Multi-Representation Ontology for Requirements Specification

Achraf Mtibaa[1], Faïez Gargouri[2]

[1]Higher Institute of Electronics and Communication, Computer Science Department, University of Sfax /MIRACL Laboratory, Tunis Road KM 10 B. P. 242 Sakiet Ezziet Sfax, Tunisia

[2]Higher Institute of Computing and Multimedia, Computer Science Department, University of Sfax /MIRACL Laboratory, Tunis Road KM 10 B. P. 242 Sakiet Ezziet Sfax, Tunisia

[1]achrafmtibaa@gmail.com; [2]faiez.gargouri@isimsf.rnu.tn

***Abstract*- The requirements specification step is considered as a crucial step involved during the requirements analysis in the life cycle of an information system. This step is considered as a contract between future users and designers. It concerns the expected characteristics: functional and non-functional requirements. However, many problems arise in this step such as the difficulty of gathering information, misunderstanding and incomplete requirements, lack of opportunities and constraints of the proposed systems, etc. In addition, requirements risk is unclear. This materializes in particular by conflict profiles, points of view and contexts among different users admitting different techniques to specify their requirements. For this reason, we propose a multi-representation ontology (MRO) for requirements specification (RS) to solve the multi-context and the multi-representation problems. This paper proposes a MRO to enhance the effectiveness of RS. It presents the complementarities between context and ontology. It exposes an approach to establish the MRO providing the formalization and the visualization of this ontology. The proposed ontology is operationally defined in ContextOntoMR prototype.**

***Keywords*- Ontology; Requirements Specification; Context; Multi-Representation**

I. INTRODUCTION

The variation of the environment and the user context of the information system (IS) can cause the change of users' requirements or even their views and attitudes towards a given situation (a decision). The user's reaction and his decision may be influenced by different aspects, namely skills, personal characteristics (profile) and the situation in which it is located (the context of use when decision-making). The profile itself can be influenced by the user's context. This multitude of contexts results different requirements expression, even contradictory. A good design of an IS must take into account certain essential aspects for the project success, such as the requirements specification, taking into account the multitude of contexts and heterogeneity of techniques used to the specification of requirements. The designer, in this case, is facing various problems including inconsistency, semantic ambiguity and the difficulty of requirements modelling.

As a solution, we propose the use of ontologies known by their incontestable contributions to the semantic level. They are typically used to address the semantic problems. For the problems of multi-context, we proceed to a requirements multi-contextual modelling. To address these problems and assist future users to express their requirements, we combine the ontology with the multitude of contexts.

In this paper, we propose an approach to establish ontology for the multi-context requirements. Our goal is to assist the users to specify their requirements in different contexts specified with several techniques. We aim to solve some of these problems. For that, we plan to cover the requirements specification step engaging users to express their requirements and analysts to specify these requirements.

The remaining of this paper is structured as follows: Section 2 presents RS problems. Section 3 shows the contextual aspect in ontology. In Section 4, we propose a MRO for RS to solve the multi-context and multi-representation problems. Section 5 shows a comparative study of related research works. In Section 6, we present an approach to establish the MRO. We suggest, in Section 7, formalization and visualization of MRO. Section 8 shows *ContextOntoMR* prototype. In Section 9, we enumerate the MRO contributions via *ContextOntoMR*. Section 10 concludes the paper.

II. REQUIREMENTS SPECIFICATION PROBLEMS

The RS is a very important step to ensure a consistent and durable system. However, it can cause some problems against the user. Among them, we mention:

- users do not know what they want;
- users do not want to write their requirements;
- users insist on new requirements after the cost and calendar have been set;
- communication with users is time-consuming;
- users lack technical competence;
- users do not understand the development process.

These problems can make some conflicts like:

- Omission: one or more requirements that should be specified are omitted;
- Inconsistency: the requirements specification is in disagreement with the expressed requirement;
- Ambiguity: the requirements specification is not clear, and could cause a misinterpretation or misunderstanding of requirements meaning;
- High cost and time-consuming spent on this step;
- Requirements collection misunderstands;

DOI: 10.5963/IJCSAI0203002

- Ignorance of opportunities and constraints of the proposed systems;
- Requirements volatility;
- Requirements evolution in time;
- Multi-representation of requirements;
- Multi-context of specified requirements;
- Etc.

