Business Integration Models in the Context of Web Services

William Song[1]*, Deren Chen[2], Jen-Yao Chung[3]
[1] *Computer Science, Durham University, Durham, UK*
[2] *Computer Science, Zhejiang University, Zhejiang, China*
3 IBM T.J. Watson Research Center, New York, USA
w.w.song@dur.ac.uk , drchen@zju.edu.cn, jychung@us.ibm.com

Abstract

E-commerce development and applications have been bringing the Internet to business and marketing and reforming our current business styles and processes. The rapid development of the Web, in particular, the introduction of the semantic web and web service technologies, enables business processes, modeling and management to enter an entirely new stage. Traditional web based business data and transactions can now be analyzed, extracted and modeled to discover new business rules and to form new business strategies, let alone mining the business data in order to classify customers or products. In this paper, we investigate and analyze the business integration models in the context of web services using a micro-payment system because a micro-payment system is considered to be a service intensive activity, where many payment tasks involve different forms of services, such as payment method selection for buyers, security support software, product price comparison, etc. We will use the micro-payment case to discuss and illustrate how the web services approaches support and transform the business process and integration model.

1. Introduction

Use of the Internet has reshaped people's life in many ways. At the beginning, people use the Web as a huge information repository to search for what they need. Nowadays, people do much more on the Web. They advertise, they publish articles, they play games, they go shopping, and they even gamble. Two decades ago, people started doing business through the Internet. EDI (electronic data exchange) has been a standard for transferring data between two large enterprises involved in business transactions.

Nowadays, the Internet and the Web are reshaping people's business life. These terms, like e-commerce, e-procurement, e-payment, e-marketplace, have illustrated a large number of different requirements, which need to be met in various e-business models and technologies [9]. In this paper, we attempt to investigate the business process and models in the context of web services.

1.1. Business process and integration model

The rapid development of the Web, in particular, the introduction of the semantic web and web service technologies enables business processes, models, and patterns to enter an entirely new stage. Traditional web based business data and transactions can now be extracted and modeled to discover business rules and business strategies, let alone mining the business data in order to classify customers or products.

A nowadays business model is a very complicated system which, in general, consists of information processing, collaborative work, enterprise analysis, and service management. These components represent a horizontal view of the business structure, while the service management component itself also provides a refinement mechanism that can be roughly expressed in three intersectional facets in vertical dimension, that is, business level, component level, and functional level. At the business level, we describe general business objectives and requirements. At the component level we refine the business objectives into a number of concepts or processes, which are modeled in information system development. These concepts or processes can be further decomposed. At the functional level, we focus on mainly on implementation issues. The high level components are now described in various functions and programming modular.

Most of the web services components are those of

* The corresponding author: Address. Computer Science, University of Durham, South Road, DH1 3LE, Durham, UK, Tel. +44-191 334 1729, Fax. +44-191 334 1701, Email. w.w.song@dur.ac.uk.

0-7695-2430-3/05 $20.00 © 2005 IEEE

concepts, processes, or functions. For example, the services, described in OWL-S, have an ontology, which contains service profile, service process, and service grounding. Here the service profile provides the concepts describing the service, such as business contacts, the service process describes how the service works (including input and output parameters), which is very close to function description of the service [8]. Another issue proposed in [11] suggests viewing software components as services. This concept is significant in the sense of modeling, communicating, and composing services. However, it is quite difficult to build up a repository to manage this huge number of software components.

The induction of the semantic web methodology, such as ontology definition, gives us a powerful tool to describe the meaning of business processes, business structures, and business rules. It is more useful to build up various specific ontologies for business objects within a business organization, such as personnel, products, and management hierarchies. For example, to model the access rights for business, RBAC has been produced [2].

1.2. Web service issues

Web Service is a piece of software that makes itself available over the Web and uses a standardized XML messaging system [1]. Its significance lies in composition of software for various purposes for users who attempt to accomplish certain tasks with the composed services through the Web. A web based application system requires various software components, such as security supporting programs and payment methods, to work together to accomplish the required tasks. In other words, it is required that a dynamic construction of various related services and software be performed to meet a particular user's specific need.

