

# An Ontology for IT Services

Jorge Freitas, Anacleto Correia and Fernando Brito e Abreu

QUASAR Research Group, Department of Informatics,  
Faculty of Sciences and Technology, Universidade Nova de Lisboa, Portugal  
j.freitas@fct.unl.pt, accorreia@gmail.com, fba@di.fct.unl.pt

**Abstract.** The survey of related work on ontological studies in the IT services domain, allowed us to identify some serious limitations on the current state-of-the-art across a set of six maturity dimensions. To mitigate that shortcoming this paper proposes a formal ontology for IT services. To grant preciseness to this ontology we have expressed well-formedness rules with the OCL constraint language. The proposed ontology is then instantiated to illustrate its validity in addressing realistic examples.

**Keywords:** Process Engineering, Model Engineering, Ontologies, IT Service Management

## 1 Introduction

The competition on the global market is already obliging many service providers to think in terms of quality of service (QoS) and service level agreements (SLA), when establishing their contracts.

On the other hand, the need to offer more sophisticated and complex services often forces service providers to participate in consortia, where subcontracting takes place. The latter implies that everybody should use well defined terms, that is, with a shared meaning within the consortia and also understood by clients. Despite the ongoing efforts on establishing a body of knowledge on IT infrastructures [1], those efforts do not yet provide a basis to formal definition and assessment of IT services in general. For that purpose we first need a shared and consistent understanding of the concepts involved in this area. This is what ontologies are for. An ontology identifies (and clarifies) the core concepts as well as their interdependencies.

Most of the work published in the IT Services (ITS) area are either centered on the identification of best practices (e.g. CoBIT, ITIL) or focus on service oriented architectures (SOA) and web services. In this paper we are concerned with a more generic and technology-independent view on IT services, applicable whenever a service can be identified and their quality characteristics can be described. We present an ontology with a large application spectrum that provides an abstract, compact and formal approach that covers aspects from services implementation to their management

This paper is organized as follows: In section 2 we describe the proposed ontology. Next, in section 3, we instantiate the ontology with a realistic case study. In section 4 we present a survey of related work, organized according to a framework that includes a set of maturity dimensions and we finalize our paper with some conclusions and the outline of future work.

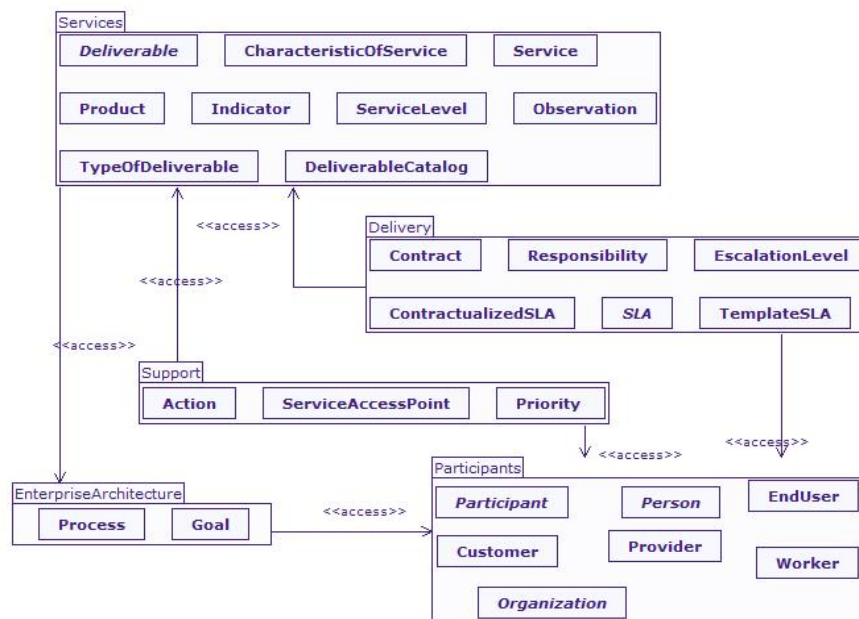
## 2 Ontology description

### 2.1 Introduction

In this section we propose a generic ontology for IT services. We aim at supporting services at different abstraction levels such as infrastructure management services, software outsourcing or even web services. We also wanted to cope with generic services concepts such as SLAs, QoS or business processes. For modeling purposes we have chosen UML since it is the most widely used modeling language in the IT industry nowadays. UML provides the required formalization to express complex constraints (using OCL) and also provides the required modularization constructs to organize a complex domain into several fairly independent parts.

### 2.2 Global view

The proposed ontology is composed of several packages, as represented in Fig. 1, where dependencies are shown. In the following subsections we will briefly describe each of the represented packages. The modeling of services integration upon enterprise architectures is currently being addressed and will be presented in an upcoming paper. A similar attempt to perform this integration is presented in [11].