These conflicts can significantly complicate the realization of a collaborative system. The problems of multiple representations and contexts have motivated the search to resolve these problems in order to assist potential users to specify their requirements.

There are few solutions proposed in the literature, to overcome the problems of multi-representation and the multitude of contexts. This type of problem appeared, initially, in databases domain. We briefly present some solutions to overcome them.

A. Solution Based on Views

The notion of view is initially adapted to solve the problem of multiple context and/or views in databases. A user view is designed for describing external schemas that give the definition of sub-schemas of original custom schema [1]. User requirements are accorded to different contexts and the notion of view can solve this multitude of contexts and/or viewpoints. The views allow the data representation from different viewpoints and based on specific objectives for different applications. However, the update operations are not applicable on all views.

B. Solution Based on Roles

The role concept has been used to model the different facets or aspects of an entity [2]. The classic example of an entity with multiple roles is an entity that passes from one state to another (student, employee, director, etc.). Each role corresponds to a facet of the real entity. The entity is represented by a common structure, which is increased by information about its different facets [3]. The role notion offers a solution for supporting the multi-representation. Indeed, it allows each actual entity to be represented by a set of objects or instances belonging to different classes that correspond to the roles of the entity. Nevertheless, this notion is rarely used because, until now, there is no standardization of this concept.

C. Solution Based on Ontologies

The solution-based ontologies occurred in particular to ensure the semantic interoperability and cooperation between different representations of the system. Several ontologies have been defined to specify a heterogeneous system admitting multiple perceptions [4]. These ontologies define a concept by a set of properties, operations and structural and semantic links with other concepts in specified field.

Few solutions are proposed to resolve RS problems but the solution-based ontologies are solicited. In our work, we focus on ontologies to overcome some problems emerged during the RS.

In fact, ontologies are well known by their contributions in knowledge representation. However, ontology should provide definitions and contextual data structures to represent the diversity of user’s perceptions and reflections. This aspect is not treated in most current research on ontologies. For that, we offer in the next section, a study of the contextual aspect in ontologies.

III. CONTEXTUAL ASPECT IN ONTOLOGY

Contexts and ontologies have strengths and weaknesses in conceptualizing a domain. On the one hand, ontology is used in some areas, such as a referential for user community. These shared ontologies define a common understanding of the field. These ontologies (as a referential) neglect the particularity of users. On the other hand, the contexts are built to be held locally and represent interpretations of unshared schemas of individuals or groups of individuals. These local contexts neglect the collaborative work of users. We can find a complementary relationship between ontologies and contexts. Ontologies which take into account the contexts allow users to represent their views and/or their contexts and provide a common and a shared view for all users. We take advantage of these two concepts by combining them into a single framework. Before presenting the complementarily between context and ontology we present some conflicts related to context and an overview of some context-aware platforms.

A. Conflicts of Context

Conflicts of context occur when concepts seem to have the same meaning but different in reality. This is due to the different contexts of definition or of evaluation. The context is a very important concept in collaborative and distributed information systems. Indeed, the same real-world object can be represented in the data sources by multiple representations in a local context to each source. These conflicts are in the context where the concepts seem to have the same meaning, but operating in different contexts. These conflicts emerge by using different names for the same concept or property (synonym) or identical names for different concepts or properties (homonymous).

Other conflicts related to values measures occur when different reference systems are used to evaluate the same value (when different units of measurement are used by different data sources). These conflicts are related to the concept value in real-world systems. They are in the case of using different units for measuring the value of properties.

The presented conflicts and the relationship found between ontology and context have led researchers to propose some modelling languages coupling context with ontologies. In the literature, there are languages for representing context coupled with the ontologyie such as CoOL [5], C-OWL [6] and CML [7]. There are a few context-aware platforms from which we quote Context Broker Architecture (CoBrA) [8] [9], Context Management Framework (CMF) [10] and Context Toolkit [11].

DOI: 10.5963/IJCSAI0203002

B. Comparison of Some Context-Aware Platforms

Table I exposes a comparison of context-aware platforms based on seven criteria including architecture, capture method, context model, context processing, resource discovery, historical data storage, security and finally confidentiality. Common property for all solutions analyzed is the separation between infrastructure of capture and the rest of the system, increasing the reusability of context sources in the system. However, each framework has its own format for the context representation and uses different principles of communication. These formats in turn make communication difficult between the frameworks and neutralize directors for, re-use services based on another software framework.

