Object-Oriented Business Solutions

Rafael Corchuelo[1], Antonio Ruiz[1], Jörg R. Mühlbacher[2], and Jesús D. García-Consuegra[3]

[1] Universidad de Sevilla, Dpto. de Lenguajes y Sistemas Informáticos
Avda. de la Reina Mercedes, s/n. E-41012 Sevilla, Spain
{corchu, aruiz}@lsi.us.es; http://tdg.lsi.us.es
[2] Johannes Kepler Universität, Institut für Informationsverarbeitung und Mikroprozessortechnik
Altenbergerstraße, 69. A-4040 Linz, Austria
muehlbacher@fim.uni-linz.ac.at; http://www.fim.uni-linz.ac.at
[3] Universidad de Castilla-La Mancha, Dpto. de Informática
Campus Universitario, s/n. E-02071 Albacete, Spain
jdgarcia@info-ab.uclm.es; http://www.info-ab.uclm.es/gsid

Abstract. This report summarises the presentations, discussions, and main results of the ECOOP'01 Workshop on Object-Oriented Business Solutions (WOOBS). It was not a pure scientific meeting, but a mixed gathering where people from the industry and the academia met to exchange ideas, experiences and build a network of relationships with others committed to the emergence of object-oriented business solutions. WOOBS had an invited talk on quality of service, twelve presentations and lively discussions during and after them. The main conclusions were on the importance of Multi-Organisational Web-Based Systems in today's e-commerce world, which justifies the study of a new multidisciplinary paradigm called Web-Oriented Programming.

Keywords. agents, aspect-orientation, coordination, databases, distribution, e-commerce, middlewares, transactions, web services, XML.

Workshop web site: http://tdg.lsi.us.es/woobs01

1 Introduction

E-commerce is gaining currency as the Internet settles as a medium for fruitful commercial transactions. Anyone with a credit card and an Internet-enabled device is a potential customer, and this is the reason why no corporation can resist jumping on the e-commerce bandwagon. To achieve this goal, web-based, attractive, technically robust applications are a cornerstone.

The e-world currently amounts to million Euros, and the underlying software industry is investing a lot of money in researching on new tools and development methods so that these applications can be built at sensible costs. This has entailed adapting existing methods and tools, which have usually been procedural, to the new resources and means the Internet provides, but at such a pace that they do not have enough time to consolidate before new proposals sprout out.

Object-orientation and component-based solutions seem to be promising and the trend towards incorporating them seems to be settling at an appropriate pace. At the

same time, the complexity of these applications increases, and one of the reasons lies in the heterogeneous nature of the run-time and development-time aspects to be taken into account: architecture, security, attractiveness, quality of service, security, robustness, and so on.

The aim of the workshop was to discuss the research carried out in universities and industries to solve the problems related to the construction of e-commerce applications using object-oriented technology, focusing on real-world industrial experiences and innovative infrastructure for building e-commerce solutions (ECS). The main conclusions of the workshop can be summarised as follows: (i) quality-awareness is an essential cornerstone of ECS; (ii) semantic certification seems a promising tool whose benefits may be extrapolated from the traditional goods market; (iii) web services are going to be the next generation building blocks for ECS; (iv) traditional middlewares are fading away in the Internet arena; (v) aspect-orientation is the key to succeed in the management and development of the different concerns that interact in an ECS; (vi) agents play an important role because they are pro-active and autonomous.

WOOBS attracted a variety of researchers interested in applying their results to producing ECS. As a result, the workshop benefited from multiple points of view and different perspectives and approaches. Cutting down the programme to fewer presentations would have allowed more and deeper discussions, but we think that bringing together researchers from such different disciplines proved useful since the exchange of ideas and experiences was helpful in identifying a number of key topics and issues that need to be further researched and will probably help PhD students to identify a clear research topic in this area.

In this report, we go into details about the discussions and presentations and show how they relate to the above-mentioned conclusions. It is divided into five sections: Sect. 2 reports on the invited talk; Sect. 3 summarises the papers that were accepted; Sect. 4 reports on the colloquium that took place after the authors presented their papers; finally, Sect. 5 shows our main conclusions.

Next edition of WOOBS will be held in the framework of the fifth International Conference on Business Information Systems (BIS) in Poznań, Poland, during April 24–25, 2002 (`http://bis.kie.ae.poznan.pl/`). In this new edition, we will go into deeper details about the hottest topics we identified during this edition and will also report on the results obtained in the interim.

2 Invited Talk

Professor Doctor Kurt Geihs was the invited speaker of WOOBS. He is a professor of Computer Science at the Technical University of Berlin, and his research and teaching focus on distributed systems and operating systems. Current research projects focus on middleware and software component technology for open distributed systems, network and system management, agent-based systems, and infrastructures for Internet applications. He received a Diplom-Informatiker degree from the Technical University Darmstadt, Germany, a M.Sc. degree in Computer Science from the University of California, Los Angeles, USA, and a Ph.D. degree from the Technical University Aachen, Germany. He worked for the IBM European Networking Centre in Heidelberg, Germany,

before joining the academia. He was a professor of Computer Science at the University of Frankfurt until June 2001.

