UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

THE FIRST INTERNATIONAL CONFERENCE ON SERVICE ORIENTED COMPUTING

DOCTORAL CONSORTIUM PROCEEDINGS

December 2003

Technical Report # DIT-03-084

Organization

Doctoral Consortium is organized by the University of Trento (Italy) and University of Hagen (Germany) as a part of The First International Conference on Service Oriented Computing.

Consortium Chairs

Alexander Lazovik (University of Trento, ITC IRST, Italy)
Alexei Bolotin (University of Trento, Italy)
Xia Wang (University of Hagen, Germany)

Table of Contents

Challenges of Totally Distributed Scenarios

Stefano Modafferi

Politecnico di Milano,
Dipartimento di Elettronica e Informazione,
Piazza Leonardo da Vinci, 20133 Milano, Italy
modafferi@elet.polimi.it

Abstract. Nowadays the anytime/anywhere/anyone paradigm is becoming very important and new applications are being developed in many contexts; the possibility of using applications along a wide range of devices, networks, and protocols raises new problems related to delivery of services. Moreover in the mobile contexts the model and management of distributed mobile workflow is still an open field. This is the background of my research which investigates both the architecture where the services have to be delivered and the problem related to workflow management in totally mobile distributed scenarios.

1 Introduction

In the last years, the design and development of information systems have significantly changed due to new network architectures and devices, which increase the number of distribution channels available for delivering of information or services. In the anytime/anywhere/anyone paradigm [17] a novel generation of applications [9] modify themselves according to the change of context, or to specific application constraints; for example, adaptive hypermedia applications [2, 3] modify data organization according to the specific client browser capability. Very interesting is the possibility of reversing this approach indicating the possibility of modifying the distribution channel with respect to services through adaptive information systems based on reflective architectures and controllable components. It is worth noting that our distribution channel is richer than the simple network (see [14] for details). For our purpose the basic concept is the principle of reflection [10, 4] that is mainly studied in the programming language community and that consists in the system's possibility of inspecting and adapting itself by using appropriate metadata.

Strongly related with mobile challenges are the studies about workflow management. Although this field has been widely investigated in the last years [6, 7, 15, 8] no one of the models are specifically oriented to a mobile scenario and lots of problems related to connection loss and/or dynamic discovery are still open. The core of my proposal is the study of totally decentralized scenario that, under opportune conditions and hypothesis, can be very interesting and useful.

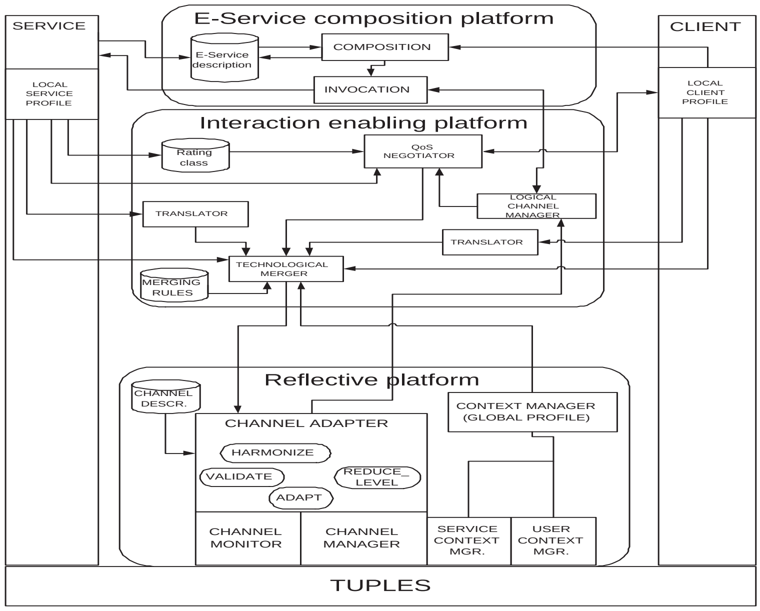


Fig. 1. General architecture

2 The Architecture

As basis for my research I want to model a reflective architecture that can support adaptive strategies at different level in a mobile scenario and that can offer events about its state useful for service delivery and workflow management. So my first work is dedicated to the description of an architecture able to support adaptive concept. In fact the middleware architecture we presented in [13] allows overcoming existing limitations of information systems by means of modification of controllable components of distribution channel, identified through their description, and according to the specific context and level of Quality of Service (QoS) requested by users. The adaptive strategies try to modify the distribution channel delivering *e*-Services by moving its current working point to a new one satisfying a given QoS. If the preferred quality level, expressed by users, cannot be satisfied then our strategies try to adapt the distribution channel to a reduced QoS level still acceptable for the user. If the reduction of QoS levels is not enough, then other ways are considered according to the users and service constraints.

The Fig. 1 shows our architecture. Three are the layers composing our architecture; they are:

– *E-Service composition platform*, which is in charge of receiving the client request, detecting the best *e*-Service(s) satisfying it, and invoking the selected *e*-Service(s).

- *Interaction enabling platform*, which is in charge of collecting constraints from *e*-Services, clients and context, determining the QoS for each *e*-Service according to the client profile and selecting the best channel along which to deliver the *e*-Service.
- *Reflective platform*, which is in charge of adapting the selected distribution channel according to the constraints obtained from the Interaction enabling platform and monitoring if the distribution channel along which an *e*-Service is delivering respects the QoS level chosen by user.

Our architecture interacts with the client, that can be a user or a software agent, and the *e*-Service. All three elements lay on the physical distribution channel and it is worth noting that the general architecture is decentralized so it is possible that all components of each layer are logically and physically distributed over a number of hosts.

The paper [13] focus its attention on the bottom layer and presents some strategies for realizing the adaptivity at technological level. Different problems are still open and the adaptivity we want to realize can be at different levels.

3 The Workflow

The management of workflow is a rich field deeply studied in the last years. Several systems manage successfully the workflow, examining closely the aspects related to specific necessity. We can cite Wide, Adept, Crossflow [7, 15, 8] or the more recent BPEL4WS [6] that are very good workflow models. Some of these models have a centralized logic and no one is specifically oriented to mobile scenario; so they do not completely support the connection loss without considering it a problem and in a mobile environment this is often true. Other models such as Exotica [1] considers the possibility of off-line work in distributed environment but there is a centralized logic.

In a mobile scenario it is necessary to consider also the device computation capability and the direction of our research is towards light solution applicable for different devices such as PDAs.

By using our architecture it is possible to receive useful information about information channel status. This information are modelled as events and we want to realize a work-flow model able to take the maximum advantage of them. The possibility of dynamic actor discovering and of run-time modification of the workflow instance are other important aspects of a mobile and distributed workflow model.

In our research we are trying to use UML as a modelling language because it is one of the best and versatile instruments for a graphical model of different situations. In [16] the authors consider the possibility of a mapping between UML and BPEL but it is a draft version. We want to study current BPEL4WS limitations trying to improve it making it functional for a totally distributed scenario.

3.1 The Open Issues

Our research is at the beginning and so lots of problems are still open.

- In a mobile scenario a coordinator has to exist? What is its role?
- How the community can establish if an actor is *transitorily* or *permanently* not connected to the net.
- How it is possible to redistribute the tasks of a dead actor?
- How to guarantee the transaction properties of tasks?

Surely many aspects of these issues have been treated in the workflow field and in the distribution database theory. Our purpose is to harmonize all these information for having a global view of totally distributed scenario.

4 A Concrete Problem

Often rich studies and models have a weak correspondence with real world. Instead we want to have a strong correspondence and so we are studying and modelling a real situation where our ideas can be validated. This example, sufficient rich, will be used as test both for architecture and workflow model. We think to Italy that probably hosts one of the most wide and significant cultural heritages in the world. Unfortunately, this abundance of cultural goods is under risk of destructions, because Italy suffers from earthquakes, has a high density of population and, like all industrialization countries, has relevant problems of pollution. All these components require the definition of an administrative and scientific instrument to manage and protect the huge cultural heritage. In 1990 the Italian government began a project to realize MARIS [5], the risk map of cultural heritage. We assume that the MARIS system can be improved by using of cooperative mobile information systems in data acquisition phase can be helped by means of a cooperative and mobile information system. In our opinion, in fact, the wide presence of goods over the Italian territory requests the use of mobile devices and networks to automate the acquisition process.

The first attempt to formalize this scenario can be found in [12].

5 Conclusion

My research wants to investigate about some interesting aspects of mobile environments. The study of an efficient reflective architecture for service delivery can give me a solid ground on which the higher level problems can lean. The study of totally distributed workflow is an open field where my ideas can be developed and my work can be productive.

Acknowledgments. My research are developing within the Italian MURST-FIRB Project MAIS (Multi-channel Adaptive Information Systems) [11].

References

1. G. Alonso, G. Gunthor, M. Kamath, D. Agrawal, A. El. Abbadi, and C. Mohan, *Exotica/fmdc: Workflow for management system for mobile and disconnected clients*, Distributed and Parallel Databases **4** (1996), 27–45.
2. G. Ammendola, A. Andreadis, and G. Giambene, *A software architecture for the provision of mobile information services*, Softcom, International Conference on Software, Telecommunications and Computer Networks (Dubrovnik (Croatia) and Ancona, Venice (Italy)), October 2002.
3. P. Brusilovky, *Adaptive hypermedia*, User Modeling and User Adapted Interaction **11** (2001), no. 1-2, 87–100.
4. L. Capra, W. Emmerich, and C. Mascolo, *Reflective middleware solutions for context-aware applications*, Lecture Notes in Computer Science **2192** (2001).
5. Istituto centrale per il restauro, *The risk map of cultural heritage*, http://www.icr.arti.beniculturali.it/rischio/CDR1INGL/index.htm, 2002.
6. S. Thatte (eds), *Business process execution language for web services*, 2003.
7. P. Grefen, B. Pernici, and G. Sanchez, *Database support for workflow management. the WIDE project*, Kluwer, 1999.
8. Cross-Organizational Workflow Support in Virtual Enterprises ESPRIT Project 28635, *Final report; crossflow deliverable*, http://www.crossflow.org/ (La Gaude), 2000.
9. J. Krogstie, *Requirement engineering for mobile information systems*, Proc. of International Workshop on Requirements Engineering: Foundation for Software Quality (Interlaken, Switzerland), 2001.
10. P. Maes, *Concepts and experiments in computational reflection*, In Proc. of Object-Oriented Programming Systems, Languages, and Applications (OOPSLA) (Orlando, Florida, USA), vol. 7, ACM Press, 1987, pp. 147–155.
11. MAIS_Consortium, *MAIS: Multichannel Adaptive Information Systems*, http://black.elet.polimi.it/mais/.
12. A. Maurino and S. Modafferi, *Challenges in the designing of cooperative mobile information systems for the risk map of italian cultural heritage*, Workshop on Multichannel and Mobile Information Systems, held in conjunction of conference on Web information Systems Engineering (Rome), 2003.
13. A. Maurino, S. Modafferi, and B. Pernici, *Reflective architectures for adaptive information systems*, Proceedings of the International Conference on Service Oriented Computing (ICSOC'03) (Trento, Italy), December 2003.
14. A. Maurino, B. Pernici, and F.A. Schreiber, *Adaptive behaviour in financial information system*, Workshop on Ubiquitous Mobile Information and Collaboration Systems (Klagenfurt/Velden, Austria), June, 2003.
15. M. Reichert and P. Dadam, *Adeptflex - supporting dynamic changes of workflow without loosing control*, 1998.
16. IBM research group, *Draft uml 1.4 profile for automated business processes with a mapping to bpel 1.0*, 2003.
17. K. Siau, E.P. Lim, and Z. Shen, *Mobile commerce: Promises, challenges, and research agenda*, Journal of Database Management **12** (2001).