**Fig. 1- ITS ontology packages**

The example shown in Fig. 1 is oversimplified since it is a general view. Next we will detail each of the represented packages. OCL [2] invariants are used to enforce ontology's well-formedness.

### 2.3 Participants package

Participants can be individuals or organizations. The latter can be customers or providers (suppliers). The notion of supply chain (subcontracting) is provided, in Fig. 2, by the unary association in the Organization. Individuals can be workers or end-users and be affiliated, or not, to an organization. For instance a home client using a video-on-demand service would be modeled as an EndUser. The hierarchical dependencies among staff members are represented in the unary association in the Person. An organization can offer several service access points. Several provider workers can be assigned to a given access point.

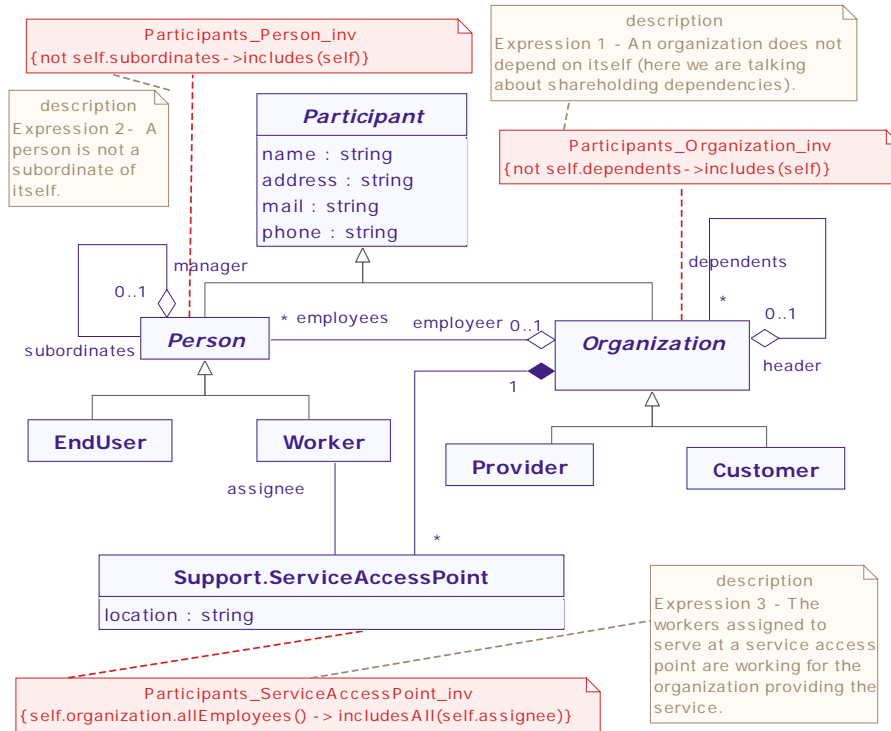


Fig. 2 - Participants package

### 2.4 Services package

Since services can be associated with product deployments, we have generalized the concept deliverable, to include both services and products. Deliverables can (i) be chosen from a catalogue of services, (ii) be hierarchically composed and (iii) be categorized according a set of types.

Deliverables can be characterized by a set of characteristics (e.g. performance, availability). The latter can be hierarchically decomposed and be quantitatively evaluated by means of a set of indicators (e.g. response-time, downtime). Each SLA defines the acceptable values of some statistics calculated upon the indicators. Those values are compared with the corresponding ones obtained from the observations of the indicators on the context of the same SLA.