During his lecture, professor Geihs talked to us about the importance of Quality of Service (QoS) in the development of next generation e-commerce applications. It constitutes one of their cornerstones and challenges the established middleware design principles because they usually provide naive abstractions of the underlying machinery that are quickly broken as soon as QoS issues such as bandwidth fluctuation or partial failures are taken into account.

He discussed how distributed object systems that rely on a standardised middleware layer, CORBA in this case, can be extended so as to deal with QoS requirements in an feasible but still abstract way. He focused on the results of two projects called MAQS and QCCS, and presented their approaches, goals and preliminary results. A short abstract of these projects follows.

A Management Architecture for Quality of Service (MAQS). A part of the invited talk focused on how the results of the MAQS project can be used to integrate QoS concerns into an object-oriented middleware such as CORBA. Usually, their most publicised feature is that they provide high-level abstractions of the underlying distributed computational machinery by means of which a programmer can work with his or her objects as if they resided on the same machine. However, such a level of transparency is quickly broken as soon as the programmer comes down to earth and has to deal with partial failures, bandwidth fluctuation, security concerns, transmission errors, and so on. All of these problems need to be addressed during the construction of an ECS, and, therefore, it would be desirable for a middleware to be able to deal with them in an elegant, robust, easy-to-use way.

The MAQS project addresses the integration of QoS concerns into ECS, or more generally, distributed systems in a broad sense, at both design/ implementation time and run time. At design/implementation time, QoS specifications have to be provided, and they must be in an adequate format so that negotiation, monitoring, and adaptation is possible and easy to map into programming languages such as C++ or Java. At run time, QoS-aware interaction can be further decomposed into three steps: Establishment of a connection, interaction itself, and shutting down. During the establishement of an interaction, both clients and servers negotiate the level of quality they require and can provide, respectively, which mainly depends on the availability of resources; during a QoS-aware interaction, monitoring is used to ensure the server provides its services at the QoS level it promised (renegotiation may be needed if it cannot achieve such a level); when a client finishes using a service, the interaction is closed and the resources it used are freed and charged if necessary.

Geihs's team has recently built a framework prototype for QoS integration based on the results of the MAQS project. It relies on MICO, a public domain CORBA 2.3 implementation initially developed by his team, and its core is an extension to CORBA IDL called QIDL that offers two new constructs for defining QoS interfaces and assigning them to service interfaces. A QoS interface defines a conceptual artefact that must be implemented to ensure a given component implements a certain quality level. Its definition consists of a number of QoS parameters that define the state of such artefacts

or help define what a client requires and a server may provide, and a number of QoS operations that can be used to retrieve information or influence the behaviour of a QoS artefact.

To demonstrate the feasibility of this approach, Geihs's team have used the framework to build several QoS artefacts that include a simple multicast and group-membership protocol that can be used for voting and pooling, as well as a mechanism for streaming and load balancing. The results about load balancing are being applied in a joint project with a major German bank because they can help improve the response times of the investment banking applications it uses.

Current work focuses on completing the infrastructure they need to build real-world applications that need both monitoring, resource control, accounting, billing and charging. Their current negotiation component is also under hot improvement because it is only suitable for closed systems under the assumption that services do not optimise their profit using any knowledge about a client's preferences. Geihs's team is researching on strategies for open distributed systems where this assumption does not hold in general.

Quality-Controlled Component-Based Software Development. Another part of the invited talk focused on the QCCS project, which aims at researching a new approach for component-based software construction in which QoS concerns can be modelled independently from functionality. Professor Geihs thinks that quality is the only way to succeed in achieving a higher degree of reusability, adaptability and composability than traditional techniques.

QCCS deals with the following quality aspects: functional behaviour, structural properties, synchronisation properties, and use constraints. Components that have been designed according to the QCCS methodology will have proven properties which are specified in contracts and can thus be safely applied to build complex services from scratch by instantiating standardised frameworks.

Contracts are a core part of the QCCS methodology. They are the means a programmer can use to specify what his or her components can do and how good the results will be, or what they expect from other components. They can be classified into the following categories: (i) Basic contracts, which consist of the definition of a syntactic interface, (ii) synchronisation contracts, which specify how components residing on different nodes can collaborate synchronously, and (iii) QoS contracts, which describe issues such as performance or reliability.

Professor Geihs used the following example to explain further these concepts: consider, for instance, a database that stores replicated data on several nodes for the sake of robustness and fault-tolerance. Data cannot be replicated instantaneously, so the information at different nodes may be temporarily inconsistent due to network delays and bandwidth fluctuation. Using a QoS contract framework, a client can choose amongst several contracts the one that best suits its needs. A contract might ensure, for instance, that a query returns always the latest data (with a presumably high delay), whereas another might ensure a query returns the most up-to-date data found in less than, say, two seconds. Depending on the quality a client needs, a different contract may be selected dynamically at run time.

