QoS in Multichannel IS: the MAIS approach

Cinzia Cappiello[*], Paolo Missier[+], Barbara Pernici[*],
Pierluigi Plebani[*], Carlo Batini[+]

[*] Politecnico di Milano, Milano, Italy
{cappiell, pernici, plebani}@elet.polimi.it

[+] Università di Milano Bicocca, Milano, Italy
{pmissier, batini}@ disco.unimib.it

Abstract. Multichannel adaptive information systems can deliver services over different types of networks and access devices. Since different providers may offer either functionally equivalent or similar services, the quality of service (QoS) is a fundamental element of the service selection and the subsequent negotiation. This paper presents a taxonomy for quality dimensions, its application to different models of a multichannel information systems, and outlines requirements for a language to describe quality elements in a multichannel environment as modeled in the MAIS (Multichannel Adaptive Information System) Project.

Index Terms— QoS taxonomy, QoS languages

1 Introduction

According to the Service Oriented Computing (SOC) paradigm [4], e-services may be accessed by different users and requestor applications, after being selected from a registry on the basis of their characteristics. In registries such as UDDI [18], service selection is based mainly on functional characteristics, categories, and providers descriptions. In a multichannel environment, it is reasonable to assume that different providers may offer either equivalent or similar services, while QoS provides a differentiator among these offerings and a user may select the most suitable service on the basis of its QoS requirements. Subsequently, the user may engage into a negotiation process with the provider of the selected service in order to define a Service Level Agreement (SLA) that contains the user's quality levels requirements.

QoS issues and SLA definition have received much attention during recent years, mostly in the networking [10, 11] and the middleware communities [7, 22] where local metrics have been defined without concern for end-to-end service metrics. Among the few studies that propose taxonomies or deal with end-to-end application

QoS requirements, we mention [5] and [16]. In [5], the authors propose an integrated framework to characterize the production, transmission and consumption of data through a single medium. In [16], a taxonomy for QoS specification is presented. The primary categories identified are metrics and policies. Metrics are performance, specifications, security levels, relative importance, while policies are level of service and management policies.

Relevant work in the area of languages for QoS management in the Web Services context includes the Web Service Level Agreements (WSLA) language and runtime architecture, and the Web Service Offerings Language (WSOL). WSLA [21] includes a language for the definition of quality dimensions and low-level metrics, the functional composition of high-level parameters, and the expression of service level objectives, i.e., logical conditions on the parameters. A Service Level is defined as a particular configuration of the dimensions, and serves as a template for a negotiated Service Level Agreements (SLA) between service user and service provider[1]. A SLA is part of the contract established between customer and provider, and contains a formal expression of the guaranteed service conditions in a given time period. The WSLA runtime enforcement architecture provides monitoring of the parameters during operations, either by the user or by a trusted third party, and invocation of recovery actions when contract violations occur. These may include the assessment of penalties, as defined in the SLA, and the activation of provider-side components such as workload balancing or admission control modules.

WSOL [17] is suitable for the definition of quality dimensions, their metrics and quality constraints. The language does not formalize the contract terms between user and provider defining service levels but contains constructs to define simple quality constraints on each quality dimension. Further, WSOL provides constructs to specify functional constraints in the form of pre- and post- conditions, which monitor the correctness and completeness of the provided service. Using WSOL, it is also possible to express access rights to regulate the usage of the different services.

The goal of the MAIS (Multichannel Adaptive Information Systems) project (January 2003 – December 2005) is the development of models, methods and tools for the design and implementation of multichannel adaptive information systems that can deliver services over different types of networks and access devices [12]. In this way the same service could be accessed, for example, using web-based systems, call centers, GSM networks, using networks and devices with different characteristics.

A goal of the MAIS project is to provide an adaptive execution environment for e-services at all levels, where adaptation goes from applications to adaptive low power consumption processors, through a series of architectural layers. The characteristics and interplay of all these architectural layers and components are being studied in the project. Quality of service definition plays a major role in defining the relationships among the components.