Many pieces of such software exist in the Web. The Web Services methods provide various supporting facilitates to describe the software for sharing and reuse. Among others, there are two issues significant for a web based application system development. The first is service description and selection of software components. Web Service Description Language (WSDL) and Universal Description, Discovery, and Integration (UDDI) are proposed standards for software component description. The purpose of the description is for the users to select the right and suitable software components. The second is service integration or composition of the selected software components. The integration issue is based on the service description. During the integration process, the service components, which fit together to serve the users requirements, are composed to form the unified software for particular tasks.

In this paper, a web based application system, the micro-payment system, is a software and service intensive activity, where many payment tasks involve different forms of services, such as payment method selection for buyers, security support software, product price comparison, etc. Using the web service technology will great enhance the quality and performance of the micro-payment system.

1.3. Business model for micro-payments

A key business issue to apply the micro-payment techniques in the context of web services is that more and more services distributed over the Web will be organized dynamically to meet a given task by a user. These services are comprised by a collection of software components or functions from various sources. Together with the data and resources provided (and changed) by the content/service providers, the services or software pieces will be also charged. For example, a customer may like to know the one day cost for her mobile phone when she was abroad.

Currently, the description of a service mainly focuses on its simple semantic description and ontology using UDDI, and profiles and processes using WSDL. While payment is concerned, a suitable description for services in terms of business processes and transactions is dispensable. In [4], BPEL4WS has been developed for description of business processes for web services. However, to implement semantic description in web services modeling and hence to represent the business objectives and transactions is crucial because we need to compose web services to form a greater task for a given goal where the description for payment is considered to be a dynamic part of the composition process of web services.

1.4. Paper overview

The paper is organized as follows. In the next section, we describe general business integration model, focusing on business integration model, business ontologies, and integration issues. Then, in section 3, we describe the case of micro-payment, discussing its procedure, system, and components. We will also discuss what the business process model issues are considered. In section 4, the web service approaches to the business model of micro-payment system are discussed, where we propose some business service patterns and semantic description model for micro-payment systems. Finally we conclude the paper in section 6 by proposing our future work in this direction.

2. The business integration model

According to [4], a business process model can be viewed to be a formal definition of composing objects, which provides the description of the behavior and interactions of a process instance relative to its partners and resources through Web services interfaces. It provides a standard XML language to express business processes consisting of functions defined. The business process model considers both design and runtime uses. At design time, development or modeling tools can use, import, or export the proposed models and objects, allowing business analysts to specify processes and developers to refine them and bind process steps to specific service implementations. At the runtime, its business workflow engine can use the models and objects to control the execution of processes, and invoke the services required to implement them.

From the viewpoint of business integration, the business process model describes a set of requirements for an *integration framework* that enables the high level requirements and the low level functions to match each other. Through web services support, a business process can be effectively mapped into their functions via a logical process model. The business process can be exported and translated into the business workflow and functions, which are further implemented as web services. In a distributed environment, web services are used to enable the business integration technology, so *un-interoperable* software components can work together.

One of the goals of web services is universal interoperability between applications by using web standards. The Business Integration technology can be used for this purpose, to deliver a loosely coupled integration model, allowing flexible integration of heterogeneous systems within an enterprise or in a variety of domains, including a business-to-business or business-to-consumer process model.

2.1. Business integration model

The business integration consists of four components, Information processing, Collaboration, Enterprise modeling, and Service self-management, which can be further decomposed into business functions. These four components are generally described as follows:

- Information processing component is to handle various sources of data from the service providers and aggregate the data for the end users. The Web services technology will be used to make the content of the data available to the customers.
- Collaboration component is to deal with the individuals, programs, and organizations to work together to make payment transactions done. Access controls and management are the key issue, which should be processed in the Web Services infrastructure.
- Enterprise component is to handle the business processes, systems requirements, and basic services for the payment systems. The entire business value chain will be modeled and supported by using the Web Service technology.
- Service self-management component is to deal with all the services that provide support to the above three components. The services will be selected and integrated based on the availability and on-demand principles.