However, we are not likely to succeed in the ever-demanding component market unless we can deliver components that implement a given quality level in a short period of time. Thus, a technique in which QoS concerns are separated from functionality can significantly enhance maintenance and adaptation. This technique is popularly known as Aspect-Oriented Programming (AOP) and it allows to develop partial implementations that address particular aspects that can be woven into a single component later.

To merge contract-based design techniques and AOP, Geihs's team use the multiple view modelling capabilities of UML. The Object Constraint Language (OCL) is used to specify invariants, and pre- and post-conditions on the operations that constitute a contract. They have also defined several domain-specific stereotypes for dealing with quality concerns.

The methodology and tools are going to be validated using two real-world applications. The first one is a workflow system that aims at supporting business processes by providing tools that facilitate cooperation; the second one is a new Internet-based information system that helps users retrieve information about a certain geographical or administrative region.

3 Contributions

Over forty papers were submitted and peer-reviewed by two or three members of the programme committee. Only thirteen were finally accepted for presentation.

3.1 On Defining and Building Object-Oriented Views for Advanced Applications

R. Wrembel
Poznań University of Technology, Poland

Abstract. Contemporary information systems should allow to have access various sources of data maintained within a company/institution for the purpose of analysis. These data are often in different formats and have different complexity, e.g. relational, object-oriented, semistructured, HTML, spreadsheets, flat files. This data heterogeneity causes difficulties in accessing them. There are two approaches to the integration of heterogeneous data sources, namely, a virtual approach and a data warehouse approach. A very important mechanism applied in data integration is a view. For the integration of data having complex structure object-oriented views are very promising. In this paper we present: the concept of an object-oriented view for the application in object-relational data warehousing system, the issues concerning object-oriented view materialisation and maintenance, the prototype system that we have implemented, and some experimental results.

3.2 CBR-BDI Agents for an E-Commerce Environment

J.M. Corchado
University of Salamanca, Spain

Abstract. This paper shows how to build deliberative agents using case-based reasoning systems. Deliberative agents and Case-based Reasoning systems are presented. The

advantages and disadvantages of deliberative agents are outlined and then it is shown how to solve some of their inconvenients, especially those related to their implementation and adaptation. The Internet has emerged as one of the most popular vehicles for disseminating and sharing information through computer networks. A multiagent-based system for e-business, in which CBR-BDI agents have been used, is also presented and evaluated in this paper.

3.3 Agentspace as a Middleware for E-Commerce Applications

S. Ambroszkiewicz and T. Nowak
Polish Academy of Sciences, Poland

Abstract. Agentspace is an emerging environment resulting from process automation in the Internet and Web. It is supposed that autonomous software (mobile) agents provide the automation. The agents realize the goals delegated to them by their human masters. Interoperability is crucial to assure meaningful interaction, communication and cooperation between heterogeneous agents and services. In order to realize the goals, the agents must create, manage and reconfigure complex workflows. Our research aims at extracting a minimum that is necessary and sufficient for providing transparency between users and services, i.e. for joining applications as services in agentspace on the one hand and for using them by heterogeneous agents (on behalf of their users) on the other hand.

3.4 Intelligent Agents and XML – A Method for Accessing Webportals in Both B2C and B2B E-Commerce

J.R. Mühlbacher, S. Reisinger, and M. Sonntag
Johannes Kepler Universität, Austria

Abstract. In E-Commerce today webportals are important and also intelligent agents grow in significance. Our approach is to combine them in designing webportals and interfaces for both users and agents. In this paper we discuss the problems in automatically accessing portals and possible solutions for them through using OOM methods. The solution selected by us, using an XML-based standard and dynamically reconfigurable protocols, is described afterwards and the methods used are shown. We also briefly present an example, a webportal for sports information to give an impression of a practical application.

3.5 Coordination Technologies for Business Strategy Support – A Case Study in StockTrading

G. Koutsoukos[1], *T. Kotridis*[2], *L. Andrade*[1,3], *J.L. Fiadeiro*[3,5], *J. Gouveia*[1], *and M. Wermelinger*[3,4]
[1]*OBLOG Software, Portugal;* [2]*Accenture, UK;* [3]*ATX Software, Portugal;* [4]*Universidade Nova de Lisboa, Portugal;* [5]*Universidade de Lisboa, Portugal*

Abstract. In today's global and highly competitive business environments, to the question of whether technology is forming business or vice-versa, organizations are replying

by integrating their business and IT strategies, thus using technology to do business. Online business and virtual organizations is the trend as e-commerce numbers are increasing and the emerging wireless data technologies are fuelling the creation of new business opportunities. Flexibility, innovation and change have become critical success factors for every business organization. Business strategies have now, more and more, a short-term time horizon and, for the first time, can be put in practice by in-formation systems alone (strategic information systems). As a consequence, there is an increasing pressure for building software systems that are dynamically reconfigurable and adaptive to changes imposed either by technology innovation or new business needs. Unfortunately, OO languages and the technologies associated with component-based frameworks have fallen short of re-dressing this situation, which may explain why software teams are still struggling to compete with the fast business and technology evolution and, as shown by numerous scientific studies on large-scale software systems, more than 80% of the total cost of software development is still devoted to software maintenance.