The goal of this paper is to provide a general framework for the definition of quality of service dimensions in the MAIS Project, that can be applied to all architectural levels. The requirements for a general quality specification language for e-services in

[1] In the most recent version, multi-party SLAs are also defined.

an adaptive environment are defined and current QoS definition approaches for e-services analyzed and compared.

This paper is organized as follows. Section 2 provides a general introduction to adaptive multichannel information systems as developed in the MAIS project. In Section 3, an overview of the different quality dimensions and technical characteristics is provided, and a taxonomy for the identified dimensions is proposed. Section 4 defines the requirements for a suitable QoS representation language in order to understand and give a precise representation of the metrics and dependencies associated with QoS dimensions in a general framework for e-services in an adaptive multichannel environment. Finally, in the last section, conclusions and future work are presented.

2 Multichannel Adaptive Information Systems

The MAIS project studies a flexible highly adaptive environment for delivering services through different distribution channels. From a user perspective, such an architecture allows the invocation of services registered and published in a private UDDI Registry owned by the MAIS architecture. A user can interact with the services through several devices (e.g., Smartphone, PDA) and a variety of wired and wireless networks. In MAIS, the device represents the user end point of the channel, i.e. the element which allows the communication with the service provider [14]. The service provided by the MAIS architecture can be simple or composite. In the former case the service can be considered as a typical software component, whereas in the latter case the service can be viewed as a composition of several simple, or even composite, services. In particular, composite services may include other MAIS services, as well as third-party services that are not part of the MAIS architecture.

In order to offer such capabilities, the architecture is composed of different modules, characterized by complex inter-relationships. The reference model of the MAIS architecture consists of four models:

- E-service Model: specifies the functional characteristics of the provided service, as well as their QoS dimensions as specified by the providers [3];
- Context Model: provides information about the user, such as location, preferences and abilities attributes;
- Channel Model: describes the technological characteristics of the user invocation environment, including device and network parameters;
- Architectural Model: provides a reference model for the definition of the architectural components of the systems. Components are reflective [1] and component parameters, including QoS parameters, may be queried and set.

The MAIS environment provides flexibility and adaptation in all its parts: at service level, through reconfigurable processes and flexible invocation of e-services, at context level, allowing the user to dynamically change his/her profile, and at channel level, allowing variable access devices and adaptive networks. The reflective architecture allows accessing such variable configurations, monitoring quality parameters, generating events to notify critical changes at different levels, and provides some basic services to improve reliability and availability.

A general framework [13] has been proposed to model quality in the MAIS architecture. The framework calls for service, network, and the device providers to organize as communities in order to define the relevant quality dimensions for services, networks, and devices respectively.

In some areas, such as in networking, quality parameters are defined with reference to available standards [11, 6]. For devices, some proposals are emerging in producer communities, such as in the System and Devices Working Group of the DMTF [6]. For application level services, QoS is domain specific, and quality needs to be defined in the specific service provider communities. Considering the same community, services, network and devices can be easily compared. The framework also considers a set of rules able to calculate the quality perceived by the user starting from the values of the quality dimensions defined by the service provider and affected by channel characteristics. This type of quality, called Quality of Experience [8], represents the basis on which user and provider build the contract agreement defining the terms and the conditions about the exploited service. Such a contract forces the provider to enact a set of adaptation strategies able to keep what promised in the contract. Consequently the adaptation, the fundamental feature of the MAIS project, is provided at all levels (context, channel, service functionalities and architecture), with the goal of globally respecting the agreed quality levels.

A detailed description of the quality enforcement architecture goes beyond the space limitations of this paper. As a preliminary issue, here we only introduce a classification of quality dimensions, and outline requirements for a quality definition language.

3 A Taxonomy of Quality of Service Dimensions

We have described in the previous section the architecture of the MAIS systems, in terms of layers, devices, strategies and services that allow flexible composition and adaptation of multichannel services for users characterized by a rich variety of profiles and preferences.