2.2. Ontologies

As depicted in Fig. 1, the business ontology group contains four specific ontologies, which inherit the characteristics from the upper level ontology. These four ontologies, described below, attempt to represent various dimensions in a business process and integration model.

- Service ontology provides various ontologies for the application domains. These domain ontologies vary from one application to another. For example, payment service has its own payment ontology, while messaging service provides its specific messaging ontology.
- Task ontology provides conceptual description for various flows, which express the relationships between business tasks and their sub-tasks.
- Organization ontology describes structural conceptions and relationships. It contains departmental issues, personnel issues, across-organizational issues, and various internal and external agent issues.
- Economy and finance ontology is actually a specific ontology for our application case, the micro-payment system. Its aim is to indicate that it is the business ontology beyond the business domain.

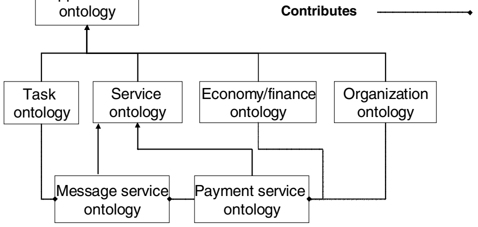

Fig. 1 Ontology group for web based business system

2.3. Integration issues

The integration issues involve the following four aspects, i.e., platform integration, ontology integration, virtual organization, and metadata model integration.

- Platform integration deals with integration of various platforms, such as e-Wallet in mobile devices, server-managers, and web service interfaces.
- Ontology integration considers different views represented in different ontologies. Examples of these ontologies, include these given in the above section.
- Virtual organization indicates a collaborative work required in order to accomplish a given task. For example, a payment transaction may need to co-use the micro-payment method and credit card payment method.
- Metadata model integration copes with different representations of business requirements and functions due to using different metadata modeling languages, such as RDF, OWL, WSRF, WML.

3. Case Study: a Micro-payment system

Let us consider how a micro-payment system works, see Fig. 2. We assume that all transactions take place over the Internet. Similar to ordinary marketplace, where people visit, select, and buy things, now we have an electronic marketplace. You can enter the e-marketplace, choose what you want and pay for them. Consequently, there are at least two basic components. One is a collection of goods (probably services or pieces of information) in the e-marketplace. The other is a wallet holding some money (usually small amount) enabling you to pay for your purchase. This wallet is called electronic wallet or e-wallet. As a matter of fact, there are another three parties involved in the e-marketplace and micro-payment. The first one is Internet Service Providers (ISPs), which enable you to connect to the Internet, such as telecom companies. The second one is Internet Content Providers (ICPs), which provide products, goods, services, or other items for buyers [3, 6]. The third one is a bank or a financial unit to certify the wallet you hold is valid and has a certain amount of electronic cash (e-money).

3.1. System structure

A micro-payment system consists of these components: an e-wallet, a billing system at ISP/bank, and a billing system at merchant. In the following, we describe these components and their connections.

The e-wallet component itself has three parts: an Internet connection to ISP and ICP, a small billing mechanism, and an interactive interface. The Internet connection allows it to access the required certificate data from ISP or a bank (of course, ISP has already provided the user the Internet services) when necessary, and to ICP for shopping. The small billing mechanism (compared with the large billing system maintained at the ISP side) lets the user know how much he has spent and how much left in the e-wallet. The mechanism also provides a list of items the user has bought and other related information. The interactive interface allows the user to communicate with the system, for example, login messages.

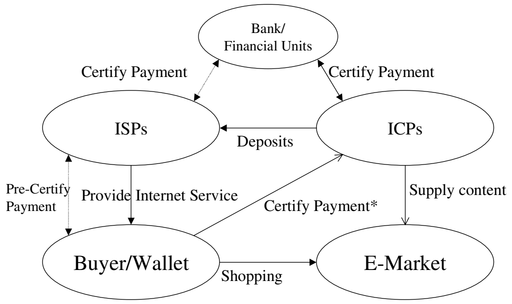

Fig. 2 Payment flow in a micro-payment system

The billing system at the ISP side performs three tasks. First, it provides the certificate of the valid e-wallet to ICP or transfers such certificate from a bank to ICP. Second, it maintains all the users' billing records. Third, it periodically checks the customers' e-wallets and billing records.