In this paper we show how a new semantic primitive (coordination contract) that we presented previously as an extension to OO modeling languages, together with the design patterns that support its implementation over component-based platforms, can be used in support for a new approach to Business Modeling based on the definition of "software strategic libraries" and (re)configuration mechanisms that will deliver "business reactive" information systems. We present our case on the basis of some simple, but real-life examples from the Stock-Trading industry. We discuss why OO techniques such as inheritance and clientship are too static and white-box in order to model and implement such volatile assets and we present how coordination contracts can provide a more dynamic modeling and implementation alternative. We, also, illustrate how contracts promote the idea of building systems using libraries of stable components and strategic evolving components, allowing organizations to cope with the volatility and turbulence of today's business environments.

3.6 A DPE Architecture Supporting the Deployment of Distributed Applications

T. Li, A. Hoffmann, M. Born, and I. Schieferdecker
GMD FOKUS, Germany

Abstract. With the unprecedented increase of the complexity of today's distributed applications the deployment of such applications onto heterogeneous middleware platforms becomes more and more a challenge to the developer and administrator of such applications. Deployment of distributed systems encompasses the distribution as well as the configuration of their components. The Eurescom project P924 targets concepts, methods, and notations for the deployment of distributed component-based applications onto target middleware platforms. In this paper, we focus on this project's platform aspects. A DPE architecture to support the deployment of distributed application is elaborated.

3.7 Transaction Services for Business Objects Middleware

J. Nummenmaa[1] *and P. Thanisch*[2]
[1]*University of Tampere, Finland;* [2]*University of Edinburgh, Scotland*

Abstract. Transaction processing systems ensure that properties such as atomicity and isolation hold for transactions executing in a distributed, multi-user environment. It is now recognised that these transactional properties are useful for a much wider class of business object applications. However, it is notoriously difficult to develop robust software that provides such services in a failure-prone distributed environment. Consequently, do-it-yourself transactional facilities have been incorporated in the leading distributed objects middleware platforms. The use of such transactional facilities comes at a computational cost. Furthermore, some distributed objects applications are too complicated to benefit from these facilities. In view of this, we produce check-lists of criteria to determine whether a distributed business objects application can usefully exploit transactional services. We also discuss some potential problems.

3.8 Transparent Distribution in the Artavan Web Application Server

F. Sánchez, J.M. Murillo, R. Rodríguez, P. Bachiller, A. Gazo, A. Gómez, J.L. González, and M. Sánchez
University of Extremadura, Spain

Abstract. In this paper we present an experience using the Aspect Oriented Programming paradigm (AOP) to obtain a transparent and middleware independent object distribution in Artavan, a Web Application Server (WAS) developed in Teleserver, a Spanish company. The aim is allowing the programmers to serve applications using Corba or JavaRmi without mixing the distribution code with the functional one, promoting reusability and adaptability. This work is being developed under a contract between Teleserver and the University of Extremadura with the aim of doing applied research. This is only a part of the whole work where we are improving the WAS using other object oriented concepts.

3.9 A Performance Assessment Model for Web-Based CORBA Applications

J. Hosszú
Budapest University of Technology and Economics, Hungary

Abstract. This paper presents how discrete event simulation can be used for performance evaluation of distributed systems. general aspects of building the multi-layered object-oriented simulation model, sample application, and measurement results for various configurations are introduced.

The analysis focuses on scalability, throughput and latency of large scale distributed systems, which are critical performance issues in e-commerce applications. Using the method, performance of a system may be assessed without deploying it. However, the technology being considered rely on the Common Object-Request Broker Architecture, the model can be adopted to other distributed technologies as well.

3.10 On Dynamic Service Composition and Its Applicability to E-Business Software Systems

V. Tosic[1]*, D. Mennie*[2]*, and B. Pagurek*[1]
[1]*Carleton University, Canada;* [2]*The Bulldog Group Inc., Canada*

Abstract. This paper discusses dynamic service composition and its applicability to e-business software systems. It also presents the ICARIS architecture for dynamic service composition and its application to dynamic, on-demand establishment of security and trust relationships. Addition of security extensions to a B2C e-commerce system is used to illustrate dynamic service composition in ICARIS.