An Access Control Framework for Business Processes for Web Services

Hristo Koshutanski*

Dip. di Informatica e Telecomunicazioni - Univ. di Trento
via Sommarive 14 - 38050 Povo di Trento (ITALY)
hristo@dit.unitn.it

Abstract. Web Services and Business Processes for Web Services are the new paradigms for the lightweight integration of business from different enterprises.

Security and access control policies for Web Services protocols and distributed systems are well studied and almost standardized, but there is not yet a comprehensive proposal for an access control architecture for business processes. The major difference is that business processes describe complex services that cross organizational boundaries and are provided by entities that see each other as just partners and nothing else.

This paper presents a framework for controlling access to business processes for Web Services. It shows a possible architecture; introduces "mobile" processes as a unified way for communicating requests; goes down to the level of basic Web Services and describes an approach for reasoning of missing credentials from partner's access policy when a request fails.

1 Introduction

Access control has been a constant security issue as the IT sector has been developed through time. At the end of the past millennium it became an inevitable security issue when the call for *integration of enterprise resources* took a main place in IT development. Middleware was a trendy word connected with products as CORBA, COM+, EJB that emerged at that time. Nowadays a new paradigm for the *lightweight integration of business resources of different enterprises* takes hold – Web Services and Business Processes for Web Services. Now everything is run over the Web. Web Services are network-accessible using standards as UDDI (discovery), WSDL (interface) and SOAP as a transport protocol that connects them.

The general idea of Web Services (WS for short) is to encapsulate enterprise resources and make them available for using by other enterprises. Moving up in the paradigm from single enterprises to orchestration of their business resources we find virtual enterprises to result.

* The work presented in the paper is a joint work with Fabio Massacci from the same institution as in the title.

Considering the nature of a virtual enterprise – orchestration and choreography of WS, global and local business processes, complex business transactions – the picture changes. Crossing of administrative boundaries is the main bottleneck in tailoring the available access control systems and models to WS business processes.

This calls for a number of differences with respect to "traditional" approaches for access control such as:

- credential vs. classical user-based access control;
- interactive and partner-based vs. oneserver-gathers-all requests of credentials from clients;
- controlled disclosure of information vs. all-or-nothing access control decisions;
- abducing missing credentials for fulfilling requests vs. deducing entailment of valid requests from credentials in formal models.

Looking at the access control field we find good approximation of most components [2, 8, 3, 1, 7, 4, 5] but not their synthesis into one access control architecture for business processes for web services.

2 The Framework

Combining the traditional proposals for distributed access control and the essential components used for Web services we propose here a security architecture for orchestrating authorization of Web Services Processes.

2.1 Architectural View

Fig. 1 shows a horizontal view of the architecture with multiple partners.

Following is a brief description of the entities shown in Fig. 1.

PolicyEvaluator is an entity responsible for achieving endpoint decisions on access control. All partners involved in a business process are likely to be as different entities, each of them represented by a PolicyEvaluator. The role of the PolicyEvaluator is to encapsulate the connected with it partner's specific authorization policy, and requirements with their internal representation and interpretation and presenting it as a Web service interface using WSDL.

AuthorizationServer decouples the authorization logic from the application logic. It is responsible for *locating*, *executing*, and *managing* all needed PolicyEvaluators, and returning an appropriate result to the ApplicationServer. Also it is responsible for managing all the *interactions* with the Client.

PolicyOrchestrator from the authorization point of view is an entity responsible for the workflow level access and release control. It decides which are the partners that are involved in the requested service (Web service workflow) and on the base of some orchestration security policies combines the corresponding PolicyEvaluators in a form of a Web process (*Policy Composition Process*) that is suitable for execution by the AuthorizationServer.

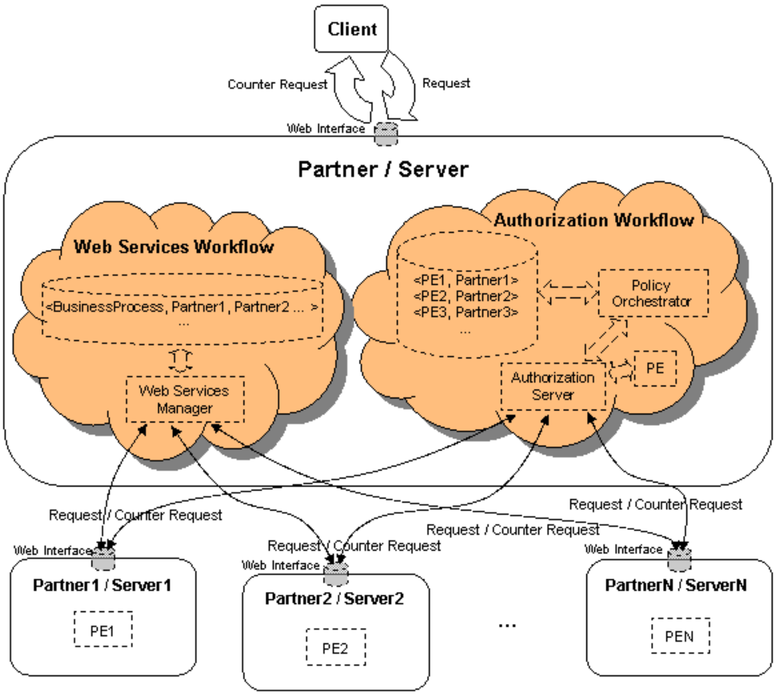


Fig. 1. Horizontal view of the architecture

To secure the entire architecture we assume authentication, confidentiality and message integrity at the transport and message levels. At the transport level, the adoption of the WS-Security specification[1] that describes enhancements to SOAP messaging to provide message integrity, confidentiality and authentication. At the message level, the W3C and IETF specification for XML-Signature[2] and W3C XML-Encryption[3], or the recently release specifications by IBM and Microsoft for WS secure conversations[4].

We free the AuthorizationServer from bothering about all the details around connections between partners and PolicyEvaluators, as well as, PolicyEvaluator's description, location, orchestration, etc. The PolicyOrchestrator is responsible for the *Policy Composition Service*: maintaining all relations between resources names (services) and policies, selecting which are the partners involved in the requested process and combining the corresponding PolicyEvaluators (as mentioned before) in a policy composition process and link them to the workflow level ac-

[1] WS-Security–www-106.ibm.com/developerworks/webservices/library/ws-secure
[2] XML-Signature–www.w3.org/TR/xmldsig-core
[3] XML-Encryption–www.w3.org/TR/xmlenc-core
[4] WS-SecureConversation–www.ibm.com/developerworks/library/ws-secon

cess and release policies. This is possible because the AuthorizationServer can just download and run a business process as we'll discuss in the next subsection.

Leading this approach at an extreme the AuthorizationServer can simply receive a business process from the orchestrator and execute it. The process may still be computationally intensive as an AuthorizationServer may have to process thousands or millions of authorization workflows, but it could be logically very simple thus reducing the TCB to the simple execution of certified processes from certified sources.

2.2 Communicating "Mobile" Processes

Assuming security at lower level, the second key component is the languages and format of communications. The typical exchange of messages in an access control system is at "data" level (credentials, policies, requests, objects, etc.) that are interpreted by the recipients. This choice makes the actual implementation of proposed access control infrastructure difficult and often not easily portable. We propose here a major innovation: exchanging messages at "source code" level and in particular at the level of business process description. It means that instead of sending just messages that have to be interpreted by entities, we truly have mobile processes passing from one entity to another indicating themselves what the recipient has to do.

The mobility of authorization processes has a number of advantages. First of all a server simply needs an off-the-shelf interpreter for business processes for a quick implementation. Second we have more flexibility for describing the process leading to an access control decision.

One of the advantages of using BPEL4WS[5] is that it is possible to implement the entire architecture using BPEL4WS. Thus we propose to use BPEL4WS itself as a language for communicating interactive requests back to a Client. This is even in the case when a Client is an AuthorizationServer waiting for a response either from a PolicyOrchestrator or from a PolicyEvaluator.

3 Abduction of Missing Credentials

For the deployment of the architecture, the PolicyEvaluator must be able to determine the set of additional credential that are necessary to obtain a service in case of failure. This problem may of course be shifted on the implementors of PolicyEvaluators, as the architecture only needs that the outcome of this derivation is mapped into some BPEL process that is then sent to the client.

Here is presented an approach based on logic that allows to reason for what is wrong if an access request fails. For the sake of simplicity let assume that the policy is expressed using Datalog rules or logic programs with the stable model semantics (if we need negation to implement some constraints like separation of duties). What we need is a logical implementation of the following process:

[5] BPEL4WS–www-106.ibm.com/developerworks/webservices/library/ws-bpel

1. the PolicyEvaluator receives the credentials and evaluates the request against the policy augmented with the credentials, i.e. whether the request is a logical consequence of the policy and the credentials;
2. if the request is granted nothing needs to be done;
3. if the request fails we evaluate the given credential against a release policy of the PolicyEvaluator to infer which are the credentials whose need can be disclosed on the basis of the credentials already received;
4. abduce the actually needed credentials by re-evaluating the request against the policy and considering the potentially disclosable credentials determined at the previous step; only the needed credential are communicated to the client.

In a nutshell, for the implementation of the PolicyEvaluator we need to implement two main inference capabilities: *deduction* and *abduction* [6]. We need to use deduction to infer whether a request can be granted on the basis of the present credentials and abduction to explain which minimum set of credentials would be necessary to grant a failed request. Here, it is not necessary to use logic, what we claim is that the underlying logical constructs, that we need for our access decisions, are these two conceptually different operations.

The Following definitions give the basic hint of the formalization.

