Decentralized Case-Based Reasoning for the Semantic Web

Mathieu d'Aquin, Jean Lieber, Amedeo Napoli

Decentralized Case-Based Reasoning
for the Semantic Web

Mathieu d'Aquin, Jean Lieber, and Amedeo Napoli

LORIA (INRIA Lorraine, CNRS, Nancy Universities),
Campus scientifique, BP 239,
54 506 Vandœuvre-lès-Nancy, France
{daquin, lieber, napoli}@loria.fr

Abstract. Decentralized case-based reasoning (DzCBR) is a reasoning frame-
work that addresses the problem of adaptive reasoning in a multi-ontology environ-
ment. It is a case-based reasoning (CBR) approach which relies on contextualized
ontologies in the C-OWL formalism for the representation of domain knowledge
and adaptation knowledge. A context in C-OWL is used to represent a particular
viewpoint, containing the knowledge needed to solve a particular local problem.
Semantic relations between contexts and the associated reasoning mechanisms al-
low the CBR process in a particular viewpoint to reuse and share information about
the problem and the already found solutions in the other viewpoints.

1 Introduction

This paper presents a research work on the application of case-based reasoning (CBR,
see e.g. [1,2]) within the semantic Web technologies and principles. CBR is a type
of analogical reasoning in which problem-solving is based on the adaptation of the
solutions of similar problems, already solved and stored in a case base. In particular,
knowledge-intensive CBR (KI-CBR [3]) relies on a knowledge base including domain
knowledge and, as well, knowledge units exploited for the retrieval and adaptation op-
erations of CBR.

Ontologies are at the heart of semantic Web technologies and OWL is the standard
language for representing ontologies [4]. An ontology is used for the conceptualization
of a particular domain and for knowledge exchange. The OWL language allows the
use of deductive reasoning mechanisms, such as classification and instantiation. In this
paper, we want to show that the classical deductive reasoning made in the semantic
Web technologies may be completed and enhanced with KI-CBR that may take advan-
tage of domain ontologies and provide an operationalization for reasoning by analogy.
Moreover, the representation of the knowledge used for adaptation in CBR (*adaptation
knowledge*) must be integrated within ontologies.

Usually, the adaptation knowledge is dependent on the application context. For ex-
ample, a Web service applying CBR for advising customers on computer sales will
consider a male customer as similar to a female customer. However, in a case-based
Web service dedicated to fashion advises, a male and a female customers have to be
considered as dissimilar. In other terms, the knowledge for CBR is dependent on the
considered viewpoint, i.e. on the type of problem that the system has to solve.

C-OWL (for context-OWL) is a formalism that has been recently proposed [5] for the representation of mappings between several OWL ontologies for the purpose of ontology alignment. A local ontology in C-OWL is considered as a context, having its own language and its own interpretation. Mappings are made of bridge rules that express semantic relations between classes, properties and individuals of the local ontologies. In this way, aligning ontologies using C-OWL allows the coordinated use of these ontologies, keeping the knowledge contained in each of them in its local context. Moreover, beyond ontology alignment, C-OWL can be used for representing modular ontologies, combining different viewpoints on the same domain, and this is how we use it hereafter.

In this paper, we propose DzCBR (decentralized case-based reasoning), a KI-CBR mechanism that exploits the decentralized knowledge represented in a C-OWL contextualized ontology. Each context of a contextualized ontology is used for representing a particular viewpoint, containing the domain knowledge and the adaptation knowledge needed for solving a particular type of problem. Several DzCBR processes are then distributed among these viewpoints, each one being carried out locally in a context and relying on local knowledge. Collaboration between these multiple local processes is implemented thanks to C-OWL bridge rules and to the associated reasoning mechanisms. In this way, decentralized problem-solving is based both on local knowledge, for a particular viewpoint, and on the combination of several viewpoints. DzCBR is a new paradigm that we have designed and that we currently use in an application in oncology. The roots of decentralized reasoning can be found in pattern recognition and distributed artificial intelligence [6], and we have extended this approach within the C-OWL formalism, to design DzCBR and to enhance problem-solving capabilities for the semantic Web.

The next section presents a motivating application of DzCBR in the domain of oncology. In the section 3, CBR and its integration in the semantic Web framework are detailed. A short introduction to C-OWL follows in section 4. The section 5 details the knowledge and reasoning models of DzCBR, and how problem-solving is carried out by combining several decentralized viewpoints represented by C-OWL contexts. An example of a DzCBR process applied to a breast cancer treatment problem is presented in section 6. Finally, the related work is discussed in section 7, and the section 8 concludes the paper.

2   Motivating Application: Adaptation Within Multiple Viewpoints in Oncology

Oncology is a complex domain where several specialties, e.g. chemotherapy, surgery and radiotherapy, are involved in several treatment phases. In most cases, the adequate therapeutic decision is given according to a protocol that associates standard patient characteristics with a recommended treatment. Even if it is designed to take into account the majority of the medical cases, a protocol does not cover all the situations. Decisions concerning patients out of the protocol are elaborated within a multi-disciplinary expert committee, and rely on the adaptation of the solutions provided by the protocol for similar cases. Specialties in oncology organize their background knowledge and

past experiences in different ways. Indeed, a protocol is structured according to the oncology specialties and, during a meeting of an expert committee, each expert from each specialty supplies a personal view on the solution as a part of a collective solution. For each specialty, a particular type of treatment is requested, in a particular treatment phase, and the patient characteristics used to find the solution change from one specialty to another. Thus, oncology specialties provide different viewpoints on oncology, and these viewpoints are related to each other. Information about a problem, e.g. finding a therapeutic decision for a patient, can be shared across specialties, and decisions taken in a particular specialty may influence decisions taken in another one.