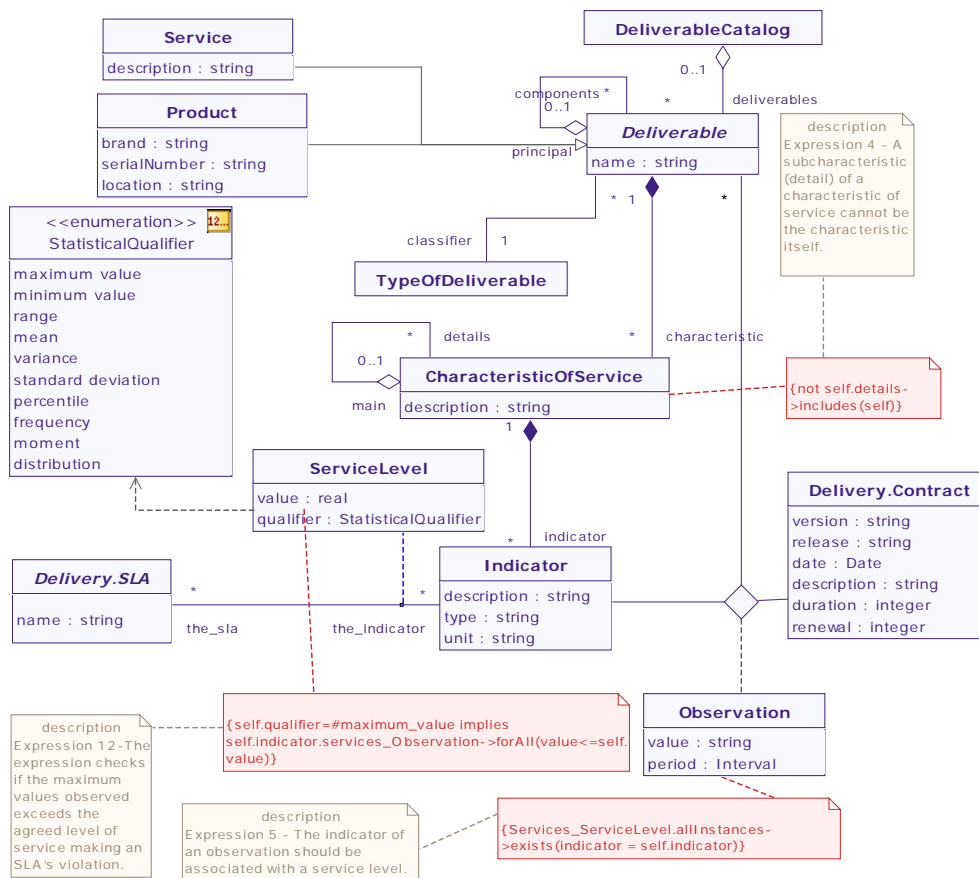
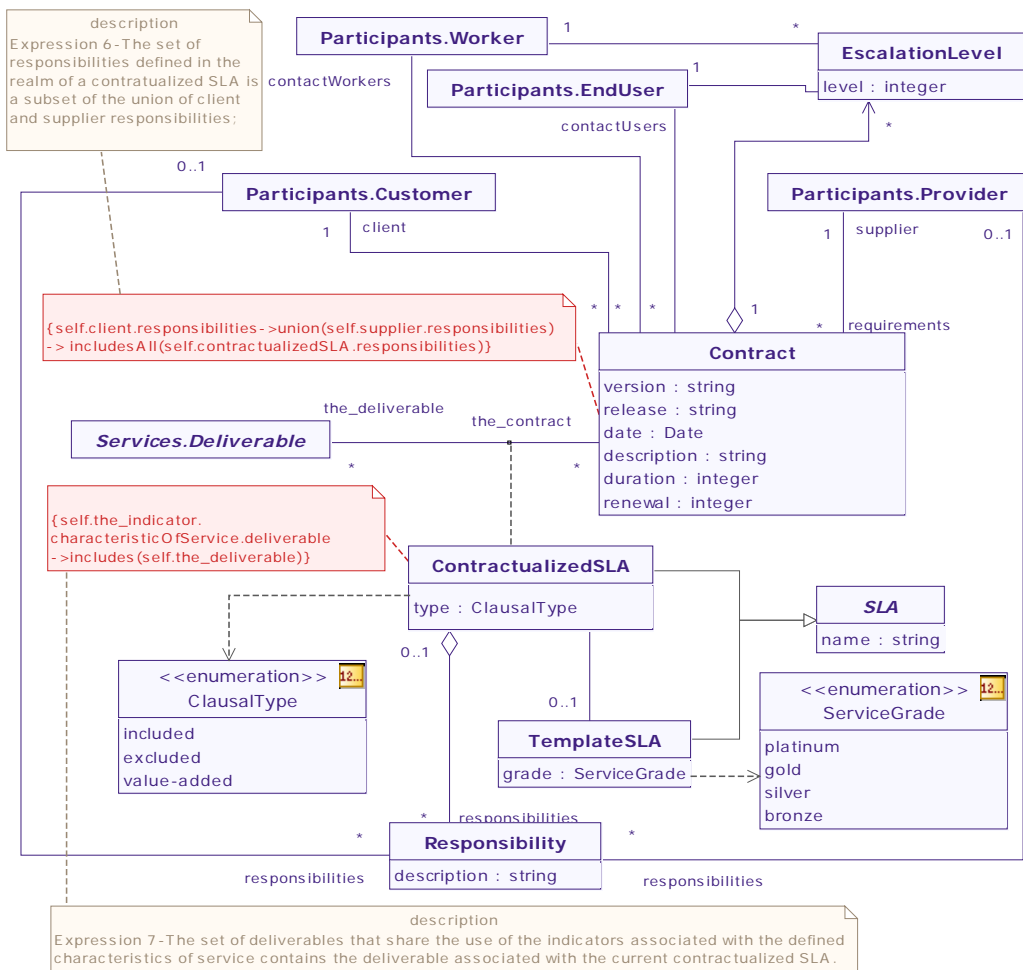