Definition 1 (Access Control). *Let P be a datalog program (or stratified logic program) representing an access control policy, let r be an atom representing a request, let C be a set of atoms representing a set of given credentials, the* request is granted *if and only if $P \cup C \models r$.*

Definition 2 (Release Control). *Let P be a datalog program (or stratified logic program) representing a release control policy, let d be an atom representing a credential, let C be a set of atoms representing a set of given credentials, the* credential d is disclosable *if and only if $P \cup C \models d$.*

Definition 3 (Access Control Explanation). *Let P be a datalog program (or stratified logic program) representing an access control policy, let r be an atom representing a request, let C be a set of atoms representing a set of given credentials, let $D_P \supseteq C$ be a set of atoms representing disclosable credentials, an* explanation of missing credentials $C_M \subseteq D_P$ *such that*

1. $P \cup C \not\models r$
2. $P \cup C \cup C_M \models r$
3. $P \cup C \cup C_M$ *is consistent*

The first conditions says that the missing credentials are indeed needed. The second condition says that they are sufficient and the last condition says that they are actually meaningful.

4 Future and Ongoing Work

As an ongoing work is the implementation of the basic system entities AuthorizationServer and PolicyOrchestrator. Collaxa[6] is used as a main BPEL manager

[6] Collaxa BPEL Manager–www.collaxa.com

for executing (managing) interactive requests between entities in the system as well as for the implementation of the entities and their algorithms.

Also we plan to do some experimental assessments using the DLV[7] system as a background formal engine for the basic functionalities of deduction and abduction.

In a research plan we want to go in the direction of formal relation with trust negotiation in our formal model.

References

1. Beznosov, K., Deng, Y., Blakley, B., Burt, C., and Barkley, J. A resource access decision service for CORBA-based distributed systems. In *Proceedings of 15th IEEE Annual Computer Security Applications Conference. (ACSAC)* (1999), IEEE Press, pp. 310319.
2. Bonatti, P., and Samarati, P. A unified framework for regulating access and information release on the web. *Journal of Computer Security* 10, 3 (2002), 241272.
3. Godik, S., and Moses, T. eXtensible Access Control Markup Language (XACML). OASIS, February 2003. www.oasis-open.org/committees/xacml/.
4. Johnston, W., Mudumbai, S., and Thompson, M. Authorization and attribute certificates for widely distributed access control. In *Proceedings of Seventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE)* (1998), IEEE Press, pp. 340345.
5. Li, N., Mitchell, J. C., and Winsborough, W. H. Design of a role-based trust-management framework. In *Proceedings of IEEE Symposium on Security and Privacy*, 2002. S&P (2002).
6. Shanahan, M. Prediction is deduction but explanation is abduction. In *Proceedings of IJCAI 89* (1989), Morgan Kaufmann, pp. 10551060.
7. Woo, T. Y. C., and Lam, S. Designing a distributed authorization service. In *Proceedings of Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. INFOCOM* (1998), vol. 2, IEEE Press, pp. 419429.
8. Yu, T., Winslett, M., and Seamons, K. E. Supporting structured credentials and sensitive policies through interoperable strategies for automated trust negotiation. *ACM Transactions on Information and System Security (TISSEC)* 6, 1 (2003), 142.

[7] DLV System–www.dlvsystem.com

Retrieval Functions and Invocation of *e*-Service in Multichannel Information Systems

Pierluigi Plebani

Politecnico di Milano
Piazza Leonardo da Vinci, 32 - 20133 Milano, Italy
plebani@elet.polimi.it

1 Introduction

In the last years, according to the Service Oriented Architecture paradigm, several standards, proposals of standards, and methodologies are released in order to define which is a Web Service and how can I use it, and how I can compose them to create process.

Focusing on the Web Service composition, the goal is to create a process in which some of the activities are performed by Web Services. In this way once the required process activities are identified, the organization has to lookup the better Web Services which are able to perform such activities.

Differently from a classical workflow approach, this kind of processes (also called cooperative processes) involves several organizations and, according to a non-pervasive approach, the Web Service is considered a black box and thus out of requester control. Moreover the communication among the organizations could occur through different channel (e.g., Web, mail, phone).

In this context the Web Services, since they are only related to the Web, are not fully suitable to support a multi-channel information systems. For this matter the Ph.D. research proposal focuses on *e*-Service, defined as an abstraction of functionality, or a set of functionalities, exported by a system through a standard interface. Unlike the Web Service, with the *e*-Service we suppose that the functionalities it performs could be invoked by different channels not only Web based. However, even if we consider the generic *e*-Services, for their specification we reuse the vast amount of work done in the Web-Service community.

This Ph.D research proposal aims at developing an adaptive environment which supports the dynamic substitution of failed or modified e-Services used inside a cooperative process. In particular, the goal is to provide retrieval functions which are able to find an *e*-Service according to the functionalities provided, the compatibility with respect to the process and the offered quality of service.

2 *e*-Service Model

A retrieval environment necessarily requires the analysis of *e*-Services from two different standpoints, i.e. the provisioning and the request.

At higher level of detail, the provisioning perspective specifies who provides the *e*-Service, what the *e*-Service does and how to invoke its functionality, according to the offered quality of service. On the other hand the request perspective specifies who requires the *e*-Service, i.e., the actor, who wants to have a certain level of quality for the required e-Service, has a particular user profile, and operates in a particular context.

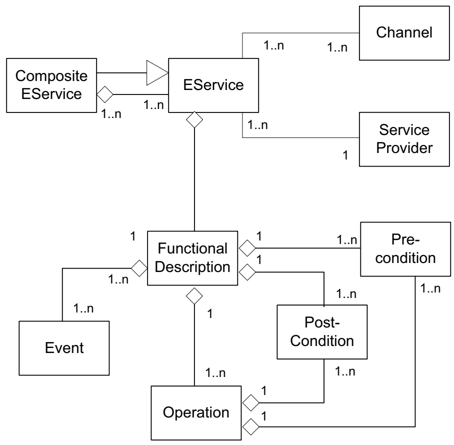


Fig. 1. Provisioning perspective

2.1 *e*-Service Provisioning

The provider is described in terms of the provided *e*-Services and available channels as shown in Figure 1. According to this description, an *e*-Service is defined by the provided functionalities. Constraints on the operation invocation can be defined in terms of pre- and post-conditions.

Moreover, *e*-Services are grouped into *CompatibilityClasses* for substitutability purposes. A compatibility class is associated with an *AbstractEService*, that is the *e*-Service required in a process execution expressed in terms of the functionality it provides. A compatibility class groups, on the basis of predefined "similarity" criteria performed through comparison between functional descriptions [1], *e*-Services that are able to substitute each other in satisfying the considered abstract service. When an *e*-Service during the execution of some tasks is not available anymore, it can be automatically substituted by another *e*-Service that belongs to the same compatibility class and that offers at least the same functionality. An *e*-Service can belong to more than one compatibility class at the same time.

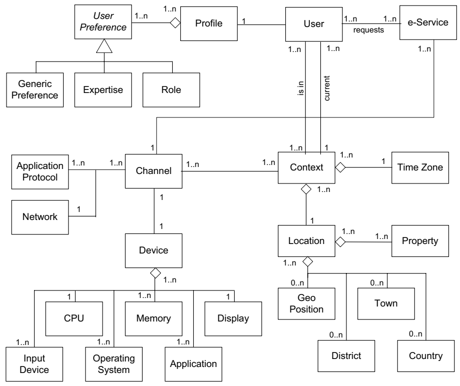


Fig. 2. Request perspective

2.2 *e*-Service Request

A user is characterized by a *context* and by a *profile* [2] (Figure 2). The context describes, among the others, the set possible available channels and the channel currently in use, in a given location and at a given time. The profile captures user preferences which depend on a role held by the user, its expertise on the service, and a set of generic preferences that enable further service-specific user profiling. Furthermore, the user could classify channels according to his preferences.

2.3 Quality

The quality model consists of (i) a *system model* defining objects and actors, and of (ii) a set of roles and rules enabling the association of quality information to objects. Quality information is expressed using *quality parameters* and *quality sets*.

System Model The proposed system model is based on the service oriented architecture (SOA, [3, 4]), which is extended with an explicit representation of channels (i.e, of networks and devices) to fit multichannel information systems. As a consequence, the SOA basic infrastructure based on service providers, service requestors, and service directories, is augmented with networks, devices, and associated actors. Figure 3 illustrates the main concept of the resulting systems model, consisting of *objects*, *actors*, *communities*, and *specifications*, described below.

Quality parameters, sets, and rules A quality parameters models a relevant and measurable non-functional aspects of an object. Quality sets are compounds of quality parameters that can be associated to objects. Quality rules define how to combine quality sets associated to distinct objects to evaluate the effects of coupling them on their non-functional aspects. In the following, we introduce the notation deemed necessary to deal with these concepts.

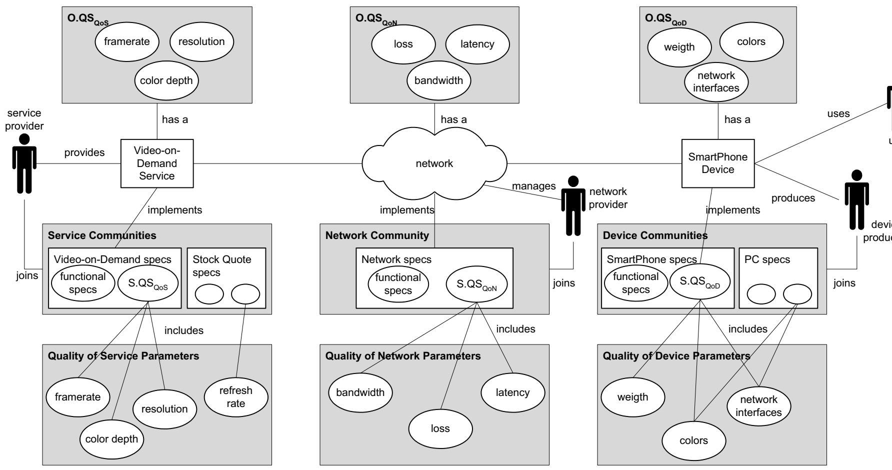


Fig. 3. Quality model

3 Related Work

At this stage several proposal can be used to support the presented environment. WSDL [5] is the de facto standard language to describe a Web Service interface in term of operation provided, the message exchanged and the protocol supported.

About the service composition research work [6–8] has recently focused on methods and tools for presenting an abstract view of internal processes to hide internal details of process execution inside organizations and for service composition [9]. Several research issues are still to be solved concerning service composition using web services [10]. BPEL4WS [11] can be used to describe the behavior of the *e*-Services and the conversation among them. DAML-S [12] can be used to create service ontology, also specifying the pre- and post-condition, whereas the UDDI Registry can be used to store references to existing *e*-Services.