A protocol contains the standard knowledge for decision making in oncology. As a standard Web formalism for knowledge representation and exchange, OWL is a well-suited language for Furthermore, reasoning mechanisms associated with OWL, such as classification and instantiation, may be used to provide intelligent access to this knowledge, for the purpose of decision support in oncology. In the perspective of decision support for out of the protocol cases, a KI-CBR mechanism relying on a formalized protocol may be applied. In this way, the knowledge used by expert committees is represented and operationalized in the form of adaptation knowledge to become sharable and reusable. Knowledge representation and reasoning have to take into account the multiple viewpoints involved in the decision, corresponding to oncology specialties. C-OWL provides a formalism for representing several alternative representations of the domain and for relating these local representations to each other. Thus, domain knowledge (contained in a protocol) as well as adaptation knowledge are represented within contextualized ontologies in C-OWL. A KI-CBR mechanism may be used with profit for exploiting such decentralized knowledge. The framework of DzCBR is proposed here for this purpose.

3   Case-Based Reasoning with OWL

3.1   Principles of Case-Based Reasoning

A case is a problem solving episode usually represented by a *problem* pb and a *solution* Sol(pb) of pb. A case base is a (usually structured) set of cases, called *source cases*. A source case is denoted by (srce, Sol(srce)). CBR consists in solving a *target problem*, denoted by tgt, thanks to the case base. The classical CBR process relies on two steps, retrieval and adaptation. *Retrieval* aims at finding a source problem srce in the case base that is considered to be similar to tgt. The role of the *adaptation* task is to adapt the solution of srce, Sol(srce), in order to build Sol(tgt), a solution of tgt. Then, the solution Sol(tgt) is tested, repaired, and, if necessary, memorized for future reuse.

In knowledge intensive CBR (KI-CBR, see e.g. [3, 7, 8]), the CBR process relies on a formalized model of domain knowledge. This model may contain, for example, an ontology of the application domain, and can be used to organize the case base for case retrieval. KI-CBR may also include some knowledge for adaptation, as explained in the following.

3.2 Reformulations: An Approach for Representing Adaptation Knowledge

Reformulations are basic elements for modeling adaptation knowledge for CBR [9]. A reformulation is a pair $(r, \mathcal{A}_r)$ where r is a relation between problems and $\mathcal{A}_r$ is an *adaptation function*: if r relates srce to tgt –denoted by "srce r tgt"– then any solution Sol(srce) of srce can be adapted into a solution Sol(tgt) of tgt thanks to the adaptation function $\mathcal{A}_r$ –denoted by "Sol(srce) $\mathcal{A}_r$ Sol(tgt)".

In the reformulation model, retrieval consists of finding a *similarity path* relating srce to tgt, i.e. a composition of relations r_k, introducing intermediate problems pb_k between the source and the target problems. Every r_k relation is linked by a reformulation to an adaptation function $\mathcal{A}_{r_k}$. Thus, the sequence of adaptation functions following the similarity path may be reified in an *adaptation path* (see figure 1).
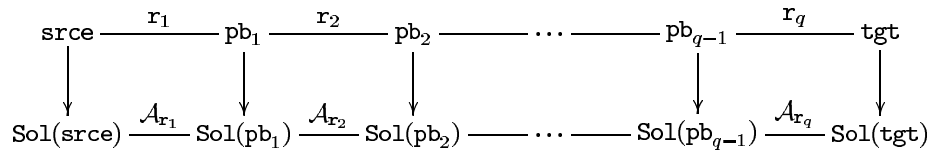
Fig. 1. A similarity path from srce to tgt (first line) and the corresponding adaptation path (second line)

The model of reformulations is a general framework for representing adaptation knowledge. The operations corresponding to problem relations r_k and adaptation functions $\mathcal{A}_{r_k}$ have to be designed for a particular application. Generally, these operations rely on transformation operations such as specialization, generalization and substitution, that allow the creation of the pb_k problems for building the similarity path and of the Sol(pb_k) solutions for the adaptation path: relations of the form pb_1 r pb_2 and adaptation like Sol(pb_1) $\mathcal{A}_r$ Sol(pb_2) correspond to applications of such transformations.

Moreover, the reformulation framework follows the principle of adaptation-guided retrieval [10]. A CBR system using adaptation-guided retrieval retrieves the source cases whose solution is adaptable, i.e. for which adaptation knowledge is available. According to this principle, similarity paths provide a kind of symbolic reification of similarity between problems, allowing the case-based reasoner to build understandable explanation of the results.

3.3 A Brief Introduction to OWL

OWL is the standard formalism for the representation of ontologies for the semantic Web. In OWL, the knowledge about a domain is represented within an ontology. An OWL ontology contains definitions of classes, properties and individuals from the represented domain. An *individual* corresponds to an object. A *property* denotes a binary relation between objects. A *class* represents a set of objects. Formally, the semantics of

an OWL ontology is given by an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non empty set called the interpretation domain, and $\cdot^{\mathcal{I}}$ is the interpretation function. This function maps a class C into a subset $C^{\mathcal{I}}$ of the interpretation domain $\Delta^{\mathcal{I}}$, a property p into a subset $p^{\mathcal{I}}$ of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and an individual a to an element $a^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$.

