Web Services
CEPIS, Council of European Professional Informatics Societies, is a non-profit organisation seeking to improve and promote high standards among informatics professionals in recognition of the impact that informatics has on employment, business and society.

CEPIS unites 36 professional informatics societies over 32 European countries, representing more than 200,000 ICT professionals.

CEPIS promotes

Monograph: Web Services (published jointly with Novática*)

Guest Editors: José Carlos del Arco-Prieto, Jesús Arias-Fisteus, Óscar Corcho-García, Jorge Cardoso

2 Presentation. Web Services from An Industrial and An Academic Perspective — Jose Carlos del Arco-Prieto, Jesús Arias-Fisteus, Óscar Corcho-García and Jorge Cardoso

5 Web Services: Introduction and State of The Art — Óscar Corcho-García, José Carlos del Arco-Prieto and Jesús Arias-Fisteus

11 W3C and Web Services Standardization — Martín Álvarez-Espinar

16 Integration and Interoperability Experiences in Healthcare — Antonio García-Landeira

21 Model-Driven Extra-Functional Property Development for Web Services: A Case Study from The Service and Client Side Perspectives — Guadalupe Ortiz-Bellot and Juan Hernández-Núñez

29 Towards an Automated Trading Process — Pablo Fernandez-Montes, Manuel Resinas-Arias de Reyna and Rafael Corchuelo-Gil

34 Semantic Web Services with WSMO — Holger Lausen, Jos de Bruijn, Uwe Keller and Rubén Lara

38 Towards Semantic Service Selection for B2B Integration — Andreas Friesen and Kourosh Namiri

46 Leveraging E-Marketplace Models for Web Service-Based Application Development — Abraham Nieva de la Hidalga, Liping Zhao and Pedro R. Falcone-Sampaio

25 From Novática (ATI, Spain)

Open Standards
Open Standards, A Renewed Impulse: The Case of Open Document Format — Miguel A. Amutio-Gómez

59 From Mondo Digitale (AICA, Italy)

ICT Security
Critical Information Infrastructure Protection — Sandro Bologna, Roberto Setola and Salvatore Tucci

68 From ITNow (BCS, United Kingdom)

Informatics Profession Professionalizing IT — Peter Skyte

70 Harmonise Project: Presentation

* This monograph will be also published in Spanish (full version printed; summary, abstracts, and some articles online) by Novática, journal of the Spanish CEPIS society ATI (Asociación de Técnicos de Informática) at <http://www.ati.es/novatica/>.
Presentation

Web Services from An Industrial and An Academic Perspective

Jose Carlos del Arco-Prieto, Jesús Arias-Fisteus, Óscar Corcho-García and Jorge Cardoso

Web Services have gained great importance in academic and industrial environments due to their focus on open standards widely accepted by industry, which guarantee interoperability between the information systems of different organizations.

The emergence of Web Services has had a major impact on the way in which organizations integrate their applications, data, and processes. The use of standards has reduced the cost of corporate integration projects by allowing the reuse of existing applications and benefiting from previous investments. Web Service adoption in the industry has allowed organizations to share information with their partners, providers, and customers in a standardized manner.

In academia, the number of papers, conferences, projects, and research lines focusing on Web Services related aspects has grown in the last years. Research is focused on the implications of applying Web Services to areas such as B2B, EAI, BPM, EC, Grid Computing, and Semantic Web. Web Services have gained great importance in academic and research lines focusing on Web Services related aspects have grown in the last years. Research is focused on the implications of applying Web Services to areas such as B2B, EAI, BPM, EC, Grid Computing, and Semantic Web.

The Guest Editors

José Carlos del Arco-Prieto is a graduate in Computer Science from the Universidad de Huelva (1998). Later his professional career began in Tecsidel (1999) where he participated in projects for Telefónica I+D and made analyses of the impact of Web Services on different markets (2002). In 2004 he worked in the Diputacion de Huelva (Huelva Provincial Council) participating in the European project "Regions on Line" (ROL). In 2005 he worked in Cibernos collaborating on the Diraya project led by Indra and participating in the development of proposals of interoperability models and Web Services-based integration. He is currently working for T-Systems, doing technical consultancy work and collaborating in the definition of interoperability models, integration for the public sector, and the design of methodologies for SOA development. He collaborates with the Universidad de Huelva in the promotion of Web 2.0 technologies. He is the father of the first Web Services Latin list (webservices-Latinos), promoter of JSWEB 2005 Workshop, Co-president of JSWEB 2006 Workshop and a member of the steering committee of European SOA & Application Architecture Conference (2007). One of his main interests is the convergence between the worlds of academia and business. <josecarlos.delarco@t-systems.es>.

Jesús Arias Fisteus works as an assistant professor in the Telematic Engineering Department of the Universidad Carlos III de Madrid. He received his MSc with honours in Telecommunication Engineering in 2001 from the Universidad de Vigo. In 2005, he received his PhD in Communication Technologies from the Universidad Carlos III de Madrid. His research topics include the application of formal methods, especially model checking, to the verification of business processes and Web Service compositions. Recently, he has also become interested in the Semantic Web and Semantic Web Services. He has worked on several European and Spanish research projects related to the field of Web Services. He has also authored more than 15 papers in national and international journals and workshops/conferences in related fields. He was a temporary research visitor at the IT Innovation Centre at Intel Ireland in 2004, and a visiting scientist under the direction of Prof. Tim Berners-Lee at the Decentralized Information Group of the Massachusetts Institute of Technology in 2006. <jaf@it.uc3m.es>.

Óscar Corcho-García works as a Marie Curie fellow at the University of Manchester. Previously, he has worked at iSOCO as a research manager and at the Ontological Engineering Group of the Universidad Politécnica de Madrid (UPM). He graduated with honours in Computer Science from UPM in 2000, obtained his MSc in Software Engineering from UPM in 2001, and his PhD in Artificial Intelligence in 2004. He received the third Spanish award in Computer Science from the Spanish Government (2001) and a PhD thesis award from the Universidad Politécnica de Madrid (2005). His research activities include the Semantic Grid, the Semantic Web, and ontology engineering. He has participated in several leading EU projects in these areas: OntoGrid, Esperonto, DIP, HOPS, SWWS, Knowledge Web, OntoWeb, and MKBEEM. He has also taken part in the HALO project, funded by Vulcan, Inc. He has published the books "Ontological Engineering" and "A layered declarative approach to ontology translation with knowledge preservation", over 30 journal and conference/ workshop papers, and he reviews papers in several conferences, workshops and journals. He has also been a research visitor at KMI (Open University) and SMI (Stanford University). He chaired the demo/industrial sessions at EKAW2002, co-organized the ISWC2003 and ISWC2004 Workshops on Evaluation of Ontology Tools (EON2003, EON2004) and was the sponsor chair of the ESWC2006 conference. <oscar.corcho@manchester.ac.uk>.

Jorge Cardoso joined the Universidade de Madeira in March 2003. He previously lectured at the University of Georgia (USA) and the Instituto Politécnico de Leiria (Portugal). While at the University of Georgia he formed part of the LSDIS Lab where he performed extensive research on Workflow Management Systems. His current interests include Workflow Quality of Service, Semantic Workflow Composition, Web Services, Web Processes, e-Commerce, and Groupware/CSCW. <jcardoso@uma.pt>.
Services have been crucial to the emergence of such paradigms as Service Oriented Architectures (SOAs).

This special issue contains a number of papers related to the industry and academic world describing the state of the art of Web Services, fundamental approaches and frameworks, work in progress, and the impact of this technology on different types of applications.

The issue kicks off with the paper "Web Services: Introduction and State of The Art", prepared by the issue’s editors with the aim of introducing the world of Web Services. This paper presents a general overview of the implications of Web Services in different industrial and academic areas.

The World Wide Web Consortium (W3C) is the most important standardization consortium in the field of Web technologies. It has developed and standardized a number of key Web technologies such as XML and the latest versions of HTML. The paper "W3C and Web Services Standardization", authored by Martín Álvarez-Espinar from the W3C Spanish Office, explains the mission of the W3C and its important role in the standardization of Web service technologies.

The paper "Integration and Interoperability Experiences in Healthcare", by Antonio García-Landeira, gives an example of how Web Services are applied in the Health sector to integrated Hospital Information Systems. This integration effort follows the basic principles of Service Oriented Architectures (SOAs) and Enterprise Service Buses (ESBs).

The paper "Model-Driven Extra-Functional Property Development for Web Services: a Case Study from the Service and Client Side Perspectives", by Guadalupe Ortiz-Bellot and Juan Hernández-Núñez, proposes a multidisciplinary approach that uses Model Driven Development (MDD), Services Component Architecture (SCA) and WS-Policy to integrate non-functional properties in service models, and the administration of such properties by service clients.

The paper "Semantic Web Services with WSMO", from Holger Lausen, Jos de Bruijn, Uwe Keller and Rubén Lara deals with the role of Semantic Web Services in the formal specification of services, allowing the automation of their localization and use. WSMO provides an infrastructure for the description of Semantic Web Services enabling the specification of services, and supports mediation in order to overcome interoperability problems.

On the subject of Service Level Agreements and Contracts we would highlight the paper "Towards An Automated Trading Process", by Pablo Fernandez-Montes, Manuel Resinas-Arias de Reyna and Rafael Corchuelo-Gil, which focuses on the automatic provision of services based on Service Level Agreements between provider and clients, and describes the activities involved in the service negotiation process and its application in different scenarios.

Service selection is a hot topic in Semantic Web Services. The paper "Towards Semantic Service Selection for B2B Integration", by Andreas Friesen and Kioumars Namiri describes a solution for the dynamic selection of Web services based on the semantic interpretation of the capabilities of the services offered and of the parameters specified at run-time.

The paper "Leveraging E-Marketplace Models for Web Service-Based Application Development", by Abraham Nieva-de la Hidalga, Leping Zhao and Pedro R. Falcone-Sampaio proposes the application of the current e-marketplace model to the development of Web Service-based applications.

We would like to thank the authors for contributing their papers and the editors of Novática and UPGRADE for their help during the edition of this special issue. We hope that this issue will be also helpful for readers of Novática and UPGRADE.

Below we provide some relevant references about the world of Web Services for those readers who wish to go into this area in greater depth.
Useful References on Web Services

The following references, along with those included in the articles this monograph consists of, will help our readers to dig deeper into this field.

Books:

Magazines and Journals:

Conferences:

Websites:
- W3C: <http://www.w3.org/>.

Discussion lists:
- www-ws@w3.org: <http://lists.w3.org/Archives/Public/www-ws/>.
- public-swss-iq@w3.org: <http://lists.w3.org/Archives/Public/public-swss-iq/>.
- soapbuilders: <http://groups.yahoo.com/group/soapbuilders/>.
- rest-discuss: <http://groups.yahoo.com/group/rest-discuss/>.

Glossary of terms:
- ACM: Association for Computing Machinery
- AOP: Aspect Oriented Programming
- API: Application Programming Interface
- ASP: Application Service Provider
- B2B: Business to Business
- B2Bi: Business to Business Integration
- BPM: Business Process Management
- BPI: Business Process Integration
- BPO: Business Process Outsourcing
- CORBA: Common Object Request Broker Architecture
- CRM: Customer Relationship Management
- DCOM: Distributed Component Object Model
- EAI: Enterprise Application Integration
- EDI: Electronic Data Interchange
- ERP: Enterprise Resource Planning
- ESB: Enterprise Services Bus
- ebXML: Electronic Business XML Initiative
- HTTP: Hypertext Transport Protocol
- IEEE: Institute of Electrical and Electronics Engineers
- JMS: Java Message Service
- JSR: Java Specification Requests
- J2EE: Java 2 Platform Enterprise Edition
- J2ME: Java 2 Platform, Micro Edition
- J2SE: Java 2 Platform, Standard Edition
- MDA: Model Driven Architecture
- MOM: Messaging Oriented Middleware
- OGSA: Open Grid Services Architecture
- OGSI: Open Grid Service Infrastructure
- OMG: Object Management Group
- OASIS: Organization for the Advanced Structured Information Standards
- QoS: Quality of Service
- RPC: Remote Procedure Call
- SCM: Supply Chain Management
- SLA: Service Level Agreement
- SOA: Services-Oriented Architecture
- SOAP: Simple Object Access Protocol
- UDDI: Universal Description, Discovery and Integration
- URL: Universal Resource Locator
- VAN: Virtual Area Network
- W3C: World Wide Web Consortium
- WS-BPEL: Business Process Execution Language
- WSDL: Web Services Description Language
- WSRF: Web Services Resource Framework
- WSRP: Web Services for Remote Portlets
- XML: Extensible Markup Language
Web Services: Introduction and State of The Art

Óscar Corcho-Garcia, José Carlos del Arco-Prieto and Jesús Arias-Fisteus

In this paper we provide a brief introduction to Web Services, including the main specifications: SOAP, WSDL and UDDI. We also describe other specifications that complement them and provide solutions to aspects needed to develop service oriented architectures based on Web Services. We also address some of the research issues open, including the semantic description of services, which is one of the issues to which more effort is being devoted currently. Finally, we list the main areas where Web Service technology is being applied successfully in the context of enterprises.

Keywords: B2B, BPM, EAI, Integration, Specifications, Web Services.

1 Introduction

In the first generations of the Web, the Web was seen as a collection of information available either in a static way (static Web documents normally generated manually by persons) or in a dynamic way (dynamic Web documents generated from databases, creating the so-called Deep Web).

Web Service technology allowed lifting the Web to a new level of service, where software applications and components could be accessed and executed using the Web as their medium for transmitting inputs needed for the execution and outputs obtained from it. Specifications (and their corresponding components) like UDDI (Universal Description, Discovery and Integration), WSDL (Web Services Description Language) and SOAP (Simple Object Access Protocol) are the basis for this new type of Web. They will be described in Section 3 of this document.

The use of this set of specifications and their corresponding components is tightly related to the development of the Service Oriented Architecture (SOA) vision, with the idea

Authors

Óscar Corcho-Garcia works as a Marie Curie fellow at the University of Manchester. Previously, he has worked at iSOCO as a research manager and at the Ontological Engineering Group of the Universidad Politécnica de Madrid (UPM). He graduated with honours in Computer Science from UPM in 2000, obtained his MSc in Software Engineering from UPM in 2001, and his PhD in Artificial Intelligence in 2004. He received the third Spanish award in Computer Science from the Spanish Government (2001) and a PhD thesis award from the Universidad Politécnica de Madrid (2005). His research activities include the Semantic Grid, the Semantic Web, and ontology engineering. He has participated in several leading EU projects in these areas: OntoGrid, Esperonto, DIP, HOPS, SWWS, Knowledge Web, OntoWeb, and MKBEEEM. He has also taken part in the HALO project, funded by Vulcan, Inc. He has published the books "Ontological Engineering" and "A layered declarative approach to ontology translation with knowledge preservation", over 30 journal and conference/workshop papers, and he reviews papers in several conferences, workshops and journals. He has also been a research visitor at KMI (Open University) and SMI (Stanford University). He chaired the demo/industrial sessions at EKAW2002, co-organized the ISWC2003 and ISWC2004 Workshops on Evaluation of Ontology Tools (EON2003, EON2004) and was the sponsor chair of the ESWC2006 conference. <oscar.corcho@manchester.ac.uk>.

José Carlos del Arco-Prieto is a graduate in Computer Science from the Universidad de Huelva (1998). Later his professional career began in Tecsidel (1999) where he participated in projects for Telefónica I+D and made analyses of the impact of Web Services on different markets (2002). In 2004 he worked in the Diputacion de Huelva (Huelva Provincial Council) participating in the European project "Regions on Line" (ROL). In 2005 he worked in Cibernos collaborating on the Diraya project led by Indra and participating in the development of proposals of interoperability models and Web Services-based integration. He is currently working for T-Systems, doing technical consultancy work and collaborating in the definition of interoperability models, integration for the public sector, and the design of methodologies for SOA development. He collaborates with the Universidad de Huelva in the promotion of Web 2.0 technologies. He is the father of the first Web Services Latin list (webservices-Latinos), promoter of JSWEB 2005 Workshop, Co-president of JSWEB 2006 Workshop and a member of the steering committee of European SOA & Application Architecture Conference (2007). One of his main interests is the convergence between the worlds of academia and business. <josecarlos.delarco@t-systems.es>.

Jesús Arias-Fisteus works as an assistant professor in the Telematic Engineering Department of the Universidad Carlos III de Madrid. He received his MSc with honours in Telecommunication Engineering in 2001 from the Universidad de Vigo. In 2005, he received his PhD in Communication Technologies from the Universidad Carlos III de Madrid. His research topics include the application of formal methods, especially model checking, to the verification of business processes and Web Service compositions. Recently, he has also become interested in the Semantic Web and Semantic Web Services. He has worked on several European and Spanish research projects related to the field of Web Services. He has also authored more than 15 papers in national and international journals and workshops / conferences in related fields. He was a temporary research visitor at the IT Innovation Centre at Intel Ireland in 2004, and a visiting scientist under the direction of Prof. Tim Berners-Lee at the Decentralized Information Group of the Massachusetts Institute of Technology in 2006. <jaf@it.uc3m.es>.
of "plug-compatible" software components that allow reducing the costs of developing software systems while at the same time allows increasing their capabilities. Though the origin of SOAs cannot be found on Web Service technology, but on previous efforts on the development of distributed systems (from Remote Procedure Calls to CORBA – Common Object Request Broker Architecture), it is true that Web Services have contributed deeply to their success and wide implementation, and constitute a good technology to realise this vision. Section 2 will describe this vision further and will focus on how Web Services contribute to it.

With the emergence and wide deployment of Web Service-based applications in different contexts, new requirements arise. These include how to represent and exchange Web Service metadata, how to ensure the reliability of services, how to deal with security aspects such as authentication and authorisation, how to compose services and coordinate their executions in complex interactions, how to manage and monitor them, how to represent and deal with their state in the case of stateful resources, etc. To address all these requirements, new specifications have been created on top of the basic set of specifications. They will be described in Section 4.

Many challenges still exist in the development of Web Service-based applications. These challenges include service discovery, selection, composition, negotiation, dynamic configuration, invocation, monitoring and recovery. A large amount of research is being devoted to studying them and proposing solutions. Current research is focused on isolated specific aspects (for instance, improving service discovery algorithms using a combination of keywords, input and output datatype similarity, etc.) or on more comprehensive approaches, such as the ones proposed in the context of Semantic Web Services, which aim at augmenting the descriptions of Web Services with formal metadata related to the application domain, the preconditions and postconditions for the service execution, etc. This last approach would permit finding solutions to most of the aforementioned challenges, although with the additional cost of requiring a higher effort for service providers who have to annotate their services.

Other distributed and non-distributed technologies are also adopting Web Services as their means to implement their functionality: this is the case of datasets, which are being made available by Web Services, or of Grid Technologies, which are also adopting (and extending) Web Service specifications and components. All these research challenges will be described in Section 5.

Finally, in Section 6 we will be giving examples of applications based on Web Service technology. Applications range several domains: public administration, financial services, B2B (Business to Business) integration, enterprise application integration, etc.
2 Architecture

Web Service technology is not monolithic. It is defined by means of a set of specifications, each of which is focused on solving different needs of service oriented architectures. Even if the main Information Technology companies have different objectives in mind with respect to Web Service technology, they collaborate in the standardisation of these specifications. As a result, they have converged into a set of widely-adopted basic specifications that have contributed to the success of this approach to application development.

Figure 1, obtained from [1], presents a commonly agreed view of Web Service Architecture, structured along a set of functional layers with pointers to the most relevant specifications.

Now we summarise the role of each layer in the global architecture, and the main characteristics of the corresponding specifications:

- Web Services define a message-based architecture. The transport layer is focused on the set of protocols used to deliver these messages. Web Service specifications allow using any protocol in this layer, though most of them rely on HTTP and HTTPS.
- The Messaging layer defines protocols that specify the format of the exchanged messages, their source and target, the systems that can process them, etc. This layer is the core of the architecture and the two main specifications are SOAP and WS-Addressing.
- The Service description layer is used to express metadata about service capabilities, both from the functional and non-functional points of view. The most important specification in this layer is WSDL.
- The Service discovery layer allows users (either humans or systems) to look for services that can accomplish their goals, according to the metadata stored for them. The most important specification in this layer is UDDI.
- The Quality of Service layer is in charge of non-functional aspects of Web Services, which are relevant to make it possible to interact with them, such as reliable messaging, security, transaction management, etc.
- The Components layer is in charge of the composition of new services from existing services, and of the coordination of interacting services.

3 Basic Web Service Specifications: SOAP, WSDL and UDDI

3.1 SOAP

SOAP is an XML-based protocol used to define the messages to be exchanged in an heterogeneous and decentralised environment, independently of the transport protocol used, the type of communication established, and the rules for the interpretation of messages.

SOAP has been standardised by the W3C consortium [2], [3], and widely accepted by industry.

3.2 WSDL (Web Service Description Language)

WSDL is a language used to describe the public interface of a service, including its public functions, the service location and the way to access it. The aspects related to service invocation, operations and messages are described in an abstract way so that they can be linked later to a network protocol and to a specific message format [4].

WSDL has been also standardised by the W3C consortium, with different versions [5], [6].

3.3 UDDI (Universal Description Discovery and Integration)

UDDI [7] is used to describe an XML-based universal registry that stores different types of information about the Web Services in a system. The UDDI specification includes a set of APIs for the discovery and publication of services in repositories.

4 Other Specifications

Besides the previous three basic specifications, now we describe others that are in the standardisation process and define other aspects related to Service-Oriented Architectures.

4.1 Addressing: WS-Addressing

WS-Addressing allows identifying nodes that exchange messages, independently of the protocol used to transport them, by means of endpoint references (EPRs). This specification is currently under the standardisation process by the W3C, and some documents are already proposed as W3C Recommendations [8][9][10].

4.2 Policies: WS-Policy

WS-Policy defines a general framework to describe and combine, in an abstract way, different types of policies regarding the service access or execution features of the services in different domains. This includes aspects such as security, reliable messaging, transactions, etc. The information provided with WS-Policy complements the WSDL functional descriptions. As with the previous one, this specification is currently under standardisation by the W3C, with some working drafts [11][12].

4.3 Metadata Exchange: WS-MetadataExchange

WS-MetadataExchange [13] defines protocols to allow endpoint references to exchange their metadata. Metadata can include policies, WSDL descriptions, XML Schema datatype descriptions, etc.

4.4 Security: The WS-Security Family

Secure transport protocols like HTTPS are not enough for building secure service-oriented applications, since they usually imply the use of multi-step messages, participation of more than two entities, etc. The WS-Security family of specifications (which includes WS-Trust, WS-SecureConversation and WS-Federation, among others) proposes an interoperable way to combine existing security techniques in an interoperable way. Some of these specifications are OASIS (Organization for the Advanced Struc-
ReliableMessaging (WS-Reliability and WS-ReliableMessaging)

Independently of whether the lower-level protocols are reliable or not, other additional mechanisms have to be used to guarantee a reliable end-to-end message exchange. WS-ReliableMessaging [15] specifies three basic semantics that can be combined: ordered delivery, delivery of each message at least once and delivery of each message at most one. WS-Reliability [16] is similar to WS-ReliableMessaging and has been standardised by OASIS.