In addition, almost all systems have well-developed component for resource discovery. They support the storage of historical data that helps them later in the context treatment. Security and confidentiality are presented in several systems, but always in the form of basic security mechanisms that should be strengthened further.

Management of historical context is an important criterion in the context-aware systems. Indeed, the history allows the implementation of learning algorithms to provide highly adaptable services to the context. Furthermore, with this kind of algorithms, proactive actions can be automatically triggered for a number of services to the user without an explicit request form.

Another important aspect in these systems concerns the security management and data privacy. Indeed, concepts must be specified to define who owns the contextual information. *CoBrA* uses *Rei* language to define security policies in terms of rights of access permissions to the context. On Context Toolkit, it implements the concept of belonging of context to a user or entity. Thus, contextual information is only accessible to the user or entity to which it belongs.

TABLE I COMPARISON OF CONTEXT-AWARE PLATFORMS

Model / Characteristics	**Context Toolkit**	**CoBrA**	**Context Management Framework**
Type of architecture	Based on widgets	Agent-based, context Centralized broker	Focused on a context manager
Method of Capture	Context widget s	Acquisition module context	Resource server
Context model	Attribute-value pairs	Ontologies (OWL)	Ontologies (RDF)
Processing context	Transformation and aggregation of context	Inference engine and a knowledge base	Interpretation Service (context recognition service)
Resource discovery	Component discover	Not available	Resource servers and subscription mechanism
Storage of historical data	Available in a server	Available	Not supported
Security and Privacy	Membership of the context	Policies with language Rei	Not available

Nevertheless, these platforms have a context sensitive limit on assistance to the user through ontology.

C. Complementarities between Ontologies and Context

According to authors of [6], ontology is built to be shared while a context is built to be maintained locally. To take advantage of both notions, they propose to combine them into a single framework. Thus, they propose the notion of contextual ontology as ontology with a local interpretation. This means that its content is not shared with other ontologies. The strengths of ontologies are the weaknesses of contexts and vice versa. Since, several approaches have been proposed to combine the two concepts to achieve the information semantic interoperability.

However, ontologies represent isolated pieces of knowledge. Putting them on a network, we can explore their interrelationships. These ontologies are known as contextual ontologies. In this case, each one represents a context and its components (concepts and relationships). Thus, for a given ontology, its components can be interpreted in different contexts by choosing the appropriate ontologies representing appropriate contexts.

In the next section, we benefit from the complementarities between context and ontology to overcome the multi-context and multi-representation problems emerged in RS.

IV. MRO TO OVERCOME THE MULTITUDE OF CONTEXTS AND REPRESENTATIONS IN RS

Ontologies are generally used to remedy the semantic problems. For multi-context problem, we proceed to a multi-contextualization of requirements using ontology. This will take into account the contexts variation and multi-representation of requirements. These latter can be specified informally (eg. text), semi-formally (eg. Use Case Diagram of UML) or formally (eg. Z language, EB3).

In most cases, when many users try to agree on a common ontology, they are already placed in different contexts. In particular, in an open environment like the Internet, it is very difficult to get an approval on a common representation of shared domain knowledge. This is mainly due to the different contexts in which participants are placed.

Ontology presents the key concepts, attributes and instances related to a given domain. For this, we specify the role of domain ontologies. The actors in the same domain must first adhere to common domain ontology. The advantage of this solution is to limit the role of domain ontology in a minimal description of common concepts by facilitating the adherence of service providers. This domain ontology does not take into account the different requirements that can be expressed in different contexts. The solution to this problem is to take into account the associated contextual ontologies containing their local semantics.

The importance of understanding the context in computer science is widely recognized. Ontology aims to

DOI: 10.5963/IJCSAI0203002

describe the context of the domain ontology concept. For a particular domain, there can be several contextual ontologies where each one is described in a particular context. These ontologies are called mono-representation ontology. Thus, for each context we associate a contextual ontology. For a quite complex domain, we can have a large number of contexts. In this case, to have reference ontology for this type of domain, we must integrate the different contextual ontologies. This integration is difficult to accomplish. For this reason, we orient our research to ontology admitting the multitude of contexts and representations.

We aim an ontology linked to several contexts simultaneously. Such ontology is called a multi-representation ontology [4]. It's an ontology that characterizes an ontological concept by a variable set of properties or attributes in several contexts. Indeed, using a single Ontology (Multi-Representation Ontology or MRO) is a request caused by the difficulty of contextual ontologies integration. With this ontology, we can represent the multi-context information in the same ontology and we have the benefit of explicit contexts (through context-dependent properties).

A. Reasons to Use MRO for RS

We performed a case study in order to assess the general issues described below. In the following we choose the e-learning domain to illustrate our contribution. The main actors of e-learning are: Student, Administrator and Tutor. Each actor can pass in the corresponding space and can access to a privileges list accorded by administrator. Depending on location and user socio-cultural environment, the context of actor can change. In fact, the academic and their organizations can vary from one academy to another and from one country to another. We take the example of three different academic locations education that wants to establish an e-learning platform in their teaching methods: *Rectorship of Sfax in Tunisia* (*RST*), *Academic Rectorship of Toulouse in France* (*RATF*) and *Training Enterprise* (*Ese*). About techniques for RS, we can use two main representations: informal (*Textual RS*) and semi-formal (*Use CaseDdiagram* (*UCD*)).

In our study carried out in [12], we identify three contexts (*RST*, *RATF* and *Ese*), three actors (*Student*, *Administrator* and *Tutor*) and two representations (*Textual RS* and *UCD*) which raise the number of RS to eighteen. We found the problem of multitude of contexts and representations. In fact, the same requirement can be expressed differently which causes a redundancy problem. In addition, inclusions or equivalences between the requirements involve their incoherence. Consequently, designers and system developers are faced to many problems such as: (i) contexts divergence, (ii) heterogeneity of RS representations and, (iii) identification of similarities between requirements.

Several searches postulate semantic is context dependent. Indeed, the concept interpretation depends especially on context in which concepts are used. Therefore, ontology should provide definitions and structures of contextual data representing the diversity of perceptions and reflections. The context used as a user view to choose an ontology subset.

To resolve problems of multi-context and multi-representation in RS step, we propose the use of MRO for the following reasons:

- difficulties in integrating contextual ontologies;

- lack of knowledge, sometimes, for users of ontologies and their contents;

- keep non-contextual definitions of concepts valid for all contexts (to model the contextual by non-contextual in multi-representation ontologies);

- integrating requirements specified in different contexts and representations;

- semantic flexibility: ability to add other definitions of a concept according to another context in the same ontology.

To understand the proposed ontology, we present the MRO characteristics and components.

B. MRO Definition and Model

Our contribution aims to assist users during the RS step, which is considered a crucial step in the system life cycle. To achieve this goal, we propose a multi-representation ontology (OMR) for multi-context and multi-representation requirements. The construction of such ontology requires the use of a process to insert the new requirements in the ontology. The proposed MRO, is involved in solving some problems related to semantic interpretation.

In this vision, we propose a model of MRO that consists of two layers: a core layer represented by domain ontology and a layer to support the multitude of contexts. We assume that the different contexts in which a requirement is established are partial but complementary. In fact, several interpretations are possible for the same requirement. Each interpretation is on a given context. Indeed, every requirement is seen differently according to the context in which it is specified. We consider a context as "*a set of parameters that gives special meaning to a requirement concept to be interpreted appropriately in a collaborative system*" [12]. Fig. 1 shows the MRO with the combination of these two layers (Core ontology + contextual layer).

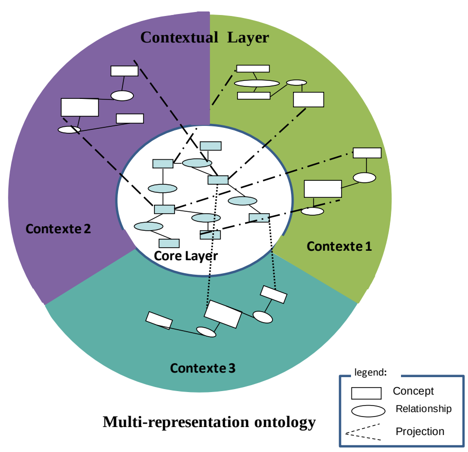

Fig. 1 MRO model

DOI: 10.5963/IJCSAI0203002

In Fig. 1, the core ontology is composed of concepts and relationships between these concepts. Each concept admits a set of properties, common to all contexts. The core ontology concepts will be projected according to interpretations of the upper layer. This helps us to say that the projected concepts in the contextual layer admit particular properties for each context. In its turn, each context admits its own contextual parameters.

C. MRO Components

Our ontology is seen as a set of concepts and relationships intended to represent world objects in an understandable form. The proposed ontology is aligned with this definition.

MRO is composed of concepts (C) and relationships (R):

- Concepts are of two types: requirement and context (C → (R, Cxt)). Requirement concept R is defined by its name and its properties (R → (NR, P)). These are subdivided into two types: the common properties (CP) existing in the core layer and the particular properties (PP) for a given context (P → (PC, PP)). The particular properties are considered as specificity of requirement from global vision (core) to contextual vision.

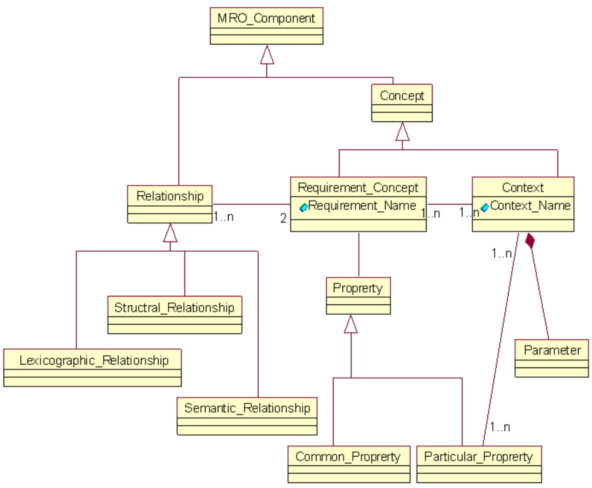

Fig. 2 Class diagram describing the MRO components

This vision is the result of requirement projection from the core layer on the contextual layer. This projection is modelled, in Fig. 2 (using UML language [13]), by the link between *Particular_Property* class and class *Context*. The latter is defined by its name and contextual parameters (Cxt → (NC, Pr)).

- Relationships that can exist between two requirements concepts are classified into three types: structural relationships, semantic relationships and lexicographic relationships. Structural relationships are dependent on the used representation. In effect, for a semi-formal model like an extended use cases diagram, we can have structural relationships that are “is-a”, “include” and “extend”. Semantic relationships can be “equivalence”, “part-of”, “identity” and “disjunction”. Lexicographic relationships included in our ontology are “synonymy” and “homonymy”. Table II presents the MRO components definitions.

TABLE II MRO COMPONENTS

Notation	Description
O → (C, R)	MRO (O) is composed of concepts (C) and relationships (R)
C → (R,Cxt)	Concepts (C) are requirement and context (Cxt)
R → (NR, P)	Requirement concept R is defined by its name (NR) and its properties (P)
P → (CP, PP)	Properties (P) are composed of common properties (CP) and particular properties (PP)
Cxt → (NC, Pr)	Context (Cxt) is defined by its name (NC) and contextual parameters (Pr)
Pr → (localization ‖ time ‖ activity ‖ user ‖ physical environment)	Contextual parameters are localization, time, activity, user, physical environment
R → Rst ‖ Rse ‖ Rlx	Relationship can be structural relationships (Rst) or semantic relationships (Rse) or lexicographic relationships (Rlx)
Rst → is-a ‖ include ‖ extend	Structural relationships (Rst) can be “is-a” or “included” or “extend”
Rse → equivalence ‖ part-of ‖ identity ‖ disjunction	Semantic relationships (Rse) can be “equivalence” or “part-of” or “identity” or “disjunction”.
Rlx → synonymy ‖ homonymy	Lexicographic relationships (Rlx) can be “synonymy” or “homonymy”

D. Contextual Aspect of Requirement Concept

The requirement contextual aspect is taken into account in the MRO through the contextual parameters to detect the context in which the requirement is specified. These contextual parameters depend on the studied domain and should be specified with care.

When defining contextual parameters, it occurs an ambiguity between what is a context parameter and what is a type or category of context. We consider that a parameter is information, which helps to understand the current context and to exploit it.

Contextual parameters and their numbers differ depending on study domain. For our chosen domain (e-learning), we use five parameters which are User, Activity, Location, Physical Environment and Time [14].

- User: represents the user’s profile, his physical and mental properties (name, function, etc.);

- Activity: represents the activities, tasks and user’s goals;

- Location: user’s geographic location;

- Physical environment: devices, network and various kinds of materials used by the user;

- Time: action historized, date and system time.

After describing the proposed MRO, we present in the next section a comparative study in order to position our contribution against other research works.

V. COMPARATIVE STUDY

We present a comparative study of four research works dealing with multi-representation ontology and context sensitive (see Table III). We have studied similar works to our contribution, namely, working with multiple ontologies,

DOI: 10.5963/IJCSAI0203002

MRO in GIS (*Geographic Information System*), MRO as a foundation of enterprise information system and MRO for RS (our contribution). We have based our comparative study on seven criteria including: supporting heterogeneity, ontological concept, ontologies integration, description language, context, ontology aim and system functionalities.

TABLE III COMPARATIVE STUDY

	Working with Multiple Ontologies [15]	**MRO in GIS** [4]	**MRO As A Foundation of Enterprise Information System** [16]	**MRO for RS** [12]
Support the Heterogeneity	Yes	Yes	Yes	Yes
Ontological Concept	Knowledge concept (for specific field)	GIS concept	Enterprise system concept	RS concept
Ontologies Integration	Difficult	Not need	Mapping technique	Not need (one ontology for multiple contexts)
Description Language	C-OWL, Context toolkit, CoOl, etc.	DL extended with stamping technique	Combing DL and Modal Logics with stamping technique and/or indexing	DL extended with projection + Global and local vision
Context	Ontology view point (or not exist)	Contextual projection	Statics and Dynamics properties	Contextual parameters
Ontology Aim	Mediation	Spatial database mediation	Mapping between concepts	User assistance
System Functionalities	Not treated	Not treated	Enterprise information	Actions and operation describing system requirements

We start with the heterogeneity which is included in the four research works. For the ontological concept, multiple ontologies use knowledge concept for specific field. To MRO in GIS, GIS concept presents spatial knowledge. For MRO as a foundation of enterprise information system, the ontological concept is enterprise system concept. In MRO for RS, the ontological concept is a requirement concept. The ontologies integration remains difficult for multiple ontologies despite the different techniques proposed. For MRO in GIS and MRO for RS, the integration is implicit because we have a single ontology for multiple contexts. For MRO as a foundation of enterprise information system the integration is achieved through the mapping technique.

For description language of multiple ontologies, we have several languages such as C-OWL, CoOL, Context Toolkit, etc. MRO in GIS uses the DL extended with stamping technique. Enterprise MRO is described with MDL (Description Logic + Modal Logic). To our MRO for RS, we extend the DL with projection and the global and local vision (core and contextual layers). The taking into account the context in the four works is done for multiple ontologies with ontology view point (not exist for some cases when context is nonexistent). For MRO in GIS, the notion of context is achieved through the contextual projection. For enterprise ontologies, they use static and dynamic properties to describe the contexts specificities. For our ontology, we have proposed contextual parameters (numbering five) describing physical context.

Mediation is the purpose of multiple ontologies and MRO in GIS. Mapping between concepts is the general goal for enterprise MRO. In our ontology, the main objective is the user assistance. Finally, the system functionalities are not covered in multiple ontologies and MRO in GIS. For MRO enterprise it can be present with enterprise information. For our work, we present system functionalities through the actions and operations describing system requirements that reflecting the use cases and system requirements (system functionalities).

We present in the next section the approach to establish our ontology.

VI. APPROACH TO ESTABLISH A MRO FOR RS

Our main objective is to assist the user to express their requirements. This assistance can overcome some problems of ambiguity, inconsistency and avoid omission of requirement information. For this purpose, we propose two models for RS acquisition and an approach to implement a MRO for multi- context and multi-representation RS.