In order to meet the requests of the user, the service provider can act on a wide spectrum of qualities, defined at all levels of the architecture. Correspondingly, the user can express its own requirements not only in terms of functional needs, but also in terms of quality requirements. We have defined a MAIS Quality of Service (QoS) Registry to collect, define, and relate all quality dimensions managed in the MAIS adaptive environment. In this section we initially provide a general overview of the Registry.

The QoS Registry has been built according to a bottom up process, extracting from the deliverables, and research reports produced in the MAIS project the quality dimensions considered relevant for each part of the adaptive environment. Over 240 QoS dimensions have been identified and described in the Registry according to the following schema, inspired by the ISO 9126 Software Quality model [9]: characteristic (name and definition), possible sub characteristic (name and definition), metrics (name and definition), method of measure, possible range of values.

Quality dimensions are not distributed uniformly among the different MAIS models, since about 45% pertain to the Architectural Model, 25% to the Channel Model, 15% to the E-service Model, 8% to the Context Model, and the remaining ones are of uncertain attribution.

Concerning the Architectural model (about 100 dimensions defined), most of the quality dimensions refer to device qualities (altogether one third of the total set of qualities), and can be classified more properly as technical characteristics. A wide set of devices is taken into account such as: CPU, memory, printer, speaker, audio controller, media access, speaker, microphone, modem, keyboard and other input devices, PDAs, and various types of assistive technologies, such as pointers, screen readers and others. Typical technical characteristics concern performance, availability, response time, power, frequency, and, when applicable, security characteristics such as authentication.

Concerning the Channel Model (about 50 quality dimensions defined), relevant dimensions are interoperability, maintainability, reconfigurability, security, reliability, throughput, bit rate, lifetime, fairness, flexibility, delay, robustness, transfer and release error, resilience, electromagnetic pollution, power consumption.

Relevant qualities of the e-service Model (about 30 quality dimensions defined) are usability, mentioned in four different sublayers of the e-service model, accessibility, adaptivity, security, availability, price, time related qualities such as response time, and different qualities related to data and information provided by the service, such as accuracy and timeliness (see also [20]).

Finally, in the Context Model (17 quality dimensions assessed), relevance is given to accessibility, adaptability, adaptivity, accuracy, perceivability, and performance issues.

In order to check the completeness of the quality dimensions set with respect to standard sets provided in the literature, we have verified that device and network technical characteristics are widely covered, together with related metrics and methods of measure, while software qualities cover only a limited set of characteristics and subcharacteristics of the ISO 9126 standard; they are accessibility, learnability, operability, related to usability, analyzability and changeability related to maintainability and fault tolerance related to reliability. In fact, quality aspects typical of the software development process, such as testability and portability, are not issues being addressed in the project.

The QoS Registry is now used in MAIS as a monitoring tool for achieving completeness in the approach to quality and real integration among the different groups involved in the project. First of all, it can be adopted in order to reach completeness, i.e. providing for each quality characteristic a definition, a metric, a method of measure, and achieve consistency, i.e. eliminating synonyms, etc. Furthermore, in the first phase of the project, quality dimensions have been defined independently, and only a negligible number of them have been related to each other in order to characterize the dependencies existing among them, such as, for example, dependencies among response times of different devices and network layers. As we will see in the next sections, we are now working to guarantee these levels of representation.

As discussed in Section 1, the quality model proposed for the MAIS project focuses on the service negotiation between service provider and user. The QoS Registry classifies the quality of service dimensions needed at different phases according to the SOC paradigm. At *service discovery time*, the goal of the negotiation activity is to find a service provider that may satisfy the required quality levels. The MAIS architecture finds the suitable provider and defines a contract. During *service provisioning (run time),* the MAIS environment provides adaptation at different levels to maintain the values of all quality parameters defined in the contract inside their admissible ranges.

We propose a *taxonomy* for Qos Registry that focuses on the negotiation phase and regards mostly the provider side (both as service provider and as invocation platform for service provisioning), analyzing the interaction with the user both at discovery time and at run time, according to the preferences contained in the user profile (Figure 1).