Transactions: The WS-Coordination Family

In complex interactions with multiple message exchanges it is difficult to guarantee that the final result of the interaction is coherent, due to the occurrence of errors, unexpected situations, etc. For this reason protocols for the coordination of transactions are being proposed. WS-Coordination specifies a general framework to define such coordination protocols. Among these protocols we can cite WS-AtomicTransaction, for applications that require the use of the classical model for atomic transactions ACID (Atomicity, Consistency, Isolation and Durability), and WS-BusinessActivity, for long-lasting business interactions where the ACID model is not adequate. These specifications are available at [17][18][19].

Composition: WS-BPEL

WS-BPEL (Business Process Execution Language, previously known as BPEL4WS) is a high-level language for defining service composition (aka business processes). Service composition descriptions can be executed and managed automatically by execution engines that are compatible with the language. The current BPEL4WS 1.1 [20] was sent to OASIS for standardisation and will be transformed into the WS-BPEL 2.0 specification.

State: WSRF

The Web Service interfaces aforementioned do not specify how service providers and requesters have to deal with the access to the resources they are wrapping when these resources are stateful. The WS-Resource [23] construct was proposed as a means to expressing this relationship, and the WS-Resource framework (WSRF [24]) is a set of Web Service specifications that define a rendering of the WS-Resource approach in terms of specific message exchanges and related XML definitions: WS-ResourceProperties (WSRF-RP), WS-ResourceLifetime (WSRF-RL), WS-ServiceGroup (WSRF-SG), and WS-BaseFaults (WSRF-BF). The aim of this specification is to allow programmers to declare and implement associations between a Web Service and one or more stateful resources, describing how the resource state is accessible through the Web Service interface and defining related mechanisms concerned with WS-Resource grouping and addressing.

The initial work on WSRF was performed by Globus Alliance and IBM (WSRF was first released in January 2004), as an initial refactoring of the concepts and interfaces developed in the OGSI V1.0 specification [25]. Now it is in the process of standardisation by OASIS.

Semantic Web Services

Current Web Service technologies (SOAP, WSDL and UDDI) operate at a syntactic level. Hence they normally require humans to search for appropriate Web Services to be used in an application and to combine them so that they fulfil the application objectives. Semantic Web Services are aimed at overcoming this human dependency by providing (semi)-automatic means to discover, select, and compose Web Services for an application, giving support as well to service mediation, execution and monitoring. The basis for providing such functionalities is the annotation (also known as markup) of Web Services with machine-understandable content that describes their preconditions and postconditions, their inputs and outputs, etc.

There are currently three leading approaches for the development of Semantic Web Services: WSDL-S, WSMO (Web Service Modeling Ontology) and OWL-S (Ontology Web Language-S). A comparison can be found in [21]. All of them are similar with respect to their design principles and to the types of problems that they aim to solve. However, they differ in the formal languages and models used to describe services, and in how they integrate with the underlying Web Service technology.

WSDL-S [26] uses the extensibility elements of WSDL in order to include semantic descriptions of Web Services. Extension attributes are used to associate WSDL entities with concepts in a semantic model, to handle structural differences between the Web Service schema elements and the semantic concepts, to specify preconditions and effects of each Web Service operation, and to specify a semantic category of the service, which can be seen as an extension to the UDDI registry information. The WSDL-S approach does not make any assumption about the formal language to be used to specify the semantics of Web Services. Hence it can be seen as complementary to the other two approaches.

WSMO [27] and OWL-S [28] propose ontology-based models for describing Web Services. The WSMO model is specified in the WSML (Web Services Meta Language) and can be dealt with by execution platforms like WSMX or IRS-III. WSMO descriptions specify preconditions, postconditions, assumptions and effects of Web Service operations, as well as non-functional properties of the service. The OWL-S model is specified in the OWL language and can be dealt with by the OWL-S virtual machine. Services are described according to their profile (what the service does), model (how to use the service and what happens when the service is used) and grounding (details on how

---

1 http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf
requesters can access the service). These approaches are under standardisation by different standardisation committees (W3C, OASIS). Open research issues in this work are still on the aspects related to automatic discovery, selection and composition of Web Services. There are also contests open for participation for these open issues.

6 Applications

The emergence of Service Oriented Architectures in organisations has spread the use of Web Services as the candidate technology to implement the design principles behind this paradigm. Below we describe the main application areas of Web Services.

6.1 Enterprise Application Integration (EAI)

Web Services have changed the way in which enterprises integrate their legacy applications, departments and information systems. Web Service standardisation is key to make it possible to share information and services with other partners, providers and customers, reducing the software development and maintenance costs and taking advantage of all the development and investments done in the enterprise.

Besides, the importance of Web Services in application integration has grown because of the popularity of Enterprise Service Buses (ESBs).

6.2 Business Process Management (BPM)

BPM solutions are linked to the maintenance of the lifecycle of business process, together with their automation, optimisation and design. Web Services constitute a key technology to achieve an open integration between inter- and intra-organisational processes. Specifications like BPEL have boosted the support for the orchestration of collaborative Web Service based processes.

6.3 Business-to-Business and E-Commerce (B2B and EC)

Collaboration in B2B requires an open and normalised exchange of information between organisations, so that they can converge into a common model and ensure the extension of their business and organisational boundaries to other organisations.

With respect to Electronic Commerce, Web Services are used in portals like Amazon.com, Yahoo.com, or eBay, using APIs based on REST. In the academic environment, analyses are being made about the impact of Web Services in virtual markets, contracts and e-provisioning.

6.4 Public Sector

Web Services are being used in all types of public administrations to offer services to citizens (change of address, tax payment, electronic notifications, certificates, etc.), and in those scenarios where the collaboration between different public administrations is required.

As an example, in the Health sector we can highlight the wide use of Web Services to make it possible for hospital information systems and other public Health-related services to interoperate. Similarly, Web Services are being used in other areas such as tourism, finance and insurance.

6.5 New Business Models

Web Services are being also considered as a gateway to new business models, since they provide an inter-connection of services that was not provided before. One example is the emergence of the concept of "Software as a Service" (SaaS) [22], which represents a new way to distribute software.

Translated by Oscar Corcho-García

References


W3C and Web Services Standardization

*Martin Álvarez-Espinar*

The World Wide Web Consortium (W3C) is a vendor-neutral forum founded in 1994 to develop Web standards and design technologies to ensure that the Web will continue to thrive in the future. This article gives a broad outline of the W3C’s organization and its main goals, and a brief explanation of the long process of standardization, in which anyone interested can be involved. W3C is focused on core standards for the Web and establishes liaisons with other organizations developing complementary technologies. In the dynamic area of Web Services, we can see that there is active collaboration between organizations and the public. W3C’s activity is growing at a rapid rate because Industry has a great deal of interest in solving its technological problems with these interoperability mechanisms. The need for standardization arises when different parties are researching similar topics.

**Keywords:** Recommendation, Semantic Web Services, Standard, SWS, W3C, Web Services.

1 The Consortium

The *World Wide Web Consortium* (W3C) [1] is an international consortium in which Member organizations (over 400 worldwide), a full-time staff (more than sixty researchers and engineers), and the public (there are many ways for people to collaborate with the W3C) work together to develop Web standards.

This consortium, led by its Director and the inventor of the *World Wide Web*, Tim Berners-Lee, is a vendor-neutral forum that designs and develops technologies to ensure that the Web will continue to thrive in a future marked by the growing diversity of people, hardware, and software.

1.1 Mission

W3C’s mission is "To lead the World Wide Web to its full potential by developing protocols and guidelines that ensure long-term growth for the Web".

W3C primarily pursues its mission through the creation of Web standards and guidelines. Since 1994, W3C has published more than ninety standards, known as W3C Recommendations [2]. A W3C Recommendation is the equivalent of a Web standard, and guarantees that a specification is stable, contributes to Web interoperability, and has been reviewed by the W3C Membership, which favours its adoption by the industry.

W3C also engages in education and outreach, develops software, and serves as an open forum for discussion about the Web.

1.2 Goals

The W3C’s main goal is to reconcile users’ increasing expectations with advances in technological capacity; W3C is already laying the foundations for the next generation of the Web. According to *Tim Berners-Lee*, "W3C is where the future of the Web is made. Our Members work together to design and standardize Web technologies that build on its universality, giving the power to communicate, exchange information, and to write effective, dynamic applications—for anyone, anywhere, anytime, using any device". In order to achieve W3C’s mission, a number of goals have been established.

1.2.1 Universal Web

W3C defines the Web as the universe of network-accessible information (available through computers, mobile devices...)

**Author**

*Martin Álvarez-Espinar* has an MSc in Computer Science. He is a Web Standards and Development technical expert at the *W3C Spanish Office*. He has experience in the development of standards-based Web applications. He has worked at the Research and Innovation department of the *Universidad de Oviedo*, Spain. He has been involved in several open-source projects and he has participated in a number of seminars, educational courses, and conferences as an invited speaker. His interests are centred on Web engineering and, more specifically, on Semantic Web, Web Services, Mobility and standardization in general. <mlvarez@w3.org>.

The Spanish Office of the W3C (World Wide Consortium), established in October 2003, is hosted by the CTIC Foundation, a not-for-profit organization that conducts and disseminates applied research in information technologies. It is located in the Technological and Scientific Park of Gijón, Asturias. The promotion of W3C standards is bound to local issues; Web uptake differs from one place to the other and sometimes language can be a major problem and regional issues can be obstacles that the Office helps to overcome by means of educational and outreach actions and the performance and coordination of the translation of technical documentation. The general mission of the Office is to promote the adoption of W3C recommendations among developers, application builders, and standards setters, and to encourage the participation of stakeholder organizations in the creation of future recommendations by joining W3C. The Office is the first local corporate and technical point of contact with the W3C, serving as a two-way communication channel: from the W3C to the Spanish community and from Spanish industry and academic institutions to the W3C. For more detailed information see: <http://www.w3c.es/>.
phones, TV sets, etc.). Today this universe benefits society by enabling new forms of human communication and new opportunities to share knowledge. W3C’s vision of the Web is that of a truly integrated environment which allows for the expression of cultural nuances. Thus, one of W3C’s primary goals is to make these benefits available to all people, whatever their technological limitations (hardware, software, or network infrastructure); whatever disabilities the users may have (visual, intellectual, sensorial, use of interfaces, etc.); and whatever cultural, geographical, or linguistic differences there may be (characters and way of writing, kinds of keyboard, date formats, identifiers, postal codes, etc.).

1.2.2 Accessible Web from Everywhere

Nowadays, there are a great number and a wide variety of mobile web-enabled devices. W3C believes that mobility has an important role to play, because the mobile phone market is extremely dynamic and growth is relentless, resulting in a large number of potential users anywhere in the world.

One reason for the importance of mobility is the fact that, while there is expected to be one computer per family in the future, each individual person will have more than one mobile device. Furthermore, in some developing countries mobile devices are the only way to access the Internet.

1.2.3 Web as A Knowledge Base

The Web is not merely an immense library which people can consult using lexical and syntactical terms. It is also a vast database which, if designed carefully, can allow all resources to be connected by their natural relationships, in a way similar to how concepts are connected in the human brain (a web page can be related to its creator, the creator to his/her friends, the friends to their personal data, their place of work, and so on). Thus, all information on the Web could be related to their represented concepts, regardless of the associated lexical terms. The Semantic Web [3] keeps data accessible to people, as does the traditional Web, but allows machines to process it in order to solve problems that make understanding, searching, reusing, sharing, aggregation, and extension of the information easier.

1.2.4 Web of Trust

The Web has transformed our way of communicating with one another, and it has also modified our social relationships. People now meet on the Web and start communicating, either for private or for business reasons, without having ever physically met. Global commerce has been affected by the web. If the Web is to be a useful medium for transactions, the parties involved must able to trust one another and the Web. Technologies that guarantee trust are being developed in order to address this issue.

Thus, one of the W3C’s current goals is to promote a Web that encourages collaboration and trust among all participants, a Web which provides for accountability, security, and confidentiality.

1.2.5 An Easy-to-Use Web

Web sites are undergoing evolution that focuses on their
ability to facilitate the use of applications and tools based on user experiences. It is intended to obtain richer interfaces which allow the user to interact with the Web in order to arrive at an attractive, intuitive, and easy way to use the Web. W3C is actively working in this new area. It is researching into new technologies that need to be standardized to allow them to be used correctly, without neglecting the rest of the objectives mentioned earlier.

1.3 Inside W3C

1.3.1 The W3C Technology Stack

The Web is an application built on top of the Internet layer and includes several catalogued Technologies based on the area for which they are developed (see Figure 1). W3C is transforming the architecture of the initial Web. The Web is in constant evolution and has undergone a major transformation since its inception (essentially HTML, HyperText Markup Language; URLs, Uniform Resource Identifiers; and HTTP, HyperText Transmission Protocol) into the architecture of tomorrow’s Web.

1.3.2 Activities

W3C’s organization is divided into Activities [4], which are formed by groups: Working Groups (for technical developments), Interest Groups (for more general work), and Coordination Groups (for communication among related groups). These groups are formed by member organization representatives, the W3C Team, and Invited Experts, who produce the bulk of W3C’s results: technical reports, open source software, and services (e.g. validation services).

There are currently sixty groups spread over twenty W3C Activities. In order to facilitate management, the Team organizes W3C Activities and other work into five Domains: Architecture, Interaction, Technology and Society, Ubiquitous Web, and the Web Accessibility Initiative.

2 Standards Development

2.1 W3C Members

Over four hundred Members spread all around the world ensure the power of W3C. These Members belong to various organizations which perform a wide range of activities. They offer their points of view while Recommendations are under development. Members appoint representatives who will be involved in the development of the standards. These representatives form a community of more than six hundred experts who pursue consensus-based goals.

2.2 Emergence of An Idea

Occasionally, the W3C Team organizes Workshops [5] on an interesting area to promote the participation of the Members and the public in new W3C activities. These workshops are organized in order to bring people together to discuss topics of interest to the W3C community. All stakeholders (members and the public) exchange ideas about particular technologies to meet certain needs that reflect Members’ interests and concerns. These Workshops are usually the starting point of the standardization of a technology.

2.3 Working Group Charter

The W3C Working Groups develop WC3 Recommendations. These groups can be formed by Member representatives, Invited Experts, and W3C Team representatives. The Team proposes the creation of a new Working Group and the Members are able to evaluate and give their opinion about the creation of a working group, or even oppose it.

2.4 Standardization Process

Working Group members who develop the specifications are continuously in contact via email or phone, and they meet regularly. As their work progresses, they publish Drafts reporting on their developments in order to share their work so the public can comment on what they are doing and offer their points of view. When the specification is ready, a Working Draft can be published. If it is successful and accepted, the document may progress to Candidate Recommendation status and a Call for Implementations is made. In this Call for Implementations, the developer community is asked use the new technology to make pilot developments so as to prove the viability of this proposed standard. The standard becomes a Proposed Recommendation and after a review and approval process undertaken by Members the Recommendation is then released.

2.5 Organization Liaisons

W3C works in collaboration with other organizations developing standards for the Internet or the Web in general, and in some cases their activities may overlap with W3C activities. It is important to identify the role and domain of the operation of each organization with respect to the Web and W3C to avoid such overlaps and ensure that communication between the two organizations will be smooth and efficient. W3C is focused on core standards for the Web and establishes liaisons with other organizations [6] specialized in specific areas (i.e. OMA - Open Mobile Alliance or ISO - International Standardization Organization), ensuring interoperability between technologies, and avoiding duplicity of efforts, since the standardization process is laborious and slow. In the area of the Web Services, there are important standardization organizations, such as OASIS (Organization for the Advancement of Structured Information Standards) or WS-I (Web Services Interoperability Organization). OASIS, which is also a W3C Member, mainly researches and develops standards for Web Services, and is centred on horizontal technologies which promote interoperability beyond the vertical technologies and provide frameworks for e-business applications in the public sector. OASIS and W3C both preach the virtues of collaborative relationships, which are essential if we are to reduce the risk of developing divergent approaches to the solution of common problems.
3 Standards in Web Services

3.1 Web Services
The original purpose of Web Services in W3C was to meet the needs of industries with specific requirements in their systems. Early in 2000, there was global excitement about Web Services based on the potential for a combination of XML, the Web, SOAP and WSDL specifications, and to-be-defined protocol stacks to address many of the problems these technologies have encountered.

When specifications were first developed, there was already some sound and efficient distributed object systems such as Microsoft’s COM family and the OMG CORBA standard. These systems, adopted by many organizations, had several important constraints. Different technologies did not interoperate and each presented numerous security and administration challenges when deployed over the Internet, and neither quite meet the scalability expectations created by the Web.

SOAP, Simple Object Access Protocol, is an essential protocol for Web Services. This is an example of the evolution of a technology which has been strongly supported by major interested companies. Since the first steps of XML-based protocols, SOAP has been a fundamental technology for e-business. The SOAP 1.2 Recommendation has evolved from the immature 1.1 version, which was released as a W3C Note a few years after the first version.

SOAP 1.0 was published in September 1999 by IETF, Internet Engineering Task Force. During the research and development process, many organizations cooperated with each other in order to achieve their goals.

3.2 Assortment of Technologies
The degree of consensus in the industry on the adoption of the Web Services in its applications has been significant. Ever since its development began in 2000, W3C has provided several fundamental mechanisms to create Web Services based applications, including an extensive range of technologies separated by abstraction levels. These technologies may have a wide range of functions; from the creation and dispatch of messages to the description of their functionalities or the composition of several services so as to behave as a unit. The abovementioned descriptions are required for the search or discovery of the requested services.

Industry’s need for specialized applications, a need based on the requirements of the domain in question, has caused a revolution in the research of new Web Services specific technologies to address concrete issues. Most of the specifications can be easily recognized since the identifier name usually begins by WS (Web Services). Aside from the technologies developed by W3C (SOAP, WS Addressing, WSDL, WS-CDL, WS Policy, etc.), there are many others proposed by various organizations which attempt to address deficiencies in existing technologies for certain issues (i.e. WS-DM for services management; WS-Eventing for event-based messaging; WS-Reliability for sending messages with guarantees; WS-Security for security in message transmission; or WS-HP which is oriented towards presentation; UDDI to discover services; and many more).

3.3 Need for Standardization
W3C pursues the goal of allowing the Web to reach its full potential. The most fundamental Web technologies must be compatible with one another and allow any hardware and software used to access the Web to work together. W3C refers to this goal as “Web interoperability.” By publishing open (non-proprietary) standards for Web languages and protocols, W3C seeks to avoid market fragmentation and thus Web fragmentation. This is applicable to any sector or activity within the Co-ordination. In the Web Services area, due to the diversity of organizations trying to develop their own technologies and solutions, there is a need to control the research and to standardize specifications in order to avoid duplicity of effort, and to ensure full interoperability between technologies.

4 The Future of Web Services

4.1 Semantic Web Services
It is in the expansion of the Web Services where the core of W3C’s work can be seen today. The standard language to define the abstract functionality and the way to use and access a service, WSDL (Web Services Description Language), does not offer the possibility of including any type of semantics in the descriptions. Thus it is possible for there to be two services with similar syntactical descriptions but with a completely different meaning. By leveraging WSDL’s capacity of extension, the W3C Semantic Annotations for WSDL Working Group is developing a generic support to add semantics to classic Web Services. This addition of semantic descriptions can be very useful when a services consumer wants to make a search (or discovery) of a service with a required functionality. If there were more than one service with a similar syntactic description, the most appropriate one will be assigned. We can achieve this by using reasoning and looking for similarities between the semantics of the service description and the semantics of the request.

4.2 Extension of Traditional Web Services
One of the main applications of the semantics to the Web Services is focused on services descriptions. To achieve this, W3C is developing a language that makes it possible to specify semantic information to the Web Services definition. In March 2006, the Semantic Annotations for Web Services Description Language (SAWSDL) Working Group was formed. Its mission was to create a standard that would allow semantics to be added to definitions created using the WSDL language.

Thanks to this initiative, most of the existing Web Services related technologies can be reused with minimum changes to the technology. One example of this is the expression of WSDL in RDF, a framework widely used in the Semantic Web to describe resources.
4.3 Public Collaboration

The purpose of the W3C Semantic Web Services Interest Group [7] is to provide an open forum for W3C Members and non-Members to discuss Web Services topics essentially oriented towards the integration of Semantic Web technology into the ongoing Web Services work at W3C.

In addition to the approach followed by W3C regarding the development of useful technologies in Semantic Web Services, W3C strives to obtain a global point of view focusing on other possible solutions based on new technologies, which may be unconnected with existing specifications in traditional Web Services. These alternative approaches can be proposed within the Interest Group and later these ideas will be evaluated and discussed.

Some examples of the ideas sent to this group are some documents developed by various organizations or assemblies related to Semantic Web Services: WSDL-S [8], for the definition of semantic annotations in Web Services, based on WSDL; Web Ontology Language for Web Services (OWL-S) [9], a proposal from nine organizations to create a language to replace WSDL and other technologies; Web Service Modeling Ontology (WSMO) [10], complements SOAP, WSDL and UDDI; and Semantic Web Services Framework (SWSF) [11].

Thanks to these kinds of groups, W3C Members or the public in general can offer their points of view on possible solutions to specific problems. In the case of the Semantic Web Services Interest Group, various submissions from many different organizations can be evaluated and considered in future activities.

Translated by the author

References
Integration and Interoperability Experiences in Healthcare

Antonio García-Landeira

The Technological environment in Healthcare has usually been defined as a group of information islands. Systems involved have a high degree of specialization in those areas which they support but, in general, they are not oriented to integration and interoperability. In this scenario Business Oriented Applications seem to be a possible solution. Unfortunately, and even when it is commonly accepted that such systems have solved a part of an existing problem, there are a number of reasons which make it impossible to accept them as the best solution. One of those reasons is they unavoidably reside within departmental applications such as Digital Imaging, Pharmacy or Laboratory. Lately, a new approach, which takes advantage of the last technological advance, Service Oriented Architecture and Integration standards in the Healthcare sector, has arisen.

Keywords: DICOM, Enterprise Service Bus, Healthcare, HL7, IHE, Integration, SOA, XML.

1 Introduction

Nowadays, Healthcare Information Systems include a large number of areas or departments. All those areas play a very important part in the daily activity of Healthcare Organizations. The work done at clinical services, Infirmary, Laboratories, Image based diagnostics or Pharmacy is supported by applications specifically designed for their daily management. This fact makes such systems essential tools for the correct performance of each department. In most cases, existing information systems are well designed to support each service workflow, and collect in an exhaustive manner the information needed for their management, or for being exported to other organizational units.

Unfortunately, and due to the large number of specific information systems that coexist in any healthcare organization, it is difficult, if not impossible, to find homogeneous software which supports the full existing functionality. This fact makes very easy to find a large collection of installed applications from different providers and supported with different technologies. Also, it is usual to find solutions from different providers in departments where the same activity is done. The reason is that such applications were acquired and installed just in the moment that they were needed. Due to this fact, it is easy for those applications to be in differing evolutionary stages.

A major part of the information that departmental applications manage is generated and used for the service or area where they are installed. However there is a significant part of that information which must be obtained from other systems, or even sent to other interest points of the healthcare environment. An example of information which is necessary to keep up to the date in any clinical application could be patient identification data, the place where a patient has been assisted and their welfare programme. These data, which are generated at the admission service, must be sent to the different systems involved in the welfare activity (laboratory, pharmacy, infirmary etc.) in order to assure an integrated service. One more example is healthcare professionals’ data and the organizational structure where they are working. This information is necessary in order to manage electronic accreditations, access permissions, etc. A third example of information generated by specific information systems and used by more general purpose systems, is the information related to important clinical information and patient treatment such as laboratory or medical image reports. Such information is generated by departmental applications, but once the result is validated, it must be accessible, from a Clinical Station, to everyone involved with patient care.

