On the Modeling of Context-Rules with WSML

Reto Krummenacher and **Holger Lausen** and **Thomas Strang** [1]

Abstract. Modeling context information based on formal descriptions is a core aspect of service integration and interoperability, in particular in pervasive computing environments. In this paper we present an improved and simplified version of the Context Ontology Language modeled in WSML-Rule to show the potential use of context rules in pervasive computing applications and in particular as part of Semantic Web service descriptions.

1 INTRODUCTION

The trend towards pervasive computing is driving a need for services and service architectures that are aware of the context of the different actors (users, service providers, or third parties and their environments) involved in a service interaction: vicinity, location, QoS, ownership, time. For instance, context information can be used to reduce the amount of required user or service-service interactions, as well as to improve the user experience. A key accessor to context information in any context-aware system is a well designed model to describe contextual facts and contextual interrelationships. The context modeling approach applied in this paper is derived from the Context Ontology Language (CoOL, [7]). CoOL is based on the Aspect-Scale-Context (ASC) model also introduced in [7]. ASC defines a very simple context model in form of an extendable umbrella vocabulary that is shown to increase interoperability on the contextual level.

In this paper we improve and simplify the context modeling language by evolving its definitions based on recent advances in the field of Web-rule languages. We look in particular at the application of CoOL in combination with rule-based WSML variants [1]. This allows us to update the well designed context model and to bind it to a language family that is part of a large framework of Semantic Web languages. The WSML family of languages is a member submission to the W3C, and although it does not have the status of an official standards recommendation, we expect to be able to easily map our results into the ongoing work of the Rule Interchange Format working group [3], which will eventually endorse an official standard. Furthermore the application of rule languages allows for a simplified Context Ontology Language through the use of meta-modeling, where a concept itself can have attribute values just like any particular instance (cf. Section 4).

The paper is organized as follows. In Section 2 a short introduction to WSML is given. Section 3 provides more information about the Aspect-Scale-Conext model and the derived Context Ontology Language (CoOL). In Section 4 we show how CoOL can be modeled using WSML-Rule and how to define context-rules. We also look at possible application areas of context-rules, in particular in the area of Semantic Web services, where the WSML family resulted from. Finally we conclude with Section 5 and provide a short outlook at where and how the ideas of this paper will be further explored.

2 WSML-RULE LANGUAGE

The Web Service Modeling Language WSML [1] provides a framework for the modeling of ontologies and Semantic Web services based on the conceptual model of the Web Service Modeling Ontology WSMO [5]. WSML defines two rule-based language variants that are of interest to the issues of this paper. The first rule-based variant, WSML-Flight, semantically corresponds to the Datalog fragment of F-Logic, extended with inequality in the body and locally stratified negation under the perfect model semantics [6]. The second, WSML-Rule, extends WSML-Flight to the logic programming subset of F-Logic which allows the use of function symbols and unsafe rules (i.e., there may be variables in the rule head which do not occur in the body).

A WSML rule has the common form of *head :- body*. We illustrate this with the following example which states that every woman (rule body) is a human being (rule head):

?x **memberOf** Human :− ?x **memberOf** Woman.

Further technical details about the language are available in [1].

3 CoOL: CONTEXT ONTOLOGY LANGUAGE

The context description language applied in this paper was described in [7] and is based on the Aspect-Scale-Context model introduced in the same dissertation. On an generic level an aspect is a dimension of the situation space that is used as a collective term for information objects having the same semantic type. A scale is then seen as an unordered set of objects defining the range of valid context information instances. In other words, a valid context information with respect to an aspect is one of the elements of the aspect's scales. This results in a number of aspects that aggregate one or more scales, where each scale aggregates one or more pieces of context information. The three concepts that constitute the CoOL-core ontology are interrelated by use of the attributes hasAspect, hasScale, hasMember and usedByScale (cf. Figure 1).

Through the combination of meta-data instances, CoOL allows the provision of higher order context information or the binding of quality measures. In [7] meanError, timestamp and hasQuality were proposed for any context information instance.