The billing system at the ICP side performs two tasks. First, it maintains its own billing records for different customers. So it can report to and get redeemed from the ISP. Second, it manages all the items sold and for sale.

3.2. Transaction flows

In this section, we describe a collection of possible certificate transactions between the micro-payment parties. See Fig. 3. First of all, we assume that the buyer's wallet, which has been initially certified by the ISP or bank, holds some e-money. This is called pre-certify payment. When a buyer has chosen some item to buy, by a click of payment, he issues a "certify payment" to ICP. Then ICP sends a message of "certify check" to ISP. This is the first round of certificate, from buyer, to ICP, and to ISP.

ISP checks the buyer's ID and the amount the buyer has in his wallet. If the buyer's ID is correct and the amount is sufficient for this payment, the ISP sends a message "certify reply" back to the ICP. This is the second round of certificate, from buyer, to ISP, and to ICP. When the ICP gets "yes" from the ISP, it delivers the item to the buyer.

During the second round, the ISP, in this order, updates the billing records at the buyer's wallet, its own billing records, and the ICP's billing records.

If the buyer's ID is wrong or the amount is not sufficient, the ISP will send messages to both the buyer and the ICP, informing "no certify reply".

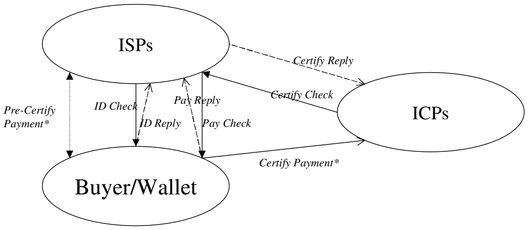

Fig. 3 Transaction flow among Micro-Payment components

3.3. Payment description

In the micro-payment system, data transfer through the Internet is most important. Therefore, how to describe the transferring data so that such description can meet the current and future requirements of the Internet data exchange and maintenance is critical. Since W3C proposed XML to be a standard for various data representation in the Web, XML has been widely accepted and used for description of a great variety of the Web resources. XML has the advantages of extensibility, separation of content from presentation, strict syntax, well-formedness, etc.

Recently, based on the XML specifications, a particular markup language specification for describing micro-payment is proposed to W3C [15]. This specification provides an extensible way to embed in a Web page all the information necessary to initialize a micro-payment (amounts and currencies, payment systems, etc). This embedding allows different micro-payment electronic wallets to coexist in an interoperable manner.

This specification defines a set of tags for description of payment related information to be transferred on the Web.

3.4. Business process in Micro-payment

The micro-payment system provides us a good case where various business issues have occurred. From the end users (such as buyers) and the experts in the finance domain, the requirements for the system are usually very general, informal, and less precise but very critical and essential. These requirements directly motivate business process and high-level service description, as well as various subjects and concepts, on one hand.

On the other hand, the micro-payment system development requires a large number of existing and reusable software components and data sources. These software components and data sources are available in various types of services via the web. An effective description framework for the business integration model is indispensable because we need these services to match our goals and these services must be coupled and integrated together in order to accomplish a given task.

Still another aspect is the middle level requirements and description, which is also modeled using the business process and integration model. In this model, we need to specify the business tasks and workflows, which are representing the general users' requirements and objectives, embodying the business strategies, structure, and natures, and motivating the functional and dynamic requirements and specifications for the software components and data structures.

In brief, from the micro-payment case, it is quite obvious that the business process and integration model with web services mechanism is required to describe a vertical business refinement process as well as a horizontal integration process. In the next section, we will discuss how these issues are coped with under the web services.