The conclusion could be that it is impossible to deploy important corporative projects, e.g. Electronic Patient Record, Digital Image management or Electronic prescription, without a correct mechanism for information exchange, consolidation and publishing. At present, this integration frame hardly exists because the different systems involved work on an independent basis in which information is manually introduced in every system. The most optimistic approach consists of the use of specific interfaces developed and designed for every particular case in order to share common information among different applications. This situation has driven the creation of information islands in

Author

Antonio García-Landeira is Healthcare Technologies Coordinator at Software A.G. Spain Healthcare Area. Since he first joined the company in 1988 he has shared his work in business application development with R&D. This led him to join the Spanish America’s Cup Challenge in 1992. He first worked with Information Technologies applied to Healthcare in 1994 during the development of the HIS CHAMAN. In 2003 he was appointed as Technology Coordinator in charge starting up the Defense Division R&D Centre where a number of R&D projects related to Telemedicine and Digital Image where developed. At present, he is responsible for Technological and Functional Coordination of the different projects carried out in Healthcare Area. <agandeira@softwareag.es>. 
Healthcare Organizations without a coherent or adequate workflow. The problem is still bigger when the objective is to coordinate information across different healthcare organizations or different care levels.

2. A New Technological Paradigm

In this scenario the Services Oriented Architecture (SOA) paradigm appears. Many people consider it as the fourth technological wave and the one which will prevail during the next decade. After the previous three waves, Monolithic applications, Client/Server and Business Oriented Applications, SOA focuses on the creation of collaborative systems using XML [1] based technologies. This emerging paradigm seems to have been created in order to solve the traditional problems in the healthcare environment, in special particular those related to clinical systems. Using SOAs, and focusing on assistance processes, it is easier to build interoperability layers, based on traditional Software Engineering concepts such as reusability or encapsulation, but oriented to a new technological environment which makes possible the integration of components on a global clinical area.

Adoption of this paradigm in any organization drives the generation of services from the components of the existing systems which can be easily used by other systems or services. It is possible to use many different approaches in order to build a SOA, but the most commonly used is Web Services implementation. In this case, SOA is usually called Web Services Architecture (WSA). Using WSA as SOA implementation has a main advantage: It makes possible the use of I*Net protocols in order to communicate between systems, which makes internal and external integration easier. This fact has a special interest when applied in Healthcare, because it implies an easy integration of the different care levels.

But, building SOAs alone is not enough in a healthcare environment. Today, any healthcare system with an integration approach, must take in account healthcare standards, such as HL7 [2], DICOM [3] and IHE [4] which have been created in order to ease the integration of the different healthcare environment components.

HL7, the acronym of Health Level 7, defines a message collection for information exchange in healthcare, from administrative data to clinical reports. Up to HL7 v.2.5, data exchange was done using text files whose information fields are separated using "pipes" and information elements are preceded by "Tags". In v.2.5 the first documents appeared using XML formats as an alternative to the primary format configuration. From v.3.0 information exchange will be done using only XML documents.

DICOM, the acronym of Digital Imaging and Communications in Medicine, is a protocol created in order to integrate Digital Image Information Systems and Equipment, which are called "Modalities" in the standard.

The meaning of IHE is "Integrating the Healthcare Enterprise". Even though it is not a true standard, it has been commonly adopted as a useful reference frame in the healthcare sector. IHE defines a group of integration profiles for each department involved in Healthcare organizations, such as Cardiology, Radiology, etc. Such integration profiles describe healthcare workflows using HL7 and DICOM.

Figure 1: Healthcare Integration Approach
3 Integration Approach

SOA and Healthcare information standards have been taken as reference bases for an Integration Model definition which is being successfully applied in many organizations (see Figure 1). The Model uses an Integration Platform to merge legacy applications with emerging technologies.

Such a Platform has the following characteristics:

- Healthcare Integration Standards based solution (XML, HL7, DICOM e IHE). It creates a common information interchange language for the whole organization which can be used to adapt and combine legacy applications and new acquisitions.
- Agility enhancement in existing systems renewal, avoiding any dependency related to solution providers.
- Cost optimization. Once the integration scenario has been defined and developed, any existing system can be included. As an example, patient demographic data updates are done once and information is forwarded to any application connected to the Platform. In addition, there is a return on investment in such an approach by making it easier to change or adding applications with a similar workflow to any other previously integrated one.
- Simplifies Software specification, its delivery to vendors, and evaluation of solutions, making sure that any offered solution is in agreement with healthcare consolidated standards, which can be checked using the appropriate conformance statements.
- The Platform creates a catalogue or dictionary of supported scenarios. Such a catalogue makes understanding and management of the Platform easier.
- It guarantees that any transmitted or received message is in accordance with the given specifications by means of its tracking mechanism which analyzes the correct implementation of messages and warns if any inconsistency is detected.
- Non intrusive solution with little impact on legacy systems development and existing workflows.
- It concentrates integration effort on a single point, the Integration Platform, isolating the legacy systems from the information exchange problems.
- Platform allows short workflow development time due to its incremental approach.
- Transparency in communications with any information system, irrespective of the development language or the platform it is running on: (Windows, Unix, 4GL, application server, J2EE, .net etc.).
- It concentrates full recovery procedures in case of a system failure.
- It supports an early warning system for every integration element to make problem management easier using a proactive approach.
- The platform supports messages persistence in the event of failure of any component or system involved. As an example, when the Integration Platform receives a delivery error from a destination system, it stores the message and retries the operation later. In this way there is no possibility of information loss.
- Exchanged messages are stored in native XML, which allows the use of XML signature.
- It manages messages delivery to more than one receiving system on a broadcast approach. Delivered messages can be equal to the received message or transformed, even with different content or structure for every destination.
- This solution allows the definition of synchronous, asynchronous and transactional scenarios, as well as automating processing and information routing, using message content or structure based rules.

This Platform is oriented to Systems interoperability, it is not related to its technology, it runs on a variety of platforms and Operating Systems and it is a SOA backbone.
This makes systems and middleware reuse easier, allowing a low cost incremental approach and giving great flexibility to intersystem communication.

Though the main purpose of the Platform is oriented to XML management, it can also handle nonXML HL7 versions, and virtually any information transference format. HL7 was elected as the dialect to use due to its healthcare orientation and the fact that it is a commonly accepted Healthcare "de facto" standard, not through any restriction related to document format.

At the same time, this solution is one further step in the search for a Single Point of Integration, allowing healthcare and administrative processes definition and integration using IHE. In addition, it is possible to describe those business processes which are executed by legacy applications and the flows they generate. This makes possible an Enterprisewide Single Point of Integration.

In relation to security, the Integration Platform supports digital signature and XML encryption which can be used for communications with internal or external systems.

### 4 Underlying Technology

A SOA usually implies a higher level of complexity compared to present architectures because the entire system which results, or at least a part of it, is based on a set of cooperative services which can use, and be used by, other different services in a networked approach. Thus, it is unavoidable to use some tools, such as Integration Metadata Repository (IMR) and Enterprise Service Bus (ESB), in order to guarantee easy SOA management and maintenance. IMRs are used to store the information needed to build and manage the SOA, and ESBs are used to define and manage "relationships" among the different services involved in SOA. Using ESBs, complexity of services implementation and resulting architecture is reduced in a radical manner, because services need just one connection with the ESB in order to talk with any other service, relying on the ESB for integration logic management.

The created Platform is based on an ESB which is the centrepiece that concentrates any XML based information exchange (see Figure 2). Such a Bus handles XML transactions over the I*Net, delivers documents to the appropriate receiving applications and changes back-end message formats and presentation styles such as HTML (Hyper Text Markup Language), PDF (Portable Document Format) and WAP (Wireless Application Protocol), when it is necessary. Any connected application send HL7 messages to the ESB which is in charge of routing, transformation, composition, distributed transaction consolidation, etc. All the information needed to execute such operations can be dynamically managed by the System Administrator using a Management Hub.

Organizations or providers of Healthcare Centres do not need to know or understand any architecture or formats used by their customers. In the same way that printed documents are managed, electronic documents need a unique entrance point to assure their right delivery to the appropriate destination. The ESB used works as an XML communications centralizer capable of deciding where documents will be sent by applying content or structure based rules. These rules can be dynamically updated in order to allow the addition of new applications without changing existing application. Input documents are received via I*Net protocols and routed to their destinations. After being processed, output routing is managed making sure that complete XML transactions are handled in a consistent way.

In the same way, the ESB adapts documents structure and content in order to deliver them with the format expected for the receiving application. Such a transformation is done by means of a high performance engine which uses XSLT (eXtensible Stylesheet Language Transformations) to handle different source and destination formats. Stylesheets can be used, as well, to transform XML documents to a variety of formats such as EDI (Electronic Data Interchange), HTML, Wireless Markup Language (WML) for mobile devices and Adobe PDF among others.

Complex transactions often need the action of multiple systems before the result is returned. The ESB used includes a sophisticated sequencer that can be used to create and update document flows, as well as a document aggregator that combines transaction results from both sides of the sequence. These functions simplify processing of documents and enhances the ESB competitive advantages, preparing it for a faster and more flexible reaction to new business needs.

As to be expected, such an ESB supports HTTP (Hyper Text Transfer Protocol), which means that the addition of organization providers or partners can be done in real time, just by adding the server URL. It is not necessary to install any proprietary communications software in the associated site. In those sites where a firewall is used, the ESB can take advantage of the Web Server security features in order to allow encrypted transactions via HTTPS (Hyper Text Transfer Protocol Secure sockets).

Industry and providers standards such as Simple Object Access Protocol (SOAP), Universal Description, Discovery and Integration protocol (UDDI), Document Object Model (DOM), Web Services Description Language (WSDL) and Simple API for XML (SAX) are supported as well.

### 5 Conclusions

The existing technological scene in Healthcare Information Systems is a natural environment to apply SOA, which seems to have been created in order to solve their integration and interoperability problems. The experience acquired in this field demonstrates that the most productive approach is the definition of an Integration Model in order to collect every specific feature of this given business environment.

Such a Model may be based on Healthcare technological standards such as HL7, DICOM and IHE framework. In the same way that it is necessary to know and understand business requirements in order to build optimal business applications, such an Integration Model is the functional and technological base in an Integration Project. If the model
Web Services

is correct, it can be easily and quickly implemented and deployed using tools such as an ESB. The ESB election is a key aspect for project success, because it must support those specific Healthcare features such as HL7 document management and transformation. At the same time it must provide the most adequate connectors for an environment made of different types of Information Systems which has been developed using different technologies at different stages of evolution.

Translated by the author

References

Glossary of Terms
DICOM: Digital Imaging and Communications in Medicine. Is a protocol created in order to integrate Digital Image Information Systems and Equipment, which are called "Modalities" in the standard.

HHRR: Human Resources.
HL7: Health Level 7. Defines a message collection for information exchange in healthcare, from administrative data to clinical reports.

HIS: Hospital Information System. A component of the Healthcare IT environment which mainly manages administrative, financial and clinical aspects of a hospital.

IHE: Integrating the Healthcare Enterprise. It has been commonly adopted as a useful reference frame in the healthcare sector. IHE defines a group of integration profiles for each department involved in Healthcare organizations, such as Cardiology, Radiology, etc. Such integration profiles describe healthcare workflows using HL7 and DICOM.

PACS: Picture Archiving and Communication System. Computers or networks and specific software dedicated to the storage, retrieval, distribution and presentation of images which are mainly managed with DICOM.

RIS: Radiology Information System. Specific Healthcare information system used by radiology departments to store, manipulate and distribute patient radiological data and imagery. It supports the complete Radiology department workflow and it’s commonly integrated with PACS.
Model-Driven Extra-Functional Property Development for Web Services: A Case Study from The Service and Client Side Perspectives

Guadalupe Ortiz-Bellot and Juan Hernández-Núñez

Currently one of the most promising technologies, Web Services are at the intersection of distributed computing and loosely coupled systems. Although vendors provide multiple platforms for service implementation, service integrators, developers and providers demand approaches for managing service-oriented applications at all stages of development. In this sense, we propose a cross-disciplinary approach in which Model-Driven Development (MDD), Service Component Architecture (SCA) and WS-Policy are assembled in order to integrate extra-functional properties in web service models and manage them from the client side. The initial platform-independent model is later transformed into a platform-specific model from which final code is automatically generated.


1 Introduction
Web Services provide a successful way to communicate distributed applications in a platform-independent and loosely coupled manner. Although development middleware provides a splendid environment for service implementation, methodologies for earlier stages of development, such as the modelling stage, are not provided in a cross-disciplinary scope [9], which, for instance, would enable the automatic model-implementation transformation or the addition of extra-functional elements. At present, academia and industry are beginning to focus on the modelling stage, where it is also pursued to conform to the notions of loose coupling and independence from the platform. Some rising proposals focus on representing services as components and others base the model on WSDL elements; two representative approaches are described below:

To start with, in Service Component Architecture (SCA) services are modelled as components linked to a given interface [2]. The model can be later implemented by using different approaches, thereby permitting adaptation to the customer’s specific needs.

As a second trend, many proposals are emerging in the literature where a Model Driven Architecture (MDA) approach is being applied to web service development [3] [10]. MDA has been proposed to facilitate the programming task for developers by dividing system development into three different phases: a Platform-independent Model (PIM), a Platform-specific Model (PSM) and the final application code.

Let us consider now that we want to provide our modelled services with extra-functional properties\(^1\). It is suggested by SCA that this type of property may be modelled at a different level; the way to do so and to include them in additional stages of development has not been approached as yet. Alternatively, MDA proposals do not consider how extra-functional properties may be included in modelled services. On the other hand, WS-Policy, which lets us describe extra-functional service capabilities by using the XML (eXtensible Markup Language) standard [1], does not determine how properties are to be modelled or implemented, and an additional mechanism would be

\(^1\) Non-functional and Extra-functional are terms which are often used indistinctly.
necessary to integrate property modelling and implementation with their description in service-based systems.

This draws us to the first aim of this paper, which consists of a model-driven methodology to deal with extra-functional properties in web service models. The second goal will be to show how these properties are modelled and handled from the point of view of web service clients.

The rest of the paper is organised as follows: Section 2 presents, firstly, our proposed profile for property modelling at PIM level; secondly it is then applied to the services in a case study and finally it is applied to its client. Then, Section 3 shows the proposed platform-specific metamodels and describes the models obtained from the case study PIM. The automatically generated code is introduced in Section 4. Other related approaches are examined in Section 5 and the main conclusions are presented in Section 6.

2 Extra-Functional Property PIM

In this section we refer to the profile used to model extra-functional properties and exemplify how it is used in a case study PIM. We do not intend to deal with either the service model or its transformation. Our proposal could be used in conjunction with any modelling WSDL-centric one as [3], [4] or in component-oriented proposals, such as [10], since all of them show service interfaces and their operations for the properties to be applied [8].

2.1 The Extra-Functional Property Profile

In order to keep our system loosely coupled when adding extra-functional properties to the model, we propose the profile in Figure 1, whose elements will be explained as follows and are described thoroughly in [8]:

- First of all, we define the abstract stereotype extra-functional property, which will extend operation metaclass or interface metaclass. The extra-functional property provides five attributes: the first one is actionType, which indicates whether the property functionality will be performed before, after or instead of the stereotyped operation’s execution – or if no additional functionality is needed it will have the value none, only possible in the client side. Secondly, the attribute optional will allow us to indicate whether the property is performed optionally, i.e. the client may decide if it is to be applied or not, or compulsorily, i.e. it is applied whenever the operation is invoked. Then, a third attribute, ack, is included: when true it means that it is a well-known property and its functionality code can be generated at a later stage; it will have the value false when only the skeleton code can be generated. Finally, we have two additional attributes, namely policyID and policyDoc. PolicyID contains the name of an existing policy or the name to be assigned to the new policy in the service side; policyDoc allows the developer to reuse an existing policy document. If the attribute value is null then the WS-Policy document could be generated at code generation stage. The policy attachment document would be generated in each case. These are the necessary attributes to define the main characteristics in any property, which may be complemented by the specific property attributes.

- In order to define actionType, an enumeration is provided with four alternative values: before, after, instead or none. In this sense, properties may include new functionality before executing the stereotyped operation, after it or they can even replace the operation’s functionality. Some properties may be included from the client side without the need for any additional functionality, in which case its value would be none.

- When we want to use the profile in a specific case study, we will extend it with the specific properties to be used or we can have a pool of predefined properties, as for instance the one in Figure 2, which will be used in the next subsections.

Figure 1: Extra-functional Property Profile.
2.2 Case Study PIM: The Service Side

Consider a simple case study in which we have a set of services related to the university administrative service (Figure 3). Let us imagine that we want to include some extra-functional properties to the services' model. We have devised four different sample properties, included in Figure 3:

- First of all, a log property, to be applied to all operations offered by the registration service to record received invocations.
- Secondly, a property called detailedInfo, which will be required at the discretion of the client when invoking bringForwardExam in ExamOpportunityService: exam dates and locations can be obtained when changing the semester in which the student is going to take the exam; regularly the change is updated and no additional information is obtained.
- Additionally, invocations to personalData in RegistrationService must be encrypted. In order to enable this functionality the de-encryption stereotype has to be applied to the offered operation.
- Finally, sendPdf from PreregistrationService can be invoked - optionally - with a digital signature. That is why the property digitalSignatureCheck stereotypes the named operation in the service model.

In order to provide all the operations in RegistrationService with Log in the PIM, we have only to stereotype the provided interface with the <<log>> stereotype. Stereotype attributes are normally attached to models as tagged values, but they have also been included as comments in the illustration in order to show their values. In it the attributes for log indicate that the property will be performed when any operation in the interface is executed, since it is a non-optional property; log will be performed after the execution of the named operations; the information will be recorded in logFile; it is a well-known property; policyID is Log_ao4ws and policyDoc is null (omitted for clarity). The function which will be invoked to add the additional functionality is myLogFunction. The remaining property values would be similarly interpreted.

2.3 Case Study PIM: The Client Side

We will have a web client which invokes the previously described service operations. Our next step is to examine each property applied in the service side to check which ones need to be included in our client:

- We can note log is not in Figure 4. This is due to the fact that log is always applied in the service side (it cannot be chosen by the client) and it does not require any additional information from the client side.
- The client selected detailedInfo to be applied, which is why it is stereotyping bringForwardExam. In this sense additional information has to be added to the invocation SOAP message for the property to be applied.
- Invocations to personalData have to be encrypted from the client side. Since it is not an optional property and no additional information has to be included in the SOAP message for its application.
- Finally, there is no evidence of the digitalSignature property, since the client decided not to include it.

3 Extra-functional property PSMs

In this Section the proposed specific metamodels and their corresponding models resulted from the PIM-PSM transformation will be shown.

3.1 PSM Metamodels

Our specific models will be based, first of all, on an aspect-
Web Services

Figure 3: PIM with Extra-functional Properties in The Service Side.
**Figure 4:** PIM with Extra-functional Properties in The Client Side.

**Figure 5:** PSM Metamodels.
Web Services

<table>
<thead>
<tr>
<th>ExamOpportunityIF_bringForwardExam_detailedInfo: AspectClass</th>
</tr>
</thead>
<tbody>
<tr>
<td>- target: string=ExamOpportunityIF_bringForwardExam</td>
</tr>
<tr>
<td>- actionType: ActionType=After</td>
</tr>
<tr>
<td>- optional: boolean=true</td>
</tr>
<tr>
<td>- ack: boolean=false</td>
</tr>
<tr>
<td>+ action(data: HashMap, Email: string): string</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ExamOpportunityIF_bringForwardExam_detailedInfo: Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>- name: string=detailedInfo_ao4ws</td>
</tr>
<tr>
<td>- optional: boolean=true</td>
</tr>
<tr>
<td>- ack: boolean=false</td>
</tr>
<tr>
<td>- targetType: string=Operation</td>
</tr>
<tr>
<td>- targetName: string=bringForwardExam</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>exam_OpportunityIF_bringForwardExam_detailedInfo: SOAPTag</th>
</tr>
</thead>
<tbody>
<tr>
<td>- target: string=bringForwardExam</td>
</tr>
<tr>
<td>- value: string=detailedInfo</td>
</tr>
<tr>
<td>- side: string=service</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Client_examOpportunityIF_bringForwardExam_detailedInfo: SOAP Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>- target: string=bringForwardExam</td>
</tr>
<tr>
<td>- value: string=detailedInfo</td>
</tr>
<tr>
<td>- side: string=client</td>
</tr>
</tbody>
</table>

**Figure 6:** Specific Models Obtained from The Transformation.

oriented [5] approach to specify the property functionality, secondly on what we have called a *soap tags*-based approach, to lay down the necessary elements to be included or checked in the SOAP message header and, finally, a policy-based one for property description. Metamodels are explained and depicted in Figure 5:

- Every `aspectClass` will have an attribute `target` which indicates the method for the property to be applied, a second attribute, `actionType`, which signals when it has to be applied; `ack` indicates whether the property is well-known or not and, finally, an `action` refers to the corresponding functionality. Besides, all additional characteristics from the particular property will be included as attributes.

- New tags are included in the SOAP Header to select – on the client side – or check – on the service side – relevant properties, when optional, or to deliver any other information which may be necessary for property application. Every `SoapTag` element will have an attribute `target` which instructs the method for the property to be applied, a second attribute, `value`, to indicate the tag to be included; finally, `side` indicates whether it is a tag to be included by the client or checked by the service.

- A policy will be generated for each property. The policy element will contain the policy `name`, whether the property is `optional`, globally-applied or domain-specific (`ack`); `targetType` indicates whether the policy is to be applied to a `portType` or an `operation` and `targetName` gives the name of the latter.

### 3.2 PSM Models: Service and Client Side

Once the transformation rules are applied to the PIM, the specific models containing information on the properties are obtained. Due to space restrictions only one characteristic ele-
ment from each metamodel has been described; we have chosen detailedInfo for the description, as shown in Figure 6:

- An aspect, examOpportunity_bringForwardExam_detailedInfo, will be generated for detailedInfo in the service side; its attributes target and actionType will have the values examOpportunity.bringForwardExam and instead, respectively. ack will be false and myAccuracyFunction action will also be included with the corresponding parameters.

- Regarding the policy element, its name will be detailedInfo_ao4ws; its optional value will be true. Finally, for policyAttachment, targetType will be operation and targetName bringForwardExam.

- Being optional, we ought to include code whose function is to check whether the detailedInfo has been selected: the corresponding SOAP Tag target will be bringForwardExam, its value detailedInfo and side service.

- An additional SOAP Tag element is necessary in the client for property selection, so that target will be bringForwardExam, its value detailedInfo and side client.

### 4 Extra-Functional Property Generated Code

Transformation rules will generate skeleton code for the three metamodel elements. However, in the case of well-known or user-defined properties, a repository with specific code may be maintained to generate additional code for the three of them. These cases are the ones in which ack is true.

AspectJ has been chosen for the implementation of the property’s functionality in the service side and, when necessary, for additional functionality in the client side, as a result properties remaining well modularised and decoupled from implemented applications, as demonstrated in [7]. In this sense, an AspectJ aspect will be generated for each aspect class in our model. AspectJ pointcuts will be determined by the execution of the target element. Regarding the advice, depending on the actionType attribute value, before, after or instead, the advice type will be before, after or around, respectively.

