

Project Number: **215219**
 Project Acronym: **SOA4ALL**
 Project Title: **Service Oriented Architectures for All**
 Instrument: **Integrated Project**
 Thematic Priority: **Information and Communication Technologies**

D3.4.7 Defining extensions to WSMO for capturing contextual information

Activity N:	Activity 2 – Core R&D Activities
Work Package:	WP3 – Service Annotation and Reasoning
Due Date:	M18
Submission Date:	14/09/2009
Start Date of Project:	01/03/2008
Duration of Project:	36 Months
Organisation Responsible of Deliverable:	OU
Revision:	1.4
Author(s):	Pierre Grenon OU
Reviewers:	Carlos Pedrinaci OU Barry Norton UIBK

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination Level		
PU	Public	PU

Version History

Version	Date	Comments, Changes, Status	Authors, contributors, reviewers
1.0	05/08/2009	Main body, focus on introduction, sketch of formal part	Pierre Grenon (OU)
	05/08/2009		Informal review by Carlos Pedrinaci (OU)
1.1	17/08/2009	Developed formal part with examples Added graphical illustration	Pierre Grenon (OU)
	26/08/2009		Review by Barry Norton (UIBK)
1.2	30/08/2009	Language corrections Corrected and simplified formal presentation using WSML only	Pierre Grenon (OU)
1.3	08/09/2009	Final formal presentation alignment with WSML	Pierre Grenon (OU)
	08/09/2009		Comments by Barry Norton (UIBK)
1.4	08/09/2009	Final correction	Pierre Grenon (OU)
1.5	14/09/2009	Final editing	Malena Donato (ATOS)

Table of Contents

EXECUTIVE SUMMARY	7
1. INTRODUCTION	8
1.1 INTRODUCTORY EXPLANATION OF THE DELIVERABLE	8
1.2 PURPOSE AND SCOPE	8
1.3 STRUCTURE OF THE DOCUMENT	8
1.4 METHODOLOGY	8
2. CONTEXT FOR THE DELIVERABLE	9
2.1 DELIVERABLE RELATION WITH THE ARCHITECTURE OF THE PROJECT	9
2.2 DELIVERABLE RELATION WITH THE USE CASES	11
3. INFORMAL DISCUSSION	12
3.1 PRELIMINARIES	12
3.1.1 <i>In a nutshell</i>	12
3.1.2 <i>Collecting contextual dimensions</i>	12
3.2 EXTENSIONS TO WSMO FOR CAPTURING CONTEXTUAL INFORMATION	13
3.2.1 <i>Implicit contextual modelling with WSMO</i>	13
3.2.2 <i>Pointers</i>	13
3.2.3 <i>Concluding remarks</i>	14
3.3 THE NEED FOR EXPLICIT MODELLING OF CONTEXTUAL INFORMATION	14
3.4 LIGHTWEIGHT KNOWLEDGE REPRESENTATION	14
4. FORMAL CONSIDERATIONS	16
4.1 PRELIMINARY: WSMO-BASED KNOWLEDGE REPRESENTATION	16
4.2 SIMPLE STRUCTURE FOR CAPTURING CONTEXTUAL INFORMATION	17
4.3 GENERIC STRUCTURE FOR CAPTURING CONTEXTUAL INFORMATION	18
4.3.1 <i>Dimensions: generic and specific</i>	19
4.3.2 <i>Types of dimensions</i>	22
4.3.3 <i>Conclusive remarks</i>	28
4.4 PROPOSED EXTENSIONS TO WSMO	29
4.4.1 <i>Goal sensitivity to a dimension</i>	29
4.4.2 <i>Complex and composed goals: sensitivity of complex of goals and services</i>	30
4.4.3 <i>WSMO elements sensitive to context</i>	31
4.4.4 <i>Agents with context</i>	32
5. CONCLUSIONS	33
6. REFERENCES	34

List of Figures

Figure 1 Position of contribution in relation to D4.1.1. D3.4.7 is conceived as mediating between WSMO and context modelling mechanism, providing extensions as required by context modelling mechanism to facilitate this mediation.	9
Figure 2 Scope of the deliverable in the context of updated DoW: providing a framework and guidelines for ontology development in relation to context information and the minimal links between context information and context information sensitive elements of WSMO-based description as well as minimal extensions in support of this linkage.	10
Figure 3 Context information space as information subspace. The information space may be conceived as a multidimensional space reflecting informational aspects. The context information space is defined by a subset of the dimensions of the information space. ...	19
Figure 4 Context information as information space. The context information space can be further reduced by selecting portions of its defining dimensions. These indicate the ranges of values in which contextual knowledge (in the sense of D4.1.1) may be picked.	20
Figure 5 Definitions of the class Magnitude and of three ordered instances (left column) that are magnitudes for values in two different dimensions of trust level (middle) and customer value (right). Dimensional values inherit the order of these magnitudes (relations between dimensional values and the axioms to the desired effect are not given).....	24
Figure 6 Values (V_i) derive ordering properties from magnitudes (M_i), where for all i , V_i is associated with M_i	25
Figure 7 Order among values. A is a dimension exhibiting no order among its values. B exhibits a partial order in which B1 is followed by both B2 and B3 which are both of actual rank and followed by B4. C exhibits a linear order in which the C_i s are ordered as the is.....	25
Figure 8 Example of definitions of age values with a year-duration unit (not defined here) and of a definition of a dimension representing the age range of adults under 25. While we see that it might still be manageable to define a reasonably large number of values with numerical magnitude, we also see that tests can become handy; suppose, for example, that age were given a unit of day-duration or suppose again that we needed to be able to refer onto values with any numerical magnitude, crafting value might just become unfeasible.	27
Figure 9 A complex dimension (V , magenta) is defined through its values, e.g. V_2 , from values (A_1 , B_2 , C_1) taken from other (simpler) dimensions (A , purple; B , red; C , orange) that are combined with an operator (represented by the centre box).	28
Figure 10 Graphical depiction summarising the sketch of an ontology of dimensions for modelling contextual information.....	29
Figure 11 Complexes of goals/services that are sensitive to contextual dimensions compound these sensitivities.	31

List of Tables

Table 1: Contextual aspects identified through use case requirements (indicative and non exhaustive).....	11
Table 2: Representation of the facts about Alan and Beth. The left column provides a possible representation in WSML that is paraphrased line by line in the right column...	16
Table 3: Alternative for the capture of contextual information.	17
Table 4: Definitions of values for the age and skill level dimensions of Figure 4.	20
Table 5: Definitions of the class Dimension of dimensionals. Dimensions include values which can be also determined (optionally) with a test. Constraints are left unspecified.	21
Table 6: Definitions of dimensions through values. The A and B dimensions are generic and represent the axes of the blue areas in Figure 4 and the -Prime variants represent the axes for the green area in the same figure.....	22
Table 7: Definitions of the class DimensionValue of dimensional values.	23
Table 8: Definition of a subclass of Dimension and DimensionValue to which are attached units.....	26
Table 9: Definitions of the concept ComplexDimension using a slot for indicating simpler dimensions used to arguments for an operation defining values in this dimension.....	27
Table 10: Alternative for the representation of the link between goals and dimension.	30
Table 11: Alternative for the representation of the concept of sensitive WSMO element. The first row gives intrusive extensions defining lasses and slots with greater degree of extension from left to write. The second gives row the minimal non-intrusive underspecified relation.....	31
Table 12: Definition of the class Agent as generic and extendable.	32

Glossary of Acronyms

Acronym	Definition
D	Deliverable
DoW	Description of Work
EC	European Commission
SOA4All	Service Oriented Architecture for All
WP	Work Package
WSMO	Web Service Modeling Ontology
WSML	Web Service Modeling Language

Executive summary

The present deliverable report is concerned with defining extensions to WSMO for capturing contextual information. The report locates its contribution in relation to the overall architecture of the project and in relation to use cases. The notion of extension is understood as an addition to the WSMO model in a way that is consistent with the general approach to knowledge representation underlying WSMO and that respects the open-ended character of SOA4All's application scenarios. In the light of these characteristics, the forthcoming treatment of contextual information is largely implicit and consists in providing ontological support for the capture of contextual information. This constitutes a first kind of extension to WSMO models in so far as such treatment of supporting domain specific information remains inherently underspecified in WSMO. Moreover, despite the implicit character of contextual information, as contextual information nevertheless requires being identified as such for the sake of the applications that use such kind of information, a second kind of extension is discussed that proposes to introduce the concepts of context sensitive WSMO elements, on the one hand, subsuming certain goals and services, and a concept of agents from the description of which contextual information may be gathered and subsuming users, providers but also software and artificial agents.

Technically, extensions can be seen narrowly, in which case, the narrow extensions to WSMO proposed here consists in the addition of few concepts and relations with underspecified constraints. Construed broadly, the extensions contain an abstraction of the context knowledge representation framework presented in support of the capture and representation of contextual information. The latter consist in generic notions of (contextual) dimensions providing a reservoir of objects which themselves stand for (non functional) features occurring in goals and Web service descriptions when these are sensitive to context.