An OWL *ontology O* is defined by a set of axioms and a set of assertions. Classes are introduced through the use of *axioms* of the form[1] C $\sqsubseteq$ D , C and D being two classes. C $\sqsubseteq$ D is satisfied by an interpretation $\mathcal{I}$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. C $\equiv$ D is a notation for C $\sqsubseteq$ D and D $\sqsubseteq$ C. *Assertions* are used to introduce individuals. The two possible types of assertions are C(a) and p(a,b), C being a class, a and b two individuals, and p a property. C(a) is satisfied by an interpretation $\mathcal{I}$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ and p(a,b) is satisfied by $\mathcal{I}$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in p^{\mathcal{I}}$. $\mathcal{I}$ is a model of O if it satisfies all the axioms and assertions defining O. OWL provides constructors for building complex classes and complex properties. For example, a class conjunction, C $\sqcap$ D, is interpreted as an intersection ($C^{\mathcal{I}} \cap D^{\mathcal{I}}$), and the existential quantifier, $\exists p.C$, represents the set $(\exists p.C)^{\mathcal{I}}$ of the objects being in relation with at least one object from $C^{\mathcal{I}}$ by the property p. The syntax and semantics of all the OWL constructors can be found in [4], but only some of them are used in the examples of this paper.

3.4 CBR Within OWL Ontologies

In OWL, problems and solutions are represented as instances of the Problem and Solution classes. The link between a problem pb and its solution Sol(pb) is materialized by a property called hasSolution. OWL axioms are used to relate Problem and Solution to classes of the domain knowledge. For example, in the application for breast cancer treatment, the Patient and Treatment classes correspond respectively to the Problem and Solution classes, and thus, the two axioms Patient $\sqsubseteq$ Problem and Treatment $\sqsubseteq$ Solution are added to the ontology. Furthermore, the hasSolution property relates patients to the recommended treatments. Problem relations, adaptation functions and reformulations are also formalized in OWL. The specific underlying mechanisms are made by Web services implementing transformation operations like specialization, generalization and property substitution on OWL individuals.

Given two classes C and D, the *subsumption test* in OWL is defined by C is subsumed by D (C is more specific than D) if, for every model $\mathcal{I}$ of O, $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. Based on the subsumption test, *classification* consists of finding for a class C, the most specific classes in the ontology subsuming C, and the most general classes subsumed by C. Classification organizes the classes of the ontology in a hierarchy. Regarding CBR, the class hierarchy is used as a structure for the case base, where a class represents an index for a source problem. Every index is considered as an abstraction of a source problem, containing the relevant part of the information leading to a particular solution.

Instance checking tests whether an individual a is an instance of a class C, i.e. if for every model $\mathcal{I}$ of O, $a^{\mathcal{I}} \in C^{\mathcal{I}}$. It supports the *instantiation* reasoning service that consists of finding the most specific classes of an individual. It is used during the retrieval step of CBR for finding index classes of source problems. A source prob-

---

[1] In this paper, we use the description logic way of writing expressions instead of the RDF/XML syntax and of the abstract syntax of OWL.

lem `srce` is an instance of its index class `idx(srce)`, and its solution `Sol(srce)` is considered to be reusable for any problem `pb` that is an instance of `idx(srce)`, i.e. `Sol(srce)` can be reused to solve `tgt` whenever `tgt` is recognized as an instance of `idx(srce)`.

Instantiation is used to infer new pieces of information about an individual on the basis of its class membership, and of constraints contained in class definitions. For example, if an individual named `bob` is an instance of the class `Man`, if `Man` is declared to be more specific than `Human` (`Man ⊑ Human`), and if the capability of intelligence is associated with humans (`Human ⊑ ∃capability.Intelligence`), then, `bob` has to be capable of intelligence. The information known about `bob` is automatically completed, thanks to constraints inherited from `Human`. This reasoning service has proved to be useful for CBR in [7], where it is called *instance completion*. Particularly, it is used in the *problem elaboration* operation, to extend the available information on the target problem with respect to the domain knowledge. Moreover, since a particular index `idx(srce)` may lead to a particular solution `Sol(srce)`, this solution can be directly attached to the index class through a *problem-solution axiom* of the form: `I ⊑ ∃hasSolution.S`. This means that, based on instance completion, any instance of the index class `I` is related to an object of the solution class `S` by the `hasSolution` property.

4 An Introduction to C-OWL

4.1 C-OWL: Contextualizing Ontologies

C-OWL is an extension of OWL for representing *contextualized (or contextual) ontologies* [5]. Contextualized ontologies are local representations of a domain, named contexts, that are semantically related with other contexts thanks to mappings. The original motivation for C-OWL is the alignment and coordinated use of ontologies made for different purposes. In our framework, C-OWL is used as a way to formalize and implement several alternative representations of the domain that we call *viewpoints*. In C-OWL, the knowledge about a domain is contained in a set of contexts. Each context O_i is an OWL ontology, with its own language and its own interpretation. Mappings are expressed by bridge rules. A bridge rule from O_i to O_j is a way to declare a correspondence between the interpretation domains of these two contexts. On the basis of these correspondences, a part of the knowledge contained in O_i can be interpreted and reused in O_j.

Formally, a C-OWL *context space* contains a set of contexts $\{O_i\}_{i \in I}$, I being a set of indexes for contexts. The indexes of I are used to prefix the expressions, associating an expression to the context in which it is defined. For example, `i:C`, `i:∃p.C`, `i:a`, `i:C ⊑ D` and `i:C(a)` are expressions of the local language of O_i.

The semantics of a context space is given by a distributed interpretation $\Im$ that contains an interpretation $\mathcal{I}_i$ for each $i \in I$. Each $\mathcal{I}_i$ is composed of a local interpretation domain $\Delta^{\mathcal{I}_i}$ and a local interpretation function $\cdot^{\mathcal{I}_i}$. A context is interpreted with the corresponding local interpretation, i.e. an axiom or an assertion of O_i is satisfied by $\Im$ if it is satisfied by $\mathcal{I}_i$.