Dynamic service composition is the process of creating new services at run-time from a set of service components. Static (design-time or deployment-time) software composition is not flexible and agile enough to accommodate frequent runtime changes of requirements and/or operational circumstances that cannot be anticipated in advance. Dynamic service composition enables run-time construction of new, potentially unanticipated, services, while minimizing both human involvement and disruption of system operation. The potential benefits of greater system flexibility, agility, and availability are crucial in many e-business systems that have to cope with the constant change of technical and business circumstances and increased diversity and mobility of users, systems, and resources.

ICARIS is a general-purpose dynamic service composition architecture based on Jini, JavaBeans, and XML. As the research goal was to explore maximal reuse of existing technologies, the base technologies were not modified. However, the Jini Lookup Service was extended to support semantically rich XML service specifications and their advanced ("fuzzy") matching. In addition, ICARIS has built-in mechanisms for preventing failed compositions and unexpected runtime feature interactions. The ICARIS architecture supports three composition techniques: composite service interface, stand-alone composite service using the pipe-and-filter architecture style, and stand-alone composite service with the single body of code. ICARIS was successfully used in the Composable Security Application to introduce security services into applications that were not originally designed with security mechanisms. This experiment showed viability and feasibility of suggested concepts and demonstrated applicability of dynamic service composition in e-business systems.

3.11 An XML-Based E-Commerce Architecture Proposal

F.J. García, M.N. Moreno, and J.A. Hernández
University of Salamanca, Spain

Abstract. We present an e-commerce architecture proposal by means of which an organisation can easily jump on the virtual commerce bandwagon, expanding its marketspace to unexpected limits, whatever its size is. Aiming at easing the implementation, we propose the development of a software tool that allows the definition, publication and update of a catalogue of products, and the setting up of a web server architecture that allows clients to have access to it. To create a uniform environment, the architecture

needs to focus on a well-defined business sector, easy to expand to others where businesses are based on product catalogues. An e-commerce server has to provide end-users with services such as searching for a product in any published catalogue, shopping cart management, selling certificates, navigation help, and so on through a uniform, intuitive interface. The whole system is built over a self-maintained platform, able to configure itself when changes are incorporated by the enterprise, with a minimum interaction with the web master. To facilitate the access to the information, a visual, intuitive tool is provided to enable the definition of business logic aspects such as catalogue of products for sale, full descriptions, selling policies, and so on. The communication between the tool and the server is implemented using XML over secure protocols. The server provides the tool with the definition pattern for the products, and it will be the tool in charge of the maintenance of the virtual enterprise by sending all necessary data in the same language.

3.12 Using the Aspect Moderator Framework For Evolving Business Requirements

F. Akkawi, M. Lee, A. Bader, and T. Elrad
Illinois Institute of Technology, USA

Abstract. While the cost and complexity of the enabling technologies for e-commerce applications decrease, the design and development of the e-commerce applications are becoming more expensive and complex. This is mainly due to the lack of a formal design methodology that stresses the separation of concerns. There are many diverse concerns that if intermixed will make the resulting software system hard to maintain and scale and consequently force re-engineering. The Aspect Moderator Framework (AMF) is an aspect-oriented approach that solves these problems. The AMF attempts to address design concerns like concurrency, security, scalability, and reliability in a way to maximize code reusability and minimize the impact of these concerns on the stability, maintainability and reconfigurability of the e-commerce applications.

3.13 BMD NTCS: From COBOL to OOP

M. Knasmüller
BMD Systemhaus Ges.m.b.H., Austria

Abstract. Introducing object-oriented programming to old-style programmers is a rather hard task. This paper presents some experiences from BMD Steyr, Austrians leading producer of accountancy software. We have more than 50 developers; most of them were maintaining a character-based COBOL-product, an integrated system supporting all business aspects. Now they are implementing the NTCS project, a complete reengineering of the product using OOP. In the first years there were two main tasks: Firstly, implementing the necessary Windows tools, secondly, updating the COBOL programmers. In a first step we chose a development environment: Delphi. Using it, we implemented our own class library, where we had a strong look on efficient input possibilities, on equal appearance of all parts and on implementing as much as possible in this library. Using this strategy the implementation of the accountancy software itself can be done

in a short time. This not only because of the earlier release date, but also because of the reduction of the time in which the new software is developed and the old software must be maintained in parallel. Another important point is to turn the COBOL programmers to object-oriented programmers and there the best method is education, education, and once more education. We divide the necessary knowledge about object-oriented programming into three courses with weekly lectures and exercises. In the first part we learned all the necessary things that make a programmer excellent: type safety, procedures, pointers and most important data abstraction. Afterwards we introduced the terms classes, inheritance and dynamic binding. In the last part we explained the different classes of the class library. More information is available in the book "From COBOL to OOP", published by dpunkt.verlag. BMD NTCS is – in Austrian terms – a very large project with a duration of about five years and a budget of about ten million Euros. Now, we have already finished the implementation of two sub-projects, an archive of documents and a balance sheet program. We chose these parts of our software, because there the advantages of a Windows product are more important than a high optimised input possibility. Future work will concentrate on implementing all the other parts of our software system.