The open-ended nature of SOA4All's application scenario demands the following interpretation of the framework for knowledge representation sketched here. This framework ought to be conceived as serving as guideline, exemplification and indication of a uniform methodology. This deliverable is thus, in the first place, intended to support of efforts deployed in use cases for the formulation the contextual dimensions of information. But, moreover, this deliverable is also intended to support the efforts related to the varied tools developed in SOA4All that will be using contextual information in the management of tasks related to Web services (selection, adaptation and so on).

1. Introduction

1.1 Introductory explanation of the deliverable

This is the deliverable report D3.4.7. on defining extensions to WSMO for capturing contextual information.

1.2 Purpose and Scope

The present deliverable report is concerned with the task of defining extensions to WSMO for capturing contextual information. WSMO is a conceptual model for describing Web services. The main purpose of the present report is to discuss the way contextual information ought to be captured in the context of SOA4All in a way based on using ontologies and under the assumption of using WSMO as a basis for the semantic description of Web services. In particular, the report is intended to contribute the background support to use cases of a generic framework as well as illustration and guidelines for the partial specification of their knowledge representation.

More technically, the purpose of the report is to set and define a generic ontology-based approach to the representation of contextual information and locate it in relation to WSMO-based descriptions of Web services.

1.3 Structure of the document

The structure of the present deliverable is as follows:

Section 2 places the deliverable's contribution within the overall project architecture and in relation to use cases.

Section 3 provides an informal discussion of the goal of defining extensions to WSMO for capturing contextual information.

Section 4 provides a formal specification of the elements that are put forward in order to support defining extensions to WSMO for capturing contextual information.

1.4 Methodology

The work reported in D3.4.7. builds on the preliminary analysis carried out in D4.1.1 (Contextual service Adaptation Framework) while also taking notice of the re-orientation that followed the updates to the SOA4All DoW. D4.1.1 approached the problem of context management in the perspective of contextualising service delivery and supporting service adaptation. As part of the mechanism, D4.1.1 anticipated a declarative approach to the capture of contextual information to support the varied tasks related to "Context Management, Context Recognition and Service Adaptation based on Context"¹.

So as to not pre-empt the implementation of WSMO extensions for capturing contextual information in the different WPs responsible for dealing with tools using context, the present deliverable will approach context modelling in the style in which WSMO itself has been delivered.

¹ <http://www.soa4all.eu/contextmanagement.html>

2. Context for the deliverable

2.1 Deliverable relation with the architecture of the project

The work presented in the present deliverable is derived from the specification of the context adaptation mechanism defined within Work Package 4 with its D4.1.1 deliverable. Based on the overview of the Contextual Service Adaptation framework envisioned in D4.1.1, the present deliverable's contribution may be situated in relation to this framework as the junction between the envisioned context modelling mechanism and WSMO (the framework's baseline ontological model of Web services), Figure 1.

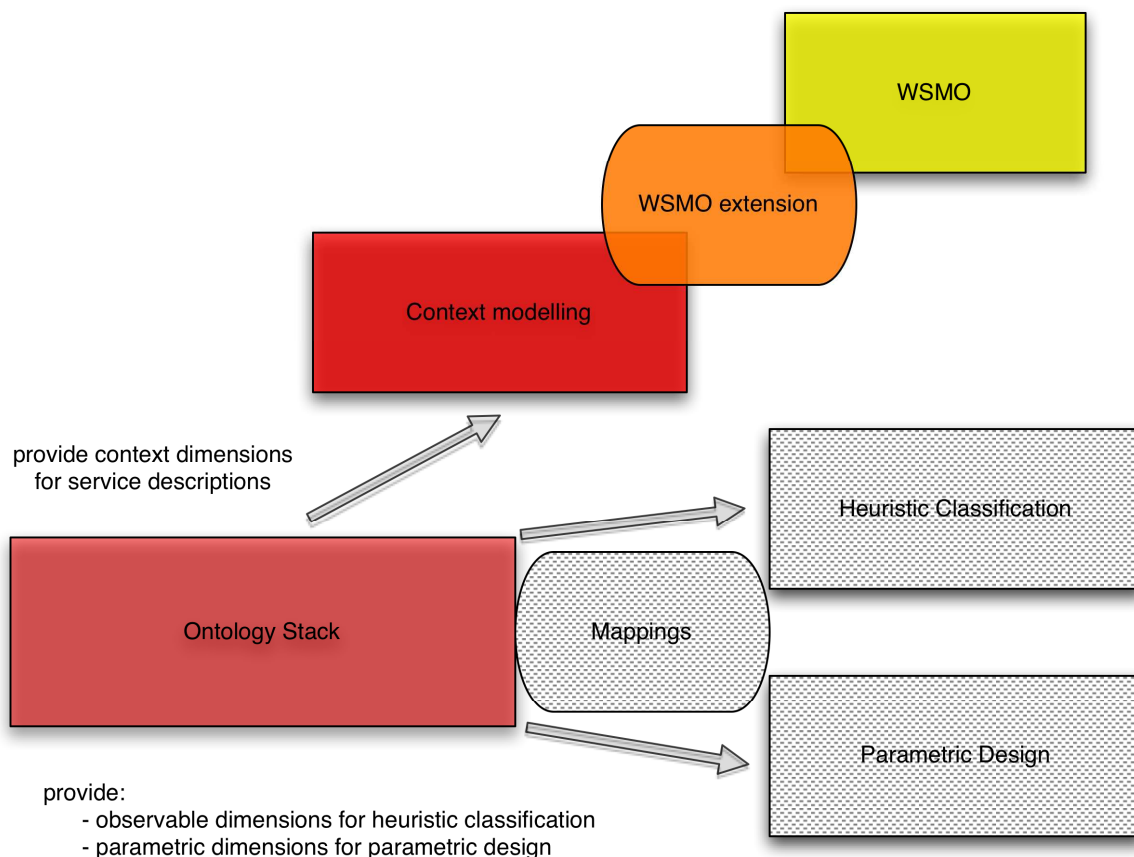


Figure 1 Position of contribution in relation to D4.1.1. D3.4.7 is conceived as mediating between WSMO and context modelling mechanism, providing extensions as required by context modelling mechanism to facilitate this mediation.

D4.1.1 envisions an ontological treatment of context information that gives a central role to the elaboration of a stack of ontologies covering parameters informing the representation of Web services's contextual information. Under this perspective, contextual information is in fact background information that has to be brought forward as relevant to a task at hand. Contextual information is any information that can be used to describe the situation of an entity. When this information is identified as relevant to a task it can take on a new special role to shape the task's realisation, in such circumstances D4.1.1. refers to "contextual knowledge". One of the requirements for a context modelling facility is therefore that it allows to reduce the amount of information at disposition to its contextually relevant number. The

task of reduction is one that has to do with context management but that is to be supported through the modelling of contextual information. The information from which to start is information that is available about a given entity subjected to description in the context of the usage of WSMO-based description of Web services. Such an entity may be prototypically a user of a service (including SOA4All users), more generally agents, goals expressed by agents, and Web services that match or are looking to match these goals. According to this picture, however, goals are prominent and natural carriers of the indicators that a given kind of information is contextually relevant---for it is primarily relevant to the realisation of a goal.

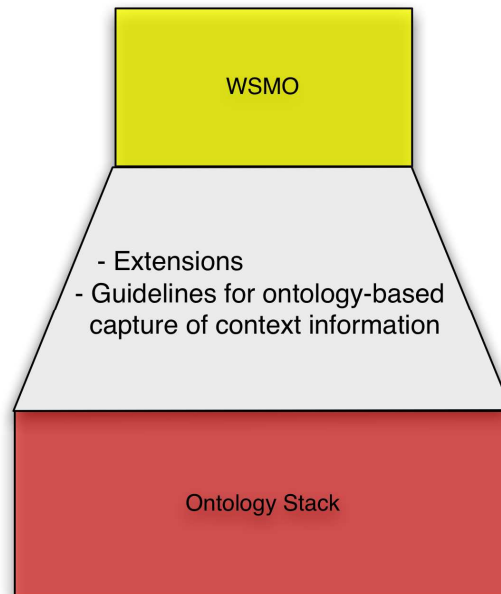


Figure 2 Scope of the deliverable in the context of updated DoW: providing a framework and guidelines for ontology development in relation to context information and the minimal links between context information and context information sensitive elements of WSMO-based description as well as minimal extensions in support of this linkage.

Ontologies covering the parameters informing the representation of Web services' contextual information have then to be linked (for example, through usage) to WSMO-based service descriptions, on the one hand, and, on the other hand, they have to be mapped to classification and adaptation (parametric design) mechanisms. These ontologies are in themselves potential extensions to the WSMO model in the very straightforward sense that they would cover grounds left unspecified by WSMO. But the interesting point as to the reason for having them is that the role of resources for capturing context becomes that of preparing the ground for serving the adequate information to context adaptation mechanism. Context modelling is then primarily a way of supporting the capture of contextual information that then can be used by different mechanisms.

Following the rework of the SOA4All DoW, relevant tasks that fell under the purview of WP4 have been passed into the various work packages dealing specifically with the corresponding service related tasks. This has affected the place of the present deliverable in the overall project, reshaping it towards assisting ontology-based representation of context information and providing minimal links between context information and the elements of a WSMO-based description of Web services potentially affected and sensitive to contextual information (Figure 2). This will be discussed in more details in the informal analysis of section 3.