Quality of service and related issues have been the topic of several research and standardization efforts crossing distinct communities during the last years. Focusing on the web-service community, [13] identifies the QoS parameters deemed useful for a service providers to characterized web-Services. In some sense, [14] integrates this proposal devising means to extend service discovery basing on QoS-related information other than on interfaces. [15] proposes a methodology enabling to evaluate the overall QoS of a composite service, i.e. a service obtained by composing several distinct services, provided that a description of the QoS parameters of the latter services is available. An interesting framework which considers the main aspects interesting the QoS definition and management for the Web-Service is provided in [16]

References

1. Antonellis, V.D., Melchiori, M., Pernici, B., Plebani, P.: A methodology for e-service substitutability in a virtual district environment. (In: Conference on Advanced Information System Engineering (CAISE 2003), Klagenfurt-Velden, Austria, June 16-20, 2003)
2. Baresi, L., Bianchini, D., Antonellis, V.D., Fugini, M.G., Pernici, B., Plebani, P.: Context-aware Composition of e-Service. (In: Technologies for E-Services : Third International Workshop, TES 2003, Berlin, German, September 7-8, 2003)
3. HP: web services concepts - a technical overview. (`www.bluestone.com/downloads/pdf/web\_services\_tech\_overview.pdf`)
4. ACM: Communications of the acm (2003)
5. Chinnici, R., Gudgin, M., J.J. Moreau, S.W.: Web Services Description Language (WSDL) 1.2. `www.w3.org/TR/2003/WD-wsdl12-20030124` (2003)
6. Casati, F., Georgakopoulos, D., Shan, M.C., eds.: Technologies for E-Services, Second International Workshop, TES 2001, Rome, Italy, September 14-15, 2001, Proceedings. Volume 2193 of Lecture Notes in Computer Science., Springer (2001)
7. Buchmann, A.P., Casati, F., Fiege, L., Hsu, M.C., Shan, M.C., eds.: Technologies for E-Services, Third International Workshop, TES 2002, Hong Kong, China, August 23-24, 2002, Proceedings. Volume 2444 of Lecture Notes in Computer Science., Springer (2002)
8. Bussler, C., Hull, R., McIlraith, S.A., Orlowska, M.E., Pernici, B., Yang, J., eds.: Web Services, E-Business, and the Semantic Web, CAiSE 2002 International Workshop, WES 2002, Toronto, Canada, May 27-28, 2002, Revised Papers. Volume 2512 of Lecture Notes in Computer Science., Springer (2002)
9. Yang, J., Heuvel, W., Papazoglou, M.: Tackling the Challenges of Service Composition in e-Marketplaces. In: 12th International Workshop on Research Issues on Data Engineering (RIDE-2EC 2002), San Jose, CA, USA, February 25-26, 2002, Proocedings. (2002)
10. Hansen, M., Madnick, S., Siegel, M.: Process Aggregation using Web Services. In: Technologies for E-Services, Third International Workshop, TES 2002, Hong Kong, China, August 23-24, 2002, Proceedings. (2002)
11. Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I., Weerawarana, S.: Business Process Execution Language for Web Services, version 1.1. `www.ibm.com/developerworks/library/ws-bpel/` (2003)
12. Service Coalition, D.: DAML-S: Semantic Markup For Web Services. `http://www.daml.org/services/daml-s/0.7/daml-s.html` (2002)
13. Mani, A., Magarajan, A.: Understanding quality of service of your Web services. IBM Developer Works (2002) `www.ibm.com/developerworks/library/ws-quality.html`.
14. Ran: A model for web services discovery with qos. ACM SIGecom Exchanges **4** (2003) 1–10
15. Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., Sheng, Q.Z.: Quality driven web services composition. In: Proceedings of the twelfth international conference on World Wide Web, ACM Press (2003) 411–421
16. Keller, A., Ludwig, H.: The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. Technical Report RC22456(W0205-171), IBM Research Division, T.J. Watson Research Center (2002)

Dynamic Matching of Services by Negotiation

Zlatko Zlatev

Department of Computer Science, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands
{zlatko}@cs.utwente.nl

Abstract. This paper presents my research project on the topic of dynamic matching of services by negotiation. The goal of the research is to develop a design theory for information systems that offer negotiation services. The result of the research will be a specification of a design framework for such systems. This framework will contain knowledge about architectural choices one has to make when designing an information system with particular properties satisfying particular requirements.

1 Introduction

This paper presents a research project in the domain of dynamic matching of services. In the next section (section 2), we narrow down the scope of the project by giving context to the project. In section 3, we position our research in the current state of the field. Section 4 presents our research goal. In section 5, we present the research framework we intend to follow to achieve our objective. Sections 6 and 7 present our research questions and research approach respectively.

2 Research Context

Businesses have cooperated via computer networks since the early 1980s. These forms of cooperation were very static and took place in the form of electronic data interchange (EDI). Since the opening of the Internet for commercial use, more dynamic forms of cooperation are facilitated. Businesses realized that the Internet offers more advantages than proprietary networks. The low costs of entry caused a bigger number of prospective partners, which in turn caused an explosion of businesses adopting Internet solutions. This new business environment is widely referred to as an e-Business environment [1] . Businesses naturally established markets as coordination mechanisms for their transactions. Through markets, they match supply and demand for products, including pure information services. Suppliers and consumers find each other through matching on price and, possibly, quality attributes. One of the strongest advantages of the Information and Communication Technology (ICT) is the enabling of a very great number of diverse businesses to cooperate in an e-business environment. At the same time, this advantage introduces new kinds of problems. One of the problems is that businesses meet difficulties finding the most appropriate partners.

The problem is caused by the great number of actors and the lack of common semantics for expressing needs. From the neo-classical economics literature about markets [8], [10], [11], we know that the execution of a business transaction in a market mechanism is connected with transaction costs [2], [10], [11]. The number of participants, the variety of offers, and the variety of means of expression increase the transaction costs in an e-market. In a traditional business environment, without extensive use of ICT, business actors reduce their transaction cost using intermediation services [8]. The costs of matching supply and demand are reduced by various intermediation services. Contrary to the early predictions that ICT will cause a shift to markets without intermediaries [5], intermediaries managed to (re)enter the e-business environment [3], [4], [6], [7]. Negotiation is one of the services offered to help reduce the costs of matching services. To our knowledge, the current state of practice offers only auctions and one-to-one negotiation. We believe that the role of negotiation in dynamic matching of services is neglected and proper support will be beneficial.

3 Current State of Negotiation Support

We made a survey of the negotiation domain [12]. The result shows that negotiation mechanisms are mainly implemented as various types of auctions. We concluded that the current state of the field is at level of accumulated 'critical mass' of implementations. We have the experience of the early adopters and their best practices. We believe that the field is mature for a design theory on intermediation technology and in particular on negotiation support. Concurrent to our research is the ebXML initiative. We differ from ebXML in the scope of the problems discussed and the levels of details. EbXML is broader in a sense of topic discussed; negotiation is only a small part of Collaboration Protocol Agreement (CPA) process. We want to research negotiation not only with respect to what protocol to use for negotiation, but also to answer the question how we negotiate and with whom.

4 Research Goal

We position our research at the intersection of the e-business and information systems (IS) domains. As far as the e-business domain is too broad, we focus on e-markets that use intermediaries to facilitate business transactions. We are interested in the role of negotiation in the process of matching supply and demand of services. From an IS perspective, we want to investigate the properties of systems that offer negotiation services. Moreover, we want to research the relation between requirements coming from the e-business domain and the properties of an IS.

4.1 Research Objective

The goal of our research is to develop a design theory for information systems that support negotiation. We want to increase our knowledge about the architectural choices one has to make when designing an IS that supports negotiation. The product of our research will contribute to that knowledge. Therefore our research objective is to specify a design framework for information systems that support negotiation.

4.2 Problem and Contribution

The source of the problem we want to address is (see section Context) the transaction costs in e-markets. These transaction costs exist not because the technology is not perfect enough, but because they are beneficial to either consumers or suppliers. For example, suppliers would like to know everything about their consumers in order to better market their products. This may include private information. Consumers will naturally impose transaction costs to prevent this. The reverse example is also true. Consumers would like to know everything about the product they want to buy. This will result in perfect competition for suppliers and they will not be able to make any profit. Our research does not aim to advance technology in a way to remove these transaction costs. Our contribution is to increase the knowledge we have about ways to reduce these costs. We strive to increase our understanding about the relation between a particular transaction cost and particular properties of negotiation support of an intermediary.

4.3 Research Framework

The specification of our research framework follows the notation proposed by P. Verschuren and H. Doorewaard in their book 'Designing a Research Project' [9]. The notation consists of rectangles and vertical and horizontal arrows. A rectangle depicts our knowledge about a phenomenon. The vertical arrow is bidirectional and connects two rectangles. The meaning of the vertical arrow is 'analyzing by confronting'. The horizontal arrow is unidirectional. It begins at a vertical arrow and ends pointing at a rectangle. The meaning of this arrow is 'derive'. Using these primitives, we can express the stages in a research process. Moreover, we can show the process of derivation of new knowledge out of existing one. In figure 1, we denote the stages in the research with Latin letters. The steps in the research we numerate with Arabic digits.

Figure 1 presents our research framework. We have four stages *a*, *b*, *c* and *d*. Stage *a* presents the current state of knowledge. Taking steps 1 and 2, namely deriving new knowledge by confronting existing knowledge, we move to stage *b*. In step 3, we repeat the same, arriving at stage *c*. Confronting again the knowledge from step 3 in step 4, we arrive at the research product (stage *d*).

We will walk through the framework starting from the end. The objective of our research is in the rightmost rectangle. Our objective is *to specify a design framework*. We identify as the main object of our research the criteria upon which we will evaluate a particular IS. These criteria are the knowledge we want to provide and towards these criteria we have to form our research perspective. We acquire that knowledge as results of the analysis of experiments with prototypes. We build these prototypes based on the knowledge we obtained in stage *b*. There, we have a certain set of hypotheses about architectures and a list of requirements. In step three, we make 'clever' guesses about the correspondence between requirements and the most appropriate architectures. We validate our suppositions with the prototypes built. The hypotheses we make about systems architecture are based on the current state of the web services and software architecture fields. This is our knowledge baseline. The upper rectangle from stage *b* depicts the requirement that we want our IS to satisfy. We derive these requirements from the economics field. We focus on Game theory, Intermediation theory, and Transaction cost theory: all three part of microeconomics theory. In the next section, we discuss the research question we answer to get the knowledge we require.

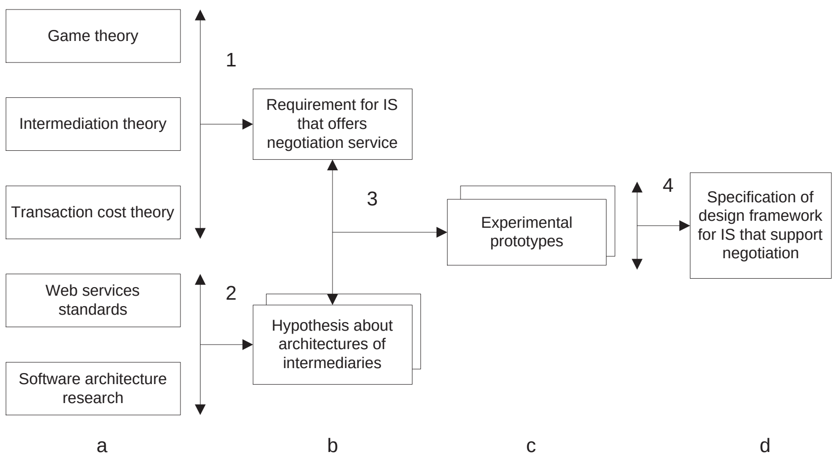


Fig. 1. Research framework.