A particular strength of the presented context model not yet mentioned is the infrastructure defined to map semantically related scales of one aspect or to combine and interlink different scales to new scales of hybrid aspects. There are two types of operations in CoOL: (1) IntraOperations that provide translations from one scale to another, e.g. from Kilometer to Miles of a DistanceAspect, and (2)

[1] Digital Enterprise Research Institute, University of Innsbruck, Austria, email: firstname.lastname@deri.org

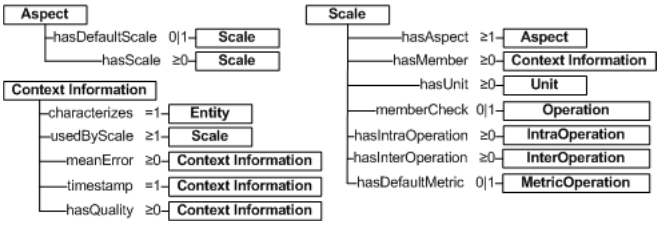

Figure 1. The ASC ontology from [7]

InterOperations that allow for example the definition of a KilometerPerHourScale of the SpeedAspect as a combination of a KilometerScale and an HourScale.

More details about the ASC model are provided in the next section where we first of all discuss some simplifications and improvements.

4 CONTEXT RULE MODELLING IN WSML

In this section we present CoOL written in WSML-Rule[2] (Listing 1). Note first of all a minor change in the model with respect to the original ontology (Figure 1): we feel that a context information is not used by a scale, but rather that the context information is encoded as given by a scale. Hence, we suggest to use the attribute inScale instead.

Listing 1. CoOL-core written in WSML-Flight.

```
concept Aspect
  hasDefaultScale ofType (0 1) Scale
  hasScale ofType Scale

axiom DefaultScaleSubScale definedBy
  ?a[hasScale hasValue ?s]
    :- ?a[hasDefaultScale hasValue ?s] memberOf Aspect.

concept Scale
  hasAspect inverseOf(hasScale) ofType (1 *) Aspect
  hasMember ofType ContextInformation
  hasUnit ofType Unit
  memberCheck ofType _iri
  hasIntraOperation _iri
  hasInterOperation ofType _iri
  hasDefaultMetric ofType (0 1) _iri

concept ContextInformation
  characterizes impliesType (1) Entity
  inScale inverseOf(hasMember) ofType (1 *) Scale
  meanError ofType ContextInformation
  timestamp ofType (1) ContextInformation
  hasQuality ofType ContextInformation
```

Based on the core concepts of CoOL it is now possible to define particular aspects, scales and pieces of context information. Listing 2 shows the necessary concepts and instances to gather information about distances in either kilometer or miles. Using the WSML language constructs like **inverseOf** keeps the definition of a domain ontology for distance measurements short and simple without loosing e.g., the aspect-scale or scale-aspect bindings. The context information containers are explicitly given by the axioms that bind them to a given distance scale (Listing 2).

CoOL has so-called *memberCheck* operations (Figure 1) that ensure correctly scaled values for context information (i.e. that they obey the type of the scale). In WSML such constraints can in simple cases directly be expressed within the conceptual syntax. In our example the values are constrained to the datatype float directly in the axiomatic definition of *KmCI* or *MilesCI*. The semantics of WSML ensures that if instances exist in a model that do not obey these constraints, such a model is inconsistent and in fact no valid model at all. For more complex value constraints it is always possible to bind an external operation to the model, as will be described later in this section.

Listing 2. An example of CoOL-WSML for distance information

```
instance DistanceAspect memberOf Aspect
  hasDefaultScale hasValue KilometerScale

instance KilometerScale memberOf Scale
  hasAspect hasValue DistanceAspect

instance MilesScale memberOf Scale
  hasAspect hasValue DistanceAspect

axiom defaultScaleKmCI definedBy
  ?kci[inScale hasValue KilometerScale, value ofType _float]
    :- ?kci memberOf KmCI .

axiom defaultScaleMiCI definedBy
  ?mci[inScale hasValue MilesScale, value ofType _float]
    :- ?mci memberOf MilesCI .
```

For a better understanding we first elaborate on the example in Listing 2. There is one aspect, the DistanceAspect, defined in the domain ontology that represents one possible context dimension: spatial distance. The default scale for distance measurements is defined to be the KilometerScale. A second possible scale would be the MilesScale. The aspects and scales are modeled as instances of the CoOL-core Aspect respectively Scale concepts, while the context information objects are implicitly defined as concepts (KmCI, respectively MilesCI) to provide containers for all collected instances, i.e. pieces of information.