2.2 Deliverable relation with the use cases

The contribution of the present deliverable is thus to provide knowledge representation support to the various task that are related to using contextual information, It thus contributes a conceptual and formal ontological support to mechanism that are intended to be reused in SOA4All application scenarios and in use cases in particular. This contribution will be exhibited to the extent that use cases present scenarios that make available contextual information to the technology and under the provision that contextual information is identified as such.

The three use cases mention the need for managing contextual information and adapting tasks related to services in the light of user and contextual information. Use cases rely explicitly heavily on the context adaptation mechanism attached to WP2 and the Context and Process language attached to WP6. Although there is no uniform and explicit definition of contextual information that is endorsed by every use case, each deliverable presenting requirements specification mentions a number of examples of what counts as contextual information. They are mostly kinds of information that pertain to the user situation when carrying a task triggering the recourse to a Web service. In that respect, there is a uniform agreement that information relating to location (e.g. geographical location such as country) is an important marker of context. But there are also specific variations on related themes that nevertheless remain vague and are difficult to pinpoint. However, important aspects that could demand modelling support for the use cases are identified in the way summarised in table 1.

Table 1: Contextual aspects identified through use case requirements (indicative and non exhaustive).

Use case	Counts as context	Reference
End-user integrated enterprise service delivery platform (WP7)	Age, Gender, Language, Computer Skills, Organisational role, Task	D7.1 Requirements Specification
Web21c BT Infrastructure (WP8)	Location (geographical), Role (business/leisure), Time (e.g. peak/off peak), Presence (at desk/away)	D8.1 Web21c Requirements (v1.62)
C2C Service eCommerce (WP9)	Geographical location, Currency	D9.1 Requirements Specification

3. Informal discussion

3.1 Preliminaries

The motivation for adding the present framework to WSMO-based description turns on the nature of contextual information. The assumption made when having recourse to WSMO in describing services is that all aspects of a service may be exhaustively represented, at least in principle if not in practice. The feature of WSMO that allows for potential exhaustiveness in WSMO-based descriptions is WSMO's ability to handle non-functional information, providing resources to representing these in addition to representing core functional aspects of a Web service. WSMO does this in particular by embedding the resource of ontologies dedicated to the representation of a given domain. In relation to a domain it is then possible to enrich the description of a Web service, in particular by adding the registration of a number of non-functional properties of Web services to their semantic description. Under the perspective taken here, contextual information is part of this reservoir of information that by some standard is deemed relevant to a given task associated with a Web service (for example, discovery, adaptation or execution; see Fensel et al, p115).

3.1.1 In a nutshell

Against this background, in first analysis, the modelling of contextual information requires no particular resource other than generic ontological engineering resources that are provided by WSMO via its provision for ontologies. The stake, however, is in recognising information as contextual in relation to a given task, application or application scenario and to allow for easy access to the relevant information. Thus, the main contribution of a knowledge representation framework dealing with contextual information, given that services are given WSMO-based description, is concerned with primarily managerial aspects of context information. The extensions to WSMO discussed here reside at the level of minimal requirements on the behalf of context representation for the support of the management of that information. Context management per se is left to the development of the task-oriented applications that will use contextual information, for example, dealing with context-based adaptation. In this connection, the present deliverable intends two contributions. The first is to minimally and generically answer the need for a declarative way of identifying information as contextual. The second is to draw the general outline of the ontological resources that will feed into this context information recollection in conformance with WSMO's approach to knowledge representation.

Since, in general, information can be recollected as properties, we will adopt a similar assumption regarding context information. Context information then can be associated to its bearer through a number of slots (binary properties that are introduced in the definition of a class or 'concept' as they are called in WSMO) and these slots allow for indicating within a range of values which property the bearer has (the bearer of the 'slot value' is an instance of the concept for which the slot is defined). That such a value of a slot --- representing a feature of the entity described -- belongs to the context of the management of a Web service has to be indicated. To afford for as much generality and flexibility as can be found, we will favour in many cases replacing the slot mechanism with a mechanism based on relations (not tied to the definition of a concept).

3.1.2 Collecting contextual dimensions

Since contextual information is nothing but information that is deemed by some standard to be relevant during the execution of a task, the proper management of contextual information is intrinsically tied to the implementation of the task in question. This fact may be seen as an underlying rationale to the absence in SOA4All of a focal point for dealing with context as an object in its own right from a knowledge representation perspective. Hence the line of work within the project departs from traditional approaches to knowledge engineering approach of context such as exemplified by McCarthy, Guha and Lenat. Nevertheless, the general trend

of seeing context as a multidimensional space for information gathering can be abstracted into dealing with contextual information as structured along a number of aspects or dimensions. This is also a line of approach that has been taken in the context of the semantic web with so-called ontology-based treatment of contextual information (Cool in particular, see Krummenacher et al.).

If we consider contextual information as information with a role, within the context of a pervasive and open ended environment such as tackled by SOA4All, we can see that provision needs to be made for any kind of information to be a priori contextual relevant for a task or an other. While there is no context as object in its own right there is still a need for identifying bits of information or information of a kind as contextual. This means that on the one hand a mechanism has to be available for i) registering bits of information or information of a kind as contextual and ii) structuring information in uniform way that is prone to making such registration.

3.2 Extensions to WSMO for capturing contextual information

This deliverable report presents elements of knowledge representations that are extensions to WSMO for capturing contextual information. WSMO is the Web Services Modeling Ontology (see Fensel et al; see also Roman et al.) which is a conceptual model formalised in an ad hoc language (WSML variant, see Steinmetz and Toma), for the description of Web services. WSMO contains four fundamental elements which are:

- *Ontologies* which are intended to provide the resources for extending Web services descriptions with relevant domain specific knowledge.
- *Web services* which are given a thorough ontological treatment in terms of functionality
- *Goals* which support the description of Web services'; capabilities
- *Mediators* which come in a variety of flavours and are tools for resolving various kinds of heterogeneity

3.2.1 Implicit contextual modelling with WSMO

Contextual information is information that may be used to describe a given situation of an entity. WSMO does not contain an explicit modelling element for addressing the representation of contextual information as such. WSMO nevertheless has the necessary provision for representing contextual information. Indeed, the state-of-the-art in context information modelling is considered to be ontology-based (see Krummenacher et al.) and to amount to the mere ontology-based representation of information that is deemed contextual.

Following this line of approach extending WSMO for capturing contextual information would merely amount to elaborating an ontology that is able to represent all aspects of any situations at the domain level. It is under this view that the present deliverable recommends the elaboration in use cases of dedicated ontologies for representing contextual information.

3.2.2 Pointers

Extending the core framework provided by WSMO, a number of ontologies for describing non functional properties have been created in the context of the WSMO Working Group (see Toma, 2006). These are concerned with providing resources for expressing such kinds of information as location, availability or price and so on. They are available from the following URL: <http://www.wsmo.org/ontologies/nfp/>

More generally, a number of 'context ontologies', ontologies explicitly destined to support the expression of contextual information in a variety of framework have been put forward over time, for example: the W3C's Delivery Context Ontology (Cantera Fonseca and Lewis, 2009), Cool (Starnig et al., 2003) or ConON (Wang et al., 2004). These, however, are sometimes difficult to find in the form of fully formalised Web resources.

3.2.3 Concluding remarks

For the purpose of capturing contextual information the framework for knowledge representation provided by WSMO is sufficient under the assumption that contextual information may be modelled implicitly. Providing means for capturing contextual information consists in developing domain ontologies that allow for the representation of such information. Hence use cases would develop ontologies or reuse ontologies such as those pointed above in order to provide for the knowledge representation means. A simple approach to context management system would consist in delineating context repositories as parts of the overall semantic space underlying SOA4All's vision as to the infrastructural support to its technology. It would be the role of applications and dedicated technologies to manage this information (WP2, WP6 and use case solutions).

3.3 The need for explicit modelling of contextual information

While the forthcoming approach to handling contextual information implicitly under a WSMO-based description of Web services is reasonable under the assumption of a definite and circumscribed application context, it becomes potentially overwhelming and non-manageable to adopt this approach in a pervasive and open-ended scenario such as that of SOA4All. If the value of context information and its relevant subset is indexed to a task of handling a Web service (discover, selection, adaptation and so on), this value is dependent on the specifics of particular iterations of the Web services management process. If the scenario is indefinite and open ended, it is virtually impossible to make for all the provisions necessary to provide support to all potential situations. In other words the contextual value of information is defined and exhibited through the applications using such information. Hence, these applications have to play an active role in indicating the modelling resource used for capturing contextual information as well as directing the way this information is kept and managed.

In reaction to this, two lessons can be drawn. The first is that the knowledge representation resources put forward as extensions to WSMO for capturing contextual information need to exhibit a high level of generality. They need to be general so as to cover a wide range of application contexts. Also they need to be generic in a manner that allows for the adequate refinements in the light of the specifics of particular applications of the SOA4All technology. This means that a generic framework for the expression of contextual information has to be specified that provides a sketch and guidelines for specification of in application contexts. The main characteristic looked for in relation to this requirement is uniformity of modelling contextual information, uniformity of method and of representational approach. To this end, the present deliverable puts forward a notion of dimension that can be utilised in order to sketch contextual dimensions that are salient to an application.