Fig. 3 - Services package

## 2.5 Delivery package

Usually, service designers define a set of standardized SLAs, for instance creating different cost/benefit alternatives (e.g. platinum, gold, silver and bronze). The former can be used as a template for specific SLAs to be contractualized in the realm of a contract between a client and a provider. A set of deliverables can be explicitly included (automatically or under request) or excluded from a contract. All the services covered by an SLA have responsibilities that must be assumed by the customer or by the provider. The escalation level provides a basis, both on the provider and clients sides, for solving conflicts on services delivery.



**Fig. 4 - Delivery package**

## 2.6 Support package

A service access point (e.g. call center, help-desk, on-line trouble-ticket system) can deal with several deliverables and the same deliverable can be dealt with in several access points. End users can generate actions (e.g. incident reporting, suggestions for improvement, queries for assistance) by using a service access point. Those actions can also be received and prioritized by a worker, who also records the action resolution or postponement, thereby affecting the action status.

## 2.7 Enterprise Architecture package

Organizational goals lead to the implementation of several processes that should somehow be aligned with those goals. Processes can have interdependencies that can be represented as a directed graph (notice the previous and next roles). Processes consume some deliverables (inputs) and produce others (outputs). Both processes and deliverables can be decomposed hierarchically. This package is still very incomplete and will be the focus of our attention in the near future.