5 Research Questions

Our research questions are the 'steering wheels' of the arrows from our research framework (see figure 1). We associate one main research question with every horizontal arrow. The answers of these questions advance our knowledge from one stage to another throughout the framework. We divide every main research question into several sub-questions. We will use the numbers of the steps from figure 1 to show how research questions correspond to transitions.

1. What are the business actors' requirements with respect to matching of services by negotiation?
   – What are the transaction costs relevant to negotiation?
   – What are the relevant efficiency criteria?

2. What are the properties of an IS that offer negotiation?
   – What are the existing architectures that support negotiation?
   – What are the supported negotiation services?
   – What are the most important services for intermediaries? Is there a classification of priority among services?
3. What are the most relevant architectural properties?
   – What are the criteria to validate a hypothesis?
   – What is the effect of different combination of services with respect to architectural properties?
4. What are the design choices?
   – What are the criteria upon which one can make an architectural choice?
   – What are the patterns of negotiation at architectural level?
   – How can we measure the quality attributes of architectures?

6 Research Approach

Our research approach is based on case study research, where the cases are experimental prototypes. Our prototypes are designed such that our hypotheses can be validated and that we can generalize the results obtained from studying the prototypes.

References

1. Amor, D., 1999, *The E-business (R)evolution*, Prentice Hall
2. Coase, R.H., 1937, 'The nature of the firm', *Economica*, new series, pp. 386-405, November
3. Hagel III, J. & Rayport, J. F., 2000, 'The New Infomediaries', *The McKinsey Quarterly*, No. 3, [online] available at: http://www.wiredcottages.com/e-commerce/newinfomediaries.htm
4. Kaplan, S. & Sawhney, M., 2000, 'E-Hubs: The New B2B Marketplaces', *Harvard Business Review*, May-June
5. Malone, T.W., Yates, J., Benjamin, R.I., 1987, 'Electronic Markets and Electronic Hierarchies', *Communications of the ACM*, vol. 30, June, p. 484-497
6. Meck, A., 2001, 'Shopbots, Powershopping, Powersales: New Forms of Intermediation in E-Commerce - An Overview', *Discussion Paper Series 203*, Universitaet Augsburg, Institute for Economics
7. Sarkar, M., Butler, B., Steinfield, C., 1995, 'Intermediaries and cybermediaries: a continuing role for mediating players in the electronic marketplace', *Journal of Computer-Mediated Communication* 1 (3)
8. Spulber, D. F., 1999, *Market Microstructure: Intermediaries and the Theory of the Firm*, Cambridge University Press
9. Verschuren, P. & Doorewaard, H., 1999, *Designing a Research Project*, Utrecht, The Netherlands, Uitgeverij Lemma
10. Williamson, O.E., 1975, *Markets and Hierarchies: Analysis and Antitrust Implications*, The Free Press
11. Williamson, O.E., 1985, *The Economic Institutions of Capitalism*, The Free Press
12. Zlatev, Z., van Eck, P., 2003, 'An Investigation of the Negotiation Domain for Electronic Commerce Information Systems, *Proceedings of the 5th International Conference on Enterprise Information Systems*, Anger, France, vol. 4, pp 386-391

Business Transactions and Conversations in Web Services Environment

Alexei Bolotin

Department of Information and Communication Technology
University of Trento
via Sommarive 14, I-38050 Trento, Italy
bolotin@dit.unitn.it

Abstract. Application integration across companies and networks plays an important role in current loosely coupled and distributed business applications. Current standards lack possibility to describe real business semantics. The present paper focuses on the requirements for Web services orchestration and execution in a stateful manner. An overview of existing specifications is also given.

1 Introduction

The complexity of current business applications grows up very fast. Without a common set of standards, each organization interested in carrying out a business in a network can develop its own proprietary protocol. This would leave little flexibility for true collaboration with other enterprises. Web services are an emerging technology in the integration of existing applications across networks and companies [2, 6].

However, with the current Web services standards such as SOAP, WSDL, and UDDI there is no way to define a business semantics of an application. Hence, today Web services are isolated and opaque [9]. This means that Web services cannot be collected together to realize more complex functionality and there is no possibility to place constraints on how operations of a collection can be used.

One of the main Web services goals is to realize a seamless interoperability among loosely coupled Web based components. Unfortunately, current business process integration models lack this. Web services must take into consideration representation and reasoning ideas from Semantic Web to achieve the real seamless interoperability [10].

The rest of this article is divided as follows. In section 2 we describe an appropriate architecture for Web services applications and the requirements that should be satisfied by new standards for Web services to enrich them with collaboration interactions and transactional behavior. In section 3 we describe some existing specifications that are candidates to be standards for Web services orchestration and for business transactions. In section 4 we take a look at the semantic problem of Web services. And, finally, we conclude with future works.

2 Architecture and Requirements

In this section we consider an architecture that provides a great flexibility for development and deployment of Web-based applications. After this we discuss some requirements that should be placed on an application for designing business collaboration of multiple applications.

2.1 Web-based Application Architecture

A workflow-based application approach is the most appropriate in a distributed heterogeneous environment [8]. It defines two strictly separated layers of application. The bottom layer represents a set of flow independent application logic blocks. The top layer is about business process, namely control and data flows inside the application.

To enable communication between multiple Web-based applications a third layer on top of the business process is introduced [6, 7, 9]. This is a conversation layer that maintains a conversation state and brokers incoming messages according to this state.

2.2 Application Requirements

There is a number of requirements that should be addressed while designing a business collaborative web-based application.

Message centric exchange. Interactions among applications should be based on message exchange contrary to direct function invocation. An advantage of this approach is that it describes the true control boundaries of the components. In this approach a service should expose only the possibility to receive a message [6].

Asynchronous interaction. As Web services interactions are inherently peer-to-peer they must support asynchronous messaging [6]. This allows a business process to invoke Web services concurrently rather than sequentially to enhance the performance [12]. Asynchronous messaging is significant in achieving the reliability, scalability, and adaptability of an application.

Conversation centric messaging. Messages in a conversation are sent within an explicit context that is maintained for the whole duration of the conversation [6]. Each message is interpreted in relation to the previously delivered messages.

Long running interaction. Most interactions between Web services are long in duration. Therefore, exception handling and transactional integrity are specially relevant [2, 12]. The system must describe its behavior in the case of an error or service timeout. If something goes wrong the system should also manage transactional integrity. However, traditional ACID-based models are not sufficient for long-running activities.

3 Existing Specifications

In this section we briefly describe major specifications that were proposed to allow Web services interoperability.

3.1 Web Services Orchestration

Web services orchestration specifications provide an open, standard-based approach for connecting Web services together to create higher-level business processes [12]. Two major specifications are BPEL4WS [5] (BPEL further) from IBM, Microsoft, and BEA and WSCI [1] from Sun, SAP, BEA, and Intalio.

BPEL A business process specifies the potential execution order of operations, the data shared between business partners, exception handling on the collection of Web services, transaction operations and other issues involving how multiple services participate into the business [9].

BPEL is an XML-based grammar to describe business process flow using WSDL operations on port type level. This is an abstract level that allows reusability of a description in different deployment scenarios.

The main focus of BPEL is an executable business process from the perspective of one of the participants ("inside-out" perspective) [12].

WSCI WSCI defines an XML-based language to describe the overall choreography of Web services messages exchange in a collaborative manner [12]. It has two levels choreography. The first level defines interfaces for one-side message flow for a single Web service. Its second level defines a global model as the composition of multiple first-level interfaces into a collaborative process.

WSCI supports message correlation, sequencing rules, exception handling, transactions and dynamic collaboration. It deals with the public message exchange between Web services in a collaborative manner.

3.2 Transactions for Web Services

There are two main specifications that provide coordination and transaction processing mechanisms for Web services. They are the pair WS-Coordination [4] and WS-Transaction [3] from IBM, Microsoft, and BEA and BTP [11] from OASIS Committee.

WS-Coordination and WS-Transaction WS-Coordination defines a framework for activities coordination using coordination protocols in a distributed and possibly heterogeneous environment. It creates a coordination context and passes it between participants. WS-Coordination is not restricted to transaction processing systems only.

WS-Transaction extends WS-Coordination and defines two distinct coordination types: Atomic transaction and Business activity. Each type specifies a set of protocols for transaction behavior. Atomic transaction deals with activities short in duration and within one trust domain. Business activity is used to coordinate cross-domain long-running activities involving business logic to handle exceptions.

BTP BTP provides a mechanism to coordinate application work between multiple participants in a distributed and heterogeneous environment using two-phase commit coordination protocol to ensure consistency of the overall outcome. BTP distinguishes two types of transactions: Atom for all-or-nothing behavior and Cohesion that allows a chosen subset of participants to confirm.

4 Semantic Web

Current Web services description is not declarative and is not encoded in a manner that facilitates symbolic manipulation [10].

Current Web services standards fully rely on XML description. XML provides a platform independent way of content definition, but it lacks the possibility to define complex relationships between services and its semantics is underspecified.

The Semantic Web community has made some important efforts in developing languages that are computer interpretable. One is RDF that provides a language for ontology definition. It enables to represent classes, properties, value domains and ranges, and hierarchies. Recent investigations introduced OWL that is more expressive. It includes a well-defined semantics and the ability to define complex relationships between properties of objects in the ontology. OWL also provides a possibility for automated reasoning about services.

5 Conclusions and Future Works

A conversational model for Web services interactions is an important area for research. It provides a more loosely-coupled and peer-to-peer environment.

Current specifications as BPEL and WSCI still take first person views on processes and their behavior is pre-defined. The interactions should be realized in a more flexible and dynamic fashion. To our knowledge the only work in this direction is Conversation Support for Web Services from IBM [6, 7].

Both BPEL and WSCI introduce a conversation notion along with process definition. However, WSCI focuses mainly on choreography aspects of message exchange, and BPEL uses the same formalism for conversation as for composition. These approaches lack some useful properties as activation, compensation and locking [2].

BPEL and WSCI are two major specifications for business process and conversation definition. They have similar goals to enable rich interactions for Web services but use different approaches. BPEL is more about executable process definition composed from multiple Web services, while WSCI is more about message choreography for an individual Web service. However, there is a notion for conversation in BPEL and there is a process definition in WSCI. It is necessary to make a distinction between conversation and process and merge these two specification into a single one or make a clear separation between the two specifications.

In designing distributed loosely coupled applications, transaction processing is a crucial point. Two specifications were proposed to address this question: WS-Coordination plus WS-Transaction and BTP. They both go beyond traditional ACID properties to satisfy the coordination needs of loosely coupled and long running activities. Both these specifications distinguish atomic and complex business patterns, but they use different approaches for their implementations. There is the necessity here to clarify expected properties and needs to develop a single standard.