The second lesson to draw is that there needs to be in addition to this general framework for expressing contextual information a mechanism for indicating in relation to a given task which information or which kind of information is contextually relevant to the task and to the elements of a WSMO-based description of Web services. In other words, there needs to be a generic way of indicating contextually relevant information. The reason why this mechanism needs to be generic is that, as the mechanism for the very expression of contextual information, the mechanism for indicating contextual information needs to allow for an unspecified and unforeseeable range of application specifications. To this end, the present deliverable sketches modelling elements that extend the overall WSMO model in simple ways while allowing for identifying the main polarities of a context-sensitive knowledge representation: a notion of agent (e.g. user) and an overarching category of context sensitive entities that in WSMO may be linked to contextual dimensions.

3.4 Lightweight knowledge representation

A further consequence that may be drawn from these requirements has to do with the

simplicity of the knowledge representation adopted for supporting the solution. If only for the sake of scalability, the resources mobilised need to exhibit a certain degree of lightness. From the standpoint of the complexity of knowledge representation, a lightweight representation is one that will use simplest recourse to situation descriptions. This can be achieved while preserving a continuity of representation with WSMO and with the general knowledge representation that it supports. This is because WSMO and the sort of linguistic resources that it requires contains at least the resources that more lightweight languages such as RDF/S will provide. In particular, it allows for describing entities via sets of properties (binary relations that are also sometimes called 'slots' or 'roles') associating property values (instances of types) to a given entity. This is in fact largely in line with the treatment of WSMO's non-functional properties that are typically recollected using a slot and its values.

The recourse to such simple means of representation of entities' features facilitates also the explicit recognition of contextual information as such. For it makes it easy to build sets of values that represent contextually relevant value ranges associated with a property slot (contextual dimensions). This will be the key to the element of the context management system sketched below in support of the task of recognising and processing context.

As a final word in relation to the lightness of representation, this deliverable has so far mentioned elements of WSMO that are of a high level. It does not require the refinements of WSMO that were left aside. As a result, it conforms to converging assumption of lightweight variant of WSMO (WSMOLite, D3.4.2, and MicroWSMO, D3.4.3) and, in particular, to the minimal model for Web Service description that these put forward.

4. Formal considerations

The formal considerations presented here are intended to help guiding the process of context information modelling in a framework that is aligned with WSMO-based description and bound by low (lightweight) expressivity.

4.1 Preliminary: WSMO-based knowledge representation

Entities in WSMO are instances of concepts and WSMO has mechanisms to associate information to entities through relations or similar mechanisms. In particular, slots specified when defining concepts can be applied to the instances of a given concept. Consider the following example set of facts:

Alan and Beth are persons.

Persons have a gender.

Alan's gender is male.

Beth's gender is female.

Using WSML (see Steinmetz and Toma, 2008), we could have support the representation of this set of facts in the way presented in the following table.

Table 2: Representation of the facts about Alan and Beth. The left column provides a possible representation in WSML that is paraphrased line by line in the right column.

WSML expression	Explanation
concept Person hasGender ofType Gender	defines the class of persons; instances have a gender property
concept Gender	defines the class of gender
instance Male memberOf Gender	defines an instance of Gender
instance Female memberOf Gender	defines an instance of Gender
instance Alan memberOf Person hasGender hasValue Male	defines an instance of Person who is male
instance Beth memberOf Person hasGender hasValue Female	defines an instance of Person who is female

Thus, to an entity is associated some information, for example that Alan is male. The prototypical case is that this information is an item, an entity itself that instantiates a concept as, for example, *Male* instantiates *Gender*. This item is related to the entity it is associated to

by a given relation (a slot) as, for example, *Alan* is related to *Male* through *hasGender*.

4.2 Simple structure for capturing contextual information

Contextual information is any information that can be used to describe the situation of an entity, its context (D4.1.1). In order to describe an entity using resources such as provided in a knowledge representation such as one based on WSMO, we use the general system of slot and slot value illustrated in the preliminaries.

Suppose that we decide that in a situation such as Alan performing a search in an online store for a piece of garment for himself, Alan is recognised as a highly valuable customer. Because of this, his order will be given priority and delivery of product will be sped up or he may receive some reward or another along these lines. The contextual element that is at work in this situation is Alan's being a highly valued customer. This is an evaluation that takes place on the side of the provider (ultimately, the store) but that is attached to the user (Alan). It may be directly recollected in a user profile that is maintained by the store or it may also be the result of a categorisation that is made on the fly on the basis of lower level information about Alan (e.g. age, available payment instrument and so on). For our purpose, however, that Alan is a highly valued customer is a contextual fact that may be represented using a slot, e.g. *has customer type*, and a value for that slot, e.g. *highly valued customer* or *high cust value* for short. Alternatively, or more generically, we can handle *has customer type* as a relation. Branches of this alternative indicate a generic mechanism for registering contextual information; they are given in the following table.

Table 3: Alternative for the capture of contextual information.

Attribute a value for a slot to an entity	Link an entity to the value object
instance Alan customer-type hasValue HighCustValue	relationInstance customerType (Alan, HighCustValue)

The required ontological resources for registering contextual information are thus as follows.

- 1) there needs to be a reserve of values that may be attached to an entity, value instances
- 2) there needs to be a slot for the kind of entity to which the contextualised entity belong that allows attaching the values, alternatively a relation
- 3) the values used to register contextual information need to be defined as instances of a type; this type allows both the definition of values and the definition of the slot or relation range.

The registering of contextual information regarding Alan in the way presented above would therefore assume a background ontological registration of the features of customer typology. For example, it could include a class of customer type with three types:

- *high value customer* who will have privileges, e.g. priority on order, fast delivery, rebate, and son on
- *medium value customer* who will not have these privileges but could receive incentives, so as to become valued customer, or advertisements of sorts

- *low value customer* for whom, for a reason or another, the company is not ready to invest in its marketing strategy, for example, they have a bad payment record with the company or they are not able to enter in legal contracts with the company.

In addition to these types, the background ontological resource ought to contain a mechanism, a slot or relation, allowing to attach these types to an instance of Person such as, for example, Alan. Thus in the example above, we could have the following ontological resources:

A kind for the definition of customer type values:

concept CustomerType

A number of customer type values instantiating CustomerType:

instance HighCustValue memberOf CustomerType

instance MediumCustValue memberOf CustomerType

instance LowCustValue memberOf CustomerType

A relation for linking instances of person to the type of customer in which they are classified:

relation customerType (ofType Person, ofType CustomerType)

Alternatively, the relation is defined as a slot for the class Person:

concept Person

customerType ofType CustomerType

Hence the basis for capturing contextual information follows the basis for doing knowledge representation in the context of WSMO-based descriptions of Web services.

Finally, this information is to be attached to the relevant entities. In order to prevent the closing of an open-ended scenario, it is safer not to constrain the domain of a relation (if it is a slot, it will be defined as part of the specification of a class under which fall the entities in question). In the general case, then, that in which the contextual information is modelled in an unspecific manner, one could argue that using relations with unspecified domains is optimal. However, on the other hand, in many cases we would expect that the contextual information be modelled with pre-existing ontological resources that may impose a domain on relation (or present them as slots for classes). In this case, the representation of an entity for which the context is being modelled has to be aligned with the chosen resources. Later, this report will introduce the concept of *Agent* (cover that of *User*) that could be conceived as a way of facilitating this alignment.

4.3 Generic structure for capturing contextual information

The distinction that the representation of contextual information draws in relation to knowledge representation at large has to do with the role of the information to be captured. Contextual information is relevant information for the description of entities given their situation. This means that within the range of information available, contextual information stands, given a task and a situation, as a portion of the overall information available. This means too that the way contextual values are aggregated together, so as to define dimensions according to which a context is specified, have to be given the resources for discriminating them from related values and features that may not be contextually relevant. For example, in order to support the representation of a contextual element (the fact that Alan is a high value customer), we define a class of customer types and few of its instances. While we could define an arbitrarily large number of such instances (and more generally of values for representing contextual elements), it is also part of the delineation of a context that only some such instances (values) will be relevant and belong to the adequate underlying contextual background. The company that wishes to contextualise its services based on

customer types may take into account the three types defined above in a reservoir of several dozens of types. The same point can be made under the provision that the three types given are all the types there are when the company takes into account only two types (for instance it treats all customer either as mainstream or valued).

4.3.1 Dimensions: generic and specific

It is useful to think of context as an information space delineated along a number of dimensions that represent as many aspects that together make for the characterisation of the context of an entity.

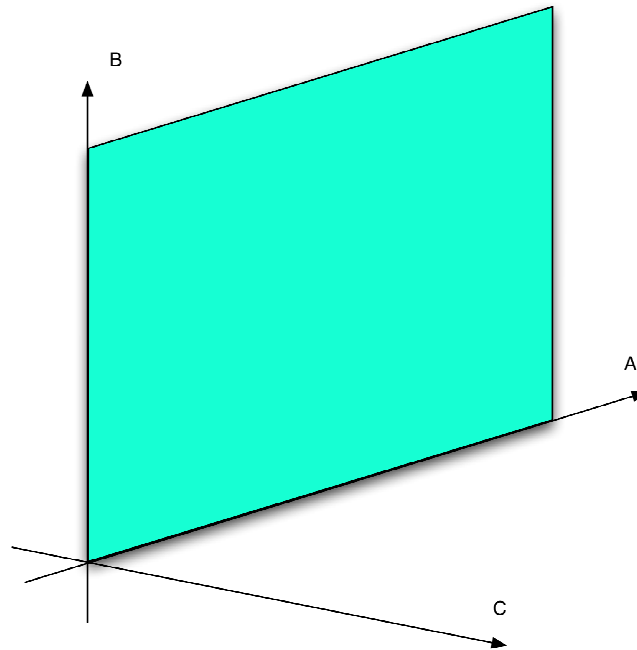


Figure 3 Context information space as information subspace. The information space may be conceived as a multidimensional space reflecting informational aspects. The context information space is defined by a subset of the dimensions of the information space.