Concerning property selection, Java code will be generated to check if SOAP tags are included in the SOAP message (service side), or the tags themselves are included in the invocation message (client side).

With regard to description, it is proposed to generate the WS-Policy and WS-PolicyAttachment documents for each property, which are integrated with the aspect-oriented generated properties as explained in [7]. In this sense, the policy is attached to the element stereotyped in the PIM model.

Figure 7 shows the code generated for detailedInfo both in the service and client sides.

### 5 Related Work

Regarding web service modeling the research presented by J. Bezivin et al. [3] is worth a special mention; in it this topic is covered in different levels, using Java and JWSDP implementations in the end. It is also worth mentioning the paper from M. Smith et al. [10], where a model-driven development is proposed for Grid Applications based on the use of Web Services. Our work differs from these in the sense that ours provides the possibility of adding extra-functional properties to the services and is not oriented to the service modeling.
itself; thus it could be considered as complementary to them.

Besides, we can find semantic proposals as the one from D. Fensel et al. [6], where extra-functional properties are modelled as pre and post conditions in an ontology description; however they do not follow the UML standard nor is their proposal compatible with Ws-policy descriptions, both facts regarded as essential for integrating properties in future service models.

On the other hand, plenty of literature can be found on model-driven development for web service compositions (for instance [4]); our proposal aims at providing support for extra-functional properties in isolated or composed services, and therefore could be complementary to the named proposals.

6 Conclusions

This paper has provided a model-driven approach to including extra-functional capabilities in the development of web services and their clients in a loosely coupled manner. Properties are included in the PIM by using a UML profile. Additionally, the initial PIM has been converted into three specific models which conform to three provided metamodels, based on SOAP tags information, aspect-oriented elements and policy-based elements. Moreover, all the code related to extra-functional property implementation is automatically generated by AspectJ, code related to description by WS-Policy, and code related to selection by Java.

7. Acknowledgments

This work has been developed thanks to the support of MEC under contract TIN2005-09405-C02-02.

Translated by the authors

References


Towards an Automated Trading Process

Pablo Fernandez-Montes, Manuel Resinas-Arias de Reyna and Rafael Corchuelo-Gil

Many software companies are using SOA (Service-oriented Architectures) as the cornerstone of their business activities. In this context, automated service provision based on the creation of service agreements is gaining importance in both cross-organizational and intra-organizational scenarios because they provide important benefits to both the service consumer and the service provider. On the one hand, service agreements provide consumers with guarantees about how a service will be provided. On the other hand, service agreements allow the providers to deploy automated service provision based on the agreements they have made with their customers. Service agreements are a prominent research field in both academia and industry. In this article, we focus on the service trading process, which is the process of locating, selecting, negotiating, and creating service agreements. This process can be applied to a variety of scenarios and therefore their requirements are also very different.

Keywords: Automated Negotiation, Automated Provision of Services, Service Agreements, Service Trading, SLAs.

1 Introduction

As SOA (Service-oriented Architectures) has evolved into a mature paradigm, new challenges have appeared on the horizon. The current standardized stack of protocols make up a solid infrastructure for interoperability. However, in order to meet new business possibilities an extension is needed. In particular, the automatic provision of services is emerging as a promising field that could lead to a new generation of enterprises that adapt "on demand" to rapid changes in their business environment.

In this article, we address a core element of the automatic provision of services: the creation of agreements that describe the rights and obligations of the service consumer and the service provider during the transaction. The terms of the agreement could refer to either functional features (such as the interface of the service) or non-functional features (such as the amount that should be paid for the service). The agreement should also be considered as a first class element of the IT infrastructure of both service consumer and provider organizations.

Generally speaking, service orientation in software development would seem to provide the perfect scenario for short term agreements that can be dynamically created to meet an enterprise’s business needs at any given moment (e.g. the cheapest service provider). Automating the creation and management of agreements, so that the human participation in the process is reduced to a minimum, brings benefits such as cutting the cost of reaching an agreement, increasing the speed of the contracting process, and allowing providers to deploy an automated provision of services based on their agreements with customers, leading to a better rationalization of the usage of its resources. Leading on from the above, from the customer’s point of view it would be interesting to be able to negotiate the agreement terms in order to obtain better offers. Such a negotiation process could also be used by the provider to fine tune the final agreement and make concessions or restrictions in order to optimize the current usage level of its resources. Additionally, given that such offers would be closely bound to the resource status at each moment, the decision making infrastructure should also take this information into account as a first level element before entering into new commitments with a customer. However, at this point, it is important to highlight that automation should be understood within the restricted context of the business world in which certain critical strategic decisions can never completely avoid human intervention.

The automatic creation of agreements requires the combined involvement of several research fields to create a holistic approach. This means, among other things: formal lan-

Authors

Pablo Fernandez-Montes is a Lecturer in Computer Science with the Department of Computer Languages and Systems of the Universidad de Sevilla (Spain) as part of the Research Group on Distributed Systems. He is also a PhD candidate whose research is focused on automated contract management and service level agreements. <pablofm@us.es>.

Manuel Resinas-Arias de Reyna is a Lecturer in the Department of Computer Languages and Systems of the Universidad de Sevilla. In 2004, he obtained an MSc in Computer Science. He is a member of The Distributed Group at the Universidad de Sevilla and he is now working on his PhD. His main research interest is the automated negotiation of service agreements in service-oriented scenarios. <resinas@us.es>.

Rafael Corchuelo-Gil is a Reader in Computer Science who is with the Department of Computer Languages and Systems of the Universidad de Sevilla, Spain. He received his PhD from this University and he has led its Research Group on Distributed Systems since 1997. His main research interests include instrumentation techniques to support the multiparty interaction model and knowledge extraction. <corchu@us.es>.
guages for the expression of requests, offers, and agreements; the search and discovery of potential partners; the evaluation of agreement proposals; decision making algorithms and strategies; provision capacity estimation; agreement negotiation; or agreement signing.

The steps of an automated contracting process have already been identified in the relevant literature [6]: The first step involves the creation of the offer by the service provider and the analysis of its functional and non-functional requirements by the consumer. This step may be considered as a preliminary stage of the contracting process. In the contracting process itself, the first step (the information phase) is to match service providers with potential consumers and vice versa. In the negotiation phase, a service agreement is created between service consumer and service provider. During the deployment phase, both service provider and service consumer set up a deployment plan to meet all the terms established in the service agreement. Finally, the fulfillment phase involves the fulfillment of the obligations established in the agreement and the monitoring of the entire process in order to ensure that both parties observe the agreement properly.

In this article, we focus on the establishment of the agreement as a subprocess of the more general contracting process. Specifically, this subprocess can be broken down into four different stages: the location, selection, negotiation, and creation of the agreement. We refer to this as the service trading process. In the rest of the paper we outline the various challenges involved in each stage of the service trading process and review the most important proposals.

2 Trading Process

The complexity of the trading process lies in the fact that it can vary greatly depending on the scenario: how the parties (service consumers and service providers) discover each other or how each party negotiates an agreement offer. Also, time constraints may vary from one scenario to another depending on the type of parties involved: In some scenarios with short fixed timeframes in which to reach an agreement, it may be necessary to carry out a quick and efficient (but not optimal) selection of the agreement from a set of proposals; on the other hand, with more relaxed timeframes the trading process can be more focused on developing complex negotiations to achieve an optimal agreement.

To address the variable nature of the trading process, we propose the Trading protocol concept as a means of orchestrating the various elements in a system while switching the choreography of the process between different systems to ensure an interaction that respects the trading process requirements of all parties. In order to clarify the idea of trading protocol perhaps we should look at a real-life example: A public auction in which an institution is looking for a service provider and devises a trading protocol that consists of the following stages: the announcement of the auction, a deadline for the submission of proposals, a pe-
riod of resolution and, finally, the communication of results.
At an abstract level, the trading protocol is defined as a set of stages (e.g. "announcement, proposal submission, negotiation, resolution, etc...") cross-linked according to some time constraints and bounded to some choreographies. In general, these elements are expressed in two sets of constraints:

- A set of potential choreographies for each stage. In some situations, this means a list of optional choreographies but, in other cases, a unique choreography can be established at a given stage. An example of this type of constraint could be "In the Negotiation stage, auction or bargaining protocols are allowed" (i.e. choreographies implementing these protocols).
- Time restrictions specify a set of constraints about the life-cycle of the trading process. This restrictions could vary from simple fixed temporal points (e.g. "The trading process should end by March 14 at 14h") or can be complex relationships between the durations of some stages (e.g. "The information stage should start in the middle of the discovery stage").

From a structural point of view, the whole trading process (see Figure 1) is composed of six subprocesses (depicted as squares) that interact to reach a specific agreement. The fulfilment of the set of processes make up trading systems (depicted as rounded squares); at an abstract level, these systems represent different parties in the trading process: i.e. some of the trading systems would be acting on behalf of service consumers while others would act on behalf of the service provider. It is important to stress that this abstraction does not imply a specific deployment in a possible implementation of the concrete system but only creates a separation of concerns at a conceptual level.

Following, we identify the behaviour of the whole process (Figure 1) taking into account the different stages involved:

1. The main goal of the trading management process is to create and specify the trading protocol that will regulate how the whole process is carried out; that is, it establishes the timing for when to start the search for parties, when to submit offers, when to wait for responses, when a negotiation must start, or when a binding offer can be sent. Thus, this process focuses on the global behaviour from a temporal point of view. Its goal is the coordination of the rest of the processes so as to implement a trading protocol.

2. The discovery process is concerned with locating potential parties requesting (or supplying) a service that other party provides (or needs). Complementarily, this process is responsible for accessing the market and disseminating the events generated by the internal active trading processes in the system. In this context, the market should be seen both as an abstract concept that includes the set of organizations looking for business relationships, and the infrastructure that acts as a facilitator for the communication between them. In [12], an analysis of the discovery problem for web services is discussed. Finally it is important to point out that the discovery process can locate potential providers or consumers according to a number of functional or non-functional requirements.

3. Discovered candidates are then passed to the information process in order to gather detailed information about the characteristics and preferences of each potential party. The aim of this process is to manage public information about user preferences and the potential candidates found during the discovery process. The amount and type of information collected from each candidate may be different; however, at a conceptual level the information should include, as a minimum, the publicly available characteristics of the service requested/supplied. In addition, some information can be harvested from external sources (e.g. information about the reputation of the candidate).

4. The data harvested in the information process is used by the selection process to create and select a set of promising agreement proposals with other parties. The aim of this process is to choose a set of candidate parties with whom a negotiation process can be started or to whom an agreement proposal can be submitted. The selection starts with a set of information about potential parties from several sources: information provided during the information process after an active search, agreement proposals received from other parties. In [3] a selection based on constraint programming is described.

5. The instructions about proposal handling are delegated to the agreement making process which is responsible for actually negotiating or proposing the agreement to the other party and for creating and signing the final agreement. Thus, the aim of agreement making is to provide a mechanism to create agreements, possibly through an automated negotiation process, that are acceptable to all parties involved. The result of the process is therefore an agreement that specifies the terms under which the service is to be executed. [10] shows a conceptual framework for automated negotiations of agreements. An example of a simple agreement making protocol is described in [1].

<table>
<thead>
<tr>
<th>Discovery</th>
<th>Information</th>
<th>Agreement Making</th>
<th>Trading</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDDI</td>
<td>WS-MetadataExchange</td>
<td>WS-Agreement</td>
<td>WS-CDL</td>
</tr>
<tr>
<td>WS-Notification</td>
<td>WS-InspectionLanguage</td>
<td>FIPA Protocols</td>
<td>BPEL</td>
</tr>
<tr>
<td>WS-Addressing</td>
<td>WS-Agreement</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Related Standards.
6. During this procedure, the agreement making process interacts with the binding process by asking for approval to make or accept binding offers. The main concern is to determine when a binding offer must be submitted and whether a binding offer that has been received should be accepted. This process must also establish when these decisions are to be made. For example, one option is to make the decision as the offers are received; another possibility is to make the decisions at some specific points in time that have been previously decided. These points may be dynamically selected, depending on changing conditions of the environment such as the frequency of arrival of offers, or statically determined based on time constraints imposed by the trading protocol, or a combination of the two. An interesting infrastructure can be found in [7] that carries out decision-making processes based on the agreements available and the amount of resources available to the service provider.

3 Architectures and Technologies

A number of abstract architectures have been proposed [2][4][5] which carry out some subset of the stages in the service trading process. An analysis [9] of the trading process support in these architectures leads us to several conclusions: (i) the discovery process is well supported by the majority of architectures, most of which provide for knowledge adaptation; (ii) these architectures do not generally include elements to support advanced decision-making; (iii) there is little support for the advanced features of service trading such as the decommitment of agreements or trading protocols. Due to these shortcomings, some complex service trading scenarios cannot be fully automated.

Several standards have emerged to enrich the basic web service stack. Table 1 shows the standards that can be used in the various subprocesses.

With regard to the discovery process, there are three specifications that can be used to implement its requirements: (i) UDDI (Universal Description, Discovery and Integration) [13] can be used as a flexible repository that can be used to store the access points of elements and the taxonomies used by the discovery process. (ii) WS-Notification can be used to subscribe and broker notification events. (iii) Lastly, WS-Adressing [13] provides a specification of the references/locations of web services by means of a standardization of the concept of endpoint references.

There are a number of standards that deal with the exchange of service descriptions, from both a functional and a non-functional point of view and they can be used in the implementation of the information process. For instance: WS-MetadataExchange [13] or WS-InspectionLanguage. Alternatively, WS-Agreement [1] uses a template-driven procedure, and those templates can be seen as a mean of expressing the preferences of a given party.

The most significant specification that covers most aspects included in the agreement making process is WS-Agreement [1]. It allows us to specify the structure of an agreement document, so that it must be used together with one or several domain-specific vocabularies to give the proper semantics to the terms of the agreement. Furthermore, it defines a protocol and a web service-based interface to create, represent, and allow the monitoring of agreements.

However, WS-Agreement only defines a take-it-or-leave-it protocol. To use more complex negotiation protocols, other specifications must be implemented. For instance, WS-AgreementNegotiation, which builds on WS-Agreement and specifies a bilateral negotiation protocol, or the negotiation protocols defined by FIPA (Foundation for Intelligent Physical Agents).

With regard to the trading management process different approaches are possible depending on the complexity of the trading protocol used. For complex co-ordinations, there are workflow standards such as BPEL (Business Process Execution Language) [1] or choreography languages such as WS-CDL (Web Services Choreography Description Language). For simple cases, an alternative to implement Trading Protocols would be the specification of ad-hoc elements in the concrete architecture built upon the conceptual framework.

4 Conclusions

In recent years the software industry has been moving from a vision of software as a product to a vision of software as a service. This change has become even more apparent with the appearance and popularization of SOA and web services.

This new vision opens up new possibilities and challenges for the software industry. In this article, we focus on the automated provision of services and, more particularly, we centre on the trading process, that is, the process of establishing a service agreement between two parties. The trading process is as a core element in the development of automated service provision.

As we have discussed, the trading process is complex and involves many different interactions between the parties involved. To deal with this complexity, we have broken it down into several subprocesses: discovery, information, selection, agreement making, binding, and trading. Then, we have analysed the main challenges posed by each subprocess and we have described different approaches and technologies that have been developed to solve each problem.

In addition, when reaching an agreement, there are explicit temporal constraints that regulate how the subprocesses that compose the trading process must be coordinated. These constraints are defined in a trading protocol, as we have discussed previously. From this analysis we conclude that it is necessary to integrate several different approaches and technologies in order to build a system capable of handling the entire trading process. Today, this integration can be carried out in an ad-hoc way. However, there are a number of different trading scenarios, and each scenario has different requirements [9]. There is therefore no general solution for each problem, but each solution de-
Depends on the specific trading scenario in question. In addition, the time constraints of the trading process create a great many difficulties in the coordination of the various parts of the trading system.

Consequently, the idea of developing an integration framework to integrate and coordinate all these partial solutions is very appealing, and forms the basis of our current research [11].

The main goal of the framework is to provide infrastructure services which are common to several trading scenarios and that enable the integration and coordination of different proposals and technologies. This will enable the parts of the framework that may be useful for different scenarios (e.g. the discovery of potential service providers) to be reused. Furthermore, this framework would also facilitate the application in real scenarios of research proposals that deal with parts of the trading process, because it would allow us to focus on the specific problem only, leaving all other details to the framework.

Translated by the authors

References

Semantic Web Services with WSMO

Holger Lausen, Jos de Bruijn, Uwe Keller and Rubén Lara

The integration of applications is one of the main challenges when building IT solutions. Integration is often achieved using costly customized solutions for every pair of applications. Web service technologies are a set of standards that allow software interfaces to be defined using XML (eXtensible Markup Language) as the message format and the Internet infrastructure for message transport. While lowering costs, Web service technologies by themselves do not ensure that two businesses use the same data structures or business protocols, neither do they provide the means to resolve potential conflicts. The lack of formal descriptions of services offered by organizations hampers automation in the location and usage of services required to perform a given business activity. Semantic Web Services enable the formal specification of services, allowing their automated, goal-driven location and usage. The Web Service Modelling Ontology (WSMO) provides a framework for the description of Semantic Web services which enables seamless integration through formal descriptions, maximal decoupling of components, and strong mediation support.

Keywords: Semantic Web Services, Service Composition, Service Discovery, Service Invocation, Service Negotiation, Web Service Modelling Ontology.

1 Introduction

One of the main challenges for the IT industry today is the integration of applications. Traditionally, integration needs mainly occurred within one enterprise between different systems such as human resources and accounting. Nowadays integration crosses the boundaries of different enterprises. Typically integration is achieved using costly customized solutions for every pair of applications and every pair of businesses. This requires an enterprise to invest in custom infrastructure for each new business partner, not to mention the effort required on a human level to agree upon data formats and interaction protocols. For these reasons,
the degree of re-usability of current integration solutions is remarkably low [3].

Web service technologies [2] are a set of standards that allow software interfaces to be defined using XML (eXtensible Markup Language) as the message format and the internet infrastructure for message transport. Many vendors have adopted Web services, so a variety of tools for various platforms is available. Web Services have the potential of reducing the cost of integrating applications because no custom communication lines need to be built, no proprietary messaging protocols need to be implemented, and no proprietary descriptions of how to communicate with applications and business partners need to be implemented and interpreted. However, Web services technologies by themselves do not ensure that two businesses use the same data structures or business protocols and they do not provide the means to resolve such conflicts.

While Web service technologies focus on the technical description of interface, the Service-oriented Architecture (SOA) paradigm provides guidelines on how to design systems, so that the integration of new components and the rearrangement of existing components can be simplified. Essentially, the SOA paradigm prescribes a loose coupling between components. The selection of a particular service should not be limited to the design time of a system, but the architecture enables services to be selected at runtime. Thereby services can be distributed with an explicit network component (e.g. Internet infrastructure). Web service technologies can be used to implement an SOA. However, Web service descriptions in WSDL are of limited expressivity. The consumer of a service has to rely on a human-language description of the Web service to decide whether the Web service offers the desired functionality. Furthermore, there is no way to cooperate if both business partners have different interaction styles or use different interface definitions, however, the fact is providers do not use standardized interfaces, but have variations in the interfaces even if they offer similar functionality. Generally speaking, we can observe three major types of differences which hamper integration [5]:

1. Differences in Vocabulary i.e. different data formats for a similar data entity such as an address.
2. Differences in Protocols which are different styles of interaction, e.g. a synchronous reply to a purchase order (like for a typical online shop) versus the multiple asynchronous replies that occur in a RosettaNet1 Purchase Order.
3. Different businesses may have differences in business processes running inside their organization, e.g. one business sells articles at fixed prices, whereas the other operates its business according to an auction model where multiple orders are gathered and only the most profitable are chosen.

A higher degree of automation in the location and use of Web services can be achieved by adding explicit semantics to Web service descriptions. Such semantically enriched descriptions are usually referred to as Semantic Web services [7], and they are expected to enable businesses to dynamically locate partners which provide particular services, and to facilitate (semi-)automated cooperation with them. Semantic descriptions add machine-processable semantics to data. The computer can "understand" the information and therefore process it on behalf of the human user. The Web Service Modelling Ontology provides a conceptual model how to formalize the semantics of Web services.

2 Limitations of Current Technologies

In a software-oriented architecture a number of central tasks have been identified. These include the discovery of new services, the negotiation of specific business terms with a given service, as well as the composition and invocation of atomic services to achieve a more complex functionality. This list of tasks is not exhaustive; in fact there are many more tasks. However, we will use these to illustrate the limitations of current technologies and later to show how WSMO (Web Service Modelling Ontology) can be used to overcome some of these limitations.

Currently the discovery of new services can be done using keyword based retrieval or using predefined taxonomies. While the former lacks the necessary precision for automating the process, the latter is inflexible since the taxonomy is maintained centrally and each change or addition requires a lengthy standardization process. Alternatively customers can try to find providers based on a given interface definition, however, the fact is providers do not use standardized interfaces, but have variations in the interfaces even if they offer similar functionality. Generally speaking, we can observe three major types of differences which hamper integration [5]:

1. Differences in Vocabulary i.e. different data formats for a similar data entity such as an address.
2. Differences in Protocols which are different styles of interaction, e.g. a synchronous reply to a purchase order (like for a typical online shop) versus the multiple asynchronous replies that occur in a RosettaNet1 Purchase Order.
3. Different businesses may have differences in business processes running inside their organization, e.g. one business sells articles at fixed prices, whereas the other operates its business according to an auction model where multiple orders are gathered and only the most profitable are chosen.

A higher degree of automation in the location and use of Web services can be achieved by adding explicit semantics to Web service descriptions. Such semantically enriched descriptions are usually referred to as Semantic Web services [7], and they are expected to enable businesses to dynamically locate partners which provide particular services, and to facilitate (semi-)automated cooperation with them. Semantic descriptions add machine-processable semantics to data. The computer can "understand" the information and therefore process it on behalf of the human user. The Web Service Modelling Ontology provides a conceptual model how to formalize the semantics of Web services.

3 Web Service Modelling Ontology

The Web Service Modelling Ontology (WSMO) [6] describes all relevant aspects related to services that are accessible through a Web service interface. Its ultimate goal is to enable the (total or partial) automation of the tasks (e.g. discovery, selection, composition, mediation, execution, and monitoring) involved in both intra- and inter-enterprise integration of Web services. WSMO has its conceptual basis in the Web Service Modelling Framework (WSMF) [5]; it refines its concepts and formalizes them as

![Figure 1: An Ontology Example to Illustrate how They Can Aid Integration.](image-url)

1 http://www.rosettanet.org/
ontology. For this purpose, WSMO defines four basic modelling elements, namely:

**Ontologies** [4] provide formal and explicit specifications of the vocabularies used by the other modelling elements. Ontologies form the backbone of the Semantic Web; they allow machine understanding of information through the links between the information resources and the terms in the ontologies. Furthermore, ontologies facilitate interoperability between information resources through links to the same ontology or links between ontologies. A key feature of ontologies is that, through formal, real-world semantics and consensual terminologies, they interweave human and machine understanding of symbols. These symbols, also called terms and relations, can be interpreted by both humans and machines. The meaning for a human is represented by the term itself, which is usually a word in natural language, and by the semantic relationships between terms. In Figure 1 we give a brief example that illustrates how ontologies can aid integration.