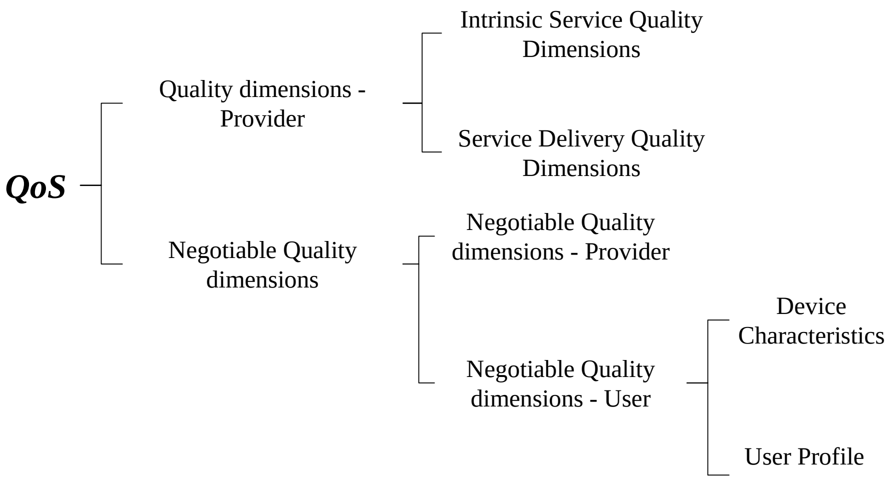

Figure 1 - Taxonomy of Quality of Service Dimensions

From the **provider perspective**, we distinguish between *non-negotiable* quality dimensions that characterize the services supplied, and *negotiable* dimensions related to the resources used by the provider to offer services. These dimensions can be contracted between the user and the provider. The user can express preferences or define constraints on these dimensions in the service selection phase. On the provider side, negotiable quality dimensions include, for example, bandwidth, price, and response time. They are used during the negotiation phase both at discovery time, in order to choose one of the functionally equivalent services, and at run time, for achieving the agreement on the desired quality level of the selected service.

On the **user side**, negotiable dimensions are used in the negotiation phase performed at discovery time, in order to determine the user-specific access configuration.

The configuration is based on the preferences contained in the user profile and according to the technical characteristics of devices made available by the user.

In Table 1, we present how the proposed taxonomy may be applied to quality dimensions provided in the e-service Model. Analogously, we might define quality dimensions for the Context, Channel, and Architectural Models.

	Quality dimensions - Provider		**Negotiable Quality Dimensions**		
			Negotiable Quality dimensions - Provider	**Negotiable Quality dimensions - User**	
	Intrinsic Service Quality Dimensions	**Service Delivery Quality Dimensions**		**Device Characteristics**	**User Profile**
e-service Model	Reliability Accountability Traceability & Audibility Non repudiation Data Reliability	Robustness Security	Bandwidth Price Security Accuracy Completeness Response time Provisioning Time Service Availability Timeliness (Data) Availability (Data)		Usability (Use context) Usability (pleasure)

Table 1 - Taxonomy for the e-service Model quality dimensions

Non-negotiable dimensions in the e-service Model are, for example, service and data reliability as intrinsic characteristics of the service and robustness and security of the application as criteria related to the resources used for the service delivery. At the e-service Model there are many general (non domain-specific) negotiable characteristics, such as the service price, security and availability. Response time and the provisioning time are two important efficiency indices of the service. Response time is referred to the time interval between the time instant in which the user sends the service request and the time instant in which the user gets a response. The provisioning time instead considers the time interval required for the complete service delivery. The two indices might coincide for services that are completely delivered in an electronic way.

As regards the user side, there are only dimensions related to usability contained in the User Profile. Indeed, the quality dimensions associated with the user devices are located at the Channel model level.