A mapping $\mathcal{M}_{ij}$ is a set of *bridge rules* from O_i to O_j. There are different types of bridge rules, occurring between classes, individuals or properties of two contexts. We are only interested here in some particular forms. An *into rule* is a bridge rule of the form $\texttt{i:C} \xrightarrow{\sqsubseteq} \texttt{j:D}$, where $\texttt{i:C}$ and $\texttt{j:D}$ are classes respectively from O_i and O_j. This type of rule means that the class $\texttt{i:C}$ of O_i is considered, from the viewpoint of O_j, as more specific than the class $\texttt{j:D}$ [11]. The *onto rule* $\texttt{i:C} \xrightarrow{\sqsupseteq} \texttt{j:D}$ means that O_j considers the class $\texttt{i:C}$ to be more general than $\texttt{j:D}$. Bridge rules are directional: a bridge rule from O_i to O_j is considered in the viewpoint of O_j, and so, $\texttt{i:C} \xrightarrow{\sqsubseteq} \texttt{j:D}$ is not equivalent to $\texttt{j:D} \xrightarrow{\sqsupseteq} \texttt{i:C}$.

Formally, the distributed interpretation $\mathfrak{I}$ of a context space is associated with a set of *domain relations*. A domain relation $r_{ij} \subseteq \Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j}$ states, for each object of $\Delta^{\mathcal{I}_i}$, the object of $\Delta^{\mathcal{I}_j}$ it corresponds to. The notation $r_{ij}(\texttt{C}^{\mathcal{I}_i})$ denotes the interpretation of the class $\texttt{i:C}$ of O_i as considered in the interpretation domain of O_j. Then, the semantics of a bridge rule is given with respect to domain relations: $\mathfrak{I}$ satisfies $\texttt{i:C} \xrightarrow{\sqsubseteq} \texttt{j:D}$ if $r_{ij}(\texttt{C}^{\mathcal{I}_i}) \subseteq \texttt{D}^{\mathcal{I}_j}$ and $\mathfrak{I}$ satisfies $\texttt{i:C} \xrightarrow{\sqsupseteq} \texttt{j:D}$ if $r_{ij}(\texttt{C}^{\mathcal{I}_i}) \supseteq \texttt{D}^{\mathcal{I}_j}$.

Another form of bridge rules is used to specify a correspondence between individuals. $\texttt{i:a} \xrightarrow{\equiv} \texttt{j:b}$ means that the individual $\texttt{i:a}$ in O_i corresponds to the individual $\texttt{j:b}$ in O_j. Formally, $\mathfrak{I}$ satisfies $\texttt{i:a} \xrightarrow{\equiv} \texttt{j:b}$ if $r_{ij}(\texttt{a}^{\mathcal{I}_i}) = \texttt{b}^{\mathcal{I}_j}$.

4.2 Global and Local Reasoning with C-OWL

Local reasoning services in C-OWL are the standard OWL reasoning services, performed in a particular context, without taking into account the bridge rules. A *global reasoning service* uses bridge rules to infer statements in a context using knowledge from the other contexts. [11] presents an extension of the standard tableau algorithm for the computation of the global subsumption test. *Global subsumption* uses the principle of subsumption propagation which, in its simplest form, can be expressed as:

> if the mapping $\mathcal{M}_{ij}$ contains $\texttt{i:A} \xrightarrow{\sqsupseteq} \texttt{j:C}$ and $\texttt{i:B} \xrightarrow{\sqsubseteq} \texttt{j:D}$
> then $\mathfrak{I}$ satisfies $\texttt{i:A} \sqsubseteq \texttt{B}$ implies that $\mathfrak{I}$ satisfies $\texttt{j:C} \sqsubseteq \texttt{D}$.

Intuitively, this means that subsumption in a particular context can be inferred from subsumption in another context thanks to bridge rules.

Similarly, we consider here a *global instance checking* based on an instantiation propagation rule:

> if $\mathcal{M}_{ij}$ contains $\texttt{i:C} \xrightarrow{\sqsubseteq} \texttt{j:D}$ and $\texttt{i:a} \xrightarrow{\equiv} \texttt{j:b}$
> then $\mathfrak{I}$ satisfies $\texttt{i:C(a)}$ implies that $\mathfrak{I}$ satisfies $\texttt{j:D(b)}$.

Instantiation is extended in order to use global instance checking. Based on bridge rules, information known about an individual in a particular context can be completed using inferences made in other contexts.

5 Decentralized Case-Based Reasoning with C-OWL

5.1 CBR and Contextualized Knowledge

Using C-OWL for DzCBR, a context is used to represent a particular viewpoint on the domain. A global target problem is represented as a set $\{\texttt{i:tgt}\}_i$ of local target problems, with a problem $\texttt{i:tgt}$ in each context O_i. In addition, a bridge rule $\texttt{i:tgt} \xrightarrow{\equiv} \texttt{j:tgt}$ is declared for each O_i and O_j of the context space, i.e. $\texttt{i:tgt}$ in O_i is viewed as $\texttt{j:tgt}$ in O_j.

A context O_i includes knowledge and cases that are used to find a local solution $\texttt{i:Sol(tgt)}$ for the local problem $\texttt{i:tgt}$. Thus, a local problem $\texttt{i:pb}$ is solved by a solution $\texttt{i:Sol(pb)}$ inside the context O_i. The adaptation knowledge used for solving a local problem $\texttt{i:tgt}$ is also represented within the context O_i. Local reformulations $\texttt{i:(r,}\mathcal{A}_\texttt{r}\texttt{)}$ are the basic adaptation knowledge units for solving $\texttt{i:tgt}$ in the O_i context.