4. Web Service in the business model of micro-payment

In this section, we will discuss how web services will support the business process and integration model in the micro-payment case. Micro-payment is a complex, software component based, distributed system. It involves different parties for different functions. To accomplish one micro-payment transaction, many services are required, such as setting up appropriate security channels, checking the payer's identity, making daily purchase records. These business services form different patterns [12], serving different users' requirements and purposes. In order to describe the businesses and services in a semantic rich manner, we need to build up a semantic description framework for the business services. This framework is also used to describe the end user's requirements and profile. In the following we will first discuss the semantic description framework for web services, and then illustrate the three aspects of business services: business combination patterns, semantic description, and requirements matching.

4.1. Semantic description framework

The proposed semantic description framework model, based on the W3C recommendation – RDF and RDFS, contains the following components: a characteristic based semantic description modeling component for representing the businesses and general services; a process based description modeling component for expressing service functions and processes; a structured requirement modeling component to represent the end users' requirements and requests, as well as their profiles. In addition, we also use an ontology model for representing the business ontology and service ontology, where the domain knowledge is structurally expressed.

The semantic description framework contains three major components: a semantic descriptor, a pattern constructor, and a user request formation. The semantic descriptor is for business, service, and function description. As we discussed earlier, business description specifies business services and services specifies service functions. In addition, ontology support for general business and service is also maintained within this component.

The pattern constructor uses the service and pattern repository to collect the matched services discovered through matchmaking and semantic mapping mechanism to form the three services patterns, integration, aggregate, and composite. For the integration pattern, the service function description will be critical because to couple different services together to form a chain of services for a specific purpose requires accurate function specifications.

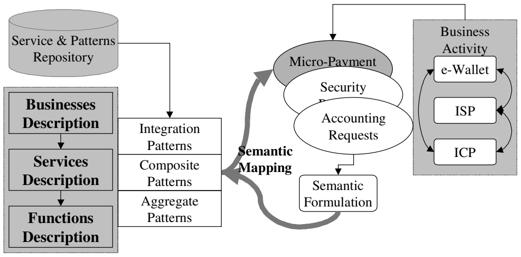

Fig. 4 The integrated architecture for services

The user request formation is to apply the criteria supplied from the semantic description framework and the ontology structure to re-structure the user requests, so that the requests will bring more semantics for later matchmaking with the service description within the registry component. Currently we are developing a graph matching algorithm, in which a user request is seen as a sub-graph and the service description schema is considered to be a big service graph [7].

4.2. Business combination patterns

In a payment environment, the users are using a number of payment services, for example pay-by-e-cash, pay-by-e-wallet, pay-by-e-card, or using payment services provided by different service providers. Each payment method can be decomposed into a number of services. These services and sub-services will work together in one way or another to meet the users' requirements. These working ways are considered to follow certain kind of service patterns.

Service patterns suggest some forms that services are organized to serve certain purposes or to accomplish certain tasks. Such form or model is peculiarly significant in the situations where many distributed services are linked, integrated, coordinated, and cooperated together with their data sources and even human power sources for achieving a tremendous target. These situations or organizations are called "virtual organization".

These service patterns can be considered to be in these three forms, a service integration pattern, a service aggregate pattern, and a service composite pattern. The service integration pattern maintains a set of services or software which is organized in parallel. When a service request comes, a "broker" checks the requirements of the request and finds a most suitable one to meet the request. To the users, it appears as if there were only one service, for example, payment service, that the users need to know.

The service composite pattern has been widely discussed and is mainly considered to be a line of services. Each service is dependent on its preceded service to provide inputs and supplies outputs to its follower. A special situation is that one service may request or call another service in order to fulfill its own tasks. The key issue here is dynamic composite of services with a secure or fault torrent mechanism.

The service aggregate pattern indicates the situations where a set of services available but one of them is selected according to the users' requirements. These services almost serve the same purposes and do the same functions. Their difference may lie in for example, low cost but light security support, better usability but less functionality, etc. This pattern emphasizes more on service availability whereas the composite pattern focuses on-demands.

How to develop these business patterns for payments is heavily dependent on a good semantic description method, which can provide rich semantics for services and better information exchange through the Web. The aim to propose these patterns is to suggest some business models of payments that better serve the users with web service techniques.