Hence, if we assume an information space that is structure along three dimensions A, B and C, but that we consider that contextual information belongs only to dimensions A and B, the contextual information space is reduced in relation to the overall three-dimensional space to a two-dimensional space carved along A and B. Suppose A is the age dimension, B is the skill level dimension and, finally, C is the gender dimension. If context is taken to include only age and skill level, but not gender, then the space of relevant information (context information) is the plan defined by positions along the age and skill level dimensions only and the gender dimension is entirely shaded for the purpose of contextualisation.

Moreover the reduction of the information space is not limited to the reduction of dimensions. It can also be a reduction within the overall dimensions of the information space that have been selected.

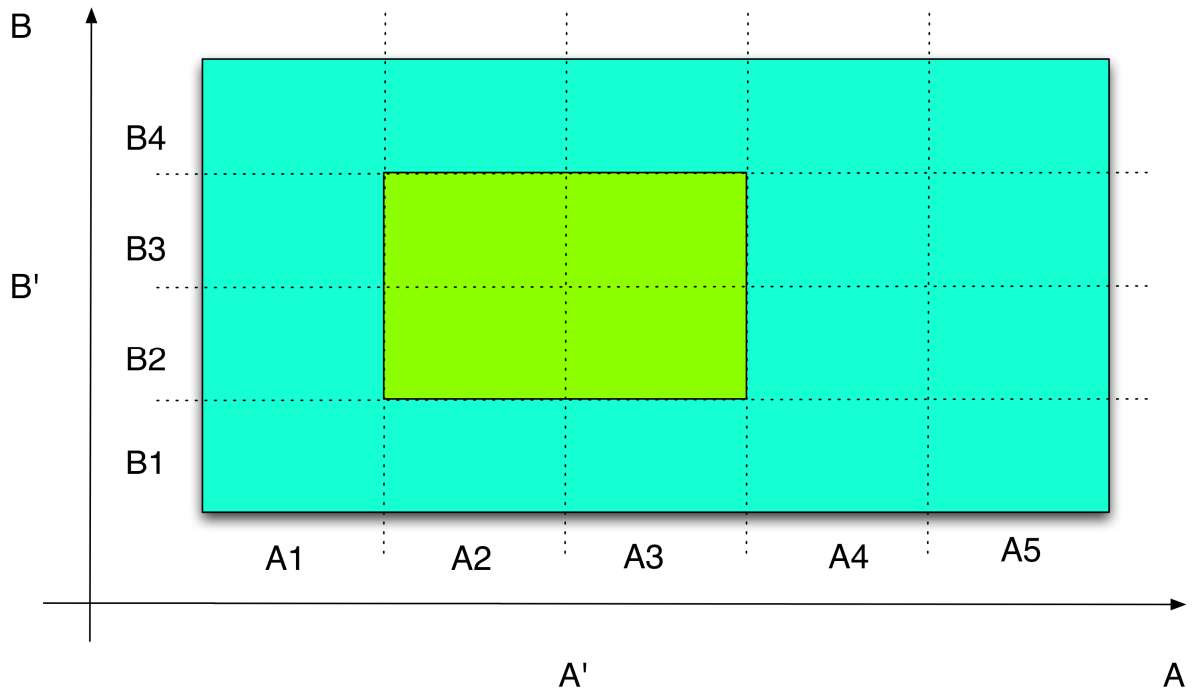


Figure 4 Context information as information space. The context information space can be further reduced by selecting portions of its defining dimensions. These indicate the ranges of values in which contextual knowledge (in the sense of D4.1.1) may be picked.

Hence, if we assume a context information space that is structured along two dimensions A and B with each dimensions being made of a number of values, we can further carve out in this space a narrower context information space that in which for the purposes of a contextualising application values along A and value along B will be taken within smaller ranges. For example, if A is the age dimension and B the skill level dimension and they are such that A is made of 5 age groups and B is made of 4 competence groups, then we can further reduce the information space to a context in which age is taken from only two age groups (A2 and A3, for example) and the skill level is linked to one of two competence groups (B2 and B3). This is illustrated by the figure above.

Consider the two figures presented above (Figure 3 and 4). We took as examples of A, B, and C the dimensions of age, skill level and gender. In order to map the space described in Figure 4 made of the age and skill level dimensions, we will first identify a number of values in the age and skill level dimensions. These can be formally introduced by defining concepts instantiated by these values (respectively *AgeValue* and *SkillValue*) and the values themselves (respectively the *AgeGroup1-5* and *SkillValue1-4*).

Table 4: Definitions of values for the age and skill level dimensions of Figure 4.

Age dimension	Skill level dimension
---------------	-----------------------

concept AgeValue instance AgeGroup1 memberOf AgeValue instance AgeGroup2 memberOf AgeValue instance AgeGroup3 memberOf AgeValue instance AgeGroup4 memberOf AgeValue instance AgeGroup5 memberOf AgeValue	concept SkillValue instance SkillLevel1 memberOf SkillValue instance SkillLevel2 memberOf SkillValue instance SkillLevel3 memberOf SkillValue instance SkillLevel4 memberOf SkillValue
--	--

The definition of a dimension is then primarily done by the provision of a number of included values within the reservoir of values that concepts, such as *AgeValue* and *SkillValue*, provide. Suppose that the extent of these concepts is completely known. There is then first a generic dimension that can be defined for both age and skill level. This is done in collecting all the values falling under *AgeValue* and *SkillValue* respectively. Generic dimensions so conceived are represented as the axes of the blue area of Figure 4. Instead of collecting the total extent of corresponding dimensional values, however, we could simply link these generic dimensions to the concept instantiated by all the values, when it exists as with the examples at hand. While this may not appear critical in the examples given so far, we will see shortly how this strategy may become handy when the range of value is not suitable to being enumerated.

In addition to generic dimensions such as described in the foregoing, we can define the more specific dimensions that pick only some of these values. Such specific dimensions would be, for example, those representing the axes for the green area of Figure 4 only. In general, it is not clear that a test will be available for automatically filling the range of included values in a specific dimension. Indeed, the very motivation for having specific dimensions that may be handcrafted is that they allow the gathering of values otherwise not very easily collected in a single sweep. In an application context, with this mechanism, we would provide for resources to narrow down the relevant values in each dimension to contextually relevant values.

On the basis of these considerations, the minimal concept of dimension that we are looking at in support of WSMO-based Web services description and the capture of contextual information is that of a collection of instances of a WSMO concept (these being the values included in a dimension) and that, from an operational standpoint, has an optional associated test for collecting its included values. A definition along these lines follows:

Table 5: Definitions of the class Dimension of dimensionals. Dimensions include values which can be also determined (optionally) with a test. Constraints are left unspecified.

Dimension
concept dimension hasIncludedValue hasInclusionTest

Using this concept and its definition, it is possible to propose formal definitions for the

dimensions (generic and specific) of the example discussed in relation to Figure 4. First, the generic dimensions A and B are those of age and skill level and pick all the corresponding known values, i.e. the instances of the *AgeValue* and *SkillValue* concepts. They have inclusion tests that are membership in these concepts. Second, the specific dimensions are made of hand pick values within these concepts; there are no known inclusion tests.

Table 6: Definitions of dimensions through values. The A and B dimensions are generic and represent the axes of the blue areas in Figure 4 and the -Prime variants represent the axes for the green area in the same figure.

Generic dimension	Specific dimension
instance AgeDimensionA memberOf Dimension hasIncludedValue AgeGroup1, AgeGroup2, AgeGroup3, AgeGroup4, AgeGroup5 hasInclusionTest AgeValue	instance AgeDimensionAPrime memberOf Dimension hasIncludedValue AgeGroup2, AgeGroup3
instance SkillDimensionB memberOf Dimension hasIncludedValue SkillLevel1, SkillLevel2, SkillLevel3, SkillLevel4 hasInclusionTest SkillValue	instance SkillDimensionBPrime memberOf Dimension hasIncludedValue SkillLevel2, SkillLevel3

Contextual information is information that belongs to an information space and which is only singled out from other information as a reflection of its relevance to a task or situation. In order to tease out this metaphor, we can think of sets of contextual values as dimension for a context. This is a broad sense of dimension can be further teased out.

4.3.2 Types of dimensions

We turn now to a narrower sense according to which certain dimensions in the broader sense are more dimensional than others, where the meaning of dimension is even closer in analogy to that of the dimension of a coordinate space. For example, there is something more dimensional to quantitative (distance, duration and so on) than more discrete and unordered sets of values, such as organisational roles or gender. There are also differences among these as to whether values come with certain units or are merely made of a magnitude and are unitless. For example, there is arguably no unit to gender values or to countries (of residence), whereas it is necessary to provide a unit to prices (in different currencies) and durations (milliseconds, seconds, minutes, days or years and so on). The importance of this later distinction comes from the fact that units themselves can be marker for contexts, such as when a customer requires a pricing in Euro rather than in Dollars.