In that example we formalize aspects of commonly used product codes, i.e. the EAN (European Product Number) and the UPC (Universal Product Code). First we establish that both are product codes, secondly we declare the concept of a bar code scanner and one of its properties (i.e. that it can read a certain product code). In fact it turns out that every scanner that can process EAN product codes can also process UPC bar codes. This knowledge has been formalized in the axiom at the bottom of the example. Once declared, this knowledge can be used, e.g. to determine during a purchase order process that, although one expects an article with EAN code, one with UPC can be accepted as well.

**Web service** descriptions document and formalize services provided by businesses. As illustrated in Figure 2, different elements describe various aspects. The value that the service can provide is captured by its capability. Within the capability one can declare for example that a particular service sells computer parts of certain brands. Non-functional properties can be used to describe aspects such as average response time or the encryption protocol used for the transaction. WSMO provides an extensible list of such properties that are based on common meta data standards such as Dublin Core [8].

The choreography describes how to interact with a service provider to request the actual performance of a service, or to negotiate aspects of its provision. A choreography defines the sequence in which different activities have to be performed; for example that for a purchase order first an order has to be created, then a number of line items can be added, and only after closing the order a confirmation about the availability will be returned.

The implementation that provides the service is transparent to us and not of interest for the description. We are concerned mainly with its style of interaction and only under certain circumstances with the services that are used to provide the value described in the capability; i.e. the way a service makes use of other services can be specified in the orchestration.

**Goals** describe the objectives when searching a Web service. They describe aspects related to the users desires with respect to the requested functionality. Goals are structured similarly to Web services and like all other elements of WSMO they use terms from ontologies in order to define all relevant aspects. Note that WSMO completely decouples the objectives that a requester has; i.e. his/her goal, from the services that can actually fulfill such a goal.

**Mediators** have a central role in WSMO, they enable differences in vocabularies, protocols and processes which may arise during the dynamic cooperation between services to be bridged. Mediators (cf. connectors in software architecture [1]) resolve such differences and enable seamless integration, overcoming heterogeneity in vocabularies, protocols, and processes. For each type of mediation, WSMO introduces different mediators, e.g. OO-Mediators (OO stands for *ontology to ontology*) resolves mismatches between vocabularies, and WW-Mediators focus on the mediation between different protocols.

**4 Usage of WSMO**

Earlier we outlined the upcoming paradigm of service-oriented architecture. The loose coupling between components enables the dynamic configuration of more complex functionality built from basic services. However with current Web service technologies the tasks involved mainly
have to be performed manually. The **different components** of WSMO can help to increase the level of automation through their formal character and the use of ontologies that interweave human understanding and machine processing.

**Discovery:** Before a service can be used in a distributed application, it must first be located. A service capabilities expresses the value a service provides. Unlike central taxonomies, the vocabulary to describe capabilities is formalized via decentralized ontologies. Interconnections in the form of logical axioms enable a given terminology to be extended and refined.

**Negotiation:** Whenever a suitable provider has been determined to serve a certain goal, it is necessary to negotiate a specific service. This includes the establishment of trust policies, the determination of payment method, etc. For the purpose of automating this task, it is important that a WSMO description not only specifies the functionality of a service (capability), but also includes non-functional aspects such as the price of a service.

**Composition:** In cases where a particular goal cannot be achieved by means of a single Web service, semantic descriptions can help to determine a promising combination of multiple Web services to achieve the requested functionality. A clever selection of services and lightweight planning or scheduling can ease the pain of manual integration. Composition requires not only the semantic annotation of the overall capabilities of a service, but also a behavioural description of how to interact with the Web service (choreography) in order to achieve a certain functionality. A composition that has been generated can be described using WSMO orchestration.

**Invocation:** After (a combination of) services have been selected, the final step is to invoke them. To this end, possible input and output values need to be extracted from the semantic goal description and adapted to the message formats and communication protocols that are expressed in the WSMO choreography.

We have outlined how the different components in WSMO can be used generally within the integration process. Let us now take a closer look into the different levels of abstraction in the descriptions that are required in the different steps. We go on to provide a proper conceptual grounding by acknowledging the difference between service discovery (finding a suitable provider in general) and Web service discovery (determining a specific service instance and the precise parameter of an interaction). It is important to understand that these two processes happen on different levels; while the former can be performed on static descriptions, the latter requires interaction with the service provider and is therefore much more complex.

Usually, a service provider does not provide just one specific service but rather a set of coherent and logically related services. For instance, an airline does not only provide the possibility to book a particular flight on a particular date, but instead it will offer (and advertise) the general service of booking flights. In order to deliver a specific service instance, a service provider usually needs certain information from the requester. An airline might require the name of the person booking, the flight number, booking class, and a valid credit card number as input information in order to create a booking. This input data will determine which specific service will be provided. While the general service (booking flights) can be statically advertised in a WSMO capability (used for discovery), the WSMO choreography describes how to interact with the service, e.g. for the purpose of negotiating a concrete flight data. Negotiation also typically requires mediation, since the data structures and protocols of the client have to be mapped to those of the provider in order to interact with a provider to access, explore and consume specific services.

### 5 Conclusions

Current technologies require a costly and custom hardware and software infrastructure for each pair of cooperating business partners. The lack of formal descriptions of services offered by organizations hampers automation in the location and usage of services required to perform a given business activity. Web service technologies and the service-oriented architecture paradigm provide a uniform infrastructure for the provision of services leveraging Web technologies, but they offer only syntactical descriptions that are hardly amenable to automation. Semantic Web Services enable the formal specification of services, allowing their automated, goal-driven, location and usage. WSMO provides a framework for the description of Semantic Web services which enables seamless integration through formal descriptions, maximal decoupling of components, and strong mediation support.

### References

Towards Semantic Service Selection for B2B Integration

Andreas Friesen and Kioumars Namiri

There are various B2B scenarios where many candidate services with the same or similar capability (provided by the same or even different service providers) can be used for integration. Hence, a requester driving a B2B integration scenario can choose among several candidate services offering a capability of satisfying the request. However, the optimal choice of the service to be invoked often depends on the parameters of the request at run-time and the preferences of the requester. In this article we describe a solution for a dynamic (at run-time) web service selection based on semantic interpretation of offered service capabilities and the parameters specifying the run-time request. The proposed solution takes into account special conditions on service usage either contractually agreed between requester and provider or specified by the requester without the knowledge of the provider. In general, these conditions restrict the interpretation of the original service capabilities as offered by a service provider and influence the choice of a service.

Keywords: B2B, Semantic Web Services, Service Discovery, Service Selection.

1 Introduction

The advent of Service-oriented Architecture (SOA) and Web Services (WS) opened new possibilities for smooth Enterprise Application Integration (EAI) in intra- and inter-enterprise scenarios in a loosely-coupled manner. In principle, Web Services enabled business systems can be used by anyone, from anywhere, at any time, and on any type of platform. The providers can offer their business functionality deployed as Web Services on a Web Server and publish their specifications to a repository offering them to potential users (requesters). The potential requesters can discover, select, compose, bind and invoke the offered Web Services in order to achieve their goals.

The traditional Web Services Technology Stack comprises at least the following core technologies: Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL) and Universal Description, Discovery and Integration (UDDI). SOAP is used for communication with a Web Service [10]. WSDL describes the Web Service interface [11]. UDDI provides publishing and discovery functionality for Web Service specifications and capabilities [12].

The lack of formally represented semantic meaning in this technology stack necessitates the tasks of discovering, selecting, composing, and binding Web Services to be manual steps performed by a human.

The recent advent of the Semantic Web and Semantic Web Services (SWS) promises new standardized means to formally capture the representation of the semantic meaning of data and interfaces. This enables the machines to automatically reason and to draw conclusions about the "intended meaning". The so-called Semantic Web Services promise a higher degree of automation concerning discovery, invocation, composition, and monitoring of Web Services. Currently several frameworks for semantic annotation of Web Services have been proposed, e.g., OWL (Web Ontology Language), OWL-S, WSMO (Web Service Modeling Ontology) and WSDL-S (Web Service Description Language) [5] [6] [8] [9]. All of them conceptually support or, at least provide, a vision concerning the phases of the Web Service usage process. However, all of them are, in a sense, rather meta-frameworks since they fall short of describing the exact realization of the single phases. This becomes obvious, if use case dependent requirements like association of the single phases with design-/run-time or impact of the realization of one phase to the sub-sequent phases has to be taken into account. In this article we focus on the interdependency between the negotiation phase (configuration) and the selection phase at run-time. Thereby, we focus on B2B integration scenarios requiring a run-time decision concerning selection of a web service from a set of available services that are potentially able to fulfill a concrete run-time request.

2 (Semantic) Web Service Usage Process

The most general notion of the web service usage process consists of the following three phases: web service discovery, web service selection, and web service invocation [7]. We argue that this approach is too straightforward since it does not take into account several additional aspects associated with the web service usage. Two of them are discussed in the following.

Authors

Andreas Friesen has graduated in Computer Sciences and got his PhD from the University of Siegen. He is now a Senior Researcher with SAP Research. His research topics include Service-Oriented Architectures, Business Process Management, Enterprise Application Integration, Security, and Semantic Web Services. <andreas.friesen@sap.com>.

Kioumars Namiri is a Research Associate with SAP Research. His research focuses on application of semantics in business process management. Before joining to SAP, he worked as a software engineer designing and developing distributed enterprise applications. Kioumars has received a diploma in computer science from the University of Karlsruhe(TH). <kioumars.namiri@sap.com>.

38 UPGRADE Vol. VII, No. 5, October 2006 © Novática
The first, and from our perspective the most important, aspect is the differentiation between the abstract and concrete service. According to [4], an abstract service is distinguished from a concrete service in that the former abstracts from the concrete service parameters which determine the latter. Thus, an abstract service describes a class of service parameter configurations that is associated with the web service capability. A concrete service describes a concrete service parameter configuration as it is delivered by the invocation of a web service. Hence, the web service capability description does not contain complete information about every possible concrete service that can be delivered by invoking the web service. Since the ultimate goal of a potential requester is to find and select a concrete service that optimally serves its needs, the selection and invocation steps inevitably overlap.

The second aspect deals with the possible negotiation of service parameters between requester and provider. This leads once again to the inevitable invocations of the web service interface. Besides the obvious overlap of the conceptually seen different phases, successful service discovery does not necessarily lead to successful service delivery, since in the set of potential service candidates there might be no one that is finally able to define a concrete service on which both the requester and provider agree.

A more sophisticated web service usage process addressing the above aspects is described in [4]. The relationship between a requester and a provider party goes through three different phases: a service discovery phase, in which potential providers are discovered; a service definition phase, in which the concrete service to be carried out is defined in all its details; and a service delivery phase, in which the value of this concrete service is actually delivered to the requester.

The lifecycle of the discovery phase presented in [2] consists of a goal discovery phase, taking into account that it needs some effort to come to a "goal description" which properly expresses the service request, a Web Service discovery phase and a service discovery phase, which map to the discovery and definition phases in [4]. The result of the goal definition is a formally specified abstract goal that can be used as a search criterion for Web Service discovery. Web Service discovery is based on comparing (in terms of relevance) the semantic description of a goal (requested abstract service) against those of provided abstract services (web service capability descriptions). Web Service discovery is performed by matching the goal against available web service capabilities. Notice that input to the Web Service discovery phase comprises the semantic descriptions of goals and capabilities only. This implies that the matching is solely based on information contained in those descriptions of abstract services. A selection of a service due to concrete parameter information, which possibly requires invocation of the service’s interface, is in the scope of the subsequent service definition phase.

Service Definition starts from an already identified set of potential web service candidates which have been identified to be relevant for a goal in the service discovery phase. It possibly involves negotiation of service parameters, and thus, invocation of the candidates’ Web Service interface. In general, final service determination is beyond the information contained in the semantic descriptions of the goal and web service capability descriptions. In this phase, the generalization to abstract services is given up and a concrete service with a concrete parameter configuration has to be defined, as it will be delivered later on.

Finally, Service Delivery comprises any steps required for the actual invocation and consumption of the concrete service selected for execution. This involves the invocation of the Web Service interface according to its behavioral interface as well as different forms of mediations concerning the communication protocol or the format of the data to be transmitted.

The service definition phase requires a tight integration with work on behavioural interface and invocation of Web Service interfaces and breaks the scope of processing semantic descriptions. Ideas that include negotiation of parameters according to preference information would require an extra encoding of preferences which is currently not present in the conceptual models of SWS frameworks or semantic service descriptions in general. Furthermore, such a negotiation is also very specific for the domain of value of the service, such that generic tool support is difficult to achieve.

3 B2B Integration Scenarios

The SWS usage process in the service discovery phase relies on the descriptions of abstract service capabilities of the provided Web Services, i.e., it enables a potential service requester to find a set of web service candidates that can potentially serve its requests. Service providers offer the same web service capabilities to all potential requesters. The choice (selection) of one web service from the set of potential candidates is left to the service definition phase. There are several possible use-case dependent scenarios. The following scenarios are taken into consideration (without pretending to be exhaustive):

Scenario 1: Service request takes place sporadically and has a "low value", e.g., flight or hotel reservations from private persons, i.e., neither requester nor provider are interested to negotiate on requester-specific service conditions. In that case, the requester accepts the service conditions (general terms) of the provider. This case fits perfectly into the described SWS usage process, so we do not elaborate on it further.

Scenario 2: Service request takes place sporadically but has a very "high value" (imagine you are going to order an aircraft at AIRBUS or Boeing). The negotiations in this case require human interaction at each concrete request. The service capability is described on a very high level of abstraction leaving a lot of space for the configuration of concrete parameters. In principle, the requester has to negotiate with all potential providers on each concrete request. Therefore, we do not consider this case either.
Scenario 3: There are several possible scenarios somewhere between the two extremes described previously. Such scenarios can be characterized by frequently occurring requests for a standardized service over a certain period of time. However, the number of requests (the total transaction value) usually justifies the efforts to negotiate about requester-specific conditions for service usage for a requester as well as for provider. The requester-specific conditions can be seen as constraints on the web service capability and must be consistently applied on the requester as well as the provider side. These constraints cannot be part of the published web service capability since they are requester-specific, i.e., in general, for each requester different.

Scenario 4: The last scenario under consideration is similar to Scenario 3 but in this case the requester sets constraints on the service capability usage according to its own preferences. Hence, service requester does not negotiate with the provider on requester-specific conditions, i.e., it accepts the general terms of the provider. This type of constraints defines business rules of the requester.

We claim that the web service usage process for B2B scenarios under consideration, i.e., the Scenarios 3 and 4, can be characterized by a negotiation phase that results in a "service contract" or "service agreement" restricting the usage of the "static part" of the web service capability and potentially influencing its "dynamic part". "Static part" means here the declaratively described and published parameters of a WS capability. "Dynamic part" comprises the parameters of the WS capability that can only be accessed through the invocation of the WS interface. The contract can be used for an automatic pre-selection of suitable WS during subsequent concrete service requests. The pre-selection criteria apply to concrete service requests as long as the respective contract remains unchanged.

A web service capability can be described as a set of service instances that can be delivered by invoking the web service. A contract between a requester and provider can be seen as a subset of this set. A concrete request during the invocation of the web service can then be seen as an instance of the contract, the capability or some more general set. Figure 1 illustrates this relationship.

The result of the discovery phase based on the abstract goal describing the set of all potentially intended concrete requests is a list of web services that can completely or partially serve this goal (Figure 2). The web service capabilities may overlap.
Hence, the result of the negotiation phase is a set of contracts that are subsets of the respective service capabilities on the one hand and subsets of the abstract goal on the other.

Similarly to WS capabilities, the contracts may also overlap and eventually do not cover the abstract goal completely (Figure 3). Therefore, at runtime, WS selection is required if a concrete request is within the scope of overlapping contracts. It may also happen that some concrete requests cannot be served at all, especially, if in the negotiation phase the abstract goal has not been completely covered by the set of available contracts.

In principle, the service selection can be implemented within the business logic of the requester’s application, e.g., through some rule-based technology or even hard coded. In the following we introduce an elegant and compact solution that is completely configurable in declarative way. The
introduced approach relies on description logics and semantic techniques derived from the approach for semantic web service discovery introduced by Li & Horrocks [3].

4 Ontological Modeling of The Set-Based Concepts

In order to apply semantic discovery techniques introduced in [3] to the discovery and selection problem described above, a domain-specific ontology is required. Based on that ontology an abstract service capability has to be built. The abstract service capability is provider-independent and covers all possible WS capabilities within the domain of discourse. For integration with a Semantic WS framework the abstract service capability must extend the WS description part, e.g., in OWL-S it’s the "Service Profile" and in WSMO the "capability" [6] [8].

WS capabilities of the service providers and the abstract request of a service requester can be modeled as sub-concepts of the abstract WS capability. The abstract request and WS capabilities are then used in the discovery phase in order to identify potential providers to negotiate with. There are different possible degrees concerning the quality of the semantic match: exact, plug-in, subsume, intersection, and disjoint [3].

In the case of successful negotiations, the contract between a requester and service provider has to be modeled as a sub-concept of the WS capability. The concrete request is then described either as an instance or as a most specific sub-concept of the abstract WS capability according to the used ontology. Figure 4 illustrates these dependencies. This means that a concrete request can be served by a Web Service if the subsumption test between the concrete request and the contract associated with a WS capability is successful. If several contracts match the concrete request a second selection step is required in order to choose between the remaining Web Services. This step usually requires invocation of the Web Services in order to get information necessary for the final selection according to the goals of the requester. The concrete selection goals are in general domain and requester-specific making a generic solution complicated or even impossible.

5 Carrier-Shipper Use Case

The application of the methodology described above is demonstrated now on a use case from the carrier/shipper domain. In this scenario, a shipper discovers potential carriers using an abstract shipment request and negotiates with them at the design time. The selection of a carrier for a concrete shipment request is performed (automatically) at run time.

Carriers, say a well known parcel carrier like UPS, offer a capability to ship some goods from one location to another (abstract shipment capability). To keep the example simple we assume in the following that each carrier serves defined geographic locations (regions, countries, cities, zip code ranges, etc.) and accepts distinct types of items to be shipped (e.g., parcels, documents, containers, etc.), i.e., it provides a carrier-specific shipment service capability. We presume the existence of domain-specific (geographic locations and logistics) ontologies describing concepts and relationships used in the following examples. A shipper negotiates with a carrier on some shipper-specific conditions and may also have some internal preferences, i.e., in the case of successful negotiations a contract is established. Having the contract a shipper is prepared for automatic carrier selection at run time depending on the characteristics of the concrete shipment, internal preferences, and special conditions negotiated with carriers. Some information, e.g., the price for a concrete shipment or the duration of the delivery, can only be accessed by invoking a web service of a carrier. Therefore, possible final selection goals could look like "take the cheapest" or "take the fastest".

Now a formally described "Abstract WS Capability" for the shipment domain has to be created. We call it "Shipment" and define as: "Ship one or more items of type ShippingItime from exactly one location of type Location to exactly one another location of type Location." The formal description in Description Logics (DL) is provided in Figure 5 [1].

Hence, a carrier expresses the capabilities of its shipment services by inheriting and refining the ranges on the properties of the concept "Shipment". Two examples in DL for carrier WS capabilities are illustrated in Figure 6.

The WS capability of CarrierA is: "Ship one or more items of type ("Parcel" or "Document") from any location within ("Europe" or "NorthAmerica") to any location within ("NorthAmerica" or Europe)".

Accordingly, the WS capability of CarrierB is: "Ship one or more items of type "Document" from any location within "Germany" to any other location (worldwide)".

Similar to carriers, a shipper can express its shipping needs, i.e., its "Abstract Request" also by inheriting and refining the ranges on the properties of the concept "Ship-

\[
\text{Shipment} \sqsubseteq (=1 \text{shipFrom}.\text{Location}) \\
\cap (=1 \text{shipTo}.\text{Location}) \\
\cap (\geq=1 \text{shipItem}.\text{ShippingItem})
\]

Figure 5: Abstract WS Capability "Shipment".
ment”. If, for instance, the shipper has to ship any items of types "Parcel", "Document" and "Container" from Germany to any locations in "Europe", "NorthAmerica" and "AsiaPacific" then its abstract request can be defined in DL as illustrated in Figure 7.

Since Abstract Request of a Shipper X has an intersection with the WS capabilities of the carriers A and B, the shipper can negotiate with the both carriers about the conditions on the usage of their shipping services.

The result of successful negotiations could then look like the contracts in Figure 8. The contract with CarrierA means: "Ship any items of type "Parcel" from "Germany" to any location in ("USA" or "Canada").

Accordingly, the contract with CarrierB means: "Ship any items of type "Document" from "Germany" to any location in the region "AsiaPacific".

Now, let’s assume that the following (semantically lifted) concrete shipment requests occur at run time (Figure 9).

In order to determine all the carrier web services capable to perform a concrete shipment request, a subsumption test between each of the available contracts and the concrete shipment request has to be executed. The results of the subsumption tests between the two sample requests and the contracts with the carriers A and B are depicted in Figure 10.

Obviously, according to the available contracts RequestA can be performed by CarrierA but not by CarrierB. Quite in contrary, the RequestB can be performed by CarrierB but not by CarrierA.

Since in the provided examples the contracts do not overlap, i.e., the intersection test (ContractA.ShipperX ? ContractB.ShipperX) is not satisfiable wrt. KB, there are no concrete shipment requests that can be potentially served by both carriers.

However, this is of course possible, e.g., if the ContractA.ShipperX would be re-negotiated to ship not only parcels but also documents then a new concrete shipment RequestC will be subsumed by both contracts (Figure 11).

In the case of multiple carrier selection, the selected services must be ranked according to some selection goal. This usually requires the invocation of the carrier web service in order to retrieve information required for the final decision, as has been already mentioned. However, DL-based technologies do not provide appropriate means for comparison operations. Hence, for this final selection step, an additional solution is required.

The introduced solution has been implemented as a WSDL Web Service mediating between a shipper application and carrier web services. The prototype implementation enables the configuration of new carriers or update of the existing carriers in a completely declarative way. The only exception is the final selection step where we used hard coded algorithms for a set of predefined selection goals; therefore, for this part a more flexible solution is required.

6 Conclusions

While the most of the work in the area of Semantic Web Services focuses on the technical interoperability, i.e., the interoperability of the interfaces (inputs, outputs, ordering of the exchanged messages, etc.) the proposed approach focuses on the business compatibility, i.e., how to ensure the consistency of business requirements in B2B integration context under the assumption that systems are technically connected in an interoperable way using Semantic Web Services. The proposed approach relies on the phases of the Semantic Web Service Usage process. While in the related state-of-the-art, negotiation and selection are recognized as important steps for the usage of Semantic Web Services, the exact definition and implementation of these phases remains an open research question. In this article we focused on selected B2B integration scenarios and provided a precise formal solution for the definition and implementation of the web service negotiation and selection phases.

