CCBR Ontology for Reusable Service Templates[1]

Charlie Abela, Matthew Montebello
Department of Computer Science and AI
University of Malta
+356 2590 7295

{charlie.abela, matthew.montebello}@um.edu.mt

ABSTRACT
We present the motivation and design of CCBROnto, an OWL Ontology for Conversational Case-Base Reasoning (CCBR). We use this ontology to define cases that can eventually be stored, retrieved and reused by a mixed-initiative approach based on CCBR. We apply this technique for retrieving Web Service Composition templates.

Categories and Subject Descriptors
I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods

General Terms
Design, Implementation

Keywords
CCBR, Ontologies, OWL, Web Services

1. INTRODUCTION
Web Services composition is usually interpreted as the integration of a number of services into a new workflow or process. A number of compositional techniques have been researched [9,10] that attempt to address service composition by composing web services from scratch while ignoring reuse or adaptation of existing compositions or parts of compositions. Furthermore composing web services by means of *concrete* service interfaces leads to tightly-coupled compositions in which each service involved in the chain is tied to a web service instance. This approach may lead to changes in the underlying workflow which range from slight modifications of bindings to whole redesigning of parts of the workflow description. Therefore we interpret services at an abstract level to facilitate their independent composition. Infact our approach is more similar to [8,11,12], which use pre-stored abstract workflow definitions or templates in their composition framework. Abstract workflows allow for more generalisations and a higher level of reusability [5]. The use of such templates can be thought of as a pre-processing stage towards service discovery and composition, whereby abstractly defined workflow knowledge can be concretely bound to actual services that satisfy a template. To make effective reuse of such templates we have considered CCBR [6]. This extends from CBR and allows for partial definition of the problem by using a mixed-initiative refinement process to identify more clearly the user's problem state.

2. RELATED WORK
In recent work relating CBR to the Semantic Web [2, 4], we find the definition of two ontologies, CaseML and CBROnto. These are both defined for CBR rather then CCBR and thus do not define concepts related to question-answer (QA) pairs, which are at the core of the CCBR process. Nonetheless we considered these when we designed and implemented our OWL-based ontology, which we call CCBROnto (this has no relation to CBROnto). We make use of this ontology within our personalised service discovery and composition framework (PreDiCtS) to define cases of best practice composition knowledge. In what follows we make explanatory references to this ongoing work.

3. CCBRONTO
In CCBROnto the basic components of a *Case* are defined by the *CaseContext*, *Problem* and *Solution* classes. This structure is motivated by the underlying methodology used in PreDiCtS. In this framework we adapt the CCBR approach to help the user refine his query for a particular service request. The problem description is defined by a set of discriminating QA pairs, which characterize a particular solution. On the other hand, the solution is a place holder for a reusable service composition template which is a container of best practice knowledge about composition of generic service components. In the following sections we will explain in more detail the basic *Case* components and illustrate by means of an example how such a case is defined.

3.1 Context
In [3], the term context is defined as "*any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves*".

We fully agree with this definition and in the *CaseContext* we have included knowledge related to the case creator, case history, and case provenance. We have also considered ideas presented in [7] and [1] which discuss the importance of context in relation to Web Services. In PreDiCtS context knowledge helps to identify, *(i)* why a case was created and by whom, *(ii)* certain aspects of case usage and *(iii)* the case relevance to problem solving. The *CaseCreator* includes a reference to the *Role* description, that the creator associates himself with, together with a *foaf:Person* instance-definition that describes who this person is. The motivation behind using foaf is to keep

Demos and Posters of the 3rd European Semantic Web Conference (ESWC 2006), Budva, Montenegro, 11th - 14th June, 2006

1This research has partially been funded by the European Commission and by the Swiss Federal Office for Education and Science within the 6th Framework Program project REWERSE number 506779

track of reputation knowledge which could be used to reliably share cases between PreDiCtS users.

```
<ccbr:Case rdf:ID="case1">
  <ccbr:CaseContext rdf:ID="cntxt1">
    <ccbr:hasProvenanceURI rdf:resource="http://www.....org"/>
    <ccbr:hasCaseCreator>
      <ccbr:CaseCreator rdf:ID="ccr1">
        <ccbr:hasRole rdf:resource="&role;#KnowledgeEng"/>
        <foaf:Person>
          <foaf:name>Joe Black</foaf:name>
          <foaf:mbox rdf:resource="mailto:joe@test.org"/>
          <foaf:homepage rdf:resource="http://www...../joe"/>
        </foaf:Person>
      </ccbr:CaseCreator>
    </ccbr:hasCaseCreator>
  </ccbr:CaseContext>
  <ccbr:Problem rdf:ID="prob1">
    <ccbr:QAPairList>
      <list:first>
        <ccbr:QAPair rdf:nodeID="quest1"/>
      </list:first>
      <list:rest rdf:resource="&list;#nil"/>
    </ccbr:QAPairList>
  </ccbr:Problem>
  <ccbr:Solution rdf:ID="sol2">
    <ccbr:hasAction>
      <ccbr:OWLSTemplate rdf:ID="tmpl3">
        <ccbr:hasServiceTemplate rdf:resource="#Trav_Serv"/>
        <ccbr:hasProcessTemplate rdf:resource="#Trav_Proc"/>
        <ccbr:hasProfileTemplate rdf:resource="#Trav_Prof"/>
      </ccbr:OWLSTemplate>
    </ccbr:hasAction>
  </ccbr:Solution>
</ccbr:Case>
```