All current specifications for Web services lack clear semantic descriptions. The specifications that are about service composition rely on strict interface declarations and a priori knowledge of participant functionality. More powerful description of interfaces with semantics and techniques for reasoning about validity is necessary to allow automatic discovery, composition and conversation.

References

1. A. Arkin and et. al. *Web Service Choreography Interface (WSCI) 1.0*, 2002.
2. B. Benatallah, F. Casati, F. Toumani, and R. Hamadi. Conceptual modeling of web service conversations. *Proceedings of the 15th Conference On Advanced Information Systems Engineering (CAiSE'03)*, 2003.
3. F. Cabrera and et. al. *Web Services Transaction (WS-Transaction)*, 2002.
4. F. Cabrera and et. al. *Web Services Coordination (WS-Coordination)*, 2003.
5. F. Curbera and et. al. *Business Process Execution Language for Web Services, Version 1.0*, 2002.
6. J. Hanson, P. Nandi, and S. Kumaran. Conversation support for business process integration. *Proceedings of the 6th IEEE International Enterprise Distributed Object Computing Conference (EDOC-2002)*, pages 65–74, 2002.
7. S. Kumaran and P. Nandi. Conversational support for web services: The next stage of web services abstraction. *IBM developerWorks*, 2002.
8. F. Leymann and D. Roller. Workflow-based applications. *IBM System Journal*, 36(1), 1997.
9. F. Leymann and D. Roller. Business processes in a web services world. *IBM developerWorks*, 2002.
10. D. Mandell and S. McIlraith. Adapting BPEL4WS for the semantic web: The bottom-up approach to web service interoperation. *Proceedings of the Second International Semantic Web Conference (ISWC2003)*, 2003.
11. OASIS Committee. *Business Transaction Protocol*, 2002.
12. C. Peltz. Web services orchestration: a review of emerging technologies, tools, and standards. Hewlett Packard Co., 2003.

Semantic-Based Discovery of Web Services also Improved by P2P Infrastructure

Xia Wang

FernUniversity of Hagen, Lehrgebiet Datenverarbeitungstechnik,
Fachbereich Elektrotechnik und Informationstechnik,
58084 Hagen, Germany
xia.wang@fernuni-hagen.de

Abstract. Services-oriented computing is increasing popularity, so the efficient discovery of relevant web services is being a significant challenge. This doctor proposal focuses on this field, do much work on study, compare and summary the technologies related to the challenge, such as semantic web, ontology-based web, DAML-S and p2p infrastructure [10]. And based on the result works of these fields this proposal tried to present a potent approach to discovery services based on semantic and also improved by p2p infrastructure.

1 Introduction

Web services technologies is an imposing way to interoperate the web-based distributed applications, Which is based on WSDL and UDDI registries to describe, publish and discover services. However their discovery mechanisms is not so efficient, as they do not support discovery based on the capability of the services and thus lead to a lot of irrelevant matches, or even fail to find the corresponding services. One reason is that they missed the support of semantics. As [1] said the location of web services is inherently a semantic problem, because it has to abstract from the superficial differences between representations of the services provided, and the services requested to recognize semantic similarities between the two. To settle this challenge, recently there are many hot researches doing in semantic-based web, ontology-based web, any kinds of web services description languages such as OWL, DAML-S and RDF, and also towards p2p-enabled semantic web. Each of them intends to describe services in a more precise way based on semantic, in which the semantic could be interpreted correctly and interoperated by services requesters and providers, and to precisely automatically discover matched services. Nevertheless in all these fields there are still many issues to be considered.

My doctor proposal also interests in how to efficiently describe and discovery web services, which was based on the previous works within semantic, ontology, DAML-S and so on. My contribution is to advance the research in the same direction. This proposal mainly discusses my current doctor research, to present the view of related fields and challenges, and my way to settle them in the future research.

2 Related Researches and Challenges

2.1 Semantic Web and DAML-S Ontology

The Semantic Web [2] describes the next generation of the web, which does not only provide information as text and graphics understandable to the human reader, but also gives a semantic description interpretable by machines. Semantic web is rapidly becoming a reality through the development of Semantic Web markup languages such as DAML+OIL [3]. These markup languages enable the creation of arbitrary domain ontologies that support the unambiguous description of Web content. DAML-S [4] is a DAML+OIL ontology to describe the properties and capabilities of Web Services, making web services computer-interpretable and enable the tasks, such as discovery specific services, invocation or execution of an identified service, composition of new services through automatic selection, composition and interoperation of existing services. Base on DAML-S specification we could build web services ontologies for specific domain which could be shared by the same kind of applications. The details mechanism of defining ontology is to see [4]. While there is an interesting situation, there are much work has been done on how to define the ontology, what is the correct ontology schema and how to use ontology in the applications, on the assumption that we have well built specific ontology repository, so the basal factor is neglected, that is we are missing the repository, who and how to fill the concrete ontology repository? It is impossible to build them manually by specialist group from the beginning. There is a great challenge, that is how to automatically create ontology repository for a given filed, we need a tool to automatically retrieve and create the ontologies and their relationships from the existing applications.

2.2 Adding The Semantic to WSDL and UDDI

From the content view, my approach to settle the challenge of discovery of web services is to expand the original web services technologies just by adding into semantic information, where the Web services are annotated based on shared ontologies, and use these annotations for semantics-based discovery of relevant services. There are some similar ideas, [6] is the latest draft release of DAML-S, which discusses more further in how to map between DAML-S (figure 1) and WSDL and grounding DAML-S services with WSDL and SOAP; [5] involves adding semantics to WSDL using DAML+OIL ontologies and using UDDI to store these semantic annotations and search for Web services based on them. My approach will consider integrating the two above ways, which involves mapping concepts in WSDL to DAML+OIL ontologies in service description and then providing an interface to UDDI that allows querying based on ontological concepts, the technical details is also to see [5].

2.3 Expanding UDDI Registries by a Hybrid P2P Infrastructure

From the architecture view, my approach proposes a hybrid peer-to-peer topology to expand the web services architecture to present a flexible and effective

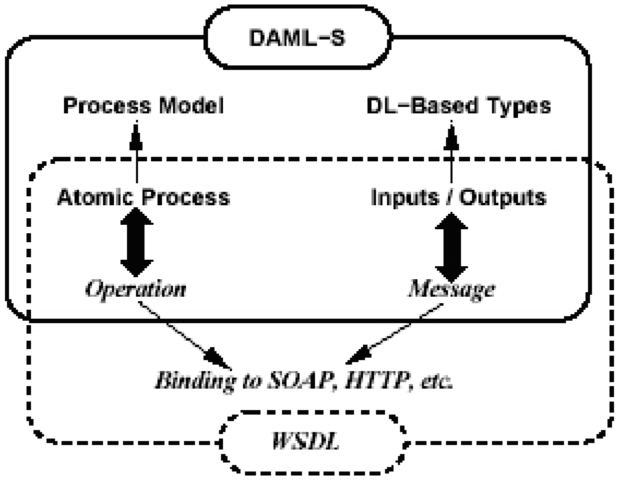


Fig. 1. Mapping between DAML-S and WSDL

discovery mechanism. Currently web services technologies support one or a few UDDI registries centers to publish and discover services, while such architecture is not scalable, easy to make the bottleneck problem and difficult to maintain to the rapid increase information. While the p2p infrastructure is rather flexible, which has advantages in directly exploiting resources present at other nodes of the network without the intervention of any central server, easy to maintain the network framework when the new node joins and leaves, especially after integrating semantic into p2p network, because p2p network could be easy to have the capabilities to find relevant set of ontologies, facilitating reuse of existing ontologies to create additional ontologies, and advertising the resulting ontologies. Considering discovery efficiency and time-consuming we prefer the hybrid p2p topology, see figure 2.

The designing thought of such architecture is to use p2p topology to expand the UDDI register center and use hierarchical peer groups to classify the different kinds of services. In figure 2, it is an abstract three-tier architecture, including the UDDI registries layer, service-syndication layer and the specific peer-syndication layer. The uppermost layer is UDDI registries layer, which function is similar to the former UDDI registry center, the only difference lies that one central registry is replaced and expanded by a group peers. Each peer acts as the UDDI registry, which maintains two kinds of information denoted by using different color tables, the blue table is the information shared by peers in the same layer, which is used to maintain the configuration of the peer group, to be used to note each other that who joins and who leaves. And you may note that in every layer there has the same blue table, they also have the same function; the yellow table is the information of its next layer's information, which has

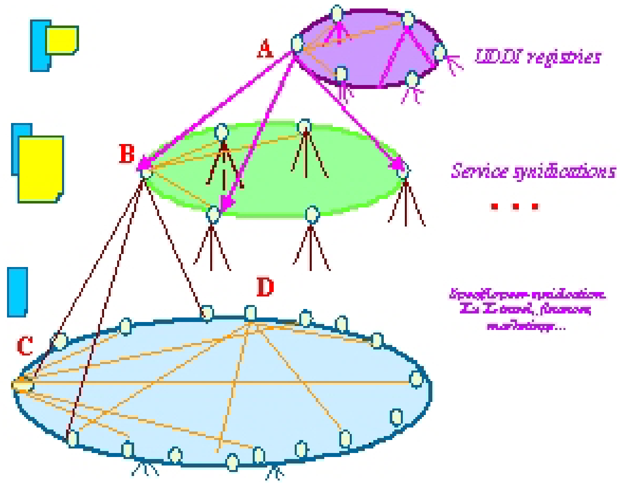


Fig. 2. The expanded UDDI registry by hybrid p2p topology

different content in different layer. To the first layer the yellow table maintains the information of service-syndication layer, that are the category information of services. The service-syndication peer group is formed by specific business areas in an e-marketplace, such as e-travel, finances, marketing and others. The second layer is service-syndication, its yellow table is the information of specific services provider, based on which the specific service provider could be located in its next layer called peer-syndication layer. So the contribution of using such hybrid p2p infrastructure is making the discovery of web services more flexible and available.

3 The Semantic Discover Process of Web Services

My web services architecture is semantic-based and expanded by p2p infrastructure, which use semantics to WSDL based DAML+OIL ontologies to define services, use ontology to organize UDDI registries, enable semantic classification of web services, store these semantic annotations in UDDI registries and search for Web services based on them. Base the result works of ([5], [6], [7], [8], [9]) a simple architecture is present to show how it works.

Figure 3 shows how this architecture works. First the publishing or requesting message will be treated by DAML-S [12] parser, the input message is a set of keywords, and the output is a list of ontologies. Match engine is used to deal with the matched services [11]. Ontology maker manipulates the ontology repository, such as creating, deleting, mapping or matching ontologies. As to the limited space the details will not be repeated here.