While in practice a concept instantiated by a number of values could be seen itself as a dimension (as suggested that the tests used for defining the previous examples referred explicitly to instantiated concepts), it is more adequate to carve out a specific construct that serves the generic representation of dimensions. This is shown in particular with dimensions

exhibiting a unit for values (say of price) could come with different magnitudes depending on their units. A dimension, however, will allow attaching to a value a single pair of magnitude and unit.

4.3.2.1 Magnitude

The values in certain dimensions are characterised by a given magnitude. For example, the age dimension that we have divided into age groups earlier could be divided in other ways, for example specifying discrete amounts of years (1 year old, 2 y.o., ... 15 y.o., and so on). We could have imagine rather that within AgeGroup1 fell all ages under, say, 15. If we are interested in discrete and separate ages, it becomes important to allow for the association of a magnitude with the values. This magnitude may be quantitative as with age or it may be more qualitative.

Table 7: Definitions of the class *DimensionValue* of dimensional values.

DimensionValue
concept DimensionValue
hasMagnitude ofType Magnitude

An example of a qualitative magnitude is the treatment of *CustomerType* given earlier. There were three values of high-, medium- and low customer value. Thus, a high value customer could be given a 'high' magnitude in the dimensional scale of customer value. These can be understood as having the corresponding *high*, *medium* and *low* magnitude. The motivation for recognising magnitude in association to certain dimensional values is that magnitudes allow for comparison and are moreover reusable across different dimensions when these exhibit similar relationships among their values.

concept Magnitude	concept TrustLevelValue	concept CustomerValue
	subConceptOf DimensionValue	subConceptOf DimensionValue
relation higherThan	instance HighTrust	instance HighCustValue
(ofType Magnitude, ofType Magnitude)	memberOf TrustLevelValue	memberOf CustomerValue
	hasMagnitude HighMagnitude	hasMagnitude HighMagnitude
instance HighMagnitude	instance MediumTrust	instance MediumCustValue
memberOf Magnitude	memberOf TrustLevelValue	memberOf CustomerValue
instance MediumMagnitude	hasMagnitude MediumMagnitude	hasMagnitude MediumMagnitude
memberOf Magnitude	instance LowTrust	instance LowCustValue
instance LowMagnitude	memberOf TrustLevelValue	memberOf CustomerValue
memberOf Magnitude	hasMagnitude LowMagnitude	hasMagnitude LowMagnitude
relationInstance HhigherThanM higherThan (HighMagnitude, MediumMagnitude)		
relationInstance MhigherThanL higherThan (MediumMagnitude, LowMagnitude)		

Figure 5 Definitions of the class Magnitude and of three ordered instances (left column) that are magnitudes for values in two different dimensions of trust level (middle) and customer value (right). Dimensional values inherit the order of these magnitudes (relations between dimensional values and the axioms to the desired effect are not given).

The advantage of having magnitude is that they are reusable across different dimensions. Hence, for example, we can define valuable customers as having High-, Medium- or Low-magnitude in the customer value dimension. Similarly, we can define levels of trust as having High-, Medium- or Low-Value in the trust level dimension. While High-value-customer is not comparable to High-trust-level, because they belong to distinct dimensions, they each can be attributed the High-magnitude and we can thus proceed with ordering each of the respective dimensions with a single scheme based on the order of magnitudes.

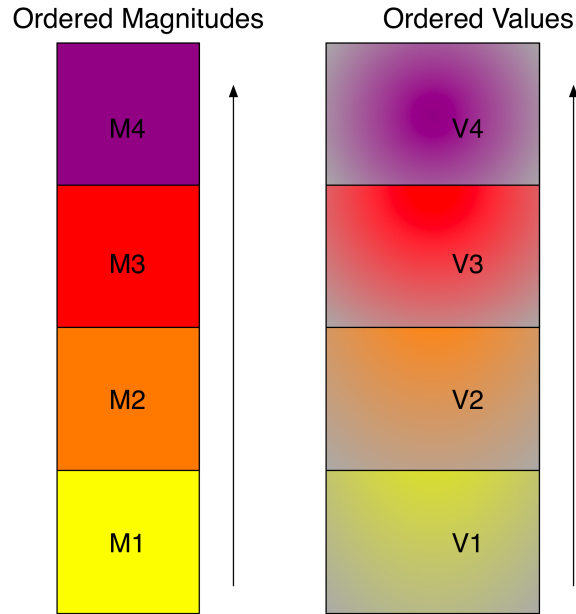


Figure 6 Values (V_i) derive ordering properties from magnitudes (M_i), where for all i , V_i is associated with M_i .

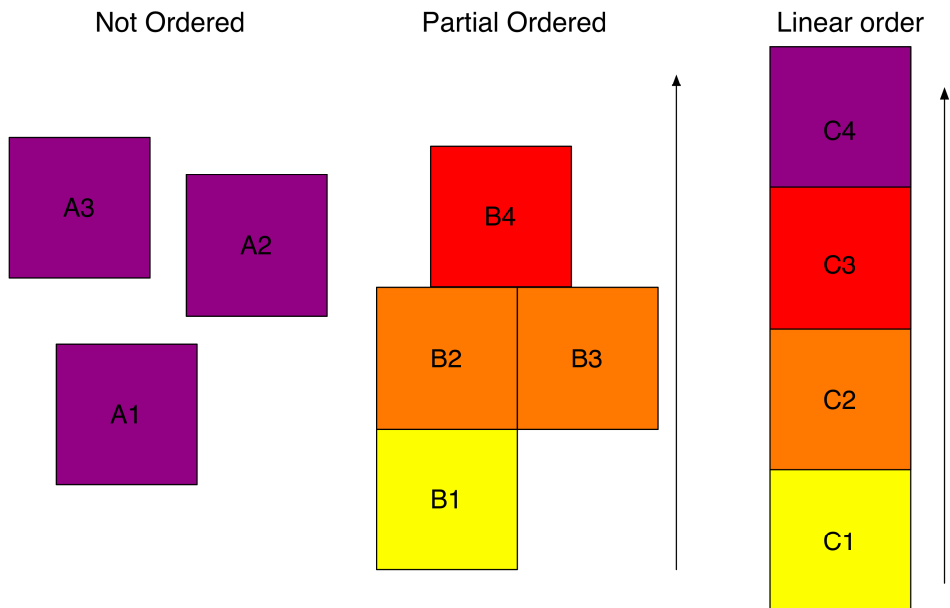


Figure 7 Order among values. A is a dimension exhibiting no order among its values. B exhibits a partial order in which B1 is followed by both B2 and B3 which are both of actual rank and followed by B4. C exhibits a linear order in which the Cs are ordered as the is.

4.3.2.2 Dimensions with units

We mentioned a number of examples motivating the introduction of magnitudes, age, customer type, trust level, and so on. Although all of these are examples of dimensions such

that values are associated with a magnitude, it is clear that for some of them providing a magnitude only is an incomplete description. Indeed, an age is not merely indicated by a number (magnitude). This magnitude has to be provided with a unit. Alan is perhaps 30 years old and not 30 months. Hence it is important that magnitudes be correlated with a unit.

Table 8: Definition of a subclass of Dimension and DimensionValue to which are attached units.

WSML expression	Explanation
concept Unit	defines the class of units
concept DimensionWithUnit subConceptOf Dimension hasAssociatedUnit ofType Unit	defines the class of dimensions with an associated unit as subclass instances are associated with a unit
concept DimensionValueWithUnit subConceptOf DimensionValue hasUnit ofType Unit	defines the class of dimension value with an associated unit as subclass instances have an associated unit

As DimensionWithUnit and DimensionValueWithUnit are subclasses of Dimension and DimensionValue respectively, they inherit the relevant slots. Hence, the instances of DimensionWithUnit may be given included values or a test for inclusion and the instances of DimensionValueWithUnit may be given a magnitude. Figure 8 provides examples.

The approach taken here thrives for simplicity and it is notable that the dimensions to which are attached a unit have a single unit attached to them. Additional sophistication may be warranted that allows for conversion between related units (e.g. seconds to minutes), even including between units that differ only in order of magnitudes (e.g. second to milliseconds).

```

instance OneYearOld memberOf AgeValueInYear
  hasMagnitude 1
  hasUnit YearDuration

instance TwoYearOld memberOf AgeValueInYear
  hasMagnitude 2
  hasUnit YearDuration
...
instance TenYearOld memberOf AgeValueInYear
  hasMagnitude 10
  hasUnit YearDuration
...

instance TwentyYearOld memberOf AgeValueInYear
  hasMagnitude 20
  hasUnit YearDuration
...
instance NinetyNineYearOld memberOf AgeValueInYear
  hasMagnitude 99
  hasUnit YearDuration
...

concept AgeValueInYear subConceptOf {DimensionValueWithUnit, AgeValue}

instance AgeDimensionAdultUnderTwentyFive1 memberOf DimensionWithUnit
  hasIncludedValue {EighteenYearOld, NineteenYearOld, TwentyYearOld, TwentyOneYearOld,
    TwentyTwoYearOld, TwentyThreeYearOld, TwentyFourYearOld, TwentyFiveYearOld}
  
```

Figure 8 Example of definitions of age values with a year-duration unit (not defined here) and of a definition of a dimension representing the age range of adults under 25. While we see that it might still be manageable to define a reasonably large number of values with numerical magnitude, we also see that tests can become handy; suppose, for example, that age were given a unit of day-duration or suppose again that we needed to be able to refer onto values with any numerical magnitude, crafting value might just become unfeasible.