In a context O_i, there is a class hierarchy where a class represents the index of a source problem to be reused. An index $\texttt{i:idx(srce)}$ is an abstraction of the $\texttt{i:srce}$ problem, retaining the relevant information according to the viewpoint of the O_i context, i.e. $\texttt{i:Sol(srce)}$ can be reused to solve $\texttt{i:tgt}$ whenever $\texttt{i:tgt}$ is an instance of $\texttt{i:idx(srce)}$ (in accordance with the solving schema described in the section 3.4).

Then, in O_i, the instantiation reasoning service is used in a *localized retrieval* process for finding the index $\texttt{i:idx(srce)}$ of the source problem $\texttt{i:srce}$ to be reused. More precisely, the retrieval process consists of finding a similarity path between the target problem $\texttt{i:tgt}$ and the index $\texttt{i:idx(srce)}$ that is composed of relations defined in O_i:

$$\texttt{i:srce} \xrightarrow{isa} \texttt{i:idx(srce)} \xleftarrow{isa} \texttt{i:pb}_1 \ \texttt{i:r}_1 \ldots \texttt{i:r}_q \ \texttt{i:tgt}$$

where the "isa" arrows mean "is an instance of". In addition, a *localized adaptation* process has to build an associated adaptation path using reformulations and adaptation functions defined in O_i for building $\texttt{i:Sol(tgt)}$. Using contextualized knowledge and cases, the CBR process is then "contained" in a context. A detailed example of this localized CBR process is given at the end of the next section.

5.2 Combining Viewpoints Thanks to Bridge Rules

Decentralized artificial intelligence, as defined by [6], is concerned with the activity of autonomous intelligent agents that coexist and may collaborate with other agents, each of them having its own goals and its own knowledge. In the same way, the DzCBR mechanism is:

1. local to a context in the sense that it is carried out in each context, not in a centralized manner,
2. collaborative in the sense that it relies on knowledge sharing between contexts.

In the following, we present an example of a DzCBR process that is distributed among contexts and that takes advantage of this distribution for building a global solution for a target problem.

Let us introduce three contexts named O_1, O_2 and O_3, where a source problem is represented by its index class, and each association between a problem and its solution is represented by a problem-solution axiom. For example, the expression `1:I1` $\equiv$ `Problem` $\sqcap$ $\exists$`p1.C1` defines a source problem in the context O_1, and `1:I1` $\sqsubseteq$ $\exists$`hasSolution.S1` associates an instance of the solution class `1:S1` to an instance of the problem class `1:I1`. In the same way, the source problems `2:I2` and `3:I3` are respectively defined in the contexts O_2 and O_3, together with their problem-solution axioms (1st and 2nd lines of the figure 2). Bridge rules have been declared between the three local target problems `1:tgt`, `2:tgt` and `3:tgt`, making precise the fact that these local problems are three views about a single problem (3rd line of the figure 2). Moreover, bridge rules between classes indicate the subsumption constraints between the contexts (4th line of the figure 2). Finally, a set of assertions is given for the three local target problems (5th, 6th and 7th lines of figure 2).

O_1	O_2	O_3
I1 ≡ Problem ⊓ ∃p1.C1 I1 ⊑ ∃hasSolution.S1	I2 ≡ Problem ⊓ ∃p21.C21 ⊓ ∃p22.C22 I2 ⊑ ∃hasSolution.S21	I3 ≡ Problem ⊓ ∃p3.C3 I3 ⊑ ∃hasSolution.S31
2:tgt $\xrightarrow{\equiv}$ 1:tgt 2:∃p21.C21 $\xrightarrow{\sqsubseteq}$ 1:∃p1.C1	1:tgt $\xrightarrow{\equiv}$ 2:tgt 1:∃hasSolution.S1 $\xrightarrow{\sqsubseteq}$ 2:∃p23.C23	2:tgt $\xrightarrow{\equiv}$ 3:tgt 2:∃hasSolution.S22 $\xrightarrow{\sqsubseteq}$ 3:∃hasSolution.S32
Problem(tgt)	Problem(tgt) C21(a) p21(tgt,a)	Problem(tgt)
Dz2. ∃p1.C1(tgt) Dz3. ∃hasSolution.S1(tgt)	Dz1. ∃p21.C21(tgt) Dz4. ∃p23.C23(tgt) Dz5. ∃hasSolution.S22(tgt)	Dz6. ∃hasSolution.S32(tgt)

Fig. 2. A DzCBR example. 1st and 2nd lines define some source problems. 3rd and 4th lines describe mappings associated with the contexts. 5th to 7th lines describe the target problem. 8th to 11th lines show 6 DzCBR inference steps.

When the DzCBR process is run in each context, the three local target problems `1:tgt`, `2:tgt`, and `3:tgt` are instantiated in their respective contexts.

Dz1. In the O_2 context, `2:tgt` is recognized as an instance of the class `2:`∃`p21.C21`.

Dz2. The bridge rules `2:`∃`p21.C21` $\xrightarrow{\sqsubseteq}$ `1:`∃`p1.C1` and `2:tgt` $\xrightarrow{\equiv}$ `1:tgt` allow the completion of the instance `1:tgt`. `1:tgt` is recognized as an instance of the class `1:`∃`p1.C1`, and thus of the class `1:I1`.

Dz3. Through the problem-solution axiom, a solution `1:S1` is associated with `1:tgt`, that in turn becomes an instance of the class `1:`∃`hasSolution.S1`.

Dz4. The instance completion process is run through the bridge rule `1:`∃`hasSolution.S1` $\xrightarrow{\sqsubseteq}$ `2:`∃`p23.C23`, and the local target problem `2:tgt` is recognized as an instance of the class `2:`∃`p23.C23`.

Dz5. As it is explained below, let us assume that the CBR process in the context O_2 builds a solution that is an instance of `2:S22` and that is associated with `2:tgt`. `2:tgt` becomes an instance of `2:`∃`hasSolution.S22` in O_2.