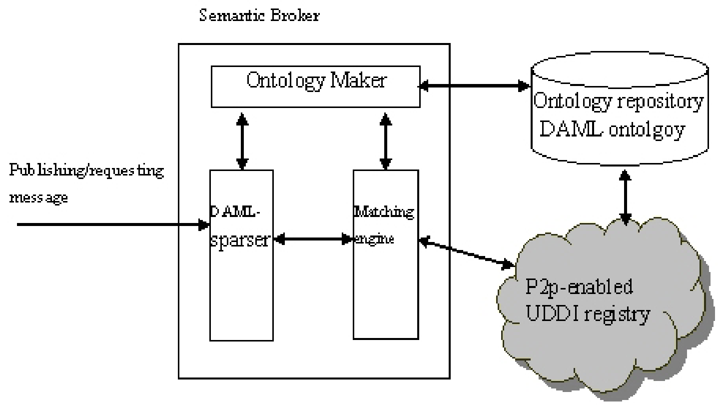


Fig. 3. How it works

4 Conclusions

This doctor proposal shows my current research in finding an effective discovery of web services based on semantic. This paper mainly showed the overview of researches dealing with such challenge. The involved works are semantic web, ontology-based web, DAML-S and p2p-enabled semantic web. Also based the some result works of these fields, this paper tried to present a potent approach to discovery services based on semantic and improved by p2p infrastructure. The doctor research is not far away from the beginning, the future work will also focus on services-oriented computing.

References

1. M. Paolucci et al.: importing the semantic web in UDDI. In Proceedings of E-Services and the semantic web workshop, 2002.
2. T. Berners-Lee, J. Hendler, and O. Lassila: The Semantic Web. Scientific American, 284(5): (2001) 34–43
3. J. Hendler and D. L. McGuinness: Darpa Agent Markup Language. IEEE Intelligent Systems, 15(6): (2001) 72–73
4. A. Ankolekar et al.: DAML-S: Web Service Description for the Semantic Web, in International Semantic Web Conference, Sardinia, Italy, pp. (2002) 348–363
5. K. Sivashanmugam, K. Verma, A. Sheth, J. Miller: Adding Semantics to Web Services Standards, Proceedings of The International Conference on Web Services, (2003) 395–401
6. DAML-S 0.9 Draft Release, 2003-05, http://www.daml.org/services/daml-s/0.9/
7. M. Paolucci et al.: Importing the semantic web in UDDI. In Proceedings of E-Services and the Semantic Web Workshop (2002)
8. M. Arumugam, A. Sheth, and I. B. Arpinar: Towards Peer-to-Peer Semantic Web: A Distributed Environment for Sharing Semantic Knowledge on the Web (2001)

9. K. Verma, K. Sivashanmungam, A. Sheth, A. Patil, S. Oundhakr, and J. Miller: METEOR-S WSDI: A Scalable P2P Infrastructure of Registries for Semantic Publication and Discovery of Web Services, Information Technology and Management, in press.
10. A. Maedche and S. Staab: Services on the Move - Towards P2P-Enabled Semantic Web Services, Proceedings of the Tenth International Conference on Information Technology and Travel & Tourism, ENTER 2003, Helsinki 2003/01/31.
11. C. Bartolini, J. Gonzalez-Castilho: A Semantic Web Approach to Service Description For Matchmaking of Services. HP Labs Technical Report. (2001)
12. A. Ankolekar, F. Huch, K. P. Sycara: Concurrent Semantics for the Web Services Specification Language DAML-S. COORDINATION: (2002) 14–21

Automatic *e*-Service Composition

Daniela Berardi

Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, 00198 Roma, Italy
berardi@dis.uniroma1.it

Abstract. Our main focus in this Ph. D. thesis is on automatic *e*-Service composition. We start by developing a framework in which the exported behavior of an *e*-Service is described in terms of its possible executions (execution trees). Then we specialize the framework to the case in which such exported behavior (i.e., the execution tree of the *e*-Service) is represented by a finite state machine. In this specific setting, we analyze the complexity of synthesizing a composition, and develop sound and complete algorithms to check the existence of a composition and to return one such a composition if one exists. To the best of our knowledge, our work is the first one that simultaneously tackles the following issues: *(i)* giving a formal and comprehensive characterization of what an *e*-Service is and of the problem of *e*-Service composition; *(ii)* laying the foundations for comparing the various approaches to composition in terms of their computational complexity characterization.

1 Introduction

Service Oriented Computing (SOC [13]) aims at building agile networks of collaborating business applications, distributed within and across organizational boundaries.[1] *e*-Services, which are the basic building blocks of SOC, represent a new model in the utilization of the network, in which self-contained, modular applications can be described, published, located and dynamically invoked, in a programming language independent way.

Research on *e*-Services spans over many interesting issues [14]. In this Ph.D. thesis, we are particularly interested in automatic *e*-Service composition. *e*-Service *composition* addresses the situation when a client request cannot be satisfied by an available *e*-Service, but a *composite* *e*-Service, obtained by combining "parts of" available *component* *e*-Services, might be used. *e*-Service composition is concerned with synthesizing such a new composite *e*-Service, thus producing a specification of how to coordinate the component *e*-Services to obtain the composite *e*-Service. Such a specification can be obtained either *automatically*, i.e., using a tool that implements a composition algorithm, or *manually* by a human.

Although an enormous interest is moving around *e*-Services ([1, 11, 2, 14, 8]), several aspects related to *e*-Services and *e*-Service composition including foundational ones, still remain to be clarified (see [9] for a survey on different approaches

[1] cf., Service Oriented Computing Net: http://www.eusoc.net/

to service oriented computing): *(i)* an agreed comprehension of what an *e*-Service is, in an abstract and general fashion, is still lacking; *(ii)* a consolidated formal definition of *e*-Service composition does not exist; *(iii)* due to the absence of a common vision, it is extremely difficult to compare the various approaches to composition, for instance in terms of their computational complexity.

The aim of this Ph.D. thesis is to define a formal and comprehensive framework for the characterization and the theoretical investigation of the problem of automatic *e*-Service composition. Although several papers have been already published that discuss either a formal model of *e*-Services (even more expressive than ours, see e.g., [8]), or propose algorithms for computing composition (e.g., [12]), to the best of our knowledge, the contribution of our research [5, 7, 4, 6] is the first one tackling *simultaneously* the following issues: *(i)* presenting a formal model where the problem of *e*-Service composition is precisely characterized, *(ii)* providing techniques for computing *e*-Service composition in the case of *e*-Services represented by finite state machines, and *(iii)* providing a computational complexity characterization of the algorithm for automatic composition.

The rest of this paper is organized as follows. In Section 2 we define our general formal framework and the problem of composition in such a framework. In Section 3 we specialize the general framework to the case where *e*-Services are specified by means of finite state machines and we present an EXPTIME algorithm for automatic *e*-Service composition in such a framework. In Section 4 we discuss future work. Finally, in Appendix A an example is discussed.

2 General Framework for Automatic Composition

In [6, 5], we give a general characterization of an *e*-Service, as a software artifact (delivered over the Internet) that interacts with its clients (possibly in a repeated way), which can be either human users or other *e*-Services. An interaction consists of a client invoking a command, i.e., an atomic action, and waiting for the fulfillment of the specific tasks and (possibly) the return of some information. Under certain circumstances, i.e., when the client has reached his goal, he may terminate the interactions. However, in principle, a given *e*-Service may need to interact with a client for an unbounded, or even infinite, number of steps, thus providing the client with a continuous service. Therefore, an *e*-Service can be characterized in terms of the sequences of actions it is able to execute, i.e., its behavior. In what follows, we refer to this conceptual vision of an *e*-Service as *e*-Service *schema*. An *e*-Service *instance* is an active occurrence of an *e*-Service effectively running and interacting with a client. In general, several active instances corresponding to the same *e*-Service schema exist, each one executing independently from the others. However, in what follows, wlog, we assume that the number of active instances corresponding to the same *e*-Service schema is equal to one. If more instances correspond to the same schema, we simply duplicate the schema for each instance.

When a client invokes an *e*-Service E, E may *delegate* the execution of some or all of its actions to other *e*-Services. All this is transparent to the client. To precisely capture such a situation, we introduce the notion of *community* [6] of e-Services, which is formally characterized by: *(i)* a (finite) common set of actions, called the *alphabet* of the community; *(ii)* a set of e-Services specified in terms of the common set of actions.

The behavior of an e-Service can be analyzed from two different points of view. From the external point of view, i.e., that of a client, an e-Service is seen as a "black box" that executes sequences of atomic *actions* with constraints on their invocation order. From the internal point of view, i.e., that of an application running an instance of E, it is important to specify whether the execution of each action is performed by E itself or it is delegated to another e-Service belonging to the same community C of E (transparently to the client of E). Therefore, it is natural to consider the e-Service schema as constituted by two different parts, called *external schema* and *internal schema*, abstractly[2] representing an e-Service from its external and its internal point of view, respectively.

The external schema abstractly represents the behavior of an e-Service as the set of all possible executions of all possible instances of that e-Service, i.e., as a tree of actions, called *external execution tree*. Any instance of the e-Service executes a path of such a tree. In this sense, each node x of the tree represents the history of the sequence of interactions between the client and the e-Service executed up to as certain point. For every action a that can be executed at the point represented by x, there is a single successor node y_a with the edge (x, y_a) labeled by a. y_a represents the fact that, after performing the sequence of actions leading to x, the client chooses to execute the action a, among those possible. Some nodes of the external execution tree are *final*: when a node is final, and only then, the client can end the interaction[3]. Notably, an external execution tree does not represent the information returned to the client, since the purpose of such information is to let the client choose the next action, and the rationale behind this choice depends entirely on the client. Additionally, our model of e-Service is oriented towards representing the interactions between a client and an e-Service. Therefore, our focus is on action sequences, rather than on message sequences as in [8], or on actions with input/output parameters as in [12].

The internal schema maintains, besides the behavior of the e-Service, the information on which e-Services in the community execute each given action of the external schema. Uniformly with the external schema, the internal schema is specified as an *internal execution tree*, where each edge is labeled by an action a, as before, and by a non-empty set[4] of instances of e-Services in the community that executes a.

In our framework, the problem of e-Service composition can be formulated as follows [5, 6]: given a community C of e-Services and the external execution tree

[2] For the moment, we are not concerned with any specification formalism.

[3] When a human interacts with an e-Service (instance) over the web, he can always abort the entire transaction. Here, we consider the abortion mechanism as orthogonal to the e-Service specification.

[4] The execution of actions can be delegated in parallel to more than one e-Service instance [7].

$T(E)$ of a target e-Service E, specified by the client in terms of the alphabet of C, find a labeling of $T(E)$ such that *(i)* each action a is labeled with a non-empty set of instances of e-Services in C that execute a, and *(ii)* each possible sequence of actions on $T(E)$ corresponds to possible sequences of actions of e-Services in C, suitably interleaved. In other words, the problem of composition amount to synthesize an internal execution tree for the external execution tree specified by the client, where each action is delegated to e-Services of C. When such a labeling exists we say that E is a *composition* wrt (the e-Services in) C.