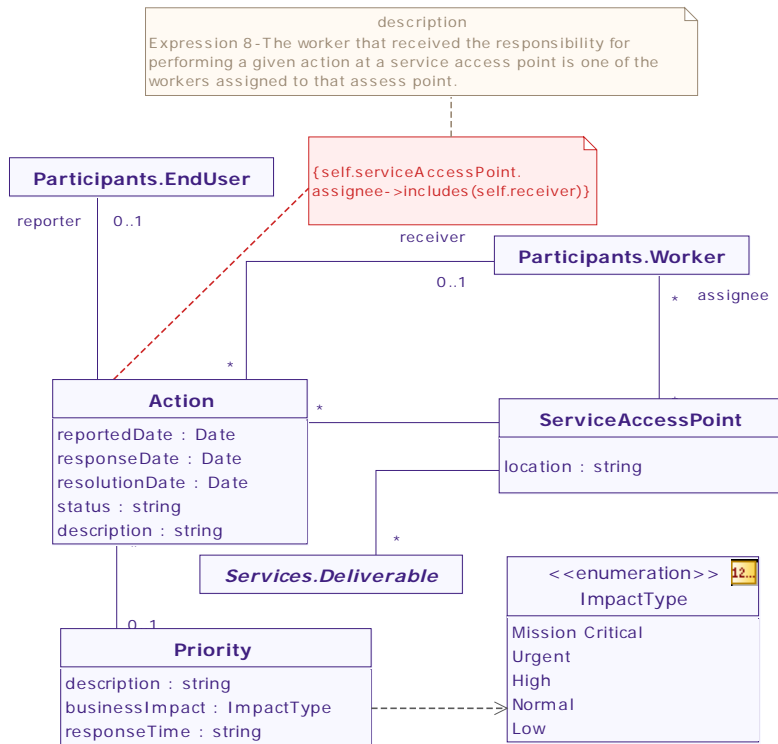


Fig. 5 - Support package

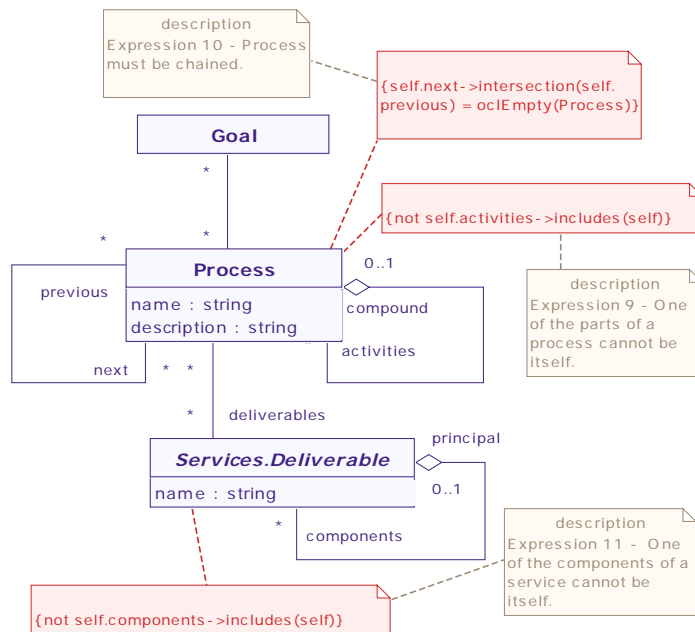


Fig. 6 - Enterprise Architecture package

### 3 Case study

In this section we instantiate the ontology entities belonging to four of the packages described in the previous section, namely the *Services*, *Support*, *Participants* and *Delivery* packages, by using UML object diagrams.

#### 3.1 SLA contract creation

The contract has some descriptive attributes (e.g. version, release, date, and comments.), as can be seen in Fig. 7. Other properties of the contract are the description (Hosting and associated IT services), starting date, duration and established renewal period. In the SLA contract negotiation there were several participants involved, such as the *Company X* from the client side and the *Information Technology Services Company* (ITSC), from the provider side. Client's representatives were the responsible by departments directly related with the contract purpose (e.g. the *CIO* (*Chief Information Technology*) and the *CEO* (*Chief Executive Officer*)). Provider representatives were the *Project Leader*, the *CIO* and the *Client Support Manager* (*Peter*). Peter is the person for problems solving on the *escalation level* number 1 (a mechanism to solve problems) from the supplier's part also works on the service access point that is a user's Helpdesk, which address is *South Street 2*. *John* is responsible for solving problems at level 1 from the customer's part (Fig. 7).

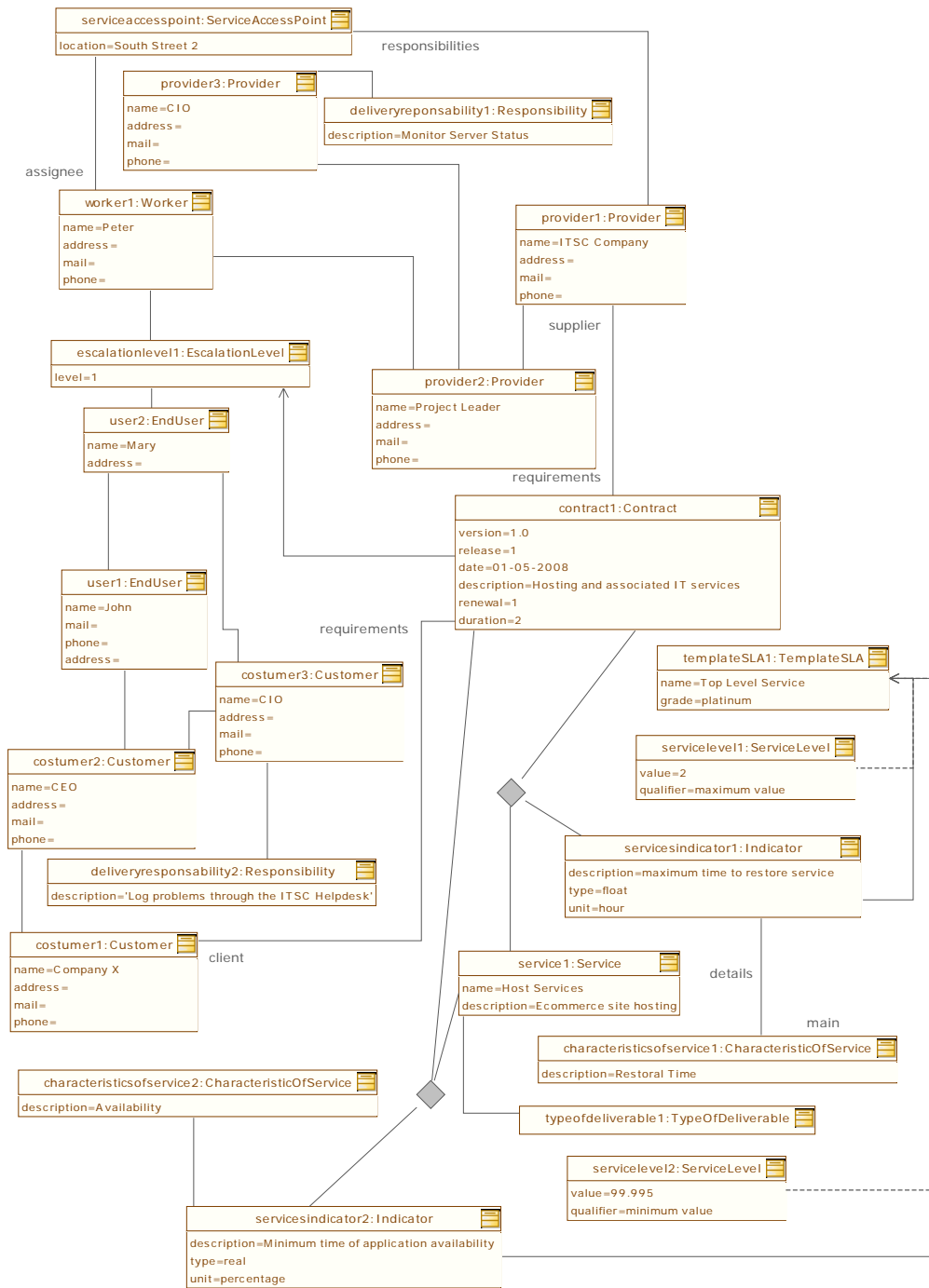
Under the terms of the contract, the client and the provider have a list of obligations that they must comply with. For the contractualized SLA named *sla2*, the client is obliged to "log problems through ITSC help desk staff" and the provider to "monitor server status".