4.3.2.3 Complex dimensions

Certain dimensions can be regarded as complex in relation to simpler dimensions. This is so when a dimension’s values are obtained as a result of combining in a more or less sophisticated way the values of two or more dimensions.

Table 9: Definitions of the concept ComplexDimension using a slot for indicating simpler dimensions used to arguments for an operation defining values in this dimension.

```

ComplexDimension
concept ComplexDimension subConceptOf Dimension
  hasComponent ofType Dimension
  hasOperator ofType Function
  
```

A simple combination is the mere conjunction of values in different dimensions. Thus, for example, a dimension representing some measure of security associated with a service (a

user could work in different contexts that require different levels of security) could be seen as a complex dimension with values corresponding to some conjunctive combinations of values in a dimension representing encryption levels and in a dimension representing the level of reputation associated with a certification authority.

More complicated combinations using operations of greater complexity may be used to define complex dimension. As a special case, we could take for example the definition of the *CustomerType* dimension discussed earlier with values serving to classify customers according to more simple attributes. The sources from which the information is obtained in order to perform such classification maybe as diverse as user profiles, logs, data mined on social website and so on. We do not provide a full fledge example as this sort of complex operation is based on heuristics that may be specific to a service provider, an application or a specific scenario, the heuristics themselves would have to be careful provided by the relevant actors.

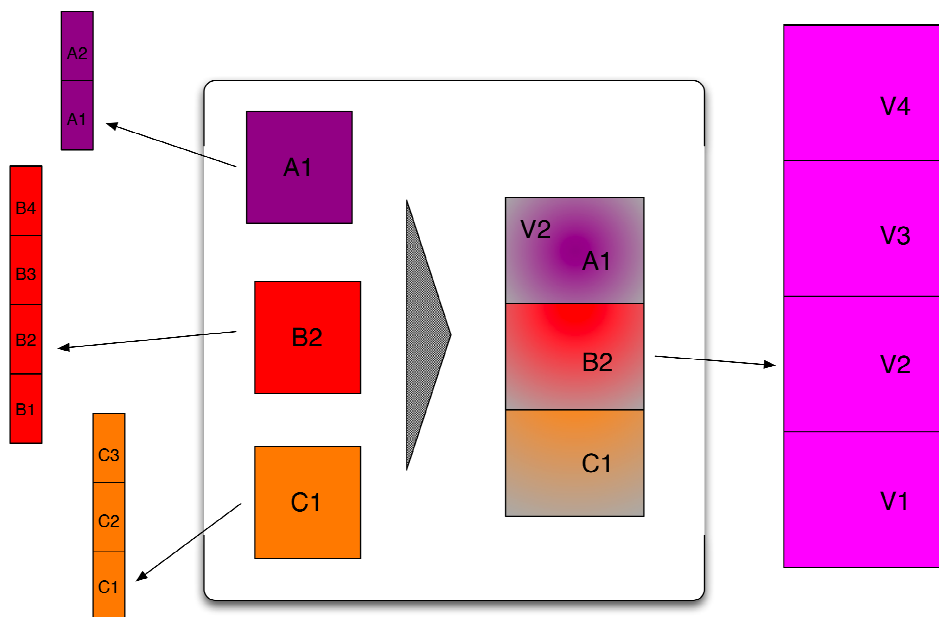


Figure 9 A complex dimension (V, magenta) is defined through its values, e.g. V2, from values (A1, B2, C1) taken from other (simpler) dimensions (A, purple; B, red; C, orange) that are combined with an operator (represented by the centre box).

4.3.3 Conclusive remarks

We have sketched mechanisms and provided guidelines for structuring contextual information in order to support capturing such information when working with WSMO-based Web service description. These elements are intended to facilitate the modelling tasks involved in SOA4All's use cases. They are also provided here in order to lay ground for supporting various tasks related to Web services that use contextual information.

What we have done here is to draw the main lines of an ontology of contextual dimensions for the representation of contextual information that will be amenable to being fed into SOA4All machinery. This is not to say that the material presented here is exhaustive, in particular, much of the management of relationship between dimensions (combination and conversion in particular but also comparison and ordering) have been left in the background. This is because these further elements are to some extent mainstream ontological

mechanisms that can be implemented or adapted in the course of developing models for contextual information. We have offered earlier pointers towards ontologies of non-functional properties for services and a number of context oriented ontologies. We can point here to the ASC model that has inspired Cool (see for a start Krummenacher et al, 2006) as well as to the work done around conversion and unit manipulation within the context of the SUPER project (see for a start, Pedrinaci and Domingue, 2009).

The following Figure provides a graphical depiction in UML style of the forthcoming ontology of dimension sketched here.

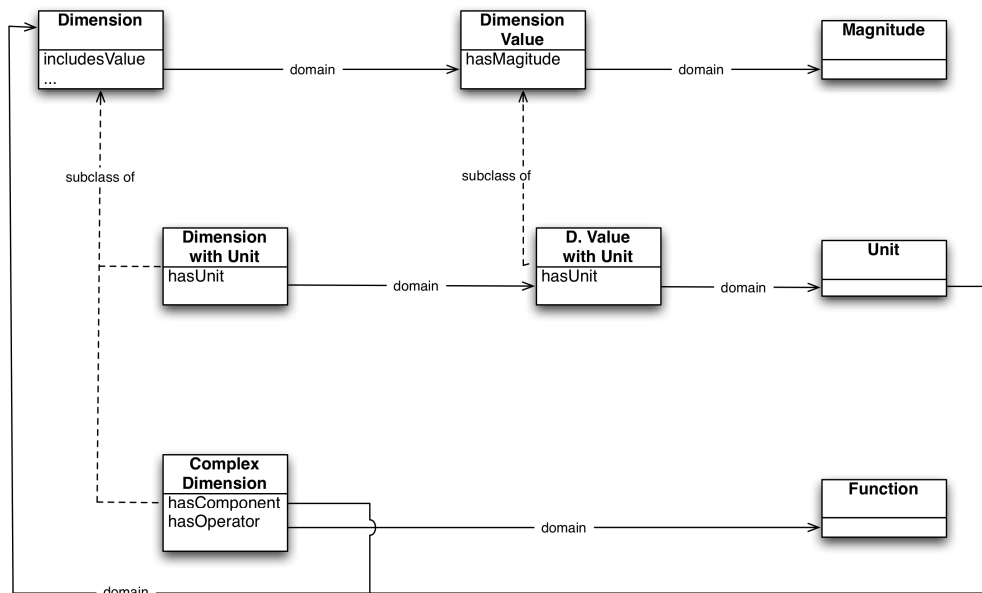


Figure 10 Graphical depiction summarising the sketch of an ontology of dimensions for modelling contextual information.

4.4 Proposed extensions to WSMO

With the resources provided by the foregoing dimensional framework, we can move on with defining a mean for registering contextually relevant dimensions in linking it up with the elements of a WSMO-based description of Web services. In a task-oriented context such as SOA4All's envisioned framework for the context-based adaptation of Web services, the carriers of useful indicators are goals (D4.1.1), but services too and their users are tied to context.

4.4.1 Goal sensitivity to a dimension

In order to indicate that some dimension provides the range for relevant contextual information in order to specify (non functional) aspects of a goal, and so as to facilitate its discriminated achievement, we have only to introduce a link between goals and dimensions. In the spirit of WSMO-based knowledge representation it is natural to make this link through either a slot included in the goal definition or a relation between goals and dimensions.

Thus we may put forward an alternative between two forms: i) a slot in the definition of the goal class and ii) a relation between goals and dimensions; this alternative is recollected formally in the following table.

Table 10: Alternative for the representation of the link between goals and dimension.

Attribute a value for a slot to an entity	Link an entity to the value object
concept goal hasGoalSensitivityTo ofType Dimension	relation isGoalSensitiveTo (ofType Goal, ofType Dimension)

As work on lightweight goals is ongoing within the SOA4All project, it is safer to adopt a conservative approach here that does not prejudice of the definition of the lightweight goals. Hence we will recommend for the time being using a relation. Moreover, using a slot here would amount to a redefinition of the native concept of Goal in WSMO and is hence the least favourable solution given that we have an alternative solution that is non intrusive and consists in a genuine, even if technically minor, extension. Also, since we may want to allow for a more general marking of goal sensitivity, we will leave open the range of this relation and not constrain it to dimension.