4.3. Service description

According to the triangle architecture for web services [13, 14], a web service involves three parties, a registry where the service is registered and "described", a service provider that also provides service specifications, a service consumer or user that makes service request together with sort of service description. The service description mainly contains business description and service description. The business description is about the business situation of the service provider and the service description is about the services that the business is providing. The proposed semantic description framework aims at improving the business services with richer semantics for better matching between the users requirements and the service description.

Payment service description. We attempt to maintain a semantic description framework to provide full description for a payment service. The framework will include a static part, which describes general transactions methods and general payment processes, and a dynamic part, which stipulates operations such as match bindings, etc. Following is an example of the semantic description framework (static part) for payment transaction.

A transaction object includes transaction ID, transaction description, transaction properties, and transaction ontology. Transaction ID is the unique number for the transaction, which is used to identify the transaction. Transaction Description is a short description for the transaction for human understanding and NL semantic processing. Transaction Properties are RDF based description for the transaction. The properties include, for example, name, relationships with other transactions, transaction types, etc. Transaction ontology describes a set of hierarchical structures, such as conceptual taxonomy, process hierarchy structure, etc.

In Fig. 5, we illustrate a payment object with its description elements. The description is based on RDF. The pay object is related to a number of values through the properties like "From" and "To". It is also related to a function property, called "PAY". The object is about a "previous" payment, which includes the payment amount, payment methods and security requirements. The above description fragment is, however, our first attempt to formalize the description for payment transactions.

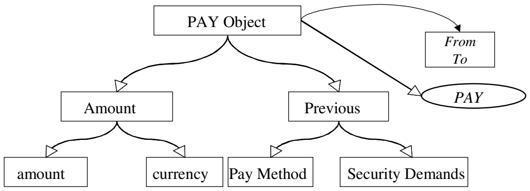

Fig. 5 A description model for payment object

Service function description. Service function is currently not well very described part within service description but it is a useful part for improving the user requirement matches. As it is difficult for a business to provide a description of its services meeting some normalized form, they are in most cases written in natural languages with very informal structures. In our framework, we also propose a formal specification for service function description. This formal description mainly concerns that the services accomplish what tasks, perform what processes, possess what features, and involve what other processes and services. For example, the functions for a payment service include access checking, identity verification, accounting, clearance, etc.

User request description. As a most important party in business service, the users description will heavily influence the quality of requirement matching. However, this work is now still left to the end users. The end users have to input some keywords and select some ontology terms for an application domain to a service registry in order to search for the services they request. In most cases, the users do not know their exact requirements for services and even do not know how to structurally form their requirements. Their expression of requests can be very vague and ambiguous. To support the end users with their payment requests, for example, the users' requirements can be transferred in a certain structure together with their profiles.

Suppose that a user wants to download a single song from a Web-based music shop and pays 50 cents for it. The user may demand a payment service for this transaction using his "e-wallet" and to be "secure". His "e-wallet" provides e-cash, e-card, phone-bill account. By "secure" he means that a security channel should be set in SSL. According to these requirements, a payment service is searched against the service registry and a matchmaker mechanism is performed to find a best match. Here we need to emphasize that to structure the users' requirements and to use a semantic framework to describe the users' profile is extremely important. The users' profile helps us with good understanding of "secure".

4.4. Service integration

Initially, the service integration process is to search and find the services from the service registry according to the users' requests. However, the users' requests are currently expressed only in terms of service names or keywords. Many approaches have been proposed to enrich the semantics for the services to meet the users' requirements [5, 7, 14]. The main feature of the approaches is to introduce ontological structure for description of services and businesses. For example, in order to provide more semantic support to the description and search of services in the service registry, a set of mapping rules can be established between the user query entries described by the semantic description framework and the service registry. Each service description item is also mapped into a registry entry. The ontology model can be used to identify services and therefore a richer semantic description for businesses and services, as well as the users requirements can be achieved using the semantic framework.

The semantic description is useful also in matching techniques, which are used for finding required services or selecting a best match among a number of service candidates. Currently, we are investigating a formal match method, which we call graph match. Suppose that *a* and *b* are two service nodes with an edge *ab* indicating a possible relationship, say, *a* depending-on *b*, between them and *G* is a service description schema. There may be a sub-graph *G'* in G which contains nodes *a* and *b*, as well as a chain of edges linking *a* and *b*, which contains *ab*. With support from the semantic relativity approach [11] and large number of services available, this graph match technique will be more effective for selection of best services.