The contract has a detailed description of supported services and responsibilities assigned to the provider, namely *monitoring server status*, etc. Each of the services has some characteristics. For instance *hosting services* has characteristics such as *availability and restore service* depicted Fig. 7. The characteristic is described by some indicators such as *maximum time to restore the service* and *Minimum time of application availability*. The latter is then constrained by a *service level mean of 99.995%* per month.

#### 3.2 SLA compliance

Two observations were added to the model, one on the indicator to the restore time and another on availability. The observed values of Qos in a given period, was 3 hours to restore the service and the availability of the ecommerce's application was 99.996%, as can be seen in Fig. 8.

In the case of the restore time the observed value was 3 hours and in the SLA's contract was defined a maximum value of 2 hours; in this case there was a SLA violation. By other hand the availability observed (99.996%) of the application was superior to what is defined on the contract (99.995%); in the case of this indicator the contract compliance is verified. The SLA compliance is verified by the OCL expression 12 in Fig. 3.



**Fig. 7 - SLA contract creation**

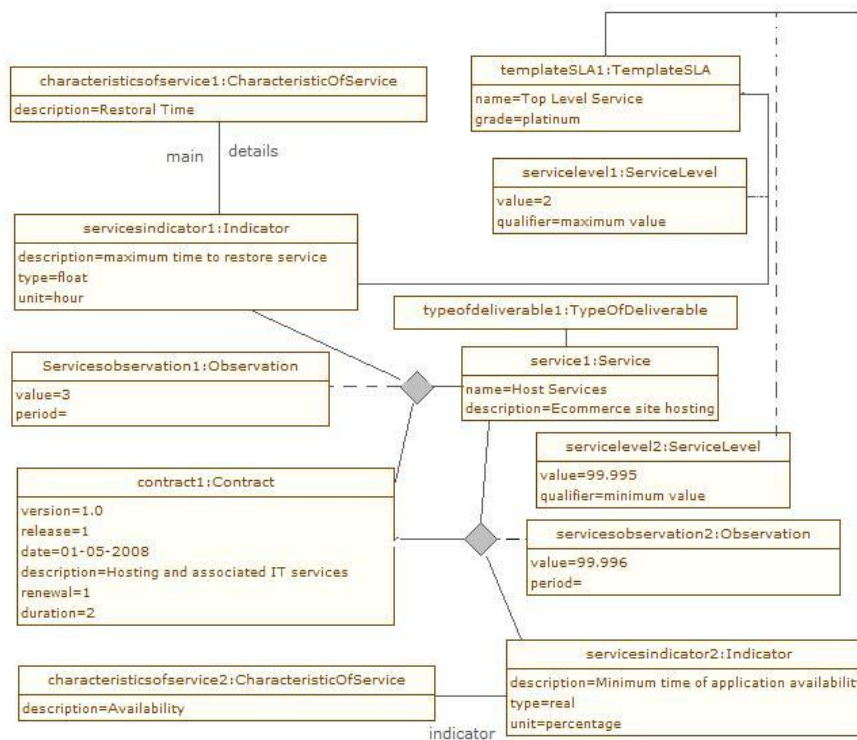


Fig. 8 - SLA observed QoS values

## 4 Related work

### 4.1 Proposed taxonomy

To provide a more systematic and objective approach to assess and compare related work proposals, we have developed a taxonomy that includes a set of ordinal scale descriptors along the following six dimensions in the range [0, 3] as seen in Table 1: Domain of Application (*DoA*), Service Definition (*SED*), SLA Definition (*SLA*), Model's Validation (*VAL*), Life Cycle Coverage (*LCC*) and Service Compliance (*SEC*).