Note that many quality dimensions belonging to different models are related to each other. For example, response time depends on both channel characteristics (i.e. networks dimensions) and architectural characteristics (i.e. dimensions related to resources involved in the provider delivery architecture). As another example, usability depends on many different characteristics such as response time, accessibility, performance, and adaptability. Future work will exploit these dependencies among different models.

4 QoS Language Requirements

As discussed in the previous sections, the quality of a service is used as differentiator in the service selection and the subsequent negotiation activity is which the user specifies quality constraints and the values of various negotiable quality dimensions. This requires a sufficiently expressive language for defining the dimensions subject to negotiation, for expressing constraints on their values, and possibly for describing inter-dependencies among non-orthogonal dimensions. In this section, we express the QoS language requirements, and, as a result we discuss the adoption of a combination of existing languages, namely WSLA [21], WSOL [17], and OWL [19].

[REQ1] The first requirement is the **definition of the quality dimensions**. As suggested in [13], a quality dimension is defined by the pair <*name, admissible_value*> where the name identifies a parameter and *admissible_value* is a set of typed values within which the parameter can vary. For some dimensions, in order to simplify the task of expressing requirements, the admissible values can be reduced to a simple discrete set, like “high, medium, low”. Examples of quality parameters are the following:

- bandwidth, [1Kps..512Kps],
- encryption, [40bit;64bit; 128bit],
- resolution, [320x200; 800x600;1024x768;1240x748],
- latency, [10ms...500ms],
- security-level, [high, medium, low].

Furthermore, considering the proposed taxonomy, a fundamental requirement in the definition of the quality dimensions is the representation of the classification structure. A possible way to describe the hierarchical relations between the dimensions is to define a reference class to describe the generic quality dimension set, and then introduce subclasses for each of the quality dimension set identified, by applying the proposed taxonomy. A first analysis performed to compare the different languages pointed out that OWL is the most suitable language for expressing the classification structure, since it has constructs able to satisfy all the requests related to the definition of the data quality dimensions. In the following, an example of the description of the relations between dimension classes coherently with the proposed taxonomy using OWL is presented:

```
<owl:Class rdf:ID="QualityDimension" />

<owl:Class rdf:ID="Negotiabledimension">
  <rdfs:subClassOf rdf:resource="#QualityDimension "
  />
   ...
</owl:Class>
```

[REQ2] The second requirement is the **definition of the metrics** associated with the dimensions together with the precise specification of the measurement function. It is also necessary to consider that the value of a dimension may be defined recursively as a function of underlying metrics, or of other dimensions. Note that the assessment functions are best expressed using the WSLA language, as they are native to the model, although in principle they may be defined using each of the other languages we have considered, defining suitable extensions.

[REQ3] We need to take also into account **dependencies among different dimensions**. The quality dimensions could be in relation with each other because a metric may be defined as a function of underlying metrics. The relationship between two dimensions could only consist in a simple dependency that allows expressing a correlation between the quality dimensions. OWL is the adequate descriptive language to express this type of dependencies. Indeed, it is possible to declare different types of correlations (i.e. symmetric or transitive) that allow maintaining all the links among the dimensions making possible an in-depth analysis of the service quality. In the following example, we illustrate how it is possible to declare dependencies between dimensions using OWL. The dependency can be declared as a transitive property:

```
<owl:ObjectProperty rdf:ID="DependentOn">
  <rdf:type rdf:resource="&owl;TransitiveProperty" />
  <rdfs:domain rdf:resource="#QualityDescriptor" />
  <rdfs:range rdf:resource="#QualityDescriptor" />
</owl:ObjectProperty>
```

For example, the service response time depends on the communication speed which in turn depends on the network bandwidth:

```
<QualityDimension rdf:ID="ResponseTime">
   <DependentOn rdf:resource="#Speed" />
</QualityDescriptor >

<QualityDimension rdf:ID="Speed">
    <DependentOn rdf:resource="#Bandwidth" />
</QualityDescriptor>
```

[REQ4] Finally, considering the negotiation phase, the languages have to be able to express the **quality requirements**, useful both at discovery time and at run time. As described in Section 1, WSLA includes definition of the Service Level Objectives, i.e., logical conditions on the parameters, to provide a particular configuration of the negotiated Service Level Agreements (SLA) between service user and service provider. The SLA contains a set of quality constraints on the selected and negotiated quality dimensions. WSOL is instead suitable for the definition of quality constraints in terms of values required for each quality dimensions. The language is suitable for expressing quality requests, but does not formalize the contract terms between user and provider.