Dz6. Finally, based on the bridge rule $2\!:\!\exists\mathtt{hasSolution.S22} \xrightarrow{\sqsubseteq} 3\!:\!\exists\mathtt{hasSolution.S32}$, it can be inferred in O_3 that $3\!:\!\mathtt{tgt}$ is an instance of $3\!:\!\exists\mathtt{hasSolution.S32}$.

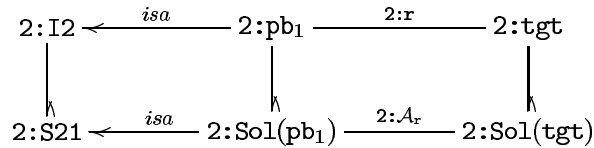The solution of the target problem, represented by the three local target problems $1\!:\!\mathtt{tgt}$, $2\!:\!\mathtt{tgt}$, and $3\!:\!\mathtt{tgt}$, is a set of local solutions, represented as instances of $1\!:\!\mathtt{S1}$, $2\!:\!\mathtt{S22}$, and $3\!:\!\mathtt{S32}$, that have been built in a decentralized way.

Relying on this example, two main operations may be distinguished in the DzCBR process:

(i) *localized CBR* that applies local knowledge for building a solution to the local problem $\mathtt{i}\!:\!\mathtt{tgt}$. The steps Dz3. and Dz5. are examples of such a local operation in DzCBR, respectively carried out in O_1 and O_2.

(ii) *case completion* represents the *collaborative* part of DzCBR. It is based on bridge rules and completes the local target case —either the problem or the solution part— thanks to knowledge sharing with the other contexts. The steps Dz2., Dz4. and Dz6. are examples of this collaboration, using bridge rules for combining viewpoints.

These two operations are run in each context, until no more inferences can be drawn. The solution set $\{\mathtt{i}\!:\!\mathtt{Sol(tgt)}\}_i$ is then delivered.

Details of the localized CBR Process Dz5. The O_2 context contains a reformulation of the form $2\!:\!(\mathtt{r},\mathcal{A_r})$ that is used in the localized CBR operation in this context (see figure 3). During the retrieval step, the $2\!:\!\mathtt{r}$ relation creates an intermediary problem $2\!:\!\mathtt{pb_1}$ from $2\!:\!\mathtt{tgt}$ such that the difference between these two individuals lies in the fact that $2\!:\!\mathtt{pb_1}$ is an instance of $2\!:\!\exists\mathtt{p22.C22}$, whereas $2\!:\!\mathtt{tgt}$ is an instance of $2\!:\!\exists\mathtt{p23.C23}$. Thus, $2\!:\!\mathtt{pb_1}$ is recognized as an instance of $2\!:\!\mathtt{I2}$, and is associated with a solution $2\!:\!\mathtt{Sol(pb_1)}$ from $2\!:\!\mathtt{S21}$, as stated by the problem-solution axiom in O_2. The $2\!:\!\mathcal{A_r}$ adaptation function is used in the adaptation step for creating the solution $2\!:\!\mathtt{Sol(tgt)}$ from $2\!:\!\mathtt{Sol(pb_1)}$. $2\!:\!\mathcal{A_r}$ is such that the difference between $2\!:\!\mathtt{Sol(pb_1)}$ and $2\!:\!\mathtt{Sol(tgt)}$ lies in the fact that $2\!:\!\mathtt{Sol(pb_1)}$ is an instance of $2\!:\!\mathtt{S21}$, whereas $2\!:\!\mathtt{Sol(tgt)}$ is an instance of $2\!:\!\mathtt{S22}$. Therefore, $2\!:\!\mathtt{Sol(tgt)}$, instance of $2\!:\!\mathtt{S22}$, becomes a solution of $2\!:\!\mathtt{tgt}$.

$$
\begin{array}{ccccc}
2\!:\!\mathtt{I2} & \xleftarrow{\;isa\;} & 2\!:\!\mathtt{pb_1} & \xrightarrow{\;2:r\;} & 2\!:\!\mathtt{tgt} \\
\big\downarrow & & \big\downarrow & & \big\downarrow \\
2\!:\!\mathtt{S21} & \xleftarrow{\;isa\;} & 2\!:\!\mathtt{Sol(pb_1)} & \xrightarrow{\;2:\mathcal{A_r}\;} & 2\!:\!\mathtt{Sol(tgt)}
\end{array}
$$

Fig. 3. The similarity path and the adaptation path of the localized CBR process in O_2

6   An Example of Application to Breast Cancer Treatment

The task of finding the right treatment for a patient ill with breast cancer is supported by a protocol. This protocol can be seen as a set of rules $Cond \Rightarrow Ttt$ where $Cond$ is a set of conditions on patients and Ttt is a description of the type of treatments

recommended for the patients satisfying *Cond*. Several specialties are involved in this decision, and the protocol is structured according to these specialties. In breast cancer treatment, the surgery specialty is mainly concerned with partial or total breast abla-tion, the chemotherapy specialty is concerned with the administration of drugs useful to stop or lower the tumor evolution and the radiotherapy specialty treats the patients by irradiation of the potentially infected zones. The global recommendation combines the decisions taken in all the specialties. The protocol rules may be directly applied in 60 to 70 % of the situations (with respect to the characteristics of the patients). In situations not considered by the protocol, the decision is taken by a multi-disciplinary expert committee. This committee adapts the protocol rules to find a solution, taking into account the characteristics of the considered patient.