**Table 1 - Maturity dimensions and corresponding levels**

Level	DoA	SED	SLA	VAL	LCC	SEC
0	Very specific and technical	Natural language	None	Nothing done	None	None
1	Specific	Structured	Informal	Partial exemplified	Covers Partially	Informal
2	Medium	Defined	Defined	Largely tested	Most part	Defined
3	General	Formal	Formal	Totally tested	Total	Formal

The *DoA* dimension refers to the extent to which the proposal fits into a global landscape of IT services ontologies, and which areas of it does it cover (e.g. Customer/Supplier, Service, SLA, QoS, etc). Since we want to measure the proposals extent of applicability, then, the broader the domain of application the better. Identified categories, by increasing level of maturity are:

0. *Very specific and technical* – The proposal is based on a very specific area (e.g. security) and covers specific technical issues; 1. *Specific* – the proposal focuses on a partial ITS domain (e.g. SLA); 2. *Medium* – the proposal covers several but not all ITS domains; 3. *General* – the proposal covers the vast majority of relevant ITS domains.

The *SED* dimension refers to the degree of preciseness used in defining the concepts used in ITS. This preciseness is required for reducing the ambiguity. The proposed scale is the following:

0. *Natural language* – concepts are presented in a rather informal, unstructured way, in natural language; 1. *Structured language* – natural language is used, but in hierarchical blocks and concepts are defined top-down; 2. *Defined* – there is an underlying ontology, but model semantics is fuzzy due the lack of formality; 3. *Formal* – a formal language such as OWL [3] or OCL [4] is used to grant precision to the ontology.

The *SLA* dimension represents the accuracy in SLA specification. This accuracy is important if we want to produce automatic compliance verification mechanisms. The proposed scale is:

0. *None* – There is no mention to SLAs; 1. *Informal* – SLAs definition is totally informal; 2. *Defined* – There is a generic model for SLAs; 3. *Formal* – A formal language is used in SLA definition.

The *VAL* dimension represents the degree of validation by example performed by the ITS ontology proponents. This validation is important to clarify the proposal's practical feasibility. The proposed scale is:

0. *Nothing done* – the proposal is only conceptual; 1. *Partial exemplified* – small examples are provided; 2. *Largely tested* – the model is instantiated with some consistent examples; 3. *Totally tested* – the model is instantiated in a general way with several consistent examples.

The *LCC* dimension represents the extent of the services' life cycle covered by the proposed ontology. The proposed scale is:

0. *None* – no reference to services' lifecycle is made; 1. *Covers partially* – at least one phase is considered; 2. *Most Part* – several phases are covered; 3. *Total* – all the life cycle is explicitly covered.

The *SEC* dimension represents how the compliance between contracted and provided services is treated. After all, SLAs would be useless if that verification could not be performed. The proposed scale is:

0. *None* – this issue is not handled; 1. *Informal* – this issue is informally handled; 2. *Defined* – this issue is modeled generically; 3. *Formal* – a formal model is provided for this issue.

## 4.2 Survey results

We have used the previous taxonomy to survey a set of related work proposals, as summarized in Table 2. Due to space constraints we can only include the grading justification for our own proposal, which is the following:

*DoA* – The application domain covers Customer/Supplier, Service, SLA, QoS, etc.

*SED* – The ontology is presented in a structured and formalized notation, using UML and OCL.

*SLA* – SLAs are a relevant part of our model and it is specifically formalized with OCL expressions.

*VAL* – Although we are working towards providing real-world application examples of our ontology, we have only presented small ones in this paper.

*LCC* – The proposal’s lifecycle coverage goes from market introduction stage or requirement definition (Enterprise Architecture package) to service completion.

*SEC* – Our ontology models expected and observed values, as well as the definition of characteristics and indicators. We have also presented an OCL-based compliance verification scheme.

**Table 2-Maturity levels of related proposals along the several dimensions**

	DoA	SED	SLA	VAL	LCC	SEC
<i>Our proposal - An Ontology for IT Services</i>	2	3	3	1	2	3
Niessink et al. [5]	0	0	2	1	2	2
Garsch. et al. [6] [7]	3	2	2	1	3	1
Skene et al. [8]	3	2	3	2	0	2
OMG [9]	1	0	1	3	0	3
Teyssié [10]	0	0	3	3	0	3
ITIL v3 [11-14]	1	1	2	1	3	2
e-TOM 7 [15, 16]	1	1	2	1	1	1
CMMI [17-19]	1	1	2	1	3	2
Su et al. [20]	1	0	2	1	3	2

A generic ITS ontology should cover all six dimensions largely. As we can see from Table 2, none of the reviewed proposals does that. Globally, our proposal appears to be the most well-balanced, namely if we exclude the VAL criterion, on which we are currently working.