As shown in Listing 3, it is possible to directly axiomatize simple intra operations within WSML, they can be modeled by rules, which transparently make values of context information available in different scales (e.g. Miles and Kilometer). The axiom *km2miOperation* infers for example the context information in the MilesScale from some in the KilometerScale. We use a function symbol to generate an identifier for inferred context information to distinguish between inferred and measured information. The rule states that every measurement that is taken using the KilometerScale is equivalent to a value in the MilesScale divided by 1.609.

Listing 3. Axiomatic IntraOperations.

```
axiom km2miOperation definedBy
  km2mi(?info)[inScale hasValue MilesScale, value hasValue ?mi,
       characterizes hasValue ?entity] memberOf MilesCI
    :- ?info[inScale hasValue KilometerScale, value hasValue ?km,
       characterizes hasValue ?entity] memberOf KmCI
    and wsml#numericDivide(?mi,?km,1.609) and naf ?info = mi2km(?i) .
```

In order to provide the same information for intra-scale operations as in [7] we suggest to use non-functional properties to annotate the mapping axioms. The *fromScale* property indicates the source scale, while *toScale* provides a link to the target scale. Most intra-operations demand a simple value transformation to cope with different units. Similar mappings exist for a TemperatureAspect where Kelvin, °C and °F would have to be interlinked to make the information compatible even though it results from heterogeneous data sources.

Before looking at the definition of context rules we shortly add some distance measurements to our knowledge base. The distance is either given by an explicit measured instance or by an inferred instance generated on-the-fly by an appropriate axiom:

```
axiom measurements definedBy
  _# memberOf KmCI[value hasValue ?d, characterizes hasValue DistAB]
     and distKM(?d,A,B) .
  _# memberOf KmCI[value hasValue 14, characterizes hasValue DistAC] .
  _# memberOf MilesCI[value hasValue 8.5,
       characterizes hasValue DistAD] .
```

[2] The complete listings are at http://members.deri.org/~retok/cool/

By now the reader should be familiar with the context modeling ontology and with the way context domains and context information are defined using WSML-Rule.

A context rule is an axiom that is defined by an implication where the body is a set of conditions using the context information in the knowledge base. Rules either infer new knowledge or return information if posted in form of queries. The following example queries distance entities that represent nearby locations. The resulting distance value shall be provided by an MilesCI instance of a scale that belongs to the aspect DistanceAspect and shall be smaller than 10 miles:

```
? − ?_c[ characterizes hasValue ?entity , value hasValue ?distMiles ,
        hasScale hasValue ?s] memberOf MilesCI
    and ?_s[hasAspect hasValue DistanceAspect] and ?distMiles < 10 .
```

The query returns for the given measurements the following matches[3]:

?entity	?distMiles
DistA2C	8.7
DistA2D	8.5

In WSMO [5] the vocabularies, constraints and logical expressions that are defined in ontologies are used to describe the functionality (capability) and interfaces of Web services. The just defined query could be used to include restrictions on the spatial distance between the service provider and requester. It could for example be envisioned that a pizza delivery service only accepts orders from clients that call from at most ten miles from the pizza store. Hence, a precondition of such a pizza ordering service would include a constraint that uses the context rule to ensure the desired maximal spatial distances.

This leads us to another interesting feature that the WSML framework provides. WSMO and in consequence WSML were developed to annotate Web service descriptions. In [7] the various operations are offered by external services that are linked into the model by use of operation bindings (Figure 1). We have already shown that many IntraOperations and member checks can be modeled by axiomatic expressions, while for the more complex ones, as well as for InterOperations and MetricOperations WSML provides us with the means of Web service descriptions within the same framework and thus based on the same notations and vocabularies.