4 Colloquium

After presenting their papers, the authors engaged in discussion of some topics that were not covered previously. Dr. Corchuelo sparked off discussion and he pointed out that all of the discussion topics identified during previous sessions need to be addressed in the context of a new kind of systems he called Multi-Organisational Web-Based Systems, or MOWS for short.

Corchuelo – In recent years, the design of reference architectures for systems distributed on the Internet has attracted the attention of an increasing number of researchers and practitioners who have focused on platforms, languages, middlewares or interoperability concerns. The main reason for such a great interest is that this network has experienced a rapid shift from information and entertainment to electronic commerce, which has gained importance and grown exponentially. That is the reason why Multi-Organisational Web-Based Systems are becoming so usual and successful. They are composed of a relatively small number of coarse-grained web services provided and supported by different organisations that need to cooperate frequently. For instance: tourist agencies, banking systems, stock traders, virtual fishmarkets, and so on.

Ruiz – They are not new at all, but I agree in that traditional development techniques or methods for addressing the issues that arise in the context of ECS need a clearly identified context that is completely different from the traditional set. The Internet has overnight changed the way applications are developed and managed, so that it is necessary to have a clear description of this new scenario. I think that investing time, effort and money on studying these systems at an abstraction level that allows to have a good perspective over them is necessary.

Geihs – Can you, please, tell us about the main features of a MOWS and why you think they deserve special attention?

Corchuelo – I think the main features of a MOWS are the following: (i) They lack a middleware layer in the traditional sense because they are interconnected by means of the Internet, i.e., they use standard protocols such as HTTP or SOAP and languages such as HTML or XML to communicate; providing high-level services such as transactions, security, or persistence is far more difficult in this scenario than in traditional middlewares. (ii) They must be economically optimal; in traditional applications this may be a complex problem, but it is even more difficult on the Internet because the availability of services and their quality may change all of a sudden. (iv) They are multi-coordinated, in the sense that most ECS cannot be easily decomposed into client/server or message passing patterns; instead, several entities on the net coordinate, exchange information and collaborate to obtain a result. (v) They must be quality-enabled and proactive, in the sense that quality (not only quantitative QoS, but also qualitative quality) is essential and needs to be achieved in a highly automated way.
Ruiz – One of the main aims of the software industry consists of producing high-quality applications at costs that should be as reduced as possible. Unfortunately, building such applications keeps on being quite a handmade activity that lacks standardised, consolidated enough automated production methods. Furthermore, the Internet has entailed a great revolution in this industry, and it is driving electronic commerce activities at an ever-increasing pace. This has implied adapting existing methods and tools to the new resources and means the Internet provides, but at such a pace that they do not have enough time to consolidate before new proposals sprout out. That is why I think that studying MOWS as a separate kind of systems may help identify key issues that will conduct the development of future methods and tools.

Attendees agreed on the necessity of studying such systems as a new paradigm and the importance of the features that make them unique. Some of them pointed then out that enabling technologies such as UMTS will, undoubtedly, play a fundamental role in next generation ECS and will surely change the way we conceive MOWS at present. However, what about the infrastructure software able to take advantage of such technology to build ECS?

Koutsoukos – MOWS are then ECS that will exploit the idea of web-services, and mobile phones will be the essential device to have access to them. I wonder why a new technology for delivering services. Was not enough with CORBA?
Corchuelo – Essentially, web-services are built on top of standard Internet protocols and languages, and that is the reason why they are becoming so successful. You do not have to buy an ORB, instead you use HTTP/SOAP as a transport protocol, XML as a data language and tools such as BizTalk to map a scheme into another if it does not suit your needs. Do not you think middlewares such as CORBA are fading away?

This question sparked off a vivid discussion. We are in no doubt that middlewares such as CORBA are widely used nowadays. However, they are more often used to build the internal part of an electronic service than to interface with others on the Internet. When such a middleware is used to interface with other systems, all of the organisations involved usually agree on using the implementation provided by the same vendor to avoid connectivity problems and performance penalties. Some attendees pointed out

important industrial experiences in which CORBA was also used to interface with other systems, but many agreed in that CORBA will be mostly used as an internal infrastructure, whereas protocols and languages such as HTTP, SOAP or XML will surely rule the future "Internet middleware".

We then moved to the problem of combining different aspects into a single ECS.