References

AbstractReqShipperX ≡ ∃ shipFrom.Germany
   ∃ shipTo.(Europe ⊃ NorthAmerica ⊃ AsiaPacific)
   ∃ shipItem.(Document ⊃ Parcel ⊃ Container)
   ∃ Shipment

**Figure 7:** Abstract Request of a Shipper.

ContractA.ShipperX ≡ ∃ shipFrom.Germany
   ∃ shipTo.(USA ⊃ Canada)
   ∃ shipItem.Parcel
   ∃ CarrierA

ContractB.ShipperX ≡ ∃ shipFrom.Germany
   ∃ shipTo.AsiaPacific
   ∃ shipItem.Document
   ∃ CarrierB

**Figure 8:** Contracts on WS Capabilities.

RequestA ≡ ∃ shipFrom.Frankfurt
   ∃ shipTo.NewYork
   ∃ shipItem.Parcel
   ∃ Shipment

RequestB ≡ ∃ shipFrom.Frankfurt
   ∃ shipTo.Singapore
   ∃ shipItem.Document
   ∃ Shipment

**Figure 9:** Examples of Concrete Requests.

<table>
<thead>
<tr>
<th>Subsumption test</th>
<th>Satisfiable wrt. KB</th>
</tr>
</thead>
<tbody>
<tr>
<td>(RequestA ⊑ ContractA.ShipperX)</td>
<td>yes</td>
</tr>
<tr>
<td>(RequestA ⊑ ContractB.ShipperX)</td>
<td>no</td>
</tr>
<tr>
<td>(RequestB ⊑ ContractA.ShipperX)</td>
<td>no</td>
</tr>
<tr>
<td>(RequestB ⊑ ContractB.ShipperX)</td>
<td>yes</td>
</tr>
</tbody>
</table>

**Figure 10:** Service Selection by Subsumption.

Figure 11: Overlapping Contracts lead to Multiple Carrier Selection.

Leveraging E-Marketplace Models for Web Service-Based Application Development

Abraham Nieva de la Hidalga, Liping Zhao and Pedro R. Falcone-Sampaio

There is a growing interest in using and developing Web Service based systems. However, current Web Service applications have been developed as highly customised and tightly coupled systems due to their architectural model and are therefore difficult to maintain and adapt. This paper proposes a new architectural model based on the concept of e-marketplace to support the design, development, reuse, and publication of loosely coupled Web Services. A survey is used to highlight the importance of e-marketplaces. An agent based model is adapted to support the basic definition of the architecture. A proposal for an architectural model for Web Services based on the e-marketplace concept is presented as a preliminary step to extend the e-marketplace model into a platform to support the development, publishing and discovery of Web Services.

Keywords: E-marketplace, Service Development, Service Discovery, Service Publishing, Web Service Architecture.

1 Introduction

Web services constitute a thriving research area in information technology. There is a growing interest in industry for the use and development of Web-Service-based systems. Practical implementations of web services have demonstrated their capabilities as integration middleware, development tools, means of communication, and wrappers for legacy systems. However, current implementations of international conferences and workshops. Liping has published over 50 technical articles on software patterns and software design in international journals and conferences. She currently holds two prestigious Global IBM Faculty Awards and one Philips EngD Award. In 2006, she is one of only six academics at the University of Manchester to become an IBM partner under a new strategic partnership with IBM and is leading the investigation of the role of patterns in services design. <liping.zhao@manchester.ac.uk>.

Pedro R. Falcone-Sampaio is a Lecturer in business technology and information systems at the University of Manchester and a former business technology consultant of McKinsey & Company. Dr. Sampaio has published extensively in the topics of e-Service Design, Business Process Management, Information Systems Architectures, Data Quality, CRM, Agile Software Processes, Deductive Database Architectures, and Query Processing. He has also developed Knowledge Reports for McKinsey & Company’s Internal CRM Practice Knowledge Base. Dr. Sampaio has led and participated in research/consulting projects both in the UK, Latin America and USA in the Telecom, Banking and IT industries in topics relating to Business Process Effectiveness, Customer Relationship Management, IT Architecture Development, Internet Strategy Development, Risk Management, Call Centre Consolidation and Internet Banking. Dr. Sampaio holds a Doctor of Philosophy degree in Computer Science from the University of Manchester, an Executive Mini-MBA Sponsored by McKinsey & Company and a Master of Science degree in Computer Science from the University of Campinas in Brazil. Dr. Sampaio has led and participated in research/consulting projects both in the UK, Latin America and USA in the Telecom, Banking and IT industries in topics relating to Business Process Effectiveness, Customer Relationship Management, IT Architecture Development, Internet Strategy Development, Risk Management, Call Centre Consolidation and Internet Banking. Dr. Sampaio holds a Doctor of Philosophy degree in Computer Science from the University of Manchester, an Executive Mini-MBA Sponsored by McKinsey & Company and a Master of Science degree in Computer Science from the University of Campinas in Brazil. Dr. Sampaio is currently the co-chair of the 1st International Workshop on Methods, Architectures and Technologies for e-Service Engineering (SMIWEPE-MATEs, Stanford, Palo Alto – CA, USA 2006) and also the co-chair of the e-Commerce Technologies Management Track of the 2006 IRMA International Conference, Washington, USA. <pedro.sampaio@manchester.ac.uk>.
Web Services

web service based applications are faced with the same development problems as other software engineering trends. These are in-house, highly-customised and tightly-coupled applications which are difficult to maintain, adapt, and update.

Most of the proposals of development of web service architectures are theoretical models. In contrast, this paper presents an architecture derived from a widely accepted e-business application. Four types of e-business applications prevailed as relevant examples: electronic commerce, e-marketplace, portal, and account access [8]. From these applications, e-marketplace is the most appropriate for inter-organisational cooperation. This paper proposes to use the e-marketplace concept as an architectural model which facilitates the design, development, deployment, and use of web services. An alternative architecture is proposed together with a high level schematization of its components.

2 Web Service Architecture

The Web Service Architecture (WSA) proposed by the W3C [15] presents a model and context to understand web services and the relationships between the components of the model. The WSA describes characteristics common to all web services and the required elements of the web service network.

The most widely accepted variant of the WSA is presented in Figure 1. This consists of three main components which are a requester entity, a provider entity and a discovery service. The architecture assumes that requester and provider entities use the discovery service to publish and search for web services. Once the requester finds a suitable provider then both parties negotiate the operational semantics governing service interactions. This model has fuelled the development of several efforts in the fields of standardisation, discovery, orchestration and choreography. The approaches and adaptations of this architecture can be widely grouped as intra-organisational development groups, systems providers and research groups.

The intra-organisational approach comprises the companies which adapted the WSA to source their necessities of integration and development, focusing mainly on decoupling existing systems into services. This approach was used by early adopters like Halifax Bank, Deutsche Post AG, Wintertur and Credit Suisse [9]. Discovery is managed manually and the discovery service is not implemented. Additionally, service connections are hard coded into the services themselves, hindering maintenance and updating.

The second approach is to use web services as integration drivers. Web services are used as a wrapping presentation layer to inter-communicate different systems inside an enterprise [10][12][13]. The main characteristic of these architectures is that registries based in the UDDI standard aid the discovery process. In this approach the middle agent is a brokering service which manages the registry facilities.

Research groups in industry and academia provide a wider range of implementation approaches. These groups focus on enabling the evolution of web services technologies towards the automatic discovery and binding of services within complex transaction scenarios. These groups adapt the WSA to different needs according with the implementation scenarios for example grid computing, peer-to-peer interactions, directories and registries. There are also different approaches to discovery such as matchmaking based in quality of service [4][18], semantic matchmaking.

Table 1: E-marketplace Definitions.

Table 2: Trading Function Offering.
Web Services

[2], and automated discovery, for instance UPnP [11] and jini [14].
According to the definition proposed by the W3C, the web services Architecture [15]:
"does not attempt to specify how Web services are implemented, and imposes no restriction on how Web services might be combined".

The flexible definition in connection with the model fostered the application of the architecture without being bound to a particular technology. However, this great openness also promoted varied approaches to the implementation of web services. A deeper analysis of the WSA [19] found three deficiencies. The first is related to the low relevance given to agents in the different models. The second problem is that agent roles are not clearly defined, complicating the definition of the behaviour of agents in different contexts. The third problem is the absence of a communication structure.

From the analysis of the practice it is possible to highlight that the main differences in the approaches are given in terms of the entity acting as the discovery service and the way in which discovery and binding occur.

3 E-marketplaces
Early analysis showed confidence in the potential of e-business applications to promote intra-organisational cooperation [1][6][7]. E-marketplace is one of such early e-business initiatives which proved its worth in practice. E-marketplaces today provide functionalities beyond the mere aggregation of pricing lists. Collaboration-oriented e-marketplaces are cited as an emerging approach to support online business-to-business transactions.

3.1 E-marketplace Defined
Table 1 presents four of different definitions of e-marketplace. Definitions 1-3 describe e-marketplace as a commercial intermediary that plays a passive role in online trading. These definitions focus on the technical characteristics of e-marketplaces and are aligned with early developments of e-marketplaces. Definition 4, on the other hand, describes an emerging approach to e-marketplaces. This definition portrays e-marketplace as an organization that plays an active role in web-based trading and collaboration.

3.2 E-marketplaces Types
E-marketplaces can be classified in different ways. This section presents three common classifications and analyses e-marketplaces listed on the eMarket Services Directory (EMSD) [5].

Domain-Oriented Classification divides e-marketplaces into horizontal and vertical [1][17]. Vertical e-marketplaces focus on commercial exchange of services and products in specialized domains. Vertical e-marketplace can be integrated into the supply chain and provide efficient management of commercial interactions. Horizontal e-mar-

---

Table 3: Service-oriented Classification.

<table>
<thead>
<tr>
<th>Type</th>
<th>Subtype</th>
<th>Num of e-marketplaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market Oriented</td>
<td>Aggregation</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>Market Matching</td>
<td>149</td>
</tr>
<tr>
<td></td>
<td>Mixed MO Functionalities</td>
<td>326</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>535</td>
</tr>
</tbody>
</table>

Table 4: Service-oriented Classification Applied to The EMSD.

---

Figure 2: The Agent Role Model.

Figure 3: Updated ARM.

Figure 4: ARM Adapted for The New WSA.
Web Services

Figure 5: High-level View of The Architecture.

ketplaces focus on distribution of services and products across different domains. These e-marketplaces supply products and services directly to buyers offering lower prices. Many e-marketplaces listed on the EMSD [5] fall into more than one category, hampering the application of this classification method.

Functional-Oriented Classification divides e-marketplaces according to the trading functions offered. Some authors recognize as little as three functions [1], while others, like the EMSD, identify up to nine (Table 2). In practice, however, most of the e-marketplaces offer a combination of two or more functions [16][17]. Classifying e-marketplaces according to functionality is arbitrary because there is no universally agreed set of trading functions.

Service-Oriented Classification is based on core services offered [3]. Core services fall into three major categories, namely transaction services, interaction services, and support services. Transaction services are concerned with online order and purchase of products, supported by search and selection of suppliers, products and prices. Interaction services include planning capabilities, product life-cycle management and capabilities for collaboration in new product design. Support services include consulting services, software integration and application service provision.

The e-marketplaces that support interaction services are also called "collaboration e-marketplaces" [3][16]. There is a simplified version of the service-oriented classification [16], which classifies e-marketplaces as market-oriented and collaboration-oriented. Market-oriented e-marketplaces support the creation of competitive markets, supporting trade relationships between buyers and sellers. These e-marketplaces are sub-divided into aggregation and market-matching. Aggregation e-marketplaces support the commercial interchange by providing product catalogues and search facilitation. Market matching e-marketplaces help participants to locate appropriate trading partners and to negotiate contract terms.

Collaboration-oriented e-marketplaces simplify the business processes between business partners. There are two kinds of these e-marketplaces: transactional and strategic. Transactional e-marketplaces focus on streamlining the order fulfilment process. Strategic e-marketplaces facilitate collaborative initiatives between trading partners. Table 3 presents this classification in detail and aligns the nine trading functions recognised by EMSD with them (adapted from [16]).

Offering market-oriented functionalities is seen as a basic requirement for e-marketplaces [16], while collaboration-oriented functionalities provide a competitive advantage. Table 4 shows how this classification is applied to the e-marketplaces listed on EMSD. This data reinforces the conception of collaboration e-marketplaces as well-established e-business practice.

4 E-Marketplace Based WSA

This Section describes how the agent role model is adapted to support the definition of a new web service architectural model based on the e-marketplace concept.

4.1 Agent Role Model

The architecture is defined in terms of the Agent Role Model (ARM) [19]. The use of the ARM provides the benefits of extensibility, consistency, low coupling and dynamic binding. Agents participating in web service execution can change behaviour dynamically adopting different roles according to the context in which they participate. Additionally, explicit behaviours can be separated and specified as roles. Figure 2 presents the original ARM. This section describes the adaptations made to ARM to support the architecture definition.

The first set of adaptations is needed because the original ARM was based on an outdated WSA definition. In the new WSA the management oriented model is no longer part of the architecture. The Intermediary was eliminated from the message oriented model. And finally, the Policy Model was redefined around the policy concept recognising two types of policies: permissions and obligations. These policies constrain the access to resources and performance of actions for all agents. As a result, the figure of the contractor became limited as it can be interpreted as a unilateral view from the requester side. As a consequence, three of the roles defined for the original version of the ARM disappear. Figure 3 presents the resulting model consisting of four basic roles. The Sender and Receiver roles are directly

Table 5: Comparison of Approaches

<table>
<thead>
<tr>
<th>Feature</th>
<th>Applied WSA model</th>
<th>Proposed Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target web service user</td>
<td>Developer/Programmer</td>
<td>Consultant/business specialist</td>
</tr>
<tr>
<td>Binding mechanism</td>
<td>Hard-coded-explicit</td>
<td>Dynamic-implicit</td>
</tr>
<tr>
<td>Interaction service</td>
<td>One requester with one provider</td>
<td>Many requesters with many providers</td>
</tr>
<tr>
<td>Participating entities</td>
<td>Undefined participants</td>
<td>Clearly recognised participants</td>
</tr>
<tr>
<td>Reuse support</td>
<td>Restricted</td>
<td>Intentional</td>
</tr>
</tbody>
</table>

© Novática
linked to the Message Oriented Model. The Requester and Provider roles are associated with the Policy, Service and Resource Oriented Models. The updated ARM is the basis for the definition of the e-marketplace architecture. For the definition of the architectural model this paper does not make reference to the message-oriented model and the associated Sender and Receiver roles given that the definition is centred on service provision and service discovery above the communication level.

The ARM was subsequently adapted to reflect the needs of the e-marketplace architecture (Figure 4). The roles were defined at three different levels: structural, execution and regulation.

Two agent roles are defined at the structural level: e-marketplace and member. These roles differentiate the basic components of the architecture, providing the basic means to distinguish the agents interacting as members from members of the e-marketplace.

At the execution level two roles are defined: provider and requester. These roles are involved in the actual invocation and execution of web services. Member agents can adopt dynamically either of these roles, while the e-marketplace can only adopt the provider role when interacting with members. Member agents interact with each other adopting provider and requester roles dynamically.

The Regulation level is concerned with the operation of the e-marketplace as a regulatory entity. Two roles are defined: policy enforcer and manager. The agent adopting the e-marketplace role undertakes the responsibilities of policy enforcer, membership and registry manager.

The definition of the architecture around the concepts of agents and agent roles fosters dynamic behaviour change. Agents can act as requesters or as providers depending on the context in which their interactions happen. Moreover the dynamic and proactive behaviour expected from the e-marketplace is better explained defining it as a set of agent roles.

### 4.2 Architecture Overview

As stated previously the structure of the e-marketplace recognises initially two agent roles, the e-marketplace and the member. Figure 5 presents a high level view of the architecture. The hexagonal shapes represent member agents playing a requester role while the circular ones represent member agents adopting a provider role. The ellipse in the middle stands for the e-marketplace. This is to reinforce the e-marketplace role as a central component of the architecture.

The core of the model proposed is the e-marketplace. The e-marketplace must provide facilities to build queries for services, negotiation, configuration, binding, and storage of service descriptions.

Clients use the e-marketplace to discover and compose business processes and value chain elements. Clients may also save queries and successful configurations for future use.

Providers use the e-marketplace functionalities basically to store their service descriptions. However, by browsing at previous queries, service configurations and service composition patterns providers can also find ideas for the development, enhancement and deployment of web services.

### 4.3 Architecture Comparison

There are a number of important differences between proposed architecture and commonly accepted architectural model. As stated earlier the architecture proposed by the W3C required the definition of a proper agent oriented model. Further more, the role of the entity providing discovery services was neglected to some degree in an attempt to preserve implementation freedom. In contrast, this proposal is aimed to actively apply an agent based role model and encourages the proper definition of the discovery service provider, the e-marketplace. This section presents a more detailed comparison of both architectures.

Table 5 describes the differences between the commonly accepted WSA and the e-marketplace based architecture according to seven features: discovery method, target user, binding mechanism, interaction scheme, discovery service, reuse support and participating entities.

In the first generation of web services the target user was a programmer. In contrast the target users of the proposed architecture are domain experts, consultants and business specialists implementing business processes through service composition. While the ideal is to foster loose coupling for the engagement and disengagement of services, the common solution is to include binding information, such as the service address and message format, into the application code. In contrast, this proposal aims to allow looser coupling by providing means of mapping amongst message formats. The architecture is aimed to support automatic and semiautomatic discovery, binding and negotiation while the conventional model assumes manual approaches to discovery and binding. The typical model privileges the interactions between one client and one provider at the time. In contrast this model recognises interactions of various providers and various users. In the proposed model active discovery services are proposed contrary to passive discovery provided by registries and directories. Recognised participants with rights and obligations opposed to undefined entities providing the discovery service. Component reuse is amongst the most important motivations for the use of Service Oriented Architectures. However in current applications reuse is limited. The proposed architecture is aimed to actively support reuse.

### 5 Summary

This paper discussed the shortcomings of some approaches to the application of web services providing an overview of the WSA and some examples on how it has been customised and used. To address some of the shortcomings identified, an alternative architecture derived from a widely accepted e-business application, the e-marketplace, was introduced. The analysis of e-marketplace classifications demonstrated its wide user base as well as the various
forms in which e-marketplaces are implemented. This also supported the selection of the e-marketplace type to support the development of the architecture.

Finally, a high level view of the architectural model for web services based on the ARM model was presented. However, supporting the implementation of business processes with web services encompasses many challenges that need to be addressed in order to:

- Allow the dynamic discovery of services and business processes.
- Support the generation of reusable services and business processes.
- Register and advertise available services and business processes in a proper structure.

The fulfillment of these requirements establishes the path to follow for the materialisation of the proposed architecture into a platform to support the development, publishing and discovery of web services.

References

Open Standards, A Renewed Impulse: The Case of Open Document Format

Miguel A. Amutio-Gómez

2006, Miguel A. Amutio-Gómez. This article is published under a Creative Commons Attribution-ShareAlike licence, available at <http://creativecommons.org/licenses/by-sa/2.0/deed.en>.

This paper was first published, in Spanish, by Novática (issue no. 181, May-June 2006, pp. 69–73). Novática, <http://www.ati.es/novatica>, a founding member of UPENET, is a bimonthly journal published, in Spanish, by the Spanish CEPIS society ATI (Asociación de Técnicos de Informática – Association of Computer Professionals).

Open Document Format is an XML-based (eXtensible Markup Language) file format for documents on electronic media. It has recently been adopted as the standard ISO/IEC 26300 (International Organization for Standardization/International Electrotechnical Commission) and has been an OASIS (Organization for the Advancement of Structured Information Standards) standard since May 2005. OpenDocument file format arouses great interest because it is non-binary, multiplatform, and can be implemented by various tools; because its specification is public, open, neutral and royalty free; and because of the impact that such characteristics have on freedom of choice, interoperability, and preservation of information. This article provides an overview of OpenDocument: what it is, who is responsible for it and who maintains it, what it is like inside, which applications support it, and why it arouses such interest. It also describes the project carried out as part of the European IDA/IDABC (Interchange of Data between Administration/(Interoperable Delivery of Pan-European eGovernment Services to Public Administrations, Business and Citizens) programme for the promotion of open formats for document exchange, in the context of public policies in the field of open standards.


1 Introduction

Documents are important for commercial, administrative, or interpersonal relations; of that there can be little doubt. They provide the evidence of a specific activity and they are the medium by which such an activity is given material form. They serve evidential and communication purposes, either internally within an organization or externally with third parties.

A document on electronic media is an identified and structured entity which contains text, graphics, sounds, images or any other kind of information which may be edited, retrieved, exchanged between information systems or users as a single discrete unit [1]. Ultimately, an electronic document is digital information represented by binary digits (bits 1 and 0). These bits form a binary stream with no spaces in between, and without punctuation or format. Characters are coded depending on the length of the byte (usually 8) and the character set used, then the
document is coded as a set of variables and data structures in memory which configure its internal representation. When the document is saved on a medium, this internal representation is converted into a storage format.

The Spanish edition of Wikipedia tells us that a "storage format" is a set of rules defining the correct way to store data in memory [2]. However, we need to understand that the term "memory" also includes any physical media on which it is possible to store and retrieve information.

When the internal representation of a document handled by a text processor is saved onto an electronic medium, a disk for example, this internal representation is converted into a specific storage format which can be automatically processed and requires a specific tool to be accessed and read. This gives rise to one of the following two situations:

- If the format can only be understood by the tool which produced it, then we have a situation of incompatibility with other tools or programs, and conversion filters will be needed in order to handle the document with alternative tools.
- If the format is compliant with a standard specification, then the document can be opened by the tool which produced it or by any other tool compliant with the same specification.

In short, the text processor works with an internal representation of the document: when the document is saved, a text processor converts the internal representation into a storage format. Afterwards, when the same tool or any other compatible tool opens the document, the internal representation is produced again. Thus, the document is coded in a specific format which is the description of the own document; the document is hostage to its coding and it only exists as such when the file is interpreted by a program which understands the file format. Consequently, and unlike paper documents, a tool is always required to access and read the document.

The arrival of the Open Document Format has aroused considerable interest and expectations because, in addition to offering the advantages of an open standard, it also has a number of interesting technical features plus all the benefits of standardization. This article provides an overview of the several dimensions of the Open Document Format (OpenDocument) describing what it is, who is responsible for it and who maintains it, what kind of files it supports, what it is like inside, which applications support it and why it arouses such interest. It also describes the work of the IDA/IDABC Programme on the promotion of Open Document Exchange Formats in the context of public policies in the field of open standards.

2 What is Open Document Format?

According to Wikipedia, OpenDocument or ODF, short for the OASIS Open Document Format for Office Applications, "is an open format for saving and exchanging office documents such as memos, reports, books, spreadsheets, databases, charts, and presentations" [3].

It is currently an OASIS and ISO/IEC (International Organization for Standardization/International Electrotechnical Commission) standard. The format was first adopted as an OASIS Open Document Format for Office Applications (OpenDocument) v1.0 standard on 1 May 2005. OASIS (Organization for the Advancement of Structured Information Standards) [4] is a not-for-profit international consortium for the development, convergence, and adoption of standards for e-business; its Technical Committee, known as OASIS Open Document Format for Office Applications (OpenDocument) TC [5], is responsible for the development, maintenance and promotion of OpenDocument.

OASIS OpenDocument TC was created in December 2003 by a number of entities with the aim of developing a standardized XML file format for office applications, based on OpenOffice.org’s file format [6]. The founding members of this TC are from companies such as Arbortext, Boeing, Corel, CSW Informatics, Drake Certivo, National Archive of Australia, New York State Office of the Attorney General, Society of Biblical Literature, Sony, Stellent, and Sun Microsystems. The complete list of members is publicly available on the OASIS web site at . The TC continues to work towards the maintenance and development of the OpenDocument file format, paying special attention to three main issues: accessibility, metadata, and mathematical formulae. In fact, three new subcommittees have been created in order to deal with these issues:

- OpenDocument Accessibility: will gather accessibility feedback from the implementers of applications for disabled people and will carry out a formal evaluation of the accessibility of the format.
- OpenDocument Metadata Subcommittee: will work on the operational requirements of metadata and proposals for improvements.
- OpenDocument Formula Subcommittee: will work to create a specification language for spreadsheet formulae.