Listing 4. A Web service description to link InterOperations

```
webService _"http :// www.example.org/interOpService"

  capability
    precondition definedBy
        ?i1 memberOf KilometerScale and ?i2 memberOf HourScale .
    postcondition definedBy
        ?o memberOf KmPerHourScale .
```

The shown service description (Listing 4) contains a capability description that uniquely indicates the constraints on the input and output parameters of the service that computes the kilometer per hour scale (Listing 5). The description of the grounding and interaction patterns with the Web service are assumed to be given in an external file, as this would exceed the scope of this paper. The goal is to reconsider the strength of CoOL and to show the advantages of modeling it with WSML, in particular with WSML-Rule.

Listing 5. A scale definition with IntraOperation binding

```
instance KilometerPerHourScale memberOf Scale
  hasAspect hasValue SpeedScale
  hasInterOperation hasValue _"http :// www.example.org/interOpService"
```

[3] WSML-Rule reasoner: http://tools.deri.org/wsml/rule-reasoner

5 CONCLUSION

Context-awareness, and as a crucial intermediate step the provision of concise context models, is a core research area of pervasive computing. Encoding context information by use of ontologies allows for formal descriptions of characteristics and states of entities. The ASC model and the derived Context Ontology Language CoOL provide a simple and extensible model based on aspect-scale-context interrelations.

In this paper we used the rule-based languages of the WSML language family to improve and simplify the language bindings proposed in [7]. The use of meta-modeling and the fact that WSML provides a set of languages that can be mapped to various types of logical formalisms which are already well integrated into the rule efforts of the Semantic Web allows for an even more concise, yet simultaneously extensible and globally applicable, umbrella framework for the modeling of context information.

This is exactly the convergence of technologies that is envisioned to be necessary to fully explore the use of context information in the field of service interoperability and information exchange on the context level. The generic character of the ASC model and the well-integratedness of WSML into the Semantic Web standardization activities allows this combined approach to become a context-modeling framework that could provide the backbone for large-scale context-aware applications on the Web. The requested and provided context information of various heterogenous information sources, sinks and services can hence be combined, processed and mapped on the machine level. In that sense, the ideas presented are expected to also improve Semantic Web services frameworks like WSMX [2] or service coordination infrastructures like Triple Spaces [4] by allowing their components to become context-aware. The upcoming work is thus concerned with enhancing the functionality-centered static descriptions of Web services to additionally consider dynamic characteristics like location, connectivity or quality to provide improved discovery, selection and coordination of services — a requirement for the access and composition of services in ubiquitous computing environments.

ACKNOWLEDGEMENTS

The work is funded by the European Commission under the projects DIP, Knowledge Web, SEKT, and ASG; by the Austrian Federal Ministry for Transport, Innovation, and Technology under the FIT-IT projects RW^2 and TSC.

REFERENCES

[1] J. de Bruijn, H. Lausen, A. Polleres, and D. Fensel: The web service modeling language: An overview, Proc. 3rd European Semantic Web Conf., June 2006.

[2] E. Cimpian, and M. Zaremba (eds.): Web Service Execution Environment (WSMX), W3C Member Submission, June 2005.

[3] S. Hawke (ed.): Rule Interchange Format Working Group Charter, W3C Semantic Web Activity, 2005.

[4] R. Krummenacher, M. Hepp, A. Polleres, Ch. Bussler, and Dieter Fensel: WWW or What is Wrong is with Web Services, Proc. 3rd European Conf. on Web Services, 2005.

[5] H. Lausen, A. Polleres, and D. Roman (eds.): Web Service Modeling Ontology (WSMO), W3C Member Submission, June 2005.

[6] T. C. Przymusinski. On the declarative and procedural semantics of logic programs. Journal of Automated Reasoning, 5(2), 1989.

[7] Th. Strang: Service interoperability in ubiquitous computing environments. PhD dissertation, LMU Munich, 2003.