Corchuelo – Let us, please, stop for a moment, and let us think about the many inter-related aspects that can influence an ECS. Just a middleware, call it CORBA or HTTP/SOAP, is not enough to engineer complex ECS.
Akkawi – Yes sure: persistence, synchronisation, security, reliability, robustness... there are so many aspects that are not functional but have an impact on how an ECS may behave that it is necessary to have tools to eject them out of the functional core to enhance modularity, maintainability, adaptation...
Corchuelo – Those aspects are very important and it is obviously necessary to deal with them independently from the functionality a web service has to implement. However, I would like to point out other interesting problems with which you can come up easily when designing an ECS. They are not usually dealt with using aspects, but I am sure they are also aspects, namely: information retrieval, high-level quality attributes such as portability or price, coordination, chiefly multiparty interaction, intelligence...
Reisinger – What do you mean by aspect-oriented information retrieval?
Corchuelo – Distribution, heterogeneity and rapid change are the key features of MOWS, because the Internet market changes with every passing minute. Never has it been so wide the spectrum of available information sources. Thus, to keep our MOWS economically optimal it is often necessary to change the web services on which they rely. The problem is that if our business rules depend on the exact primitives we need to use to retrieve information from such heterogeneous sources, we are then likely to spend a lot of time and effort every time they change. As Akkawi pointed out, the idea is to eject the code responsible for retrieving information out of the functional code, thus keeping it clean and independent from the actual information sources it uses.

We then turned to the important role of agents in such problems.

Nowak – Intelligent agents may help a lot in achieving such a goal.
Ruiz – Right, but there is an important problem to be addressed: how to bridge the gap between ontologies that have been developed independently on the Internet? I mean, how can an agent retrieve information from a new site whose ontology is new or does not exist at all, as is the usual case?
Reisinger – That is the problem of interoperability in a broad sense, and I think the solution consists of having ontology repositories in which we can store, update and manage ontologies in different application domains together with mappings that allow us to bridge the semantic gap between them, if possible. ebXML is a good example. It is a modular suite of specifications that enables enterprises to conduct business over the Internet because it provides them with a standard method to exchange business messages, conduct trading relationships, communicate data in common terms and define and register business processes.

Geihs – Having those ontologies is obviously a part of the problem, but I think that mobility is also necessary to achieve such goals in an efficient manner.
Akkawi – That is then a pretty complex aspect.
Wrembel – Yes, and homogenising views may help a lot.
Nowak – That justifies the need for agent platforms supporting the creation, management and reconfiguration of complex workflows. It is not enough with infrastructure services, but agents need more advanced platforms that bring them higher-level services that allow them to interoperate meaningfully. Introducing homogenising views is a good approach, by the way.

Mobile agents led to a new issue: semantic certification.

Tosic – Another interesting point is that of composing the services each agent provides. In this case, dynamic composition techniques are essential.
Corchuelo – Sure, and I think your paper on dynamic service composition is a valuable contribution in this sense because MOWS can be basically understood as "big" services composed of many building blocks that can be viewed as lower-level services. The problem is how can you assure a composition of services, being them provided by web artefacts or agents, performs the tasks it is supposed to perform and delivers its functionality at a given quality level.
Ruiz – The key here may be certification. Guaranteeing that a service implements a given functionality at a given quality level is, in my honest opinion, absolutely impossible unless we test it. There might be people who would call me heretic because I do not use a state-of-the-art theoretical method, but I think testing is the only method you can use to certify a service conforms to its specification. I think that certifying a component with regard to a syntactic, protocol, semantic, and quality interface is a good solution. In the future, there might be new companies specialised on certification. Thus, component assemblers might build applications from certified components and services, and thus interoperability checks would be easier. However, it is necessary a great deal of standardization to achieve this goal, i.e., we need standardised ontologies and maps between different versions of the same ontology, if possible.
Corchuelo – Remember that is the approach SETCO uses to certify a product implements the SET protocol.
Koutsoukos – It is an interesting idea to certify components in such a way. You might have a guarantee certificate as if you were buying a traditional good. However, I am not aware of any initiative that uses it, except for the SET case.
Nummenmaa – Certificates are also a good idea in the context of transactions because you can certify a system performs its task coherently and you have a guarantee certificate you can use to ask your providers for responsibilities.
Tosic – I think certification has a good potential in conjunction with a registry such as UDDI. This is the name of a group of web-based registries that expose information about web services different vendors have developed (or are developing). Including such

certificates might influence how a programmer trusts a service, and I agree on guarantees being important in the software industry.

This raised many questions about quality concerns.

Ruiz – Thus, a certificate may be viewed as the description of a number of high-level quality attributes such as robustness, reliability, security and the like. I mean that all of these attributes are valuable and then must be an essential part of every component or web-service. Thus certificates must be in a form so that an ECS can seek for the web services it needs in a highly automated way.
Akkawi – Thus, you mean that they should be described by means of an aspect-language and woven to build components.
Corchuelo – It would be nice to be able to encapsulate those aspects into independent packages that can be woven. However, the problem here is not to implement them, but to expose information about quality features so that every time a web service needs to use another, it can decide which implementation to use according to how they maximise a number of quality criteria. It is, we need a language for implementing quality aspects and another language for "documenting" them and making them available to others. Professor Geihs told us about a proposal in this direction, although the quality aspects with which he works are relatively low-level with regard to quality attributes such as complexity, price, resolution, availability or such.
Akkawi – Which are, in your opinion, the key problems incorporating such "reflective information" about quality aspects may solve?
Corchuelo – (i) Automatisation, i.e., the ability of an ECS to seek for the web services it needs autonomously; (ii) conformance, i.e., its ability to select a set of services that fulfill the quality level they need so as to achieve the quality level they promise; (iii) and economic optimality, definitely.
Ruiz – Sure. However, you have forgotten several important issues that need to be addressed: (i) temporal awareness, because quality requirements may change depending on the day, hour or month; (ii) negotiation mechanisms, so that clients can weak or strength their requirements according to the offers of a server; (iii) multiparty awareness, because typical problems in the context of ECS usually involve several parties that need to agree and cooperate to achieve a goal.