Recently, a Technical Committee, OASIS Open Document Format Adoption Technical Committee (ODF Adoption) TC, was also created to work on the demand for applications and solutions regarding the use of OpenDocument.

After its adoption as an OASIS standard, the OpenDocument specification underwent an ISO/IEC standardization process resulting in the standard ISO/IEC 26300 Open Document Format for Office Applications (OpenDocument) v1.0 [7]. The standardization process started in September 2005 and went straight into the DIS (Draft International Standard) stage. The voting process, which closed in April 2006, resulted in a majority of votes for, a few abstentions, and no votes against. Thanks to this result, the FDIS (Final Draft International Standard) stage could be avoided and the standard went straight to publication as ISO/IEC 26300.

3 What Is OpenDocument like Inside?

The approach followed by the OpenDocument specification [8] is to...
Convocatoria de Asamblea Territorial Ordinaria 2006

Capítulo Territorial de Madrid de ATi

Conferencia-coloquio previa (a las 18:00)

Figure 1: Contents of An OpenDocument ZIP File.

(legend: Carpeta = Folder; Documento = Document; Desconocido = Unknown)

Which Applications support OpenDocument?

One of the outstanding aspects of OpenDocument is that its specification can be implemented by a great many applications; it is currently implemented by products such as OpenOffice.org, KOffice, Abiword, TextMaker, Writely, StarOffice, IBM Workplace, NeoOffice, ... [9]. I.e. OpenDocument can be supported not only by office applications but by any

reuse existing XML open standards as much as possible and to create new markups only when none of the available open standards provides the required functionality. Thus OpenDocument makes use of Dublin Core XML for metadata, MathML for mathematical formula, SVG for vectorial graphics, SMIL for multimedia, etc.

OpenDocument provides strong separation between content, layout, and metadata. The internal format is a ZIP compressed file which includes a number of files and folders (see Figure 1).

- **content.xml**: stores the document's actual content, except for binary data such as images. An example of the content is the following:

```
<text:p text:style-name="Standard"/>
<text:p text:style-name="P2">Convocatoria de Asamblea Territorial Ordinaria 2006</text:p>
<text:p text:style-name="P3">Capítulo Territorial de Madrid de ATi</text:p>
<text:p text:style-name="P3">Conferencia-coloquio previa (a las 18:00)</text:p>
```

- **Pictures/** stores the document's images. The file content.xml references images with the markup `<draw:image>`. For instance:

```
<draw:image xlink:href="Pictures/00000000000028A000000504D769EDA.jpg" xlink:type="simple" xlink:show="embed" xlink:actuate="onLoad">
</draw:image>
```

**Figure 2**: Some Applications which Implement OpenDocument (Source: Wikipedia).
other application which supports this format. This offers a great potential for
new ways of processing either individual documents or large collections
of documents. (See Figure 2 and Figure 3.)

5 How Widely Is OpenDocument
Used?

Growth in the use of OpenDocument is the result of the wide
range of products which handle it, es-
pecially OpenOffice.org. In this par-
ticular case, there are several dissemi-
nation channels, as I can vouch for
from my own personal experience:

■ by means of downloads from the
main web site or mirror sites, for in-
stance RedIRIS in the case of Spain.

■ by means of Linux distributions,
including OpenOffice.org, or other of-

ce products such as KOffice.

■ by means of software that is of-
ten bundled with technical IT maga-

azines.

■ by means of software delivered
with new equipment.

■ by means of other channels such as
software included as part of the
documentation of OSS conferences
and events.

■ and, why not, through informal
interpersonal channels.

In order to have an idea of just how
widespread OpenDocument is, some
indicators may be helpful, such as the
statistics for downloads of programs
like OpenOffice.org, or the dimension
of known deployments.

For example, in terms of known
deployments of OpenOffice.org in
Spain, the following figures may be
suggestive though not exhaustive:

■ Extremadura: 70,000 units, in-
cluded as part of the GNU/Linux dis-
tribution, in the fields of Education,
Health Service, Public Libraries, and
Public Administration of Extremadura.

■ Andalusia: 255,000 units, in-
cluded as part of the GuadalLinex dis-
tribution, in the fields of Education,
Guadalinfo Centres, Public Libraries,
and other centres.

■ Catalonia: 60,000 units with
OpenOffice.org, in state schools.

In terms of downloads, still in ref-
ence to Spain, recent informa-
tion from RedIRIS, the mirror web site for
downloading OpenOffice.org, shows
that from the second half of 2005 to
May 2006 1,408,748 downloads of
OpenOffice.org were made, plus
133,737 downloads of other distribu-
tions that included copies of
OpenOffice.org.

International milestones include the
explicit adoption of OpenDocument by
the Administration of Massachusetts in
September 2005 and by Australia Na-
tional Archives in 2006 as the file for-
mat of choice to ensure the preserva-
tion of information on electronic me-
dia. Also, on 3 March 2006 the Open
Document Format Alliance initiative
was launched with 35 founding mem-
bers from the public and private secto-
rs [22].

6 Why Does OpenDocument
Arouse so Much Interest?

The reasons why OpenDocument
arouses so much interest are many and
varied:

■ It is the state-of-the-art in docu-
ment formats: there are no significant
alternatives in terms of document for-

mats or alternative tools to process
them. Existing formats are insufficient
and/or lack standardization.

■ The features of OpenDocument
as a non-binary, multi-platform (hard-
ware and software) format imple-
dmented by a large number of alterna-
tive tools. It provides separation be-
tween the format and the tool, and sup-
ports text, spreadsheets, presentations,
graphics, and others.

■ OpenDocument has several ref-
currence implementations which act as
an aid to its dissemination and wide-
spread use on different platforms.

■ The specification of

Figure 3: This Article with OpenOffice.org and KOffice.
OpenDocument is public, open, developed through an open process, visible, neutral, royalty-free, and open to the user community. It is not controlled by a single player but by a working group open to new membership, and it is subject to formal change control and adoption procedures. Its specification is free from legal restrictions (e.g. licenses or patents) and it can be implemented by anyone.

These aspects benefit two very important fields: eGovernment documents and the preservation of information:
- Public IT policies on eGovernment place great importance on freedom of choice and interoperability.
- Long term preservation of documents on electronic media requires the availability of open standards providing technological independence to help preserve information in the face of successive technological waves and future commercial policies, as well as providing freedom of choice in the way documents are accessed and viewed, together with ease of access of those documents.

7 The Effect of The IDA/IDABC Programme on the Promotion of Open Document Formats

In 2003 the European Union Programme IDA/IDABC (Interchange of Data between Administration/Interoperable Delivery of Pan-European eGovernment Services to Public Administrations, Business and Citizens) [10][11] began to actively promote open formats for document exchange [12]. The reasons behind this initiative were: a) the lack of interoperability between office suites was considered to be unsatisfactory for the development of eGovernment, and b) the lack of support of standardized open formats. Also, after studying the state-of-the-art at the time, it was decided that any documents to be exchanged between public administrations and citizens should be in a format that did not require the use of specific software products, thereby ensuring permanent accessibility to the documents.

The following timeline outlines the sequence of events of that year:
- In May 2003 a need for open document formats for the development of eGovernment was identified.
- In January 2004 the findings of the Valoris Report [13], a comparative analysis commissioned by the IDA Programme, were adopted. This analysis studied document standards available at that time and, in particular, existing or emerging open document formats and the possible market trend in that field.
- In March 2004 the major market players (Microsoft and SUN) were invited to comment on the Valoris Report and to debate with IDA Programme experts while presenting and defending their point of view.
- In May 2004 the IDA management committee issued its recommendations on the use of open document formats by public administrations [14].
- Later, the EU Enterprise Directorate General invited the major software manufacturers to work towards a greater interoperability of document formats. In response to this invitation, IBM, Microsoft and SUN expressed their commitment to working in that direction [15].

Of the abovementioned actions, two stand out as true milestones: the Valoris Report and the IDA recommendations on the promotion of open document formats.

7.1 The Characteristics of An Ideal Document Format

Among many valuable contributions, the Valoris Report identifies eight criteria or features to be used in the analysis of existing document formats to determine what features make up the ideal document format. These eight aspects are:
- Open: the specification is publicly accessible, it can be distributed freely and the format can be implemented in programs without legal restrictions and royalty-free.
- Non-binary: the content of the document, together with its mark-ups, is saved as plain text and not as a binary stream.
- Modifiable: the document can be modified, unlike formats which produce only readable documents.
- Format fidelity: the format ensures the original layout of elements within the documents (i.e. indents and spacing, place of graphics, etc.) regardless of hardware and software platforms.
- Cross-platform interoperability: the format can be used with full preservation of its semantics on various hardware and software platforms.
- Support for current word processor features: the format supports common functionalities available in currently available word processing tools.
- Support for emerging word processor features: the format can meet possible new requirements, such as those related to electronic signature.
- Widely adopted: there is a wide base of users and tools which contribute to the sustainability and use of the format; this does not necessarily mean market dominance or the universal acceptance of the format.

7.2 IDA Recommendations on The Promotion of Open Document Formats

On 25 May 2004, the management committee of Programme IDA, including representatives of 25 Member States, endorsed the recommendations on the promotion of the use of open document formats forwarded by its group of experts in this field.

These recommendations acknowledged:
1. The responsibility that the European public sector has to ensure accessibility to its information, the need to improve citizen and business interaction and the importance of the public sector as a purchaser of IT products and services.
2. Secondly, and as a result of this analysis, the conclusion was reached that:
   a. The industry had taken some steps, citing as examples the publication of the OpenOffice.org and WordML formats
   b. That it is not necessary for all documents to be modifiable, but if documents should be modifiable, XML is the best option for the separation of content, structure, semantics and presentation
   c. That the public sector should avoid making use of a specific prod-
uct mandatory
d. That a format should be promoted that is able to be implemented on various platforms, that is not discriminatory against market players, and that offers equal opportunities for its implementation
e. That the standardization of OASIS’s OpenOffice.org format is welcomed by OASIS.

3. Thirdly, recommendations took into account the limitations of existing document formats at that time and were addressed to players with some influence in these issues:
- To Industry: to become involved in the standardization of document formats; to provide conversion filters between formats; to provide tools and services so that the public sector can migrate documents to XML formats.
- To Microsoft: to commit to the publication of Word XML specifications; to propose their formats to standardization bodies; to eliminate the non-XML components of WordML.
- To OASIS: to propose the Open Document Format (ODF) to ISO/IEC.
- To the Public Sector: to make use of different formats to publish information.

8 OpenDocument in The Context of Public Policies on Open Standards

A few open standards such as OpenDocument are attracting most of the attention and act as a kind of flagship in representation of all the others. The fact that OpenDocument is an open standard is highly appreciated.

The Communication of the European Commission on Interoperability [16] recognizes that open standards, specifications and interfaces are the key to interoperability. In the past, when the eEurope 2005 Action Plan asked the European Commission to develop the European Interoperability Framework (EIF), "a set of recommendations and guidelines for eGovernment services so that public administrations, enterprises and citizens can interact across borders, in a pan-European context", it was specified that the EIF would be based on open standards and that it would promote the use of open source software.

This European Interoperability Framework [17], developed by the IDA and IDABC programmes, also recognises open standards as a key element for interoperability and identifies the characteristics that a technical specification should have in order to be considered an open standard:

"The standard is adopted and will be maintained by a not-for-profit organisation, and its ongoing development occurs on the basis of an open decision-making procedure available to all interested parties (consensus or majority decision etc.).

The standard has been published and the standard specification document is available either freely or at a nominal charge. It must be permissible to all to copy, distribute and use it for no fee or at a nominal fee.

The intellectual property - i.e. patents possibly present - of (parts of) the standard is made irrevocably available on a royalty-free basis.

There are no constraints on the reuse of the standard."

These actions come in the wake of the open systems strategies developed when Decision 87/95/CEE was issued [21].

Along similar lines and within the framework of the Millennium Development Goals, the UN Joint Inspection Unit considers [18] that public administrations should ensure that all citizens have equal opportunity access to information on electronic media and are not obliged to purchase specific solutions to exercise their rights. It also considers that public administrations should adopt appropriate measures in pursuit of open standards and procurement policies and practices which do not lead to vendor lock-in.

In the Spanish General State Administration, the "Criteria for security, standardization and preservation of information of applications used by the Administration for the exercise of its public competences " [19] and the "Proposal of recommendations for the General State Administration on the use of open source software and open standards" [20] consider that documents should be available in a format enabling them to be accessed using alternative products. This approach provides citizens with greater freedom of choice in terms of interacting with the public administration and in the use of information technologies that are in their own language and are better adapted to their needs as users.

All these actions assume that, in the provision of eGovernment services, the relationship between the different players - whether citizens, enterprises, or public administrations - by electronic means, takes place in a heterogeneous technological environment which includes, among other possible interactions, the exchange of documents on electronic media. Given this scenario, the adoption of open standards would appear to be the only way to ensure that all these players can interact using their preferred technical solutions, thereby maintaining freedom of choice and interoperability. The importance of interoperability is not only in the vertical relationship between infrastructures, services, content, and accessibility, frequently presented as a pyramid model, but also in its behaviour in practice as a chain model, which predicates a horizontal relationship between actors, so that any lock-in of proprietary protocols, specifications and formats is inevitably passed on from one player to another.

9 Conclusions

OpenDocument arouses great interest because its specification is already an ISO/IEC and OASIS standard and also for a number of reasons relating to freedom of choice, interoperability, and the preservation of information on electronic media. OpenDocument contributes to freedom of choice because it offers an alternative which can be supported by a variety of applications running in alternative hardware and software environments. It contributes to interoperability because it is a non-binary format, and a hardware and software multipurpose tool that can be used in technologically heterogeneous environments. Those qualities plus the nature of its specification (open, publicly available, free from legal restraints and royalties), together with the availability of other standards such as ISO 19005-1:2005 [23], may help to clarify current uncertainties regarding
the use of file formats for the preservation of information on electronic media in the long term.

Translated by Steve Turpin

References


[22] OpenDocument Format Alli-
Critical Information Infrastructure Protection

Sandro Bologna, Roberto Setola and Salvatore Tucci

© Mondo Digitale, 2006

This paper was first published, in its original Italian version, under the title "Le infrastrutture critiche informatizzate", by Mondo Digitale (issue no. 4, December 2005, pp. 20-31, available at <http://www.mondodigitale.net/>). Mondo Digitale, a founding member of UPENET, is the digital journal of the CEPIIS Italian society AICA (Associazione Italiana per l’Informatica ed il Calcolo Automatico, <http://www.aicanet.it/>.)

The global shift in the socio-economic paradigm calls for the adoption of new policies to ensure the security of the infrastructures that use ICT (Information and Communication Technologies) technologies to distribute essential services for the wellbeing of populations (Protection of Critical Information Infrastructures). The correct definition of such instruments cannot ignore the existence of interdependencies between the various infrastructures. This is a new phenomenon, one which is complex and strongly interdisciplinary, requiring the development of new methodologies and tools for the analysis of systems comprising a number of heterogeneous and interdependent infrastructures.

Keywords: Complex Systems, Critical Information Infrastructure Protection (CIIP), Critical Infrastructure Protection (CIP), Interdependency.

1 Introduction

The technological development of industrialized countries and their social welfare and prosperity will depend more and more on the availability and correct functioning of technological infrastructures such as grids for the transmission and distribution of energy (electricity, gas etc.) telecommunication networks, computer networks, transport networks (railways, railways and airways etc.), health-care systems, banking and financial systems, water systems etc. Due to their importance, these infrastructures are generally referred to collectively by the term "critical infrastructures". In this field he has co-authored several scientific publications and books. <bologna@casaccia.enea.it>

Salvatore Tucci received the MSc degree in Physics from the University of Pisa, Italy. After his military service, between 1973 and 1974 he held a postgraduate fellowship in Computer Science at IEI-CNIR, Pisa. In 1974 he joined the Department of Computer Science at University of Pisa as an assistant professor. He held a visiting position in 1977 at IRIA, Paris, working on the Sirius project headed by Erol Gelenbe. In 1981/82 he spent one year at IBM T. J. Watson Center, Yorktown Heights, NY as a visiting scientist in the performance evaluation group headed by S.S. Lavenberg. In 1987 he joined the Department of Electronics at the Università di Roma "Tor Vergata" as a Full Professor in Computer Engineering. From 1990 to 1999 he was the Dean of Computer Engineering Curriculum at the Faculty of Engineering. Since 1999 he has been the Director of the Department of Informatics and Telecommunications of the Presidency of the Council of Ministers. His research interests include performance evaluation of computer/communication systems, parallel and distributed systems, multimedia applications, internet modelling, conception and design of large information systems for decision-making and government, critical information infrastructures modelling. He has served as general chair and president of the organizing and program committee of various national and international conferences. <s.tucci@governo.it>

© CEPIS
term Critical Infrastructure because if they do not function correctly, even for a limited period of time, they can have a negative impact on the economy of individual systems or groups, involving economic losses, not to mention directly jeopardizing the security of property and people.

Until a few decades ago, each of these infrastructures was considered to be an autonomous system, substantially independent and managed by vertically integrated operators. Various converging technical, economic, and social causes have profoundly modified such structures. In fact infrastructures tend to be increasingly more closely connected, almost to the point of being interdependent. Consequently a failure (whether of an accidental or malicious nature) in one of them can easily spread in a domino effect, provoking dysfunctions and malfunctions to a larger and unforeseeable set of users.

In recent years there have been several incidents that have demonstrated the present level of interdependence between the various technological infrastructures. The most famous of these must surely be what happened to Galaxy IV in 1998. Galaxy IV is a telecommunications satellite in geostationary orbit over the west coast of the United States. Its failure meant that around 40 million pagers immediately went out of service, around 20 United Airline flights suffered take-off delays due to a lack of information about high-altitude weather conditions, some radio broadcasts went off the air, and the most surprising thing of all was the effect on the highway transport system. Due to the impossibility of processing credit card payments at service areas along the highways (as they use satellite communication to connect with credit distributors) drivers had considerable difficulties in refuelling [1].

Another example is the Slammer worm which on January 25, 2003 rapidly spread over the Internet [2]. This worm took advantage of a well known bug in the Microsoft SQL 2000 server and produced an anomalous increase in IP traffic. Apart from the foreseeable accessibility problems to many websites and e-services, it also affected banking and financial systems. In the USA around 13,000 automatic teller machines (ATMs) crashed, in Italy it was impossible for 11,000 post offices to carry out financial operations, and the whole banking and financial system of South East Asia was in a state of almost complete collapse. It also affected air transport and emergency systems: a number of flights out of Houston airport suffered delays or were cancelled, and the Seattle emergency call centre went out of service leaving 165,000 people unattended. The worm also caused problems in tele-control systems; due to the saturation of the connectivity backbone, the telecontrol system of a US utility company was blocked.

A more recent episode occurred on January 2, 2004, when a failure at a service plant in an important Telecom Italia node in Rome paralysed both fixed and mobile traffic (the same thing happened to other telecom operators) for several hours over a huge area of Rome. The incident also had repercussions on the financial system (around 5,000 bank branches and 3,000 post offices were left without any telematic connection) and on air transport (70% of the check-in desks at Fiumicino airport were forced to resort to manual procedures for check-in operations).

And then there were the power outages which affected more than 140 million people in a number of countries over a 6 week period in 2003. These outages dramatically underscored the fact that industrialized societies are almost totally dependent on the availability of electrical power. Nevertheless, consumers tend to adopt an ostrich mentality and assume that an uninterrupted supply of this resource will always be available.

On the subject of outages, it is interesting to note how in the course of the last 40 years there has been a constant decrease in the number of incidents, evidencing the improved reliability of the electrical system, while at the same time there has been an increase in the size of the areas affected by them. This is proof of the increasing complexity of the electrical grid (or to be more precise, of the info-electrical system) and the enormous difficulty of controlling it and, specifically, of isolating the consequences of some (rare) catastrophic events.

An analysis conducted by the Canadian government in March 2003 [4], which was found to be applicable to many other countries, shows that ability of these infrastructures to absorb accidental events (failures, human error accidents, etc.) could be increased, and therefore the number of incidents of this nature could be reduced. However, other types of threats are becoming increasingly important; those caused by ‘natural’ events related to ever more extreme and violent climatic phenomena, and those connected with malicious actions, or, to be more precise, terrorism.

Critical Infrastructures will become increasingly important in the coming years, as a target for terrorist acts, conducted by traditional methods and via cyberspace (cyber-terrorism), or by combined actions (swarming attacks, believed by analysts to be the most likely scenario). This latter type of attack would involve scenarios in which the aim of cyber-terrorism would be to slow down and diminish the efficiency of the response of emergency services to traditional terrorist acts, thereby augmenting the impact of the attacks.

At an international level the prob-
problem as a whole is generically referred to by the term (CIP) Critical Infrastructure Protection. When CIP focuses on aspects of vulnerability and interdependency caused by the existence of cyberspace, we tend to talk of (CIIP) Critical Information Infrastructure Protection. In reality there is a very fine line between CIP and CIIP due to the very close interrelationship between the physical (tangible) world and the virtual world (of information) to the extent that, regardless of the original cause, a breakdown tends to affect both worlds. For this reason, the two terms are often used almost as synonyms in literature. Nevertheless, such a subdivision is useful to stress the necessity of considering both the physical security aspects and the problems caused by the existence of cyberspace.

In 1996 the United States was the first country to identify the problem and, in 1998, under the Clinton administration, a programme aimed at the safeguard and protection of these infrastructures (Presidential Decision Directives 62 and 63) was developed, with the ambitious goal of ensuring that "Any interruptions or manipulations of these critical functions must be brief, infrequent, manageable, [and] geographically isolated".

The events of September 11 brought about a dramatic boost to all activities connected with national security. The defence of critical infrastructures has become one of the cornerstones of the Department of Homeland Security System and strategic lines to step up critical infrastructure security were set out in two documents issued by the White House in February 2003 [5, 6].

In almost all industrialized nations similar initiatives have been undertaken, aimed at understanding the problem, its contextualization in specific realities, the individualization of strategies to reduce the vulnerability of the country’s systems, and the readiness to intervene in case of emergency, all characterized by a strong public-private cooperation. Also, international bodies such as G8, NATO, and the UN (with UN resolution No. 58/199 on the "creation of a global culture of cybersecurity and the protection of critical information infrastructures" adopted by the General Assembly of the United Nations on December 23, 2003 [7]) have focused attention on the problem, inviting supporting countries to define strategies and instruments to increase protection levels of such infrastructures, to increase the restorative capability of service levels following negative events, to develop R&D activities and to promote international cooperation.

In Italy the Gruppo di Lavoro sulla Protezione delle Infrastrutture Critiche Informatizzate (Working Group for the Protection of Critical Information Infrastructures) was formed, in March 2003, by the Dept. for Innovation and Technology of the Presidency of the Council of Ministers. In March 2004 the working group drew up a report on the national situation [8].

This report calls for the adoption of initiatives to reduce such dangers. Among these initiatives, research and development activities occupy a priority role given that, as also pointed out in the CIP Handbook [9], research worldwide is still at an extremely early and immature phase. This calls for initiatives involving the academic and research world.