4.4.2 Complex and composed goals: sensitivity of complex of goals and services

It is credible that goals will enter into complex made of goals and services. In this case, we want to allow for transferring the sensitivity of the part to the whole. Moreover, this suggests that there may be a use for registering the sensitivity of services themselves in a way similar to that of goals. The first move would consist in introducing a similar to *isGoalSensitiveTo* for service, e.g. *isServiceSensitiveTo*. Again, we have the alternative between this relation and a slot in the definition of the *Service* concept. Introducing the slot would amount to a redefinition rather than an extension of the WSMO model, and we would favour having a relation.

relation isServiceSensitiveTo (ofType Service, ofType Dimension)

The same remark as with the corresponding relation having the concept *Goal* for domain can be made here. Since we may want to allow for more generic ways of stating sensitivity, we will hold off the constraining of the range of this relation to *Dimension*. The meaning of this relaxing of constraint can be seen if we consider that dimensions may be regarded as non functional properties in the sense of WSMO. However, WSMO remains rather vague on the topic of the very nature of non-functional properties and their representation, therefore so as to not preclude applicability, we prefer to leave the relation's range unconstrained.

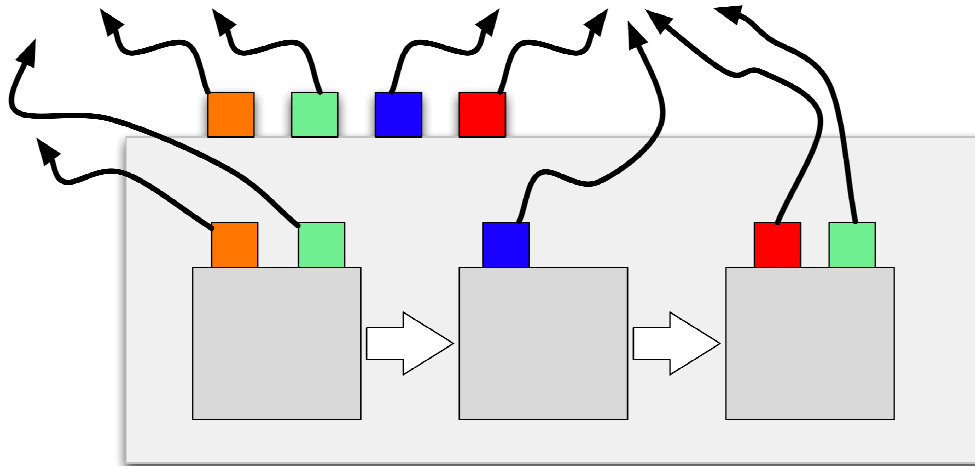


Figure 11 Complexes of goals/services that are sensitive to contextual dimensions compound these sensitivities.

4.4.3 WSMO elements sensitive to context

One way of achieving genericity and simplicity is through underspecification. This strategy which not foreign to WSMO's methodology and which has given it its flexibility (especially in its lightweight variants) can be put to use in the face of the recommendation put forward in the previous two sections. Moreover, as a further step towards generalisation, the propositions for treating goal and service sensitivity to contextual information militate towards crystallising the proposed extensions to WSMO in a generic class of elements that are sensitive to contextual information. This allows providing for both goals and services a uniform treatment of the commonalities of the mechanism supporting the capture of context sensitivity. Moreover, this allows providing a uniform treatment for the varied degrees of complexity and combinations that can be associated with an analysis of the relations between goals and services in the spirit of a process representation. This later aspect could provide support for the work developed in WP6, in particular, but in all generality for all the support provided to the different tasks that deals with such complexity throughout SOA4All.

Thus, a generic approach consists in defining a generic class of context sensitive WSMO elements, which tentatively includes both goals and services but which more conservatively includes a specialisation of *Goal*, named e.g. 'context sensitive goal', and one of *Service*, named e.g. 'service sensitive goal'. It is then possible to define slots, rather than relations for each kind of entity that would allow reflecting the sensitivity to contextual information. The fact that we have specialisations also allows for regulating different behaviours depending on whether the context sensitive element is a goal or a service, as *prima facie* it remains credible that the distinction could make at least an operational difference.

Alternatively, in order to make the solution less intrusive, a minimal implementation could maintain a single generic relation (rather than slot) of sensitivity. Because the domain would include at least a subclass of that of goals and at least a subclass of that of services but also because the range, in the light of the previous considerations, might be better off left unspecified, a conservative recommendation would be to leave the domain and range of this relation undefined. The table that follows recollect the two possible variants.

Table 11: Alternative for the representation of the concept of sensitive WSMO element. The first row gives intrusive extensions defining lasses and slots with greater degree of extension

from left to write. The second gives row the minimal non-intrusive underspecified relation.

Basic extension	Addition
concept context-sensitive-element hasSensitivityTo	concept sensitive-goal subConceptOf context-sensitive-element hasGoalSensitivityTo concept sensitive-service subConceptOf context-sensitive-element hasServiceSensitivityTo
Minimal and non-intrusive	
relation isSensitiveTo	

4.4.4 Agents with context

Finally we come back to the mention of agents that we said covered users as well as providers of services and which more generally in relation to the uses of context information would cover artificial agents (in the line of D4.1.1 generalisation of contextual information). Context information is indeed, for a start, information that relates to a prospective user and her situation. However, there is no explicit concept in WSMO that allows for representing a user and no anchoring for contextual information centred on users or any of the actors involved in the realisation of tasks related to Web service. The extension consists in introducing a generic concept of agency covering any entity from which context information may be gathered (including artificial agents and software). It is sufficient to not specify this concept but to allow for application specific (use case specific, in particular) to further specify and develop the concept and the modelling artifacts that may prove to be required or useful. Hence we will merely register the concept in the following table.

Table 12: Definition of the class Agent as generic and extendable.

Agent
concept Agent

5. Conclusions

This deliverable report discusses extensions to WSMO for capturing context information. The purpose and scope of the deliverable are defined in relation to SOA4All and its use cases. SOA4All's overall approach to context information is based on the methodology of WSMO that serves (in its lightweight variants, in particular) the purpose of producing description of Web services. The overall approach is one according to which contextual information is information with a contextual value. Hence from the standpoint of knowledge representation extending the model for the description of Web services consists in providing ontological support for the representing context information.

The present deliverable has presented a generic framework allowing the uniformisation of the treatment of contextual information as dimensions. Guidelines and examples were provided so as to guide the process of providing ontological support in the context of use cases and their use of what counts for them as contextual information. The generality of the framework is motivated by the open-end character of SOA4All application scenarios.

In addition, despite the overall implicit approach, minimal extensions to WSMO have been suggested that are intended as undistruptive as is manageable for WSMO-based models of Web services so as to preserve a degree of lightness. These suggestions highlight the main elements of WSMO that are sensitive to context (goals, services and their composition such as is forthcoming of a process oriented composition language) and attempt to provide a hook to involve the agents (conceived in a generic way) that are the focus points of contextual information.

In respect of the open-endedness of SOA4All scenarios and of the diversity of support deployed within the overall project, the extensions provided here have been kept to a large extent underspecified so as to not pre-empt refinement and specification through implementation of the application that use the captured contextual information.

6. References

- Cantera Fonseca, J. M. and Lewris, R. (eds.) Delivery Context Ontology, W3C Working Draft, 2009, <http://www.w3.org/TR/dcontology/>
- Fensel, D. et al. *Enabling Semantic Web Services: the Web Service Modeling Ontology*. Springer, 2007
- Kopecky, J., Vitvar, T. and Fensel, D. D3.4.2 WSMO-Lite: Lightweight Semantic Descriptions for Services on the Web, SOA4All Deliverable, 2008.
- Kopecky, J., Vitvar, T. and Fensel, D. D3.4.3 MicroWSMO and hRESTS, SOA4All Deliverable, 2009.
- Krummenacher, R., Lausen, H. and Strang, T. On the Modeling of Context-Rules with WSML, in Proceedings of the Workshop on Contexts & Ontologies: Theory, Practice and Applications at ECAI'06, 2006.
- Pedinaci, C. and Domingue, J. Ontology-Based Metrics Computation for Business Process Analysis. In: Martin Hepp, Knut Hinkelmann, and Nenad Stojanovic (Eds.): Proceedings of the 4th International Workshop on Semantic Business Process Management (SBPM2009), ACM International Conference Proceeding Series, ACM Press, *in press*, 2009.
- Pedinaci C., Mehandjiev, N., Moreno C R.. and Grenon, P. D4.1.1 Contextual Service Adaptation Framework, SOA4All Deliverable, 2009.
- Roman, D., Lausen, H. and Keller U. (eds.) D2v1.3 Web Service Modeling Ontology (WSMO), WSMO Deliverable, 2006. <http://www.wsmo.org/TR/d2/v1.3/>
- Strang, T., Linnhoff-Popien, C. and Frank, K. CoOL: A Context Ontology Language to enable Contextual, Proceedings of 4th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems (DAIS2003). Volume 2893 of Lecture Notes in Computer Science (LNCS), 2003. <http://www.kn.op.dlr.de/~strang/paper/dais2003/DAI>
- Steinmetz, N. and Toma, I. (eds.) D16.1v1.0 WSML Language Reference, 2008. <http://www.wsmo.org/TR/d16/d16.1/v1.0/>
- Toma, I. (ed.) D28.4v0.1 WSMO Non-functional Properties in Web services, WSMO Deliverable, 2006. <http://www.wsmo.org/TR/d28/d28.4/v0.1/>
- Wang, X. H., Gu, T., Zhang, D. Q. and Pung, H. K. Ontology Based Context Modeling and Reasoning using OWL, IEEE International Conference on Pervasive Computing and Communication (PerCom'04), 2004.