5 Conclusions

During presentations and discussions we identified ECS as a special kind of so called Multi-Organisational Web-based Systems (MOWS). The increasing demand and complexity of ECS based on the Internet, the need for reducing development and running costs, and the new opportunities that web services offer are the main reasons behind the forecast popularity of MOWS. In such systems, several organisations offer end-user web services such as banking, booking or information retrieval, whose functionality relies on a number of back-end web services provided by other organisations. To achieve optimal MOWS, they must be open systems that must be able to search for the services on which they rely automatically and dynamically.

Summing up, the rapid evolution of the Internet demands an ever-increasing ability to adapt to a medium that is in continuous change and evolution, thus automatisation is the key to succeeding. Studying MOWS from an abstract point of view and adapting current methodologies and tools to the their specific features may help identifying key problems that are inherent to these systems.

Next, we summarise some of our conclusions, some of the problems we identified, and the solutions we sketched at WOOBS:

Infrastructure:

1. Traditional middlewares such as CORBA will keep playing a prominent role in the design of back-ends.
2. It seems that protocols such as HTTP or SOAP will replace traditional middlewares in the Internet arena.

Development:

1. Aspect-orientation is one of the keys in succeeding in the development of complex web-services in which many mutually-influencing aspects are merged and combined.
2. Agents may be viewed as an enabling technology to build ECS that can search for information on the Internet.
3. Information retrieval may be expressed as an independent aspect, and will help evolving software. However, it is necessary to adapt agent services to the Internet setting so that they can develop their full potential in ECS.
4. The semantic gap between the ontologies different data sources use is one of the main problems to be addressed. Banks of ontologies and mappings between them are necessary.
5. Multiparty coordination models are necessary, chiefly in ECS in which several parties need to negotiate and reach agreements on-line.

Certification:

1. It is necessary to guarantee a service implements what its specification states. Certification seems a promising tool.
2. Certification allow clients to use guarantee certificates as if they were buying traditional goods. This concept is novel in the software arena.

Quality:

1. Optimality and conformance seem to be the cornerstones of quality.
2. Quality documents must be formalised so that they allow automatic checking. Furthermore, quality documents must include both QoS clauses on attributes such as delays, jitting or mean time to fails, and higher-level clauses on attributes such as cost per connection, security level or binding policies.
3. It is necessary to extend current architectural description languages so that they can take quality requirements into account, and it is necessary to extend current distributed run-time platforms such as CORBA, COM+ or .NET so that quality documents are first-class elements.
4. It is necessary to include quality monitoring mechanisms so that it is possible to check if the provider of a web service fulfills the quality level agreement it reached with a customer.

The importance of MOWS in the current e-world sufficiently justifies the study of a new paradigm called WOP (*Web-Oriented Programming*). The goal is to provide software architects with a number of tools to solve critical problems such as *optimality* and *conformance*, which are both closely related to the automatic management and negotiation of quality requirements.

Resources

- BIZTALK. `http://www.biztalk.org`
- ebXML. `http://www.ebxml.org`
- MAQS project. `http://www.vsb.informatik.uni-frankfurt.de/maqs/Project.html`
- QCCS project. `http://www.qccs.org`
- SET (Secure Electronic Transaction). `http://www.setco.org`
- UDDI (Universal Description, Discovery and Integration of Business for the Web). `http://www.uddi.org`
- UMTS (Universal Mobile Telecommunications System). `http://www.umts-forum.org`
- Web services (by Microsoft). `http://msdn.microsoft.com/msdnmag/issues/0900/WebPlatform/WebPlatform.asp`
- Web services (by Sun). `http://www.sun.com/software/sunone`

Acknowledgements. First and foremost, the organisers of WOOBS would like to thank all of the participants of the workshop, including the members of the programme committee, for their hard work before, during and after the workshop; they are responsible for its great deal of success. We also thank the organisers of the ECOOP conference for their help and continuous support. If it had not been for them, WOOBS would not have been possible.

Finally, we would also like to thank Atef Bader, because we got inspiration for our evaluation form from a previous one he designed, Microsoft and Telvent (Abengoa Group) for their generous financial support, Fundación Dintel for their generous supply of material, Sadiel for their support in project GEOZOCO, and Ciber Sur for their coverage of the event.