One of the most interesting research aspects is the development of models and instruments to help in the study and comprehension of various critical infrastructures in which there are dependencies and interdependencies. But the very presence of such dependencies and interdependencies gives rise to such an enormous complexity that the currently available analytical and/or simulative tools and methods [10] appear to be overloaded. This creates a need to look at different paradigms, for example those used in the field of biological science which have historically needed to work with highly complex systems. Thus the Gruppo di Lavoro [8], while still basing its work on the experience of the American NISAC (National Infrastructure Simulation and Analysis Centre), proposed the creation of a virtual simulation centre with the purpose of sharing information from various sources and leveraging a multi-disciplinary and multi-sector approach.

In the following sections, some of the proposed methodologies for the study and simulation of this type of problems are described.

2 Modelling of Interdependent Infrastructures

Although the modelling and simulation of single infrastructures is a mature field of research and there are available analytical tools capable of characterizing their behaviour on different time scales and with different degrees of abstraction, tools for analysing heterogeneous and interdependent infrastructures are largely immature.

One of the first proposed approaches is based on the generalization of Leontief’s economic model for the study of the equilibrium of economic systems [11]. Just as the production levels of "seed workers" influence the production dynamics of final assets, so too can a failure in an element influence the service levels in other infrastructures and to develop R&D activities and to promote international cooperation.

In the following sections, some of the proposed methodologies for the study and simulation of this type of problems are described.

<table>
<thead>
<tr>
<th>Electric grid</th>
<th>Transport</th>
<th>Hospitals</th>
<th>Telecom</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electric grid</td>
<td>0</td>
<td>0.2</td>
<td>0</td>
</tr>
<tr>
<td>Transport</td>
<td>0.4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hospitals</td>
<td>0.6</td>
<td>0.8</td>
<td>0</td>
</tr>
<tr>
<td>Telecom</td>
<td>1</td>
<td>0.2</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1: Matrix of Leontief Coefficients.
pital, a highway transport system, and a telecommunications system actively engaged in a geographical area, while taking into account their reciprocal dependencies and interdependencies, it is necessary to estimate a Leontief matrix (see Table 1).

In the case of Table 1, the first column indicates that the total destruction of the electric grid (to an unworkable level equal to 1) has a direct impact on the highway system rendering 40% of it inoperable, 60% of the hospital system is blocked, meanwhile the telecommunications system is completely out of use.

Other than these "direct" consequences, it is also necessary to take into consideration the secondary consequences, for which the total impact of a failure is represented by the steady-state solution of the following equation:

$$x(n+1) = Ax(n) + c$$  \hspace{1cm} (1)

Where $A$ is the matrix $m \times m$ of Leontief coefficients ($m$ is equal to the number of infrastructures taken into examination), $c$ is a vector column of length $m$, which represents the failure caused by external factors, and $x$ is the vector column of length $m$, with the probability of inoperability associated to each element.

If we want to take into account, for example, effects on the system induced by an event which renders 80% of the electrical network inoperative from (1) above, we have:

$$\tau = (I - A)^{-1}c = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \end{bmatrix}$$

Resolving (2), the total inoperability of the electrical network is equal to 89%, that of the transport system is 46%, that of the telecommunication system is 98%, whereas the hospital system is completely inoperative.

In spite of its extreme simplification, this example clearly demonstrates how the presence of interdependencies acts as an exacerbation of the event’s consequences due to feedback effects induced by the presence of (inter-)dependencies between the various infrastructures.

One of the most critical aspects of this model lies in the calculation of the elements which characterize the interdependencies (the coefficients of the Leontief matrix) due to the fact that many dependencies are poorly understood.

It should be noted that, although for some types of relationships it could seem more correct to speak of dependence, in this context we will often use the term interdependence to stress the presence of a huge number of interactions and functional dependencies existing between the different infrastructures. Thus, for every pair of infrastructures there are mechanisms, direct or indirect (i.e. intermediary by means of dependencies on other infrastructures), for which the evolution of one influences the other and vice versa.

In the formulation proposed by Rinaldi, Peerenboom and Kelly in [1], interdependencies are analysed considering six different "dimensions" to bring together the various elements which characterize both the behaviour linked to the presence of the interdependence, and its appearance.

In particular they stress the need to analyse:

**Environment**: i.e. the structure in which the owners and operators establish purpose and objectives, creating security systems to define their business etc. Obviously, the operational condition and the conditions of each infrastructure influence the surrounding environment and, in turn, the environment exerts pressure on the infrastructure itself.

**Types of Interdependence**: an interdependence can be classified as:

- Physical: two infrastructures are physically interdependent if the status of one is dependent on the material output (i.e. physical) of the other. For example, a coal-fuelled electric power plant and its railway network have a physical interdependence, since each of the two systems depends on the output of the other: the plant needs the railway network for fuel and the delivery of spare parts for the generator, while the generated electrical power is necessary for the operation and control of the railway network.

- Cyber: an infrastructure has a cyber-interdependence if its status depends on information transmitted across cyberspace.

- Geographical: two or more infrastructures are geographically interdependent if a local environmental event can provoke a simultaneous change in the state of all infrastructures. This happens when the various infrastructures share the same physical site, such as a bridge, a room etc. In such a way, a natural or malicious event can provoke a communal failure of the various infrastructures.

- Logical: two infrastructures are logically interdependent if the status of each of them depends on the status of the other by means of a mechanism other than the ones previously described. This type of interdependence, typically related to the exchange of services between infrastructures, permits the modelling of links connected with socio-economic or cultural phenomena, or regulatory or legislative issues. Generally, human decisions are predominant in the creation of such types of interdependence.

Note that, unlike the other types, Cyber interdependence is an absolute, not a relative, property. If an infrastructure is cyber interdependent it has an
extended interdependence with basically every other infrastructure which makes use of cyberspace.

Operational state: To fully understand the interdependencies between the different infrastructures it is necessary to determine what each infrastructure depends on under normal operating conditions, in anomalous situations, and during the restoration phase following a failure.

Infrastructure characteristics: In this dimension one must consider the spatial scale, through a hierarchy of elements:

Part: the smallest component distinguishable in an analysis.

Unit: a set of parts functionally correlated (for example a heat generator).

Subsystem: a unit line (for example a cooling system).

Infrastructure: a complete set of similar systems.

This spatial scale is closely connected to the geographical scale, given that the infrastructures can be considered at a city, regional, national, or international level according to the purpose of the analysis. In every case the spatial scale has important implications for the way in which interdependencies are considered in the analysis. Generally speaking, at a parts and unit level interdependencies play a minor role compared to all the other cases.

Another important characteristic is the time scale. The time horizon of interest can vary from seconds (for example for energy system operations), to hours (for operations connected with the supply of water or gas, or the transport system), to years (for improvements or capacity increase of an infrastructure).

By focusing the analysis on different time horizons, some of the types of interdependency will acquire a greater or lesser importance. For example, if we examine a very fast process such as the propagation of a failure in an electrical network, cyber interdependence will be particularly relevant, while interdependencies of a logical nature will not play any kind of role. Conversely, the latter plays a primary role in understanding the consequences produced by modifications in legislative frameworks (and in this case cyber interdependencies might be ignored in any analysis).

Alongside these characterized elements of an infrastructure there are also considerations regarding operational factors and organizational factors which characterize the functioning of each infrastructure.

Types of failure: interdependencies between infrastructures may be the means by which a failure can propagate itself.

Cascade: when a malfunction provokes a failure in a second infrastructure, which in turn causes an anomaly to appear in a third and so on.

Escalating: when a disruption in one infrastructure exacerbates an independent disruption of a second infrastructure.

Common cause: when two or more infrastructures suffer damage at the same moment due to the same reason.

Level of coupling: as a function of the degree of coupling (tight or loose), both the propagation time and the intensity transmitted from a possible malfunction vary. Such interactions can be of a linear nature if they are the result of a designed process (generally known, visible, and generated by a programmed sequence of operations) or complex when they occur unexpectedly following a sequence of unprogrammed operations.

While reducing all of these aspects to a single mathematical model is not simple, [12] proposes the adoption of a model based on a Hierarchical Holographic Modelling (HHM) approach. The HHM methodology is based on the assumption that when one wants to model a complex system, more mathematical or conceptual models can emerge. Each of these models focuses on a specific aspect and by taking into consideration a set of models, it is possible to obtain an acceptable representation that reflects the reality of the system as a whole.

To better understand the problem of dependencies between the various elements of an infrastructure and interdependencies between different infrastructures, it is necessary to model each infrastructure as an object composed logically of three distinct layers: organizational, cyber, and physical [13] (Figure 1). Within any infrastructure all elements interact beyond the elements of their own level, even with elements located in contiguous levels (by means of functional relationships known as intra-layer links). Meanwhile, the homologous layers of the various infrastructures interact among one another through connections known as inter-layer links.

This schematization helps to show how the current tendency towards convergence in telecommunication is producing an increase in the dependencies between the various infrastructures.

![Three Layer Model For The Critical Infrastructures](image)

Figure 1: Layered Modelling of A Generic Infrastructure and Its Interdependencies.
3 Simulative Analysis of Interdependent Infrastructures

The approaches described in the previous paragraph provide us with a perspective on the problem and its complexity. However, they are not in a position to provide the necessary quantitative elements to be able to define the impact of different events, which is indispensable for prioritizing the actions to be undertaken and determining which are the priority investments. In this context, due to the rapid and profound infrastructural shift which characterizes all Critical Infrastructures, the use of tools based on historical analysis is not always adequate, especially with regard to predicting the consequences of catastrophic events or terrorist actions.

It is therefore necessary to develop models and simulation tools which are able to predict a system’s behaviour in different hypothetical scenarios. However, attempts to develop such tools have run up against the complexity of the systems they are meant to model, a complexity that seems to exceed the modelling capability of present day tools.

One of the more promising approaches for the analysis of the interdependencies between complex networks is the agents paradigm (Agent-based modelling) [14]. The fundamental idea which drives these models is that the complex behaviour is the result of interactions between autonomous and elementary individuals which operate on the basis of simple rules, and that by interacting together cause the collective behaviour of the system to "emerge".

In other words, the behaviour of an interdependent infrastructure is analysable by resorting to a bottom-up approach which models the whole system, starting with the (local) behavioural knowledge of single components and going on to study how the system behaves when these components interact with each other.

This is the transposition of the biological and sociological behavioural approaches which emphasize how the aggregation of individuals of a certain type gives rise to the emergence of behaviours that cannot be predicted by the study of a single individual in isolation. The adoption of strategies such as agent-based modelling, which has been largely developed in the field of bio-complexity, is a response to the realization that modern technological infrastructures have reached a complexity comparable to the biological systems and therefore need to study the techniques used in that world.

In agent-based modelling, single constituents are modelled as an entity characterized by their location, capabilities, and memory. The location of an entity defines where it is in a physical or virtual space. The capabilities of an entity express what the entity is able to do. An entity can modify its internal representation of data (perception capability), it can modify its environment (behaviour capability), it can adapt itself to changes in the environment in which it finds itself (intelligent reaction capability). It can also share knowledge, information, and common strategies with other entities (cooperation capability), and it can carry out actions without external intervention (autonomous capability). The data representing the state of an entity and the history of its experiences (for exam-
A different strategy is to make
interoperable simulation tools developed
for the study of single isolated infrastructures. For example in [15], in
order to study the dependencies existing between the energy grid and the
communications network, the authors have succeeded in making
interoperable a great many different simulators covering a large number of
related fields (EMCAS, NS2).

The main advantage of this type of
approach lies in the possibility of using
codes already widely validated and
verified experimentally. Another
important aspect is the limited amount
of information that needs to be shared
among the different players; in fact,
only information regarding interde-
dependency elements is necessary. In
this way it is possible to overcome much
of the resistance which various oper-
ators have to sharing detailed information
about their own infrastructures.

4 Topological Analysis

Another interesting topic concerning
the study of infrastructures is their
topology and the implications which
this has on the various network owners
and, specifically, on the issue of
robustness.

The pioneering work of Strogatz
and Watts [16] around the end of the
1990s and the work of Barabasi [17]
early in 2000 has shown that in many
networks the schematicisation of com-
plex networks, such as random graphs,
does not tally with empirical data.

In particular the work of Barabasi
clearly shows that in many networks
the number of links per node, while
following a Poisson distribution (as is
predictable in a random graph) have a
distribution which varies with the
power laws (Figure 3). Therefore, these
graphs have a structure which is not
very "democratic": some nodes, ge-
erically referred to as hubs, have more
connections than others and for this
reason they play a fundamental role in
the topology of the network [17].

In this system, known as the Scale
Free network, the typological charac-
teristics of the network derive directly
from the laws that govern their evolu-
tion. It is, in fact, possible to demon-
strate that any network built by the suc-
cessive aggregation of new nodes un-

Figure 2: Dynamics of The Macro-components of The Model.
under the principle of preference aggregation (i.e. so that the new node has a strong likelihood of being connected to junctions that are already well connected) has Scale-Free properties.

The nature of Scale-Free networks make them particularly robust against random failures: the number of nodes which need to be removed before (statistically) the graph loses its connection is significantly higher than for the random graph. The price to pay is its vulnerability to the selective elimination of the most important nodes of the network. Indeed, if the order of elimination is proportional to the number of links which are connected to the single node (node degree), a scale-free graph can become disconnected by eliminating fewer nodes than can a random graph [19].

At a more operational level, a scale-free graph is more robust against accidental failures (natural or, at any rate, random events), and this explains the prolificacy of this type of structure in nature. On the other hand, it is considerably more vulnerable to targeted (perhaps we should say terrorist) attacks or, to be more precise, to actions consciously aimed at reducing the efficiency of a network (i.e. attacks performed with the capability/possibility of striking at the most vulnerable points).

Another important aspect regarding the spreading of an "epidemic" is the fact that a scale-free system does not evidence the classical threshold effect which characterizes classical epidemiological models. This means that an epidemic can spread to the whole of the network even if the number of initially infected elements is an extremely low proportion of the total number of elements. In extreme cases, it is possible for even a single node (as happens, for example, with an Internet virus) to trigger a rapid spread of an epidemic if it is able to infect one of the hubs. This calls for the need to adopt strategies to limit the spread of such "epidemics" [20]. While we still need to study and understand better the implications of applying such theories to defence and security strategies, it would seem evident that strategies based on random topological models may turn out to be inefficient.

5 Conclusion
It is a must for all developed nations to improve the robustness of their critical infrastructures. On the one hand, the role which these infrastructures play for the wellbeing of nations is vitally important and, on the other, infrastructural interdependencies and the rapid dissemination of ICT in sectors historically unaware of information security issues, constitute vulnerability risks. These considerations, along with the evidence of increased threats, both natural (above all due to extreme climate phenomena) and malicious (terrorism and crime), call for the adoption of suitable strategies to improve the resilience of these infrastructures. A fundamental step in this direction is to try to improve our knowledge of the behaviour of the system resulting from the interconnection of the various critical infrastructures. The methodological and analytical tools we are currently using to study the risks do not seem to be capable of handling the complexity which characterizes these types of systems. In our article, some of the proposed approaches to model and simulate such systems have been illustrated. However, it is evident that there is still a lot of work to do on this subject given that, to quote [9], "the research community is just beginning to single out the correct and the most important questions that need to be asked".

Translated by the authors

References
February 2003; http://www.whitehouse.gov/pcipb


Informatics Profession

Professionalizing IT

Peter Skyte

© 2006 ITNOW

This paper was first published by ITNOW (Volume 48, issue # 5, September 2006, pp. 32-33). ITNOW, a UPENET partner, is the bimonthly magazine for BCS members published, in English, by Oxford University Press on behalf of the British CEPIS society BCS (British Computer Society, <http://www.bcs.org/>). The full September 2006 issue of ITNOW is available to read at <http://www.bcs.org/server.php?show=ConWebDoc.7008>

It’s taken for granted that occupations generally understood to be professionals, such as teachers, doctors, nurses, architects, lawyers, accountants and others are regulated either by professional bodies (RIBA, ACA and others) or independent regulatory organizations (General Teaching Council, GMC, NMC). However it is easy to forget that this was not always the case, and such professionalism has in many cases had a long history of development, often arising from public scandal at the time and the resultant public outcry for ‘something to be done’. Now it’s the turn of IT.

Keywords: Careers, Professionalism, Qualifications, Standards.

"It is the essence of professionalism to apply objective standards of craftsmanship and accomplishment to one’s work rather than business criteria." Peter Drucker, ‘Management and the professional employee’, Harvard Business Review, 1952.

"Process and procedure are the last hiding place of people without the wit and wisdom to do their job properly." David Brent, The Office, BBC TV.

Everyone in today’s world would take it for granted that occupations generally understood to be professionals, such as teachers, doctors, nurses, architects, lawyers, accountants and others are regulated either by professional bodies (RIBA, ACA and others) or independent regulatory organizations (General Teaching Council, GMC, NMC). However it is easy to forget that this was not always the case, and such professionalism has in many cases had a long history of development, often arising from public scandal at the time and the resultant public outcry for ‘something to be done’.

Information technology is no longer merely a stand alone product sitting on the desk or in the pocket. It is woven into the fabric of business, private and public services and our lives. Information technology systems and services are now often business critical, safety critical or in many cases both.

Financial control, procurement, supply chains, product development, and sales are all reliant on IT systems. Trains, airplanes, (nuclear?) power stations, military equipment and defence systems, criminal records, tax and national insurance, and healthcare all now depend on the IT software and systems at their core.

And yet most private and public organizations rely on IT people to design, develop and operate these complex systems without in many cases any accreditation, certification or verification of any objective standards. With the possible exception of journalism and the press, we do not allow other professionals to practise without objective standards of craftsmanship or accomplishment.

So why IT professionals? Must we await some public or private sector disaster and the resultant public inquiry before action is taken?

It is for these reasons that Amicus, the second largest union in the UK with many IT users and professionals

Author

Peter Skyte, National Secretary of the Information Technology Professionals Association (ITPA), has a degree in Chemistry and completed a research doctorate on Physical Properties of Metalloporphyrins. He is President of the ICT Working Group of Euro-UNI, the European section of the international labour organisation representing professional, commercial and clerical workers worldwide, which brings together unions covering the IT and communications industries across Europe. He is also Vice-President of the Industry, Business and Information Technology Section of UNI and National Officer of Amicus. <peter.skyte@amicustheunion.org>
amongst its members, supports the call by BCS, Intellect and others to professionalize IT. There are also other reasons, and consequences that need to be understood:

1. The need for a career structure and qualification framework to allow IT professionals to develop their skills and expertise, not just at the technical level but in leadership and management. As with procurement and supply of products, we have developed a ‘just-in-time’ approach to people skills in this country, which will in the future undermine the UK’s ability to survive and thrive in the increasingly competitive and interconnected world.

2. Understanding that risk and regulation are intertwined and that professionalization will require regulation. There is a cost to regulation, both in setting up an agreed body of knowledge, and in verifying, certifying, accrediting, monitoring and enforcing in achieving and operating standards to accord with this body of knowledge.

3. Acceptance that higher skills and professionalism will demand increased recognition and reward, individually and collectively. Whilst IT professionals may not generally be low-paid in absolute terms, they are low paid in many organizations relative to their skills and responsibilities as measured against other professionals such as lawyers and accountants. Their stature and status is often unrecognized. How many organizations have their chief information officer on the main board or governing body? In contrast, how many organizations fail to include their chief finance officer on the main board or governing body?

4. The impact of outsourcing, both domestic and international (offshoring), is real and growing. The only way ultimately to compete with companies in India, China and elsewhere is to move up the value chain and increase levels of quality, competency, skill and professionalism. However, there is no invisible hand that will automatically do this, and organizations must continue to invest in people, skills and technology in the UK. Professionalizing IT must form a key component of this strategy.

In India, Nasscom (the industry body for IT software and service companies) is in the process of establishing a National Skills Registry, which will contain verified personal, qualification and career information about IT professionals. Whilst there are good reasons to be critical of how Nasscom has chosen to go about this in terms of its independence from employers and the use of biometric identification, this initiative does nevertheless indicate the desire to raise and monitor standards and to professionalize the Indian IT and BPO industry.

Britain must recognize the need to drive up standards of craftsmanship and accomplishment and, as such, professionalize IT. Contrary to the wit (and occasional wisdom) of David Brent in The Office, this is not about process and procedure being a hiding place from getting the job done, but ensuring that the job exists in the first place and that it is done competently and capably to the highest standards and quality.

Amicus, with its Parliamentary Group of 114 MPs at Westminster in both Houses and links into government and parliament, is fully committed to working with BCS, Intellect and other IT sector stakeholders to work towards this end.

We cannot have a 19th century approach with 21st century technology!
Harmonise Project

Presentation

(Information provided by François-Philippe Draguet, Project Manager of Harmonise)

The Information and Communications Technologies (ICT) profession has a significant impact on employment, business and society. ICT qualification schemes are of significant importance in the development of the ICT Industry and Industry in general. The global competitiveness of European industry depends on both effective use of ICT for industrial and business processes and the knowledge, skills and competences of existing and future ICT practitioners.

It is important for ICT practitioners to be able to seek employment wherever they want within the European Union. With no formal barriers to mobility between Member States and a growing use of English within the ICT workplace, increasing the integration of national labour markets for ICT Professionals for the benefit of the competitiveness of European enterprises is vital. There must now be a focus on improving employers (and customers, where ICT contractors seek work) appreciation of applicants’ knowledge, skills and competence for the job through better understanding of the ICT practitioner qualifications (or "certifications").

For now, Europe’s citizens who want to improve their "e-skills" face a wide range of competing qualification and certification schemes, both vendor and non-vendor specific, public and private-commercial ones. In addition, in what has been a rather turbulent labour market, few employers appear to attribute much weight to applicants’ qualifications, focusing their requirements more on direct experience in particular software environments.

CEPIS, as representative of the European Network of Informatics Professionals, is focusing on taking part in European initiatives to represent its Member Societies and to help improve the development of ICT Skills coordination at the European level. In order to reach its main objectives, CEPIS is highly involved in a 36 month European funded project: Harmonise. The main aim of this project is to develop a framework on which to improve the widespread recognition and "transparency" of ICT practitioner qualifications. Support is needed to put in place an acceptable framework for clarifying the meaning and value of individual qualifications from Member States and, ideally, European-wide certification schemes.

CEPIS acts as coordinator of the project. Institut AIFB (Institut für Angewandte Informatik und Formale Beschreibungsverfahren, University of Karlsruhe) and IFS (Institute for Future Studies) are the scientific leaders and are helped by the BCS (British Computer Society), AICA (Associazione Italiana per l’Informatica ed il Calcolo Automatico), GI (Gesellschaft für Informatik eV), NISZT (John v Neumann Computer Society), the ECDL Foundation (European Computer Driving Licence Foundation) and EITS (Estonian Information Technology Society). In order to concentrate on the different scopes of the certification schemes, the study has been divided in four areas:

- Demand and Supply analyses the situation, the need and the importance of the certifications in the labour market.
- e-Skills certification studies the certifications available at national or European level concerning their value, their importance, their specifications…
- Market concentrates on the organisation of the certification market in Europe.
- Quality Assurance of the certification schemes.

The Harmonise Consortium is also collecting information (as part of e-skills certification study area) about the different national qualification schemes in order to have a snapshot of the actual situation. This phase is currently on-going and a questionnaire is available for people interested in providing CEPIS with the Harmonise visit <http://www.cepis-harmonise.org/php/> or email <draguet@cepis.org>.