## 5 Conclusions and future work

We have proposed herein an ITS ontology, that:

(i) intends to cover the vast majority of relevant ITS domains, such as ITS participants (e.g. clients and providers), SLA definition, service delivery, service support or enterprise architecture integration;

(ii) defines services and SLAs precisely with the help of the OCL constraint language;

(iii) is validated by the instantiation of an example that clarifies ontology semantics;

(iv) covers most parts of the services lifecycle, from their inception to their deployment;

(v) allows the formal definition of service compliance verification, also by using OCL.

We are currently addressing the issues of integration of services in enterprise architectures in the line of the work of other proposals such as [21]. We also intend to instantiate the ontology with real-world SLA contracts and observations of indicators describing service characteristics, to assess large-scale service compliance.

## 6 References

1. Office\_of\_Government\_Commerce: The Official Introduction to the ITIL Service Lifecycle Book. TSO (The Stationery Office) (2007)
2. OMG: UML 2.0 OCL Final Adopted specification. Technical report (2003)
3. McGuinness, D.L.Harmelen, F.v.: OWL Web Ontology Language Overview. Technical report (2004)
4. OMG: Unified Modeling Language: OCL (version 2.0). Technical report (2003)
5. Niessink, F.Vliet, H.v.: Software maintenance from a service perspective, *J. Softw. Maint: Res. Pract.*, vol. 12, pp. 103-120 (2000)
6. Garschhammer, M., Hauck, R., Hegering, H.-G., Kempter, B., Langer, M., Nerb, M., Radisic, I., Roelle, H., and Schmidt, H: Towards generic service management concepts - A service model based approach. In: 7th International IFIP/IEEE Symposium on Integrated Management (2001)
7. Garschhammer, M., Hauck, R., Kempter, B., Radisic, I., Roelle, H., and Schmidt, H: The MNM Service Model – Refined Views on Generic Service Management, *Journal of Communications and Networks*, pp. 297-306 (2001)
8. Skene, J., Lamanna, D., and Emmerich, W.: Precise Service Level Agreements. In: 26th International Conference on Software Engineering (ICSE'04) (2004)
9. OMG: UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms - OMG Available Specification, version 1.0. Technical report (2006)
10. Teysié, C.: UML-based Specification of QoS Contract Negotiation and Service Level Agreements. In: ICNICONSMCL Internacional Conference on Systems and Internacional Conference on Mobile Communications and Learning Technologies (2006)
11. Iqbal, M., Nieves, M.: ITIL Service Strategy. TSO (2007)
12. Lacy, S., MacFarlane I.: ITIL Service Transition. TSO (2007)
13. Loyd, V., Ruud, C.: ITIL Service Design. TSO (2007)
14. ITIL - IT Infrastructure Library, <http://www.itil.co.uk>
15. Forum, T.: Enhanced Telecommunications Operations Map - eTOM - Estrutura de Processos de Negócio - Versão 6.0. Book Enhanced Telecommunications Operations Map - eTOM - Estrutura de Processos de Negócio - Versão 6.0. TeleManagement Forum (2006)
16. Forum, T.: Enhanced Telecom Operations Map (eTOM) - The Business Process Framework v7.1. In: Forum, T.(ed.). Book Enhanced Telecom Operations Map (eTOM) - The Business Process Framework v7.1. TeleManagement Forum (2007)
17. Team, C.P.: CMMI® for Development, Version 1.2. Book CMMI® for Development, Version 1.2. CARNEGIE MELLON® UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE CMU/SEI-2006-TR-008, ESC-TR-2006-008 'edn.' (2006)
18. Team, C.P.: CMMI® for Acquisition, Version 1.2. Book CMMI® for Acquisition, Version 1.2. SEI Administrative Agent (2007)
19. Team, S.U.: Appraisal Requirements for CMMI®, Version 1.2 (ARC, V1.2). Book Appraisal Requirements for CMMI®, Version 1.2 (ARC, V1.2). SEI / CARNEGIE MELLON UNIVERSITY (2006)
20. Su, X., Bolzoni, D., and Eck, P.v.: Understanding and Specifying Information Security Needs to Support the Delivery of High Quality Security Services. In: Int. Conference on Emerging Security Information, Systems and Technologies (SECUREWARE'2007), pp. 107-114 (2007)
21. Braun, C., and Winter, R.: Integration of IT Service Management into Enterprise Architecture. In: 2007 ACM symposium on Applied computing, pp. 1215 - 1219 (2007)