5. Conclusion

Through an application case to analyze business models, we can see clearly the various components in the business model. In this paper, we have generally proposed a business process and integration model initiated by IBM [4] with a view of web services technology, which will extend the business infrastructure. Through an analysis of the development of micro-payment we described some business and service patterns for the business model of micro-payment system. We believe that with gradual deployment of web service techniques, business integration model and its implementation can be better achieved as web services will well bridge the gap between the general business requirements and the descriptions of software components and functions.

However, as this is our initial investigation we have realized that there are many holes in the model. For instance, the stream of vertical refinement is unclear to us yet. We know, the process of turning a general end users' requirement into a business process description is very difficult, let alone defining a model or framework to introduce a formal and precise method for this process. Nevertheless, we consider this is a move.

Our next step is to study further business models with web services and investigate how the web services methodologies will support business process and integration. More concretely, we will pursue the work in the following three aspects. First, we will develop a semantic based web service description model, which is able to cope with both high level requirements and low level functions. Second, we intend to describe a workflow or business process in this description model. Third, we will build a business and service pattern repository where each pattern corresponds to a pool of software components and functions.

References

[1] E. Cerami, **Web Services Essentials – Distributed Applications with XML-RPC, SOAP, UDDI & WSDL**, The O'Reilly Network, February 2002.

[2] David Ferraiolo and Rick Kuhn, **An Introduction to Role-based Access Control**, NIST/ITL Bulletin, December, 1995

[3] A. Herzberg and H. Yochai, **Mini-Pay: Charging per Click on the Web**, in Proceedings of the sixth International Conference on WWW, 1997

[4] M. Keen, J. Cavell, S. Hill, C. K. Kee, W. Neave, B. Rumph, H. Tran, **BPEL4WS Business Processes with WebSphere Business Integration: Understanding, Modeling, Migrating**, IBM Redbooks, Dec. 2004

[5] M. Paolucci, T. Kawamura, T. R. Payne, K. Sycara, **Semantic Matching of Web Services Capabilities** in Proc. of 1st First International Semantic Web Conference, Sardinia, Italy, I. Horrocks, J. Hendler (Eds.) June 9-12, 2002.

[6] R. Rivest and A. Shamir, **PayWord and MicroMint: Two simple micropayment schemes**, in Proc. of 5th Int'l Conference on WWW, May, 1996, Paris, France.

[7] W. Song, **Semantic Issues in the Grid computing and Web Services**, in Proceedings of International Conference on Management of e-Commerce and e-Government, Nanchang, China, Oct. 2003

[8] W. Song and Y. Zhang (2004), **Semantic Description and Matching of Grid Services Capabilities**, AHM2004, Nottingham, UK

[9] W. Song, D. Chen, J-Y. Chung, **An investigation on using web services for micro-payment**, Proc. of European Conference on Web Services, Erfurt, Germany, 27-30 September, 2004, Lecture Notes in Computer Science Volume 3250 (2004)

[10] W. Song, **Semantic Object Relativity in Schema Integration**, in Proceedings of CAiSE, Utrecht, The Netherlands, 1994

[11] M. Turner, D. Budgen, and P. Brereton, **Turning Software into a Service**, IEEE Computer, October 2003 (Vol. 36, No. 10).

[12] G. Vasudeva, **Patterns for business**, IBM White Paper, 2003.

[13] L. Zhang, Q. Zhou, and J-Y. Chung, **Developing Grid computing applications**, IBM Report, IBM T.J. Watson Research Center, New York, USA.

[14] Y. Zhang, W. Song, K. Bennett, **Semantic Enrichment for Web Service Description and Matchmaking**, Department Report, Computer Science, University of Durham, UK. 2004

[15] Thierry Michel, Common **Markup for micropayment per-fee-links**, the World Wide Web Consortium Working Draft 25 August 1999.