In our research work, decision support for breast cancer treatment relies on DzCBR, where a problem is a description of the characteristics of a patient, and a solution is a treatment proposition. The case base and the domain model rely on a formalized repre-sentation of the protocol in C-OWL. In the following example, three different contexts, namely O_r, O_s, and O_c, standing for the radiotherapy, surgery and chemotherapy view-points, are considered. These contexts correspond respectively to the O_1, O_2 and O_3 contexts of the example of section 5.2. A protocol rule $Cond \Rightarrow Ttt$ is represented and implemented as a problem-solution axiom of the form PC $\sqsubseteq$ $\exists$hasSolution.T, where PC and T are classes respectively representing the *Cond* and *Ttt* parts of the protocol rule. For example, O_r contains a problem class corresponding to the patients having a tumor that is smaller than 4cm. For the members of this class, a radiotherapy of the internal mammary chain is recommended. Therefore, the problem solution axiom 1:I1 $\sqsubseteq$ $\exists$hasSolution.S1 of the preceding example is restated as:

r:Patient $\sqcap$ $\exists$tumorSize.lessThan4cm $\sqsubseteq$ $\exists$hasSolution.IntMamChainRadio

In the same way, O_s contains the problem-solution axiom:

s:Patient $\sqcap$ $\exists$hasTumor.($\exists$size.moreThan4cm) $\sqcap$ $\exists$radiotherapy.IntMamChain
    $\sqsubseteq$ $\exists$hasSolution.TotalAblation

meaning that, for patients having a tumor greater than 4cm and for whom a radiother-apy of the internal mammary chain may be applied, a total ablation of the breast is recommended. In O_c, the axiom:

c:Patient $\sqcap$ $\exists$lymphNode.infected $\sqsubseteq$ $\exists$hasSolution.PreSurgicalChemo

means that for patients having infected lymph nodes, some cures of chemotherapy should be applied before the surgical treatment in order to prepare the patient for a partial ablation.

The bridge rules of the example of the section 5.2 are now redefined on the classes of O_r, O_s and O_c:

s:$\exists$hasTumor.($\exists$size.lessThan4cm) $\xrightarrow{\sqsubseteq}$ r:$\exists$tumorSize.lessThan4cm

r:$\exists$hasSolution.IntMamChainRadio $\xrightarrow{\sqsubseteq}$ s:$\exists$radiotherapy.IntMamChain

s:$\exists$hasSolution.TotalAblation $\xrightarrow{\sqsubseteq}$ c:$\neg\exists$hasSolution.PreSurgicalChemo

The first one allows the surgery context to share the information about the size of the tumor with the radiotherapy context. Problem-solving in surgery can reuse the solution found in radiotherapy thanks to the second bridge rule. The third bridge rule expresses that, when a total ablation is recommended, a chemotherapy must not be applied before surgery.

Moreover, the O_s context contains some adaptation knowledge in the form of a reformulation $s:(r, \mathcal{A}_r)$. The $s:r$ relation holds between an instance of Patient having a little-sized tumor (less than 4 cm) that covers a large part of the breast (more than 60%) and an instance of Patient having a larger tumor (more than 4cm). In other terms, a patient with a small tumor in a small breast is considered *for surgery* to be similar to a patient having a large tumor. The $s:\mathcal{A}_r$ adaptation function simply consists in a copy of the solution.

The target problem is represented by three local target problems denoted by r:tgt, s:tgt and c:tgt, that are linked by bridge rules. Each of these individuals is an instance of the patient class, i.e. the assertions r:Patient(tgt), s:Patient(tgt) and c:Patient(tgt) are stated in the O_r, O_s and O_c contexts respectively. Moreover, s:tgt is described as a patient having a small tumor in a small breast, i.e. the assertion s:∃hasTumor.(∃size.lessThan4cm ⊓ ∃cover.MoreThan60%)(tgt) is stated in O_s.

The DzCBR process for solving this problem corresponds to the six steps of the section 5.2 example. The information about the tumor size is first shared between surgery and radiotherapy, and so, a radiotherapy of the internal mammary chain is recommended in O_r. In O_s, the reformulation $s:(r, \mathcal{A}_r)$ is applied, considering s:tgt as similar to a patient having a large tumor. According to the problem-solution axiom contained in O_s, the solution for a patient with a large tumor is a total ablation. This solution is copied through $\mathcal{A}_r$ for s:tgt. Finally the solution found in surgery, the total ablation, implies that no chemotherapy has to be applied before surgery. It must be remarked that the target problem is treated differently in O_s and O_r. Indeed, it has been considered as a patient with a small tumor for radiotherapy, whereas it has been considered as a patient with a large tumor in surgery.

7   Discussion and Related Work

A CBR system based on the reformulation model has been implemented in the form of a generic Web service manipulating OWL ontologies. This architecture based on Web services is very helpful in the implementation of localized CBR. For global reasoning in C-OWL, we are using the system described in [11] that is currently under development. A complete protocol for breast cancer treatment has also been formalized in C-OWL. This particular representation was made of 4 contexts, each of them containing between 50 and 100 classes, and about 50 bridge rules have been described between these classes. The lesson learned from this experiment is that building and managing multiple contexts that reflect existing viewpoints in the domain appear to be simpler than finding and maintaining a consensual representation for these viewpoints all together. Moreover, even if bridge rules are generally related to domain expertise and have to be

built manually, this task can sometimes be semi-automated, on the basis of ontology alignment techniques.