Figure 1: CCBROnto Case instance definition

The *CaseContext* also provides a place holder for *CaseHistory*, which becomes important when it comes to case ranking and usage, since it allows users to identify the relevance and usefulness of a case in solving a particular problem. It is also important for the case administrator when case maintenance is performed. Cases whose history indicates negative feedback may be removed from the case base. Case *Provenance* is also used in conjunction with reputation issues, since it associates a case with a URL indicating the case-origin.

3.2 Problem

The *Problem* state description in a PreDiCtS case is based on the taxonomic theory of [6]. Every problem is described by a list of QA pairs rather than a bag. This is required since QA pairs have to be ranked when they are presented to the user. Each QA pair consists of a *CategoryName*, a *Question* and an *Answer*. Since the taxonomic theory requires that QA pairs are defined in a taxonomy during the case creation stage, each question description is associated, through the property *isRelatedTo*, with an ontological concept defined in the domain of discourse. This relation is not intended to fully capture the natural semantics of the QAs, rather it is important when calculating similarities.

A typical QA pair example from the traveling domain might include the question, "*What type of transportation?* This is related, by means of the *isRelatedTo* property, to the concept *Transportation*, which is defined in the Traveling domain. On the other hand, we assume that *Answers* could have either a binary or nominal value and are respectively defined in the ontology by the *YesNoAnswer* and *ConceptAnswer* classes. The former points to the binary literals, while the latter is used to represent answers that are associated to a concept in a domain ontology through the previously mentioned *isRelatedTo* property.

3.3 Solution

The solution in PreDiCtS provides a hook where composition templates can be inserted. Each *Solution* is defined to be an *Action* which has a description and a composition template. A template can be sub-classed by a description such as that defined by OWL-S, as shown in Figure 1, though in practice it can be specialized also by other service descriptions.

4. CONCLUSION

Through the use of CCBROnto we are able to define cases whose solutions are composition templates. This allows our PreDiCtS framework to retrieve such templates by consulting the user in every stage and presenting her with the most suitable composition knowledge available to choose from. The user can then decide whether to reuse as is, or possibly adapt this to fit her personal needs.

5. REFERENCES

[1] Bry, F., et al, *Context Modeling in OWL for Smart Buildings*, in proceedings of GvB2005.

[2] Chen, H., et al, *CaseML: RDF based markup language for CBR on the Semantic Web*, ICCBR 2003

[3] Dey, A.K., *Understanding and Using Context*, in proceedings of Personal and Ubiquitous Computing, issue on Situated Interaction and Ubiquitous Computing, February 2001.

[4] Diaz-Agudo, B., et al, *On Developing a Distributed CBR Framework through Semantic Web Services*, OWL Experiences and Directions Workshop, Galway, 2005

[5] Goderis, A., et al, *Seven bottlenecks to workflow reuse and repurposing*, 4th Int. Semantic Web Conf., Galway, Ireland, 6-10 Nov. 2005

[6] Gupta, K., *Taxonomic Conversational Case-Based Reasoning*, 4th International Conference on CBR, 2001

[7] Maamar, Z., et al, *Context for Personalized Web Services*, 38th Hawaii International Conference on system Science, 2005

[8] Rajasekaran, P., et al, *Enhancing Web services description and discovery to facilitate composition*, First International Workshop, SWSWPC, July 2004

[9] Sirin, E., Parsia, B., et al, *Filtering and selecting semantic web services with interactive composition techniques*, IEEE Intelligent Systems, 19(4): 42-49,2004

[10] Sirin, E., et al, *Planning for web service composition using SHOP2*, Journal of Web Semantics, 1(4):377-396, 2004

[11] Sirin, E., et al, *Template-based composition of semantic web services*, AAAI Fall Symposium on Agents and the Semantic Web, Virginia, November 2005.

[12] Weber, B., et al, *CCBR-Driven Business Process Evolution*, Proc. 6th Int. Conf. on Case-Based Reasoning (ICCBR'05), Chicago, August 2005