Table 2 provides a summary of the strengths of the different languages along the identified requirements:

	Quality Dimension definition	Metrics definition and functional composition of metrics	Semantic relationship among dimensions (e.g. transitivity)	Quality requirements expression
OWL	Supports the description of the inheritance relation between quality dimension classes	Extensible	- Extensible set of semantic relationships - Built-in functional properties (e.g. transitivity)	Extensible
WSLA	Supported	Built-in functional composition language	Extensible	- Built-in first-logic predicate expression language - Management of service level agreement
WSOL	Supported	Extensible	Extensible	- Built-in first-logic predicate expression language - Definition of functional constraints (pre and post-conditions)

Table 2 - Comparison of OWL, WSLA, WSOL along the QoS language requirements

The requirements described as *supported* are satisfied by the language with built-in language constructs. The requirements described as *extensible* should be satisfied by the language with suitable language constructs built as ad-hoc extension. In order to take advantage of the strengths of each language, a first solution could be using the three languages, relating them with the name-space technique. Alternative solutions could consist in choosing one of the languages and extending it to satisfy all the identified requirements.

5 Concluding remarks and future work

The MAIS project aims at designing and implementing a multichannel adaptive information system that can deliver services over different types of networks and access devices. The MAIS reference model is composed of four different models that analyze different and specific issues of the system. Currently, several parts of the MAIS environment are being designed and developed. The current focus is in providing adaptive user environments for e-service execution, designing user interfaces and user interaction of the basis of quality and context parameters, and in the development of an adaptive e-service orchestration environment, focusing on flexible e-service invocation, selecting e-services on the basis of their functional and quality characteristics (channel characteristics such as negotiable bandwidth and interface parameters have been considered). The prototypes are based on Java enabled portable devices, cell phones, and wireless connections such as Bluetooth, WiFi, GPRS to provide context-dependent e-services, such as videos on demand, and supporting emergency notification procedures.

Particularly, as regards the QoS, we have considered and represented uniformly in the MAIS QoS Registry all the quality dimensions identified for each level identified in the Reference model and proposed a taxonomy focused on the negotiation task. The definition of the quality of service is one of the main non-functional parameters to consider for the selection of the provider and the specific service. A survey of the suitable languages for quality description has been conducted and different alternatives have been identified along project requirements. Future work will focus on:

- Selection and formalization of a complete quality description language
- Construction and experimentation of a complete taxonomy of dependencies and functional relationships among quality dimensions in all selected vertical MAIS applications, i.e. tourism, risk management, cultural assets
- Design of the agreement monitoring and enforcement environment;
- Definition of a methodology to model functional and quality aspects of e-services in the adaptive MAIS environment, and in particular composed e-services
- Design and realization of the complete MAIS run time service environment, that takes into account quality dimensions together with all the other dimensions of the MAIS environment, i.e. user profile, context, cost of service. A first contribution, addressing the choice of services with optimal ratio between cost and quality in data quality research area is [2]. Architectural issues have been discussed in [15].

Acknowledgements

This work has been partially supported by the Italian MIUR FIRB Project MAIS.

The authors wish to thank Ms. Franca Durante for her contribution in extracting quality dimensions from MAIS documents and all MAIS participants for discussions on issues related to quality aspects in the project.