Considering related work, description logics have been used for KI-CBR in several systems (see e.g. [7, 12]). These systems consider a single knowledge model, and take into account a single way of interpreting and using cases. DzCBR combines several viewpoints on the problems and solutions, thanks to multiple inter-related contexts. Some systems use several views on cases to retrieve several local best cases. Generally, a single global case is built from these sub-cases. For example, in [13] a choice is made between cases that are retrieved using different case representations, called perspectives. In [14], several agents retrieve local best cases that are assembled in a global best case thanks to negotiation between agents. Since there is no centralized mechanism in DzCBR, a CBR process is carried out in each context and collaborates with the other contexts through bridge rules. In this way, among contexts, several local source cases are retrieved and used independently for adaptation. If one want to apply our approach to *distributed* CBR, i.e. problem-solving by several agents with the same set of goals (by contrast to *decentralized* CBR), it would be necessary to incorporate in the reasoning process a mechanism for managing conflicts.

Our interest for a DzCBR process exploiting semantic Web technologies and principles has started with the design of a semantic portal for oncology [15]. The goal of this portal is to give an intelligent access to standard knowledge for a geographically distributed community of oncologists. There are many other situations, like adaptive query answering, case-based ontology alignment or flexible Web service invocation, where CBR would be useful for the semantic Web. Some studies have been interested in defining markup languages for case representation, on the basis of XML [16] or RDF [17]. But, to our knowledge, there is no work concerned with the design of CBR systems in the semantic Web framework. Our aim here is not to build a general theory concerning the use of CBR in the framework of the semantic Web. However, we hope that the work presented in this paper will provide a guideline for practitioners to apply such techniques.

8   Conclusion

In this paper, a KI-CBR mechanism that exploits decentralized knowledge represented by contextualized ontologies in the C-OWL formalism has been proposed. This framework, called DzCBR, addresses the problem of adaptive reasoning in the multi-ontology environment of the semantic Web. The process of DzCBR takes advantage of the distribution of knowledge into multiple contexts and of the semantic relations between these contexts for solving problems. The motivation for a decentralized KI-CBR system comes from an application in the multi-disciplinary domain of oncology. Particularly it has been applied for the problem of breast cancer treatment recommendation. In this application, different specialties, like surgery, radiotherapy and chemotherapy, correspond to several viewpoints that must be taken into account and combined. A viewpoint is implemented as a C-OWL context, and semantic mappings between contexts are used for collaboration between viewpoints.

References

1. Lenz, M., Bartsch-Spörl, B., Burkhard, H.D., Wess, S., eds.: Case-Based Reasoning Technology: From Foundations to Applications, LNAI 1400. Springer (1998)
2. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. Artificial Intelligence Communications **7** (1994) 39–59
3. Aamodt, A.: Knowledge-Intensive Case-Based Reasoning in CREEK. In Funk, P., Gonzàlez-Calero, P.A., eds.: Proc. of the European Conference on Case-Based Reasoning, ECCBR'04, Springer (2004) 1–15
4. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D., Patel-Schneider, P., Stein, L.A.: OWL Web Ontology Language Reference. W3C Recommendation (2004)
5. Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L., Stuckenschmidt, H.: Contextualizing Ontologies. Journal of Web Semantics **1** (2004) 1–19
6. Demazeau, Y., Müller, J.P.: Decentralized Artificial Intelligence. In Demazeau, Y., Müller, J.P., eds.: Decentralized A.I. – Proc. of the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World, North-Holland (1989) 3–13
7. Gómez-Albarrán, M., Gonzàles-Calero, P., Díaz-Agudo, B., Fernàndez-Conde, C.: Modelling the CBR Life Cycle Using Description Logics. In Althoff, K.D., Bergamnn, R., Branting, L., eds.: Proc. of the International Conference on Case-Based Reasoning, ICCBR'99, Springer (1999) 147–161
8. Lieber, J., Napoli, A.: Correct and Complete Retrieval for Case-Based Problem-Solving. In Prade, H., ed.: Proc. of the European Conference on Artificial Intelligence, ECAI'98, John Wiley & Sons Ltd, Chichester (1998) 68–72
9. Melis, E., Lieber, J., Napoli, A.: Reformulation in Case-Based Reasoning. In Smyth, B., Cunningham, P., eds.: Proc. of the European Workshop on Case-Based Reasoning, EWCBR'98, Springer (1998) 172–183
10. Smyth, B.: Case-Based Design. PhD. thesis, Trinity College, University of Dublin (1996)
11. Serafini, L., Tamilin, A.: Local Tableaux for Reasoning in Distributed Description Logics. In Haarslev, V., Moeller, R., eds.: Proc. of the International Workshop on Description Logics, DL'04. (2004) 100–109
12. Kamp, G., Lange, S., Globig, C.: Related Areas. [1] chapter 13
13. Arcos, J.L., Lopez de Mántaras, R.: Perspectives: a declarative bias mechanism for case retrieval. In Leake, D.B., Plaza, E., eds.: Proc. of the International Conference on Case-Based Reasoning, ICCBR'97, Springer (1997) 279–290
14. Nagendra Prassad, M., Lesser, V., Lander, S.: Retrieval and Reasoning in Distributed Case Bases. Journal of Visual Communication and Image Representation **7** (1996) 74–87
15. d'Aquin, M., Brachais, S., Bouthier, C., Lieber, J., Napoli, A.: Knowledge Editing and Maintenance Tools for a Semantic Portal in Oncology. International Journal of Human-Computer Studies (IJHCS) **62** (2005) 619–638
16. Coyle, L., Doyle, D., Cunningham, P.: Representing Similarity for CBR in XML. In Funk, P., González Calero, P., eds.: Advances in Case-Based Reasoning (Procs. of the Seventh European Conference), LNAI 3155, Springer (2004) 119–127
17. Chen, H., Wu, Z.: CaseML: a RDF-based Case Markup Language for Case-based Reasoning in Semantic Web. In Fuchs, B., Mille, A., eds.: From structured cases to unstructured problem solving episodes for experience-based assistance. Workshop at ICCBR-2003. (2003)