3 Finite State Composition

Till now, we have not referred to any specific form of e-Service schemas. In what follows, we consider e-Services whose schema (both external and internal) can be represented using only a *finite number of states*, i.e., using (deterministic) Finite State Machines (FSMs) [5]. Intuitively, this means that we can factorize the sequence of actions executed up to a certain point into a finite number of states, which are sufficient to determine the future behavior of the e-Service.

FSMs can capture an interesting class of e-Services, that are able to carry on rather complex interactions with their clients, performing useful tasks. Indeed, several papers in the e-Service literature adopt FSMs as the basic model of exported behavior of e-Services [9, 8]. Also, FSMs constitute the core of statecharts, which are one of the main components of UML and are becoming a widely used formalism for specifying the dynamic behavior of entities.

A FSM A_E^{ext} is an external schema in the sense that it specifies an external execution tree $T(A_E^{ext})$. As far internal schemas, we consider those that can be represented as a Mealy FSM (MFSM) [5][5]. The MFSM A_E^{int} is an internal schema in the sense that it specifies an internal execution tree $T(A_E^{int})$.

In this specific setting we may wonder whether: *(i)* it is always possible to check the existence of a composition; *(ii)* if a composition exists, there exists one which is a finite state machine, i.e., a *finite state composition*; *(iii)* if a finite state composition exists, how to compute it. Our approach [5] is based on reformulating the problem of e-Service composition in terms of satisfiability of a suitable formula of Deterministic Propositional Dynamic Logic (DPDL), a well-known logic of programs developed to verify properties of program schemas [10].

DPDL enjoys three properties of particular interest: *(i)* the *tree model property*, which says that every model of a formula can be unwound to a (possibly infinite) tree-shaped model; *(ii)* the *small model property*, which says that every satisfiable formula admits a finite model whose size is at most exponential in the size of the formula itself; *(iii)* the EXPTIME-completeness of satisfiability in DPDL.

We represent the (FSM) external schema of both the target e-Service E_0 and the e-Services $E_1, \ldots, E_n$ of community C, as a suitable DPDL formula Φ. Intuitively, for each involved e-Service $E_i, i = 0 \ldots n$, Φ encodes *(i)* its current

[5] Intuitively, the subset of e-Services executing an action is returned by the output function.

state, and in particular whether E_i is in a final state, and *(ii)* the transitions that E_i can and cannot perform, and in particular which component e-Service(s) performed a transition. Additionally, Φ captures the following constraints: *(i)* initially all e-Services are in their initial state, *(ii)* at each step at least one of the component FSM has moved, *(iii)* when the target e-Service is in a final state also all component e-Services must be in a final state.

The following results hold [5]:

1. From the tree model property, the DPDL formula Φ is satisfiable if and only if there exists a composition of E_0 wrt $E_1, \ldots, E_n$.
2. From the small model property, if there exists a composition of E_0 wrt $E_1, \ldots, E_n$, then there exists one which is a MFSM of size which is at most exponential in the size of the external schemas of E_0, $E_1, \ldots, E_n$.
3. From the EXPTIME-completeness of satisfiability in DPDL and from point 1 above, checking the existence of an e-Service composition can be done in EXPTIME.

From a practical point of view, because of the correspondence between Propositional Dynamic Logics (which DPDL belongs to) and Description Logics (DLs [3]), one can use current highly optimized DL-based systems [3] to check the existence of e-Service compositions.

In [4, 7] we address the problem of computing e-Service composition by following another different though deeply related approach. Specifically, we use formalisms developed for Reasoning about Actions to represent e-Services, and show that again we can use logical reasoning, and in particular, logical satisfiability, to characterize the problem of e-Service composition.

4 Future Work

In the future we plan to extend our work both in practical and theoretical directions. As to the former, we are developing a DL-based prototype system that implements the composition technique presented in the paper. Such system will enable us to test how the complexity of composition in our framework impacts real world applications. On the theoretical side, we will address open issues such as the characterization of a lower bound for the complexity of the composition problem. Additionally, we plan to extend our setting, by taking into account the possibility that either the target e-Service or the behavior exported by the e-Services in the community is not completely specified. Another possible enhancement to our work is to consider an unbounded number of active instances for each e-Service in the community. Finally, far-reaching future work may be identified along several directions. For example, it is interesting to study if and how our mediated approach can evolve towards a peer-to-peer one.

References

1. M. Aiello, M.P. Papazoglou, J. Yang, M. Carman, M. Pistore, L. Serafini, and P. Traverso. A Request Language for Web-Services Based on Planning and Con-

straint Satisfaction. In *Proceedings of the 3rd VLDB International Workshop on Technologies for* e-*Services (VLDB-TES 2002)*, Hong Kong, China, 2002.
2. A. Ankolekar, M. Burstein, J. Hobbs, O. Lassila, D. Martin, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, T. Payne, and K. Sycara. DAML-S: Web Service Description for the Semantic Web. In *Proceedings of the 1st International Semantic Web Conference (ISWC 2002)*, Chia, Sardegna, Italy, 2002.
3. Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.
4. D. Berardi, D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Mecella. *e*-Service Composition by Description Logic Based Reasoning. In *Proceedings of the Int. Workshop on Description Logics (DL03)*, Rome, Italy, 2003.
5. D. Berardi, D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Mecella. Automatic Composition of *e*-Services that Export their Behavior. In *Proceedings of the First International Conference on Service Oriented Computing (IC-SOC03)*, Trento, Italy, 2003.
6. D. Berardi, D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Mecella. A Foundational Vision of *e*-Services. In *Proceedings of the CAiSE 2003 Workshop on Web Services, e-Business, and the Semantic Web (WES 2003)*, Velden, Austria, 2003.
7. D. Berardi, D. Calvanese, G De Giacomo, and M. Mecella. Composing e-Services by Reasoning about Actions. In *Proc. of the ICAPS 2003 Workshop on Planning for Web Services*, 2003.
8. T. Bultan, X. Fu, R. Hull, and J. Su. Conversation Specification: A New Approach to Design and Analysis of E-Service Composition. In *Proceedings of the WWW 2003 Conference*, Budapest, Hungary, 2003.
9. R. Hull, M. Benedikt, V. Christophides, and J. Su. E-Services: A Look Behind the Curtain. In *Proceedings of the PODS 2003 Conference*, San Diego, CA, USA, 2003.
10. Dexter Kozen and Jerzy Tiuryn. Logics of Programs. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science — Formal Models and Semantics*, pages 789–840. Elsevier Science Publishers (North-Holland), Amsterdam, 1990.
11. S. McIlraith and T. Son. Adapting Golog for Composition of Semantic Web Services. In *Proceedings of the 8th International Conference on Knowledge Representation and Reasoning (KR 2002)*, Toulouse, France, 2002.
12. S. Narayanan and S. McIlraith. Simulation, Verification and Automated Composition of Web Services. In *Proceedings of the 11th International Conference on World Wide Web*, Hawaii, USA, 2002.
13. Mike P. Papazoglou. Agent-Oriented Technology in Support of e-Business. *Communications of the ACM*, October 2003. To appear.
14. J. Yang, W.J. van den Heuvel, and M.P. Papazoglou. Tackling the Challenges of Service Composition in *e*-Marketplaces. In *Proceedings of the 12th International Workshop on Research Issues on Data Engineering: Engineering E-Commerce/E-Business Systems (RIDE-2EC 2002)*, San Jose, CA, USA, 2002.

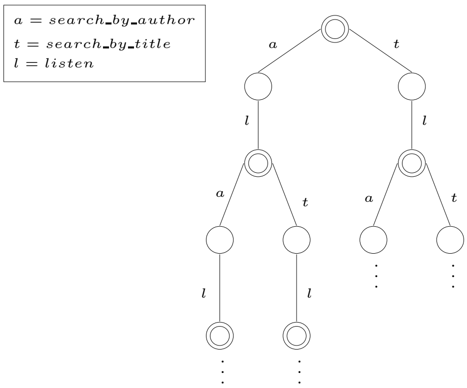
A Automatic Finite State Composition

Figure 1(a) shows (a portion of) an (infinite) external execution tree representing a target *e*-Service E_0 that allows for searching and listening to mp3 files[6].

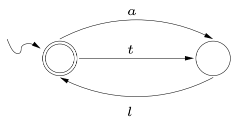
[6] Final nodes are represented by two concentric circles.

In particular, the client may choose to search for a song by specifying either its author(s) or its title (action `search_by_author` and `search_by_title`, respectively). Then the client selects and listens to a song (action `listen`). Finally, the client chooses whether to perform those actions again. Figure 1(b) shows the external schema of E_0, specified by the client as a FSM A_0. Figures 1(c) and 1(d) show the external schemas of the *e*-Services E_1 and E_2 of community C, represented as FSMs A_1 and A_2, respectively. In other words, E_1 and E_2 are the *e*-Services that should be composed in order to obtain a new *e*-Service that behaves like E_0. E_1 allows for searching for a song by specifying its author(s) and for listening to the song selected by the client; then, it allows for executing these actions again. E_2 behaves like E_1, but it allows for retrieving a song by specifying its title. Wlog, we assume that the community C is composed by E_1 and E_2 only, and therefore, the alphabet of actions of C is $\{\texttt{search\_by\_author}, \texttt{search\_by\_title}, \texttt{listen}\}$.

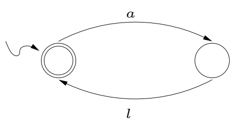
Figure 2(a) shows (a portion of) an (infinite) internal execution tree, associated to the external execution tree of *e*-Service E_0. Our algorithm for automatic composition synthesizes such an internal execution tree, where all the actions are delegated to *e*-Services of the community. In particular, the execution of `search_by_author` action and its subsequent `listen` action are delegated to an instance of E_1, and the execution of `search_by_title` action and its subsequent `listen` action to an instance of E_2. Figure 2(b) shows a finite state representation for the internal execution tree of Figure 2(a). Note that in general, it is not possible to find a labeling for the target *e*-Service represented as FSM. Indeed, in our example, the labeling of the `listen` action on A_0 depends on which action between `search_by_title` and `search_by_author` is executed, since they are delegated to different *e*-Services.



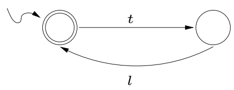
(a) External execution tree for E_0



(b) FMS for E_0

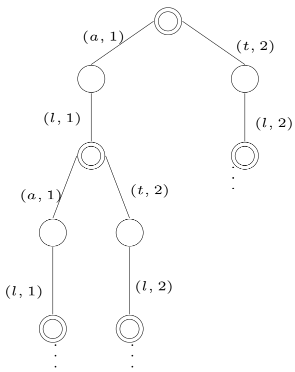


(c) FSM for E_1

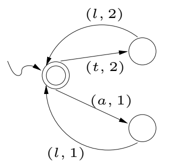


(d) FSM for E_2

Fig. 1. *e*-Services participating to the composition.



(a) Internal execution tree for E_0



(b) MFSM for E_0

Fig. 2. Composition of E_0 wrt E_1 and E_2.