References

1. Adorni, M., Arcelli, F., Campanini, S., Limonta, A., Melen, R. Raibulet, C., Simeoni, N., Tisato F. *The MAIS Reflective Architecture*. Deliverable Work

Package 3.1.1, Politecnico di Milano, 2003, http://www.mais-project.it (in Italian).
2. Avenali A., Bertolazzi, P., Batini C., Missier P. A formulation of the data quality optimization problem in Cooperative Information Systems. *Proceedings of International Workshop on Data and Information Quality (DIQ 2004)* in conjunction with CAiSE 04, Riga, Latvia
3. Baresi, L., Bianchini, D., De Antonellis, V., Fugini, M.G., Pernici, B., Plebani, P. Context-aware Composition of E-Services. *Proceedings of TES03 (Technologies for e-Services) Workshop* in conjunction with VLDB 2003, Berlin, Germany, September 2003.
4. Communications of the ACM (CACM), *Special issue on Service Oriented Computing*, October 2003
5. Cambpell, A., Aurrecoechea, C., Hauw, L. A review of QoS architectures. *Proceedings of the 4th IFIP International Workshop on Quality of Service (IWQS '96)*, Paris, March 1996.
6. Distributed Management Task Force Standards (DMTF), http://www.dmtf.org/standards.
7. Frølung, S., Koistinen, J. *QML: A Language for Quality of Service Specification*. Technical Report HPL98-10, HP Labs, HP Software Technologies Laboratory, 1998.
8. Khirman, S., Henriksen, P. Relationship between Quality-of-Service and Quality-of-Experience for Public Internet Service. *Proceedings of the 3rd Workshop on Passive and Active Measurement*, Fort Collins, Colorado, USA, March, 2002.
9. International Organization for Standardization. *ISO 9126, Standard for Software Engineering – Product Quality*. http://www.iso.org.
10. International Telecommunication Union (ITU). *Terms and definitions related to quality of service and network performance including dependability*. ITU Recommendation E.800 (08/94), 1994.
11. International Telecommunication Union (ITU). *Communications Quality of Service: A framework and definitions*. ITU Recommendation G.1000 (11/01), 2001.
12. MAIS (Multichannel Adaptive Information Systems) Project, http://www.mais-project.it
13. Marchetti, C., Pernici, B., Plebani, P. A Quality Model for Multichannel Adaptive Information Systems. *Alternate Tracks Proceedings of 13th International World Wide Web Conference (WWW2004)*, New York City, NY, USA, May 2004.
14. Maurino, A., Pernici, B., Schreiber F. A. Adaptive channel behavior in financial information systems. *Proceedings of Ubiquitous Mobile Information and Collaboration (UMICS 2003)* in conjunction with CAiSE 03, Velden, Austria
15. Maurino, A. Modafferi, S. Pernici B. Reflective Architectures for Adaptive Information Systems. *Proceedings of Service-Oriented Computing - ICSOC 2003, First International Conference,* Trento, Italy. pp. 115-131
16. Sabata, B., Chatterjee, S., Davis, M., Sydir, J., Lawrence, T. F. Taxonomy of QoS Specifications. *Proceedings of the 3rd Workshop on Object-Oriented Real-Time Dependable Systems - (WORDS '97),* February 1997.

17. Tosic, V., Pagurek, B., Patel, K., Esfandiari, B., Ma W. Management Applications of the Web Service Offerings Language (WSOL). *Proceedings of CAiSE 03 (15th International Conference on Advanced Information Systems Engineering)*, Velden, Austria pp. 468-484.
18. Universal Description, Discovery and Integration (UDDI), www.uddi.org.
19. W3C, OWL Web Ontology Language Overview, available on http://www.w3.org/TR/owl-features/
20. Wang R. Y., Strong D.M. Beyond Accuracy: What Data Quality Means to Data Consumers. *Journal of Management Information Systems*, Spring 1996, Vol.12, No.4, pp.5-34.
21. WSLA Project, http://www.research.ibm.com/wsla.
22. Zinki, J.A., Bakken, D.E., and Schantz, R.E. Architectural support for quality of services for CORBA Object Systems, 1997, vol. (3)1.