Web service interfaces for subscription management and delivering XML-formatted event notification messages.

There are some major changes from traditional event notification systems to Web services-based event notification systems [3]. *First*, the format of notification messages is XML-based SOAP message format. XML has been widely adopted as a standard data exchange format. Comparing with name-value pair messages in traditional systems, XML messages provides both semantic and structural information on the data and it is supported by major software vendors. However, XML messages are larger and more complicated than name-value pair messages. Parsing these messages is much more computation-intensive. *Second*, interoperability among different administration domains on the Internet becomes a major concern since event sources and event consumers are less likely to be under the control of the same

administrator or implemented by the same vendor. *WS-Messenger* [4] is a research project on unique challenges in Web services-based publish/subscribe systems.

2. Problem statement

Notification brokers are key components for scalable, loosely coupled publish/subscribe systems. They provide an abstraction layer between event sources and event consumers. Broker network consists of two or more notification brokers to take advantage of locality, improve scalability and reduce network bandwidth usage. Internet-scale XML publish/subscribe services are getting more and more attentions since they are key components for large-scale distributed applications [5, 6]. Handling notifications on the Internet-scale faces many new challenges that are not addressed on the Intranet-scale, such as message routing, higher message rate and subscription populations, variable network conditions, unreliable resources and firewalls.

Existing researches on Internet-scale brokering system assume fully cooperative broker networks. Each broker in a broker network needs to trust every other broker and perform all required responsibilities, such as message routing, query processing, message transformation and message aggregation. Communications among them are proprietary protocols. However, to achieve a practical Internet-scale brokering system, the brokers usually need to be deployed in different administrative domains. Due to different economical and political concerns, cooperation among different organizations across the world is very hard and hoping everyone using brokers from the same vendor is impractical. Current Web services-based event notification specifications only define publish/subscribe interfaces for clients. They do not address interoperability among notification brokers from different organizations. An *open interoperable* brokering network model that can link different organizations is desired for Internet-scale brokering systems. How to achieve this is a research challenge.

Expressiveness and scalability have been treated as trade-offs in publish/subscribe systems [7]. Topic-based subscriptions are simple and easy to scale, but they are not expressive. If publishers cannot provide fine-grained topics, subscribers have to subscribe to all messages on that topic. Content-based subscriptions are more expressive. They allow subscribers to specify content characteristics. But they usually require more processing time in the message filtering process and their scalability has been a research problem [7]. Content-based XML message filtering aggravates this problem. XPath expression [8] is the desirable subscription language since it allows fine-grained selection by both structure and content. However, parsing XML messages and evaluating XPath query are much more expensive than processing traditional content-based name-value pair messages. How to achieve both the rich expressiveness in XPath subscriptions and scalability has been a challenge [5]. When brokers are distributed across Internet in different organizations, how to achieve them in an open and interoperable way without losing security is an even harder challenge.

Two major competing specifications on Web services-based event notification systems are WS-Eventing [1] and WS-Notification [2]. They are incompatible with each other and both have implementations from different projects. In many application integration scenarios, it is not feasible to change the specification

implementations used in existing systems, especially if the change involves two or more organizations [9]. In order to keep existing Web services-based event notification systems unchanged and achieve interoperability among various implementations of competing specifications, notification brokers should be able to transparently reconcile the differences among event notification specifications. No previous work can achieve this. Also, due to different specifications and different versions, there are potentially a large amount of different operation messages or notification messages in different formats, how to reconcile them in a scalable way is another challenge.

This thesis will address the above mentioned three problems by proposing an open model and by separating different concerns to different layers in this model.

3. Related work

XML message filtering and transformation in publish/subscribe systems. Many content-based notification broker projects are available. Traditional content-based publish/subscribe systems with broker networks include SIENA [7] , Gryphon [10], HERMES [11], JEDI [12]. They usually filter messages with a simple name-value pair structure. Research efforts on efficient XPath-based filtering for more complicated XML message contents include XFilter [13], YFilter [14], Xaos [15], XSQ [16]. These XPath filters are designed for processing XPath evaluations in a single machine, mostly for XML database, and are not designed to support notification broker networks for scalability.

PsEPR project [6] deploys an Internet-scale XML message brokering system, but it uses topic-based message filtering. Currently, there are three broker networks that support XPath-based filtering, ONYX [5], XNet [17] and Naradabrokering [18]. Only NaradaBrokering has Web services implementation and only ONYX provides message format transformation based on XQuery. Unlike *WS-Messenger*, every broker in these projects needs to inspect XML message contents to make routing decisions which is ineffective and lack of security. Neither mediation on message formats from competing Web services specifications nor interoperability among event notification brokers from different vendors is addressed.

Open interoperability model for messaging systems Several open interoperability approaches are available. Some specifications, such as JMS, define *API-level* interoperability. Some communication protocols, such as TCP protocol and IP protocol, define *binary-level* interoperability. Recently, *message-level* interoperability is gaining momentum. Web services achieve interoperability by defining common XML formats. WS-Eventing [1] and WS-Notification [2] define interfaces between clients and brokers. However, they do not define an interoperable model for message filtering and delivery among brokers. Advanced Message Queuing Protocol (AMQP) specification [19] is a recent draft specification that aims to create a “totally open, platform agnostic, interoperable” messaging infrastructure for collaboration among queuing services in different organizations. The message-level open interoperability approach of this specification is similar to the OPS model in this paper, but its domain is not publish/subscribe messaging system.

3. OPS model

In this thesis, I propose Open Publish/Subscribe (OPS) model as an open and generic model for Internet-scale event notification broker networks for XML messages. This model is similar to the open TCP/IP model that Internet is built upon but it is at a higher level. This model separates different concerns in publish/subscribe systems to different layers. Five layers are included in this model: event transportation layer, event distribution layer, query processing layer, event transformation layer and event application layer. Each layer can achieve its responsibility independently without much effect on the other layers. Table 1 summarizes the responsibilities of each layer in the OPS model. In the process of delivering a notification message from a publisher broker to a consumer broker, intermediary brokers in a broker network only need the bottom two layers. This means much simpler architecture and much less processing load for intermediary brokers.

Table 1. Responsibilities of each layer in OPS model

Layers	Responsibilities
Event Application (layer 5)	Generates/consumes notification messages, including Complex Event Processing
Event Transformation (layer 4)	Transforms notification messages for common processing or for individual end consumers
Query Processing (layer 3)	Filter/check message according to subscriptions
Event Distribution (layer 2)	Determines list of next routing brokers and/or list of consumers for messages
Event Transport (layer 1)	Delivers messages between brokers

4.1 Features of OPS model

OPS model has two major features. **First, it is an open model for interoperability.** OPS model defines open and simple message formats at the event distribution layer for communications among notification brokers. Interoperability is based on messages on the wire. We believe that in order to have a wide deployment of collaborative brokers from different organizations on the Internet, a *least responsibility* approach is needed. Collaborative brokers only need to perform its core task which is message routing. Other responsibilities should be handled by higher level applications as services, possibly provided by different vendors. In this way, different organizations can achieve easier collaboration on creating an Internet-scale event delivery network without too much burden on resources. An analogy of this approach is the success of the Internet. We only need to trust the routers in other organizations to forward our messages based on IP address headers, and DNS servers can provide us with correct naming information. We do not require them to perform security checks on our IP packets or remove duplicates.

Second, OPS model uses a layered model to achieve separation of concerns for scalability and security. Each layer in OPS model can be scaled out independent of other layers. Different layers may have different scalability concerns and need

different scaling approaches. For example, scaling out event distribution layer is different from scaling out query processing layer as described in next section.

The security benefit of the separation of concerns is that it enables routing of encrypted messages in content-based filtering. This is not feasible in previous content-based message filtering systems. In OPS model, only publisher brokers and consumer brokers need to check message content for content-based message filtering, intermediary brokers just need to check FRS (filtering result summary) created by the query processing layer in the message headers to determine the message routing. Messages can be encrypted by publisher brokers for traversal on the Internet through different administrative domains. In this way, secure communication channels can be established between publishing organizations and consuming organizations.

4.2 Applications of OPS model

As sample applications of OPS model, I will address the other two research problems describe in section 2 using the open interoperability model. They can be solved at different layers in OPS model.

Expressiveness vs. Scalability problem for XML messages This problem will be addressed by separating the scalability of event distribution with the scalability of event filtering. At the event distribution layer, scalability concern is how to route messages efficiently from publisher brokers to consumer brokers. Scaling out approaches include adding more delivery agents for a broker and adding more brokers in different locations Routing decisions can be made by checking only the Filtering Result Summary (FRS) generated by the query processing layer and attached as the header of an XML notification message. Neither XML parsing nor XML content inspection is needed. The scalability concern in the query processing layer is how to efficiently matching subscriptions with XML messages. Since the query processing happens at the publishing brokers, scalability can be achieved by using more query processing servers in a cluster locally for a high-rate publisher. In our implementation, XPath evaluation will be based on YFilter [14] and FRS format will be based on a simple XPath canonicalization protocol we designed.

Scalable mediation among competing Web services specifications To solve this problem, the transformation layer in OPS model is used to transform message formats automatically based on the specifications used in subscription messages. Specifically, a Normalization-Processing-Customization (NPC) approach is used in our mediation solution [9]. The contribution of this solution is that it uses normalized information set to simplify mediation among Web services and makes mediation scalable. Mediation among N different formats only needs 2N mediation rules, instead of N^2.

5. Conclusions

In conclusion, researches in the *WS-Messenger* project concentrates on unique challenges in Web services-based event notification systems. This thesis proposes an Open Publish/Subscribe model for Internet-scale XML message brokering. It studies how to achieve scalable XPath-based message filtering and how to reconcile

competing Web services specifications under this open interoperability model. We will deploy our system in wide-area networks, e.g. PlanetLab [20], and conduct various performance tests in real-world network environments.

References

1. Box, D., et al. Web Services Eventing. Available from: http://ftpna2.bea.com/pub/downloads/WS-Eventing.pdf.
2. OASIS. WS-Notification (v1.2). Available from: http://docs.oasis-open.org/wsn/2004/06/.
3. Huang, Y. and D. Gannon, A Comparative Study of Web Services-based Event Notification Specifications. ICPP Workshop on Web Services-based Grid Applications 2006.
4. Huang, Y., et al., WS-Messenger: A Web Services based Messaging System for Service-Oriented Grid Computing. International Symposium on Cluster Computing and the Grid (CCGrid06).
5. Diao, Y., S. Rizvi, and M.J. Franklin, Towards an Internet-Scale XML Dissemination Service. Proceedings of VLDB 2004, 2004.
6. Brett, P., et al., A Shared Global Event Propagation System to Enable Next Generation Distributed Services. Workshop on Real, Large Distributed Systems (WORLDS), 2004.
7. Carzaniga, A., D.S. Rosenblum, and A.L. Wolf, Achieving scalability and expressiveness in an internet-scale event notification service. Proceeding of Nineteenth ACM Symposium on Principles of Distributed Computing (PODC 2000), 2000.
8. Clark, J. and S. DeRose. XML Path Language (XPath) Version 1.0. Available from: http://www.w3.org/TR/xpath.
9. Huang, Y. and D. Gannon, A Flexible and Efficient Approach to Reconcile Different Web Services-based Event Notification Specifications. Proceedings of International Conference on Web Services (ICWS), 2006.
10. Banavar, G., et al., An efficient multicast protocol for content-based publish-subscribe systems. Proceedings of the 19th International Conference on Distributed Computing Systems (ICDCS'99), 1999.
11. Pietzuch, P. and J. Bacon. Hermes: A Distributed Event-Based Middleware Architecture. in Workshop on Distributed Event-Based Systems (DEBS). 2002.
12. Cugola, G., E.D. Nitto, and A. Fugetta, The jedi event-based infrastructure and its application to the development of the opss wfms. IEEE Transactions on Software Engineering, 2001.
13. Altinel, M. and M.J. Franklin., Efficient Filtering of XML Documents for Selective Dissemination of Information. Proc. of VLDB 2000, 2000.
14. Diao, Y. and M.J. Franklin, High-Performance XML Filtering: An Overview of YFilter. IEEE Data Engineering Bulletin, 2003(March, 2003).
15. Barton, C.M., et al., Streaming XPath Processing with Forward and Backward Axes. Proc. of ICDE, 2003.
16. Feng Peng, S.S.C., XPath Queries on Streaming Data In Proc. of SIGMOD, 2003.
17. Chand, R. and P.A. Felber, A Scalable Protocol for Content-Based Routing in Overlay Networks. In Proc. of Symposium on Network Computing and Applications, 2003.
18. Fox, G. and S. Pallickara, NaradaBrokering: An Event-based Infrastructure for Building Scalable Durable Peer-to-Peer Grids, in Grid Computing: Making the Global Infrastructure a Reality. 2003.
19. Advanced Message Queuing Protocol Specification. Available from: http://www.iona.com/opensource/amqp/.
20. Peterson, L., et al., Experiences Building PlanetLab. Proceedings of USENIX OSDI'06, 2006.

Empowering Scientific Discovery by Distributed Data Mining on the Grid Infrastructure

Haimonti Dutta

Department of Computer Science and Electrical Engineering,
University of Maryland, Baltimore County
1000 Hilltop Circle,
Baltimore, MD 21250, USA.
hdutta1@cs.umbc.edu,
WWW: http://www.csee.umbc.edu/∼hdutta1

Abstract. The grid-based computing paradigm has attracted much attention in recent years. Computational Grids focus on methods for handling compute intensive tasks while Data Grids are geared towards data-intensive computing. This dissertation considers research in *grid-based distributed data mining*. While architectures for mining on the grid have already been proposed, the inherently distributed, heterogeneous nature of the grid, calls for *distributed* data mining. This necessitates the development of services that can perform distributed data preprocessing, mining, visualization and workflow compositions. As a proof of concept, the feasibility of executing distributed data mining algorithms on astronomy catalogs obtained from two different sky surveys Sloan Digital Sky Survey (SDSS) and The Two Micron All Sky Survey (2MASS) is explored. In particular, we examine a technique for cross-matching indices of different catalogs thereby aligning them, use a randomized distributed algorithm for principal component analysis and propose to develop an outlier detection algorithm based on a similar technique. The algorithms developed are to be implemented as services on the Data Grid. Finally, we propose an analytical and empirical framework for optimization of resources on the grid.

1 Introduction

In many scientific domains[1] large data repositories are being generated. Researchers needing to access the data may come from different communities and are often geographically distributed. The *Data Grid* [15] has emerged to provide an architecture for distributed management and storage of large scientific data sets. Several projects such as Knowledge Grid [4], Grid Miner [5] and TeraGrid [3] have focused on the creation of middleware / systems for *data mining and knowledge discovery* on top of the Data Grid. Motivated by this research, we

[1] e.g. Astronomy, High Energy Physics, Climatology, Computational Genomics, Medicine and Engineering

propose to develop service based architectures for *Distributed* Data Mining on the grid infrastructure.

The paper is organized as follows: Section 2.1 briefly summarizes the need for Distributed Data Mining (DDM), Section 2.2 discusses related work on data mining on the grid. The Section 3 summarizes Research Contributions and Section 4 introduces Preliminary Work. Finally section 5 concludes the work.

2 Related Work

2.1 Distributed Data Mining (DDM)

A primary motivation for DDM discussed in literature, is that a lot of data is inherently distributed. Merging of remote data at a central site to perform data mining will result in unnecessary communication overhead and algorithmic complexities. Simply put, DDM is data mining where the data and computation are spread over many independent sites. Typically, in a DDM environment, each site has its own data source and data mining algorithms operate on it producing local models. Each local model represents knowledge learned from the local data source, but could lack globally meaningful knowledge. Thus the sites need to communicate by message passing over a network, in order to keep track of the global information. DDM is thus not just about "distributing" the centralized data mining task on several compute nodes. The reader is referred to ([16]) for a survey in DDM.

In a data grid, the repositories may contain several different data distribution schemes. These include: (1)*Centralized Data Source:* This is one of the simplest scenarios since the data can be thought of as residing in a single relational database, a flat file, or as unstructured data (XML). (2)*Distributed Data Source:* When the data is assumed to be distributed among different sites, two different scenarios can arise. (a)*Horizontally Partitioned Data:* The horizontal partitioning ensures that each site contains exactly the same set of attributes. (b) *Vertically Partitioned Data:* The vertical partitioning requires that different attributes are observed at different sites. The following section discusses related work on data mining on the grid.

2.2 Data Mining on the Grid

Several research projects including the Knowledge Grid [4], Grid Miner [5], Discovery Net [6], TeraGrid [3], ADaM (Algorithm Development and Mining) [7] on NASA's Information Power Grid, and the DataCutter project [8] have focused on the creation of middleware / systems for mining on top of the data grid. None of these projects consider "distributed" data mining algorithms. The following section introduces the research contributions of this dissertation.

3 Research Contributions

1. Develop *distributed* data mining (DDM) algorithms implemented as services on the Grid. The application area is astronomy. This includes development of:(1) Distributed Principal Component Analysis (PCA) algorithm. (2) Distributed Classification (using decision trees)(3) Distributed Top-K Outlier Detection.
2. Provide an analytical framework for showing the trade-off between communication cost versus accuracy for DDM algorithms.

4 DDM on Federated Astronomy Catalogs

This section illustrates preliminary work done for the dissertation. In particular, it first describes a data integration problem and is followed by development of basic *distributed* data mining algorithms including Principal Component Analysis, Outlier Detection and Classification. Finally it describes usage of these algorithms as services on a Data Grid.

4.1 The Data Matching Problem

The problem of schema integration and data matching has been studied extensively in literature. This thesis does not propose to develop a novel technique for solving these problems, but uses existing solutions to solve the more complex problems related to *distributed* data mining. We illustrate the problem with two archives: the Sloan Digital Sky Survey (SDSS) [1] and the 2-Micron All-Sky Survey (2MASS) [2]. Each of these has a catalog containing records for a large number of astronomical point sources, upward of 100 million for SDSS and 470 million for 2MASS. Each record contains sky coordinates (ra,dec) identifying the sources' position in the celestial sphere as well as many other attributes (460+ for SDSS; 420+ for 2MASS). While each of these catalogs individually provides valuable data for scientific exploration, together their value increases significantly. In particular, we are interested in efficient analysis of the *virtual catalog* formed by joining these individual catalogs.

To form the virtual catalog, records in each catalog must first be matched based on their position in the celestial sphere. Consider record t from SDSS and s from 2MASS with sky coordinates $t[ra, dec]$ and $s[ra, dec]$. Each record represents a set of observations about an astronomical object *e.g. a galaxy*. The sky coordinates are used to determine if t and s match, *i.e.* are close enough that t and s represents the same astronomical object.

For each pair of surveys T and S, a distinct pair of match indices must be kept, one at each survey. The i^{th} entry in $T's$ list points to a tuple t_i and the i^{th} entry in $S's$ list points to s_i such that t_i and s_i match. Tuples in T and S which do not have a match, do not have a corresponding entry in either index. Clearly, algorithms assuming a vertically partitioned virtual table can be implemented on top of these indices.

Creating these indices is not an easy job. However, the indices do not need to be created each time a data mining task is run. Instead, each pair of indices only need be created *once*. Then any DDM task can use them. The net result is the ability to mine virtual catalogs at low communication cost.

4.2 Distributed PCA

PCA is a well-established data analysis technique used in a large number of disciplines. The main motivation for doing a PCA is to reduce the dimensionality of a data set while preserving as much variation as possible in the original dataset. To the best of our knowledge, the problem of vertically distributed PCA computation was first addressed by Kargupta *et al.* [13] based on sampling and communication of dominant eigenvectors. Later, Kargupta and Puttagunta [14] developed a technique based on random projections. Our method is a slightly revised version of this work. We describe a distributed algorithm for approximating the covariance matrix in [18].

4.3 Distributed Classification

Decision tree ensembles are frequently used in data mining and machine learning applications. Boosting, Bagging, Stacking and Random Forests are some of the well-known ensemble-learning techniques. Many of these techniques often produce large ensembles that combine the outputs of a large number of trees for producing the overall output.

In the distributed heterogeneous astronomy catalogs, consider the problem of classification of galaxies and stars across multiple sky surveys. To exploit information from different catalogs, there needs to be a mapping between an object in one catalog to an object in another (cross-match) as described earlier, thus creating a virtual catalog. If the indeces of these cross-matched records are stored at each individual site, then the computation of decision trees over all the sites can be done with very little communication. The idea is presented here: Each site computes a decision tree on its cross matched data. The model can be shipped to central site / co-ordinator site which combines the decision trees (for e.g. by a voting algorithm). Instead of sending data, communicating just the model saves considerable communication overhead. A further reduction in the communication can be achieved if the model can be compressed in some way. One interesting idea is to use the Fourier transform to represent the decision trees[2]. Our work on Orthogonal Decision Trees [19] aims to address this issue.

4.4 Distributed Outlier Detection

Most commonly the first principal components are used in data analysis. However, the last components also carry valuable information. Some techniques for

[2] Note that a decision tree can be treated as a function and it is possible to take its Fourier transform

outlier detection have been developed based on the last components [20]. These algorithms look to identify data points which deviate sharply from the "correlation structure" of the data. Since the covariance matrix and hence the principal components can be obtained in a distributed manner we use the covariance matrix obtained from the distributed computation to project the data at individual sites onto the last principal component, to identify outliers in the data. This is a simple extension of the work referred to earlier and is currently work in progress.

4.5 Analytical Framework for DDM on the Grid

Since DDM is not just "distributing" a centralized data mining algorithm, we are developing an analytical framework for illustrating the trade-off in accuracy versus communication complexity for performing DDM on the grid. The idea is to pose an optimization problem for resource management on the data grid. This is work in progress.

4.6 Service Oriented Computing for DDM on the Grid

Existing work for data mining on the grid has mostly been geared towards running known data mining algorithms over several compute nodes of a grid. While this distributes the algorithm, the intricacies involved are much more complex. For instance, issues related to data integration of vertically partitioned data, distributed resource management, distributed algorithms for data mining need to be studied in more detail. This dissertation has developed algorithms for several data mining tasks and the idea is to provide them as services on the grid. This will enable the development of completely decentralized data mining services for use in the scientific community. The services available as part of the DistributEd Mining of Astronomy Catalogs (DEMAC) system [18] are a first step in this direction.

5 Conclusion and Future Work

This dissertation is aimed at development of an infrastructure for distributed data mining on the grid. It developed algorithms for DDM including a distributed principal component analysis algorithm, distributed classification and distributed outlier detection which are to be implemented as services on the Data Grid. It further considers development of an analytical framework for optimization of resources distributed on the grid. The application area is astronomy. Possible directions for future work include development of algorithms for distributed association rule mining and other data mining tasks (in particular for astronomy) and implementing them as services and development of distributed workflow compositions for data mining tasks. It must be noted that while we have selected astronomy as an application area, the algorithms and system being developed can be used for other scientific domains such as bio-informatics, medicine etc.

References

1. The Sloan Digital Sky Survey, http://www.sdss.org.
2. The Two-Micron All Sky Survey, http://www.ipac.caltech.edu/2mass/
3. TeraGrid, http://www.teragrid.org/about/
4. Mario Cannataro and Domenico Talia. The knowledge grid. Communications of the ACM, 46(1):89–93, 2003.
5. Peter Brezany, Juergen Hofer, A Min Tjoa, Alexander Woehrer.GridMiner: An Infrastructure for Data Mining on Computational Grids. Accepted for the APAC Conference and Exhibition on Advanced Computing, Grid Applications and eResearch, Queensland Australia, 2003.
6. V. Curcin, M. Ghanem, Y. Guo, M. Kohler, A. Rowe, J. Syed, P. Wendel. Discovery Net: Towards a Grid of Knowledge Discovery. In Proc of The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD-2002, Edmonton, Alberta, Canada, July 23 - 26 2002.
7. J. Rushing, R. Ramachandran, U. Nair, S. Graves, R. Welch, and H. Lin. ADaM: a data mining toolkit for scientists and engineers. Computers and Geosciences, 31:607–618, jun 2005.
8. Michael Beynon and Renato Ferreira and Tahsin M. Kurc and Alan Sussman and Joel H. Saltz. Datacutter: Middleware for filtering very large scientific datasets on archival storage systems. In IEEE Symposium on Mass Storage Systems, pages 119–134, 2000.
9. Juergen Hofer and Peter Brezany. DIGIDT: Distributed Classifier Construction in the Grid Data Mining Framework GridMiner-Core. In Proceedings of the Workshop on Data Mining and the Grid (GM-Grid)2004.
10. http://grid.deis.unical.it/weka4ws.
11. http://smi.ucd.ie/ rinat/weka/.
12. http://users.cs.cf.ac.uk/Ali.Shaikhali/faehim/.
13. Kargupta H., Huang W., Sivakumar K., and Johnson E. Distributed clustering using collective principal component analysis. Knowledge and Information Systems Journal, 3:422–448, 2001.
14. Kargupta H. and Puttagunta V. An Efficient Randomized Algorithm for Distributed Principal Component Analysis from Heterogeneous Data. In Proceedings of the Workshop on High Performance Data Mining, 2004.
15. Ann Chervenak, Ian Foster, Carl Kesselman, Charles Salisbury, and Steven Tuecke. The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets. Journal of Network and Computer Applications, 23:187–200, 2001.
16. H. Kargupta and K. Sivakumar. Existential Pleasures of Distributed Data Mining. Data Mining: Next Generation Challenges and Future Directions.Editors: H. Kargupta, A. Joshi, K. Sivakumar, and Y. Yesha. AAAI/MIT Press, 2004.
17. Mara S. Prez, Alberto Snchez, Pilar Herrero, Vctor Robles, Jos M. Pea. Adapting the Weka Data Mining Toolkit to a Grid Based Environment. In Lecture Notes in Computer Science, volume 3528, pages 492–497, January 2005.
18. Chris Giannella, Haimonti Dutta, Ran Wolff, Kirk Borne and Hillol Kargupta. Distributed Data Mining in Astronomy Databases, The 9th Workshop on Mining Scientific and Engineering Data Sets, 2006.
19. H. Kargupta, B. Park, H. Dutta. Orthogonal Decision Trees IEEE Transactions on Knowledge and Data Engineering, volume 18, number 7, pp. 1028-1042, 2006.
20. Jolliffe I. Principal Component Analysis. Springer-Verlag, 2002.

A Study of Language-Action Perspective as a Theoretical Framework for Web Services

Karthikeyan Umapathy

College of Information Sciences and Technology
The Pennsylvania State University
323E IST Building
University Park, PA – 16802, USA
{kumapathy}@ist.psu.edu

Abstract. This dissertation contributes to the services science discipline by examining appropriateness of Language-Action Perspective (LAP) as a theoretical framework for web services, the technology component of services science. This study is conducted through three essays. The first (completed) investigates whether LAP constructs can describe and explain the web services architecture. Findings from this essay indicate that there is lack of mechanisms to generate conversation policies that guide interactions between applications. Conversation polices are crucial for developing large-scaled enterprise integration solutions using web services. The second (work-in-progress) builds on this finding. This essay demonstrates appropriateness of LAP constructs to access design knowledge to develop web services solutions for enterprise integration. The third (work-in-progress) evaluates usefulness of LAP constructs to develop effective web services solutions (artifact developed in the second essay).

Keywords: Web Services, Language-Action Perspective, Services Science, Reference Framework, Enterprise Integration, Service-Oriented Computing

1 Introduction

The current global economy is dominated by services sector. Services sector accounts for approximately two-thirds of global economic activity and 80 percentage of the United States economy [1]. The underlying phenomenon of the service economy and its implications are explored by the services science discipline [2]. In order to perform with greater efficiency in the services economy, organizations need to transform their businesses at the fundamental level, requiring changes both with the business models and the underlying technologies [3].

Service-Oriented Computing (SOC) paradigm provides critical information technology (IT) infrastructure needed for organizations to cope up with shift in the business environment [4]. The fundamental element to develop software applications in the SOC paradigm is "service" [5]. Services are autonomous, platform-independent computational entity which can be combined in particular ways to achieve business

goals [4]. Web services, is the current most promising technology based on SOC concepts.

Web services

Web services, following SOC and its architectures, provides interoperability among applications from different platforms and loosely couple disparate systems through standardized mechanisms to achieve business goals [4]. Web services supports interoperability among applications through a set of standards, each dealing with different aspects. However, rapid evolution and continued development of these standards has lead to several complexities. Complexity such as competing standards[1] confuses web services developers with regards to selection of appropriate standards for implementation. Complexity such as overlapping standards[2] is affecting the strategic nature of the web services standardization process. While for managers complexities are with decision making process since the existing set of rules and principles are inadequate and inappropriate [10]. Moreover, due to barriers created by above mentioned complexities, organizations have been slow in adopting web services [11].

Traditional development models and tools are inadequate to overcome above mentioned barriers since existing rules and principles are inappropriate for current business environment [3]. In order to overcome this drawback, theoretical frameworks for web services need to be developed. Development of appropriate theoretical framework is one of the key challenges for web services and its broader discipline - services science [12]. An anticipated benefit of theoretical frameworks would be that they will allow development of theoretically informed solutions to web services problems. Thus addressing this key challenge and possible practical contributions of a theoretical framework are the motivations for this research.

Any theoretical framework selected for web services should match with its core SOC principles, i.e., supporting communication among disparate applications to achieve business goals. Web services utilize communication to accomplish actions such as publishing, finding and binding with other applications. Therefore, any theoretical framework for web services should have communication at its core, provide understanding of interrelationships between web services functionalities, and provide mechanisms to construct web services artifacts to achieve business goals.

Language-Action Perspective (LAP)

A candidate theoretical perspective is Language-Action Perspective (LAP) [13], because LAP focuses on communication to facilitate co-ordination and interaction among information systems to achieve business goals. Premise of LAP is that much work in today's organizations is performed through language to achieve their goals [13]. LAP recognizes that language is not only used for exchanging information, (as in reports or statements) but also to perform actions (as in promises, orders, requests, declarations, etc). LAP emphasizes that such communicative actions should be the foundations for creating effective information systems.

LAP suggests that design of information systems should be based on analysis of communication among systems and people [13]. This attracted much interest in the

[1] WS-Notification [6] and WS-Eventing [7], with regards to notifying occurrence of an event
[2] WS-BPEL [8] and WS-CDL [9], with regards to coordination among services

90s and lead to wave of LAP based frameworks, models and systems building on theoretical foundations such as speech acts [14] and communicative action [15]. The LAP perspective has lead to the development of a number of methodologies and computer based tools[3] to support work practices in organizations [13].

When LAP was originally formulated, its purpose was to support human work in organizations. The foundations of LAP can, therefore, be traced to human communicative theories. Adoption of the LAP perspective in the web services domain would, therefore, require modifications that support machine to machine and humans to machine communications.

2 Research Questions

My central hypothesis is that the *Language-Action Perspective (LAP) would be an appropriate theoretical framework for web services*.

I test this hypothesis through three questions: first question attempts to investigate the relevance of LAP constructs to the web services domain, second question attempts to demonstrate the applicability of LAP constructs to solve problems in the web services domain and third question attempts to evaluate the utility of LAP constructs for solving web services problems. Following are the research questions addressed in this research:

RQ 1. *Can LAP constructs describe and explain the web services phenomenon?*

RQ 2. *How can LAP constructs help in designing web services solutions?*

RQ 3. *Are LAP constructs effective for designing web services solutions?*

Thus, the endeavor of this research is to *investigate, demonstrate and evaluate the usefulness of LAP as a theoretical framework for web services*. RQ 2 and 3 are explored and applied in enterprise integration domain. This dissertation research is conducted in a three essay approach (see Fig. 1) which is described below.

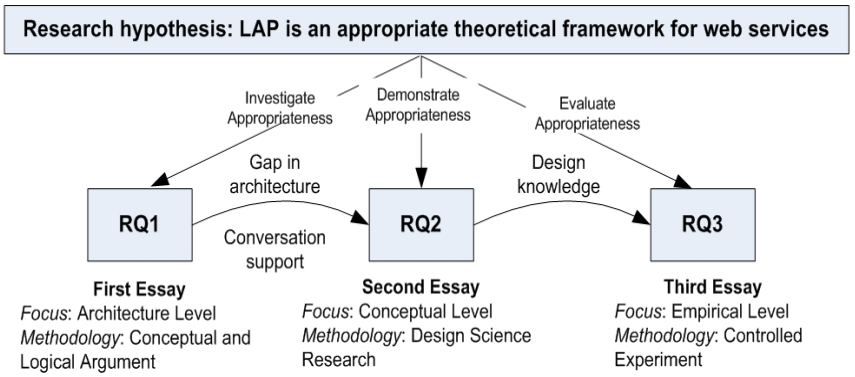


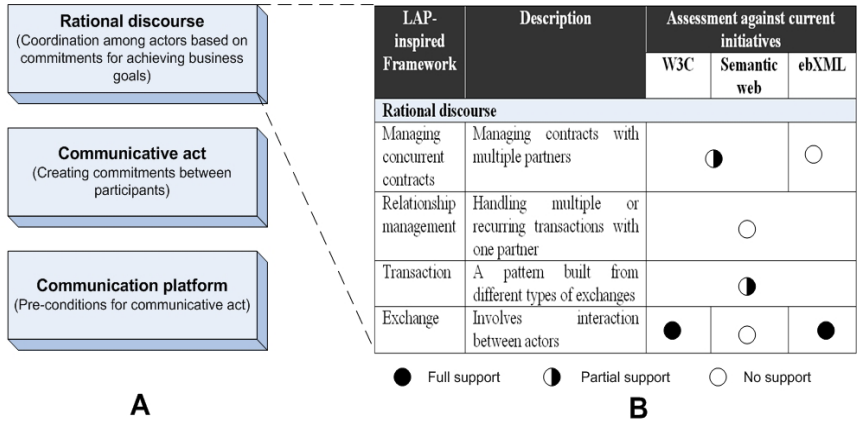
Fig. 1. An overview of the research approach

[3] See Communications of ACM (CACM), May 2006, Special Issue on LAP.

3 Research Approach

First essay – Investigating appropriateness of LAP for describing and explaining web services

The *first essay* (completed, published in Information Systems Frontiers, 2006) is conducted at the architecture level to investigate appropriateness of LAP to describe and explain web services architecture (addresses RQ1). The essence of this essay is to build a reference framework (shown in Fig. 2) based on LAP constructs which identifies specific functions that need to be addressed in the interoperable web services architecture. The purpose of the reference framework is to provide guidance to all stakeholders of web services to make sense of its framework. This essay uses conceptual development and logical argument as the research method [16]. One of the findings from the first essay indicates that there is lack of standards and mechanisms to develop conversation policies [17], which guides interactions among web services.



LAP-inspired Framework	Description	Assessment against current initiatives		
		W3C	Semantic web	ebXML
Rational discourse				
Managing concurrent contracts	Managing contracts with multiple partners	◑		○
Relationship management	Handling multiple or recurring transactions with one partner	○		
Transaction	A pattern built from different types of exchanges	◑		
Exchange	Involves interaction between actors	●	○	●

Fig. 2. An LAP-inspired framework for web services. Part A of the figure shows three major levels of the framework, while part B shows assessment of three existing web services initiatives against the LAP-inspired framework for Rational Discourse level.

Second essay – Demonstrating appropriateness of LAP for designing web service solutions

The *second essay* (work-in-progress) builds on the finding from the first essay. This essay is conducted at the conceptual level to demonstrate appropriateness of utilizing LAP constructs to develop web services solutions. The domain I use for this investigation is "enterprise integration" (addresses RQ2). This essay uses design science research [18] as the research methodology to design and develop a design tool to assist integration designer with selection of appropriate integration tactics as integration solutions and translating those tactics into conversation policies. This essay utilizes LAP constructs to develop a methodology (shown in below Fig. 3(A)) and corresponding knowledge base (based on enterprise integration patterns [19]) that would contain heuristic information on usage of integration design knowledge. The

methodology will semi-automate the process of using enterprise integration patterns (EIPs) and generate conversation policies (CPs) for web services participating in a business process (see Fig. 3).

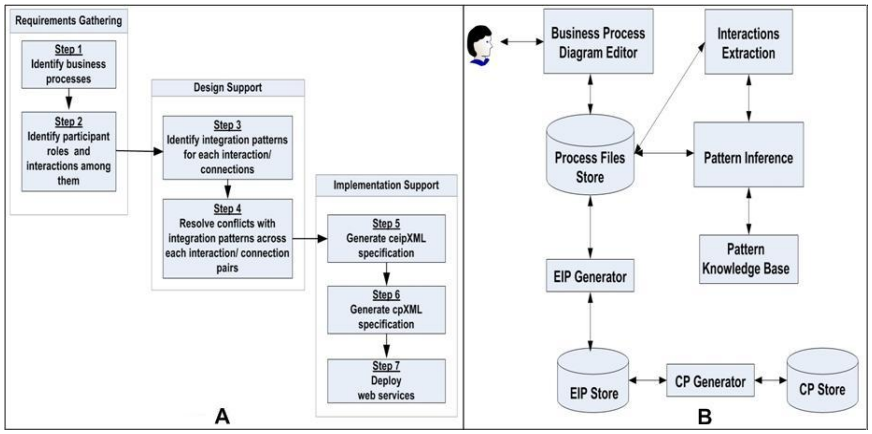


Fig. 3. A methodology (part A in the figure) and design tool (part B in the figure) to compose web service solutions for enterprise integration.

Third essay – Evaluating effectiveness of LAP for designing web service solutions

The *third essay* (work-in-progress) is conducted at the empirical level to evaluate appropriateness of LAP constructs used for developing design tool in the second essay (addresses RQ3) through a controlled experiment [20]. This evaluation would aid in justifying the theoretical constructs used as well as identifying the weaknesses in the constructs [18]. This essay will use quasi-experiment [20] with a panel of experts in enterprise integration to evaluate an artifact that implements the methodology developed in the second essay. This experiment would be conducted with two set of groups. One group will attempt to develop web services based solution for an integration problem using the design support developed based on LAP constructs; a control group will attempt to solve the same problem without LAP-based design support.

4 Anticipated Contributions and Conclusions

This dissertation argues that LAP is an appropriate theoretical framework for web services and tests appropriateness via three-essay approach. Expected contributions from this dissertation are an LAP-inspired reference framework for web services (from essay 1), a methodology and design tool for designing enterprise integration solutions (from essay 2), an evaluation of the effectiveness of LAP constructs for designing web service solutions (from essay 3). Apart from these specific contributions, the significance of this research is its applicability to the emerging services science discipline. One of key challenge in this discipline is development of appropriate theoretical framework that can answer central questions of the discipline

[2]. Web service is a critical sub-set of services science. Results from this research have the potential for extension and application in the services science discipline.

5 References

1. ABCDoha: Services – The Largest and Fastest Growing Sector in the World Economy. American Business Coalition for Doha (ABCDoha), (2006). http://www.abcdoha.org/lobby_kit/services3.pdf
2. Chesbrough, H., Spohrer, J.: A Research Manifesto for Services Science. Communications of the ACM **49** (2006) 35-40
3. IBMSummit: Services Science: A New Academic Discipline? The Architecture of On Demand Business Summit Report, IBM's T.J. Watson Research Center, New York, (2004). http://domino.research.ibm.com/comm/www_fs.nsf/images/fsr/$FILE/summit_report.pdf
4. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F., Krämer, B.J.: Service-Oriented Computing: A Research Roadmap.: Dagstuhl Seminar Proceedings: Service Oriented Computing (2005)
5. Curbera, F., Khalaf, R., Mukhi, N., Tai, S., Weerawarana, S.: The next step in Web services. Communications of the ACM **46** (2003) 29-34
6. WS-BaseNotification: Web Services Base Notification (WS-BaseNotification). OASIS, (2004). http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-03.pdf
7. WS-Eventing: Web Services Eventing (WS-Eventing). W3C, (2006). http://www.w3.org/Submission/WS-Eventing/
8. WS-BPEL: Web Services Business Process Execution Language (WS-BPEL). OASIS, (2005). http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel
9. WS-CDL: Web Services Choreography Description Language (WS-CDL). W3C, (2005). http://www.w3.org/TR/ws-cdl-10/
10. Truex, D.P., Baskerville, R., Klein, H.: Growing systems in emergent organizations. Communications of the ACM **42** (1999)
11. Senf, D.: Web Services Caution Abounds. CIO Magazine, (2005). http://www2.cio.com/analyst/report3235.html
12. SSME-IBMARC: Challenges, Frameworks and Call for Participation. Services Sciences, Management and Engineering, IBM Almaden Research Center, (2006). http://almaden.ibm.com/asr/SSME/coursematerials/noteschall.pdf
13. Weigand, H.: Two Decades of the Language-Action Perspective: Introduction. Communications of the ACM **49** (2006) 44-46
14. Searle, J.R.: Speech acts: An essay in the philosophy of language. Cambridge University, Cambridge, England (1969)
15. Habermas, J.: The Theory of Communicative Action: Reason and the Rationalization of Society, Vol. 1. Beacon Press, Boston (1984)
16. Weick, K.E.: Theory Construction as Disciplined Imagination. The Academy of Management Review **14** (1989) 516-531
17. Hanson, J.E., Nandi, P., Levine, D.W.: Conversation-enabled Web Services for Agents and e-business. International Conference on Internet Computing (IC) (2002) 791-796
18. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly **28** (2004) 75-105
19. Hohpe, G., Woolf, B.: Enterprise Integration Patterns. Addison-Wesley (2004)
20. Shadish, W.R., Cook, T.D., Campbell, D.T.: Experimental and Quasi-Experimental Designs for Generalized Causal Inference. Houghton Mifflin Company (2001)

Adaptive QoS Management for Web services using Asynchronous Distributed Searches

Xuan Thang Nguyen

Swinburne University of Technology, Faculty of Information and Communication Technologies,Melbourne VIC 3122, Australia
xnguyen@ict.swin.edu.au

Abstract. Web services technology is becoming popular for business-to-business integration. While a number of approaches have been proposed for Quality of Service (QoS) management of Web service compositions, those approaches are limited to an individual composition and management functionalities executed by a single organization. The PhD thesis presented in this paper aims at addressing these limitations. It specifies a QoS adaptive management framework for multi-related Web service compositions by employing asynchronous distributed search techniques from Artificial Intelligent field. The management framework is also designed to take into account the dynamic and untrusted nature of the Web service environment.

1 Introduction

Web service technology has emerged as a popular interoperable tool for distributed applications. It exposes the resources and applications in an existing infrastructure via a standard interface and hence makes the infrastructure more accessible, reusable, and composable. Different Web services can be combined to form a new value-added Web service which is referred as a composition or a composite Web service. Composite Web service management is the process of ensuring the satisfaction of functional and non-functional (i.e. QoS) requirements of a composite Web service during its execution. Management of a composite Web service is difficult because its managerial system must work across organizational boundaries.

The main objective for the PhD outlined in this paper is to: *develop new approaches for coordinated provision and adaptive controls of e2e (end-to-end) QoS of composite Web services during execution.* The main focuses are on new management algorithms for *multi inter-related composite Web services* and a *distributed* management mechanism in which different organizations can take part. An execution monitoring system to detect any QoS violation is not in the scope of this work.

The contributions of this PhD thesis are as follows. The first contribution is to provide a new *approach* to solve the distributed QoS management problem for multi inter-related compositions. This problem has not been well investigated in the current Web service research. We use fresh ideas from Distributed Constraint

Satisfaction (DisCSP) research and practically applies them into the Web service environment. The second contribution of the thesis is *a pair of new DisCSP algorithms* to solve the QoS management problem. The algorithms use fuzzy set to model different service provider preferences and can work in dynamic environments where compositions and constraints on QoS variables can be added or removed. The last contribution of the thesis is *a novel verification mechanism* which can check the conformance of service providers to the algorithm hence allows the algorithms to be practically used in an untrusted environment like the Web services'.

In the next section we present important related work and motivation for the QoS management problem. We describe the contributions of the thesis in Section III. Conclusions and future work are discussed in Section IV.

2 Related work and motivation

There have been a number of works focused on QoS management for Web services. In [4, 8] the author propose a QoS management framework which performs the refinement of existing Web service services through continuous monitoring of service execution. A QoS adaptation mechanism, based on Service Level Agreement, is presented in [1]. QoS (re-)composition can be considered as an important step in QoS management and can be found in [3] and [6]. In [6] a method for selecting optimal sub-providers from a list of service providers is proposed. In [3], the authors model the QoS requirements as an optimization problem and employ a special centralized CSP technique to solve it.

We argue that there are two major problems with those approaches. Firstly, they are limited to the management functionalities carried out by an individual organization and secondly they tend to rectify QoS violations of a single composition.

To elaborate more on these, let us revisit the concept of Web service compositions. A Web service composition can be considered as a *choreography* or an *orchestration* of Web services from different viewpoints. A *choreography* describes a composition from a global viewpoint of all participants (i.e. Web service providers who participate in the composition) whereas an *orchestration* has the local viewpoint of a single provider. The increasing popularity of Web service choreography as a mechanism for *multi-party* contracts opens up the possiblities for *multi-provider* compositions in which *every participant has some vested interest in a composite service and may actively take part in managing the composite service*. While collaboration between service providers for QoS management is necessary, it is not a requirement that every provider has to expose its management capability and related resources to each other. In other words, some privacy needs to be respected. Existing Web service management frameworks lack of a coordination model which allows collaboration between service providers to manage the QoS together. Instead, the QoS composition and management of the composite Web services in existing approaches rely on a single trusted QoS broker or QoS management centre. Another shortcoming of those approaches is

the lack of focus on *multi-related compositions*. We note that since each provider may be involved in more than one composition with different resource requirements, there is a relationship of QoS levels between these compositions. Existing works consider each composition separately. They neither consider nor exploit this relationship.

In our approach, we employ DisCSP algorithms as the technique for collabration between service providers to manage the QoS of related compositions while preserving the privacy of the providers' own constraints. DisCSP algorithms are search techniques to find a solution for a problem which has its variables and constraints on the variables' values distributed among a number of communicating agents. The overall idea of our management framework is that if a component service violates the requirements of its QoS levels, all providers collaborate using DisCSP algorithms to fix the violation before any replacement take place. For a motivation example to illustrate how a violation might be handled with *multi related compositions and collaboration between service providers*, we refer to Fig.1. This figure shows a scenario of five component Web services for travel information: *Mel-Transport*, *Mel-Attraction*, *Syd-Transport*, *Syd-Attraction*, and *Aus-Weather* which make up three composite online booking services with their e2e QoS shown in the right hand side of Fig.1. For the sake of clarity, we assume that response time and cost are our only considered QoS parameters. Also every composition is a sequential combination (SEQ) of its component services and hence its *e2e* response time can be computed as a sum of the component services' response time. To handle the violation, providers collaborate using our DisCSP

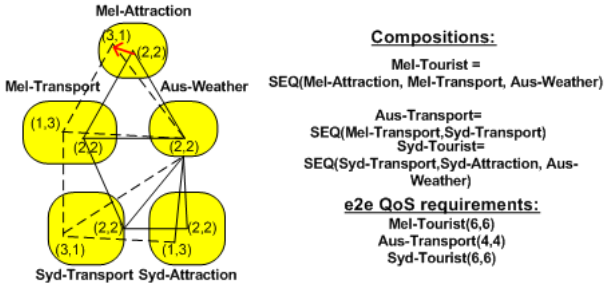


Fig. 1. An example of QoS management for multiple compositions. Initially each component service initially has the response time of 2(ms) and the cost of 2($), denoted as (2,2) in the Figure. Suppose that a violation happens to the *Mel-Tourist* service and this violation is caused by a change in the response time of the *Mel-Attraction*, which is now 3(ms). Due to this degradation, i.e from 2(ms) to 3(ms) delay, *Mel-Attraction* drops its cost (e.g. pays the penalty and hence the cost is reduced) to 1($). This violation is illustrated as a move from (2,2) into (3,1) by the Mel-Attraction.
It can be handled as shown by the shift from the continuous lines to the dotted lines.

protocol to find new values of response time and costs of the component services.

These new values supported by the service providers and satisfy QoS requirements for all the composite services. The DisCSP algorithms play an important role but are omitted here. The thesis focuses on designing such an algorithm which can work in more complex situations with uncertainties involved.

3 A QoS management framework using DisCSP

Align with the objective of the thesis described previously; the following works have been carried out:

3.1 Modelling QoS management for multi-Web service compositions as an instance of DisCSP

In our management framework, each provider has its own constraints on the supported values of its QoS parameters. Some of these constraints can be revealed, for examples through different advertised *classes of service* and others must be completely private such as resource limitations, business rules, organizational policies and service composition structures. The goal of our management mechanism is to quickly find new assignments of QoS values to all providers so that every provider's constraints are satisfied if a violation is detected. This management can be considered as a distributed search process using DisCSP algorithms. However, before any such an algorithm can be applied, we must *be able to construct/formalize constraints for QoS parameters from the e2e QoS requirements and the available resource*. To address this problem, we propose an algorithm to express composite services' e2e QoS as formulas (i.e. constraint formulas) of component services [7]. In reference to other efforts on QoS aggregation [5, 6], our constraint formulas can be considered as a mean to compute the QoS aggregations. However, our work is more general as we construct the formulas to aggregate QoS instead of compute a value. The main idea of our algorithm is to use simple composition patterns for QoS attributes [5] in combination with dynamic programming to iteratively construct the formulas for an composition with any nested levels. Our algorithm can be found at [7].

3.2 QoS management with providers' preference

Existing DisCSP algorithms are often limited to satisfactability based whereas a Web service provider may have different preference levels over different QoS values. In other words, a provider may have a different level of satisfaction on a solution. To encourage collaboration between different providers, it is desirable to maximize the global satisfaction level for all providers. We provide the Fuzzy constraint satisfaction Algorithm for Distributed Environment (FADE) algorithm is to solve this management problem in which providers can make preferences over QoS values. The main idea behinds FADE is that a Fuzzy CSP can be modeled as a collection of crisp CSPs at different levels of constraint satisfaction (i.e. α-cut) and hence FADE can solve them iteratively at these α-cut levels. This work can be found at [7].

3.3 QoS management in the dynamic Web service environment

The Web service environment is dynamic. In particular, compositions can be formed and disbanded any time. Service providers may have their own QoS constraints changed during their service lifetime. QoS requirements for a composition may also change (e.g. changes in user's requests).

To take into account the dynamic nature of the Web services environment. In the DisCSP framework, the appearance of a new composition indicates that new constraints and possibly new variables and agents (i.e. Web service providers) are added into the constraint network. Dissolving of a composition means that some existing constraints are removed and possibly some existing variables or agents are also removed. We are working on a new extension of FADE for dynamic environment. The extension is based on an indexing technique called *eliminating explanation* which had been proposed in centralized DynCSP [10]. Note that (nogood based) CSP algorithms in general generate and test solutions, and record *nogoods* (i.e. invalid solutions). The main idea of the *eliminating explanation* technique is simple enough: to index every *nogood* against the minimal set of constraints that create the *nogood*, and remove the *nogood* if a constraint in the constraint set is removed. Our extension uses this idea but for problems in distributed environments. The preliminary of this ongoing work and its supported experiments can be found at [9][1].

3.4 Verification mechanism

Our last contribution is the introduction of a novel verification mechanism, based on cryptography, for FADE. This explicit verification mechanism eliminates the impractical assumption of fully collaborative agents in DisCSP model and hence enable our DisCSP based QoS mediation to be practically used in the real Web service environment where full collaboration between providers may not be always guaranteed.

The overall idea of our verification mechanism is to employ a distributed monitoring system installed at engaged service providers. The monitoring system is to capture every incoming message of a provider A and simulate A's execution (by following FADE specification) for this input message. The simulation output of messages are used to compare against A's output to detect any inconsistency. However, instead of operating directly on the variable domains of A, the simulator operates on the encrypted values of those domains. This helps A to protect its private information from the monitoring system. Also it is important to note that the simulator does not attempt to search for any solution during its execution, but to verify the correctness of such a solution reported by A. The verification of a CSP solution in general is simpler and requires less resource as compared to a solving process. This work can be found at [2][2]. .

[1] This work will appear in the ICSOC06 conference proceedings

[2] This work will appear in the WI/IAT06 (Web Intelligence and Intelligent Agent Technologies) conference proceedings

4 Conclusions

We have discussed in this paper the limitations of current approaches in managing QoS for Web service compositions and outlined a new approach for QoS management of multi related compositions as a DisCSP problem. We also describe a new algorithm to model different service providers' preferences and its extension for dynamic environment where unexpected events and changes can happen. The thesis is six months from completion. Our future work focuses on an extension of our existing verification mechanism for this dynamic extension and a formal architectural description of our framework.

References

1. R. Al-Ali, A. Hafid, O. Rana, and D. Walker. An approach for quality of service adaptation in service-oriented grids: Research articles. *Concurr. Comput. : Pract. Exper.*, 16(5):401–412, 2004.
2. *An agent based QoS conflict mediation framework for Web services compositions.* http://ciamas.it.swin.edu.au/personal/xnguyen/paper/Verification.pdf, 2006.
3. B. Benatallah, F. Casati, and P. Traverso, editors. *Service-Oriented Computing - ICSOC 2005, Third International Conference, Amsterdam, The Netherlands, December 12-15, 2005, Proceedings*, volume 3826 of *Lecture Notes in Computer Science*. Springer, 2005.
4. Y. L. Chintan Patel, Kaustubh Supekar. A qos oriented framework for adaptive management of web service based workflows. *Lecture Notes in Computer Science*, 2736:826 – 835, Sep 2003.
5. M. C. Jaeger, G. Rojec-Goldmann, and Mühl. QoS aggregation for service composition using workflow patterns. In *Proceedings of the 8th International Enterprise Distributed Object Computing Conference (EDOC 2004)*, pages 149–159, Monterey, California, USA, 2004. IEEE CS Press.
6. Y. Liu, A. H. Ngu, and L. Z. Zeng. Qos computation and policing in dynamic web service selection. In *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 66–73, New York, NY, USA, 2004. ACM Press.
7. X. T. Nguyen, R. Kowalczyk, and M. T. Phan. Modelling and solving qos composition problem using fuzzy discsp. In *ICWS '06: Proceedings of the IEEE International Conference on Web Services*, pages 55–62, Chicago, USA, 2006. IEEE Computer Society.
8. C. Patel, K. Supekar, and Y. Lee. Provisioning resilient, adaptive web services-based workflow: A semantic modeling approach. In *ICWS '04: Proceedings of the IEEE International Conference on Web Services (ICWS'04)*, page 480, Washington, DC, USA, 2004. IEEE Computer Society.
9. *Using Dynamic Asynchronous Aggregate search for Quality guarantees of multiple Web services Compositions.* http://ciamas.it.swin.edu.au/personal/xnguyen/paper/dynamic.pdf, 2006.
10. G. Verfaillie and T. Schiex. Dynamic backtracking for dynamic constraint satisfaction problems. In *Proceedings of the ECAI'94 Workshop on Constraint Satisfaction Issues Raised by Practical Applications, Amsterdam, The Netherlands*, pages 1–8, 1994.

Towards B2B Automation Via Coalition Formation Among Service Agents

Hrishikesh J. Goradia

University of South Carolina, Columbia, SC 29208, USA,
goradia@sc.edu

Abstract. The modeling and enactment of business processes is being recognized as key to modern information management. However, current approaches are inadequate for adoption in open, dynamic environments such as the Internet. These approaches take a logically centralized view of processes instead of treating the individual business entities (realized via services) as peers. Also, the efforts are directed towards the low-level implementation issues of the composite services, rather than towards the interactions between the businesses - a higher level of abstraction appropriate for open systems. Consequently, existing approaches fail to adequately accommodate the autonomy, heterogeneity, and dynamism of the business partners in a process. Our research focus is to facilitate the enactment of Internet-based workflows by addressing the limitations in the current approaches and standardization efforts towards the coordination and composition of Web services. We focus on the coordination protocols (business protocols), which model the intractions between various (simple or composite) Web services, and propose a multiagent approach for enacting the corresponding business processes. We show that such a mechanism meets the fundamental requirements of businesses for Internet-based business collaborations.

1 Introduction

There is a fundamental shift in the way enterprises conduct their businesses today. Traditional integrated enterprises with centralized control are giving way to loosely-coupled networks of applications owned and managed by diverse business partners that interact via standard protocols. A standards-based approach helps reduce both development and maintenance costs for integrated systems. Web services simplify the interoperability problem between systems by presenting an application integration mechanism based on standard Internet protocols and languages. In the long term, Web services could become the basis for a seamless and almost completely automated infrastructure for electronic commerce and wide-area, cross-enterprise application integration [1].

Our vision for B2B automation [2, 12] is that complex projects involving multiple services to be performed by multiple enterprises will be accomplished by the formation of dynamic alliances among the best businesses available at the time of project execution without any human intervention. Individual businesses will

focus only on their core competencies, thereby reducing their costs and time-to-market, while also increasing their flexibility and market access, and improving efficiency for their customers. These businesses will rely on other businesses for handling complex projects successfully. These projects or customer requests will typically be represented as business protocols using some standard business process specification language, which identifies the activities along with the order of their execution to satisfy the business process. So, whenever a customer submits a new request, the potential business partners that can handle individual activities in the corresponding business process will coordinate their actions and form coalitions on-the-fly to handle the request. There will be no long-term commitments on the part of the business owners that come together to handle a project; these coalitions will be temporary and exist only for the duration of the project. The rewards generated out of performing the request will be mutually distributed between the coalition members.

There are many algorithmically complex issues that need to be addressed before B2B automation is feasible. The businesses are selfishly interested in maximizing their individual profits, but they have to cooperate with each other as without each other's help they would not be able to successfully perform any project, and thus generate no revenue for themselves. Under such circumstances, how do the businesses determine a project's actual worth for them? Therefore, how do they decide which projects to participate in? Businesses vary in many significant ways like the quality of service they provide, the share of the total reward that they demand, etc. How do the businesses select their partners for a particular project, such that the required quality standards for the project are met without adversely affecting their profit margins? How should the businesses distribute the revenue accrued among themselves?

We propose a multiagent approach to address the research issues mentioned above. Our approach is based on a programming model where agents represent the various businesses. These agents, by nature, are autonomous and preserve the interests of their owners during the negotiations (business interactions) with other agents (business partners). These agents selfishly try to maximize the benefits of their owners by applying their local policies and preferences while respecting the pre-defined negotiation protocols (business protocols) during agent interactions. The negotiation process is fully distributed, asynchronous, and flexible to adapt to the continuously changing business environment commonly observed in electronic commerce.

The organization of the rest of the paper is as follows. The following section presents a sampling of the related research work conducted by other researchers in this area. Section 3 provides further details about our programming model and our negotiation mechanism, comprising of a negotiation protocol and a candidate agent strategy. We also present our views on potential opportunities of incorporating our research work on multiagent systems into service-oriented computing in this section. Finally, we conclude and discuss future research directions in section 4.

2 Related Work

Business process [1] management has been an active research area for many years. Initiatives such as RosettaNet and ebXML have defined many standards that facilitate dynamic and flexible trading relationships between businesses over the Internet. RosettaNet starts with a business model that describe how business partners interact to accomplish various tasks, and re-engineers it to produce partner roles that dictate how individual partners have to interact in the form on Partner Interface Process (PIP) specifications. ebXML also allows trading partners to publish information about their business processes, including the roles they assume in the exchanges, using Collaboration Protocol Profiles (CPP). The complete business protocols in ebXML are derived by aggregating the individual partners' Collaboration Protocol Agreements (CPA). With the advent of Web services, the vision of B2B integration and automation seems increasingly close to realization. The concept of composing Web services to handle complex applications, including business-to-business (B2B) collaborations, is gaining increasingly widespread acceptance [1, 3]. Web service composition approaches based on process flows (WS-BPEL, WS-Choreography, etc.), semantic service description (OWL-S), process algebra, petri-nets, model-checking, finite-state automata, etc. have been proposed in the recent past [11, 6].

Although current approaches provide the foundation for managing cross-enterprise business processes, there are several issues that still need to be addressed. One, the development of composite Web services still largely requires time-consuming hand-coding, which entails a considerable amount of low-level programming. Since a composite service's components can be heterogeneous, distributed, and autonomous, service composition requires a high-level approach. Two, the number of services to be composed can be large and continuously changing. Consequently, approaches that require the selection and binding of component services at service definition time are inappropriate. Three, although the components that contribute to a composite service can be distributed, existing techniques usually employ a central control point. In a B2B environment, no business would allow being dictated to by others in any way. Also, businesses would want to apply their local policies and preferences while interacting. Therefore, there is a pressing need for novel mechanisms with completely decentralized execution of services, where the confidentiality, autonomy, and heterogeneity of the businesses are preserved. Four, the current standards and tools are about how a process or composed service is implemented; what we need is a declarative approach with protocols that specify *what* action should be performed, rather than *how* it should be performed. The focus should be more on the interactions between the business parties, rather than the composition per se. Five, workflow management systems today typically assume that the participants in the workflow are cooperative. However, this does not hold true for e-businesses. We need systems that are robust to manipulation by the participants. Our approach towards B2B integration addresses all these issues.

[1] We use the terms business process and workflow interchangeably in the document

3 Multiagent Solution for Dynamic Coalition Formation

Coalition formation has been studied by social scientists, economists, game-theorists, and multiagent researchers for decades [4]. While game-theorists and social scientists have provided us with various solution concepts for establishing stable coalitions (coalitions where no participant has any incentive for defecting), multiagent researchers focus on devising computationally tractable solutions for forming stable coalitions. Our research contribution to date has been the design of a negotiation mechanism for multiagent systems that allows selfish agents to pursue their agenda and maximize their profits, and yet the resultant coalition is stable and performs admirably to the empirically tested optimal solution. The negotiation mechanism comprises of a negotiation protocol, which is akin to a business protocol in service-oriented applications, and an agent strategy, which dictates how individual businesses interact in the negotiation process. We now provide some further insights into our research work.

3.1 Coalition Formation Problem in Multiagent Systems

The coalition formation process in multiagent systems involves the following steps:

1. Agents determine their values for all coalitions that they can participate in, i.e. all businesses determine their rewards from participating in the various projects with all combinations of potential partners.
2. Agents rank and select their preferred coalitions and generate the coalition structure for the system, i.e. all businesses determine their preferences for the projects and business partners, and this information is aggregated in a distributed fashion via message exchanges to determine their assignments.
3. Coalition members internally resolve the task distribution issues in the coalitions, i.e. once the business partners for a particular project are decided, the individual businesses negotiate over the roles that they will play in the business process.
4. Coalition members internally distribute the generated revenue such that the coalition is stable.

Ideally, we would want to arrive at a coalition configuration that is optimal (social welfare maximizing solution), stable (Nash equilibrium [7] or any other game-theoretic solution stability concept [4]) and fair (each agent is satisfied with its share of the reward). However, the computational complexity required for such solutions is exponential [8]. All the steps described above are interdependent, and require a combinatorial search. The fact that systems of our interest are open and dynamic add to the uncertainty and further exacerbate the problem. Therefore, considering the computational limitations, the best that we can do is use heuristics for addressing the above research issues and devise solutions that compromise on some of the ideal properties. We focus on solutions that guarantee stability because it is important for the business partners engaging

in electronic commerce to know that they cannot better their current profits in the current business environment. We present a solution [2] that has its roots in human psychology and behavioral sciences [10]. Our empirical results show that the heuristics also lead to solutions that are very close to the optimal solution.

3.2 Solution Approach

As mentioned earlier, we adopt a negotiation based approach for the dynamic coalition formation problem. Our mechanism comprises of a simple, unbiased negotiation protocol to form coalitions, and a candidate strategy for the agents that allows them to bargain over their payoff based on their negotiating power within coalitions. The negotiation protocol allows the agents to concurrently negotiate in all the different coalitions that it can potentially participate in. The choice of the coalitions to participate in is left with the agents. Our candidate stategy presents a heuristic that performs creditably in our experiments. The negotiation protocol facilitates the agents to come to a concensus, agreeing to participate in one of the coalitions which yields them the maximum profit after splitting the associated reward satisfactorily. Unlike some of the other work in this area, we do not assume a pre-decided coalition configuration [13] and/or payoff distribution [9, 5], or truthful, cooperative agents [9] in our research; our mechanism facilitates the agents to determine the best coalition and payoff for themselves via negotiations with other agents over a period of time. A heuristic for distributing the revenue among selfish agents is described in our candidate strategy for the agents. It is based on the issuance of threats and counter-threats such that agents eventually reach an equilibrium state where they cannot do any better than what they currently get. Our test results show that the strategy is stable and it also allows the agents to maximize their benefits in proportion to their negotiating power. Hence, it is reasonable to assume that the agents will use the suggested strategy in lieu of searching for others.

4 Conclusions and Future Work

In this paper, we present a novel mechanism to address the B2B integration problem in service-oriented applications by adopting a multiagent approach. We discuss the limitations of current approaches, which are mostly based on Web service composition, for application integration in open, dynamic environments, and discuss how we plan to address them in our work. We summarize some of the salient features of our solution. One, it provides the companies complete autonomy over its participation and provisioning of services for the projects. Two, it safeguards the confidentiality and vested interests of the companies as it is based on a peer-to-peer model and there is no central authority to monitor the negotiations. Three, it navigates the interactions towards a deal that all

[2] We skip most of the details due to space constraints. We have submitted a paper describing these details at the Autonomous Agents and Multi Agent Systems (AAMAS) 2007 conference.

participants prefer, if there exists one; otherwise it results in a conflict deal where nobody makes any profit - a situation that rational agents would want to avoid. Four, it allows businesses to be adaptive and take advantage of new opportunities that might arise in dynamic market environments like the Internet by allowing dynamic selection of business partners at runtime. Five, it adheres to the business protocols like PIP specifications in RosettaNet or CPPs in ebXML, thereby facilitating the migration from the existing, tried and tested solutions to the new multiagent solution with Web services. We expect to see widespread adoption of this methodology in the near future.

Our current contributions are only to the field of multiagent systems, where we have devised our negotiation mechanism for open, dynamic, fully distributed environments. Our next goal is to build a prototype system for B2B integration based on these principles. Our current solution only handles simple, sequential workflows. We would like to build on it to support more complex workflows with exception and fault handling capabilities. We hope that our research ideas would eventually lead to standards upon maturity.

References

1. Gustavo Alonso, Fabio Casati, Harumi Kuno, and Vijay Machiraju. *Web Services: Concepts, Architectures and Applications*. Springer-Verlag, 2003.
2. Hrishikesh Goradia and José M. Vidal. Multiagent workflow enactment using adaptive pricing mechanisms. In *AAAI Planning and Scheduling for Web Services Workshop*, 2005.
3. Richard Hull and Jianwen Su. Tools for Composite Web Services: A Short Overview. In *ACM SIGMOD Record*, 34(2), June 2005.
4. James P. Kahan and Amnon Rapoport. *Theories of coalition formation*. L. Erlbaum Associates, 1984.
5. Sarit Kraus, Onn Shehory, and Gilad Taase. Coalition formation with uncertain heterogeneous information. In *Proceedings of the Second International Joint Conference on Autonomous Agents and MultiAgent Systems*, pages 1–8. ACM, 2003.
6. Nikola Milanovic and Miroslaw Malek. Current solutions for Web service composition. *IEEE Internet Computing*, 8(6):51–59, November/December 2004.
7. John F. Nash. The bargaining problem. *Econometrica*, 18:155–162, 1950.
8. Tuomas Sandholm, Kate Larson, Martin Anderson, Onn Shehory, and Fernando Tohmé. Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 111(1-2):209–238, 1999.
9. Onn Shehory and Sarit Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1-2):165–200, May 1998.
10. Samuel S. Komorita. An equal excess model for coalition formation. *Behavioral Science*, 24(6):369–381, November 1979.
11. Biplav Srivastava and Jana Koehler. Web Service Composition - Current Solutions and Open Problems. In *AAAI Planning and Scheduling for Web Services Workshop*, 2003.
12. José M. Vidal, Paul Buhler, and Christian Stahl. Multiagent systems with workflows. *IEEE Internet Computing*, 8(1):76–82, January/February 2004.
13. Gilad Zlotkin and Jeffrey S. Rosenschein. Coalition, cryptography, and stability: Mechanisms for coalition formation in task oriented domains. In *National Conference on Artificial Intelligence*, pages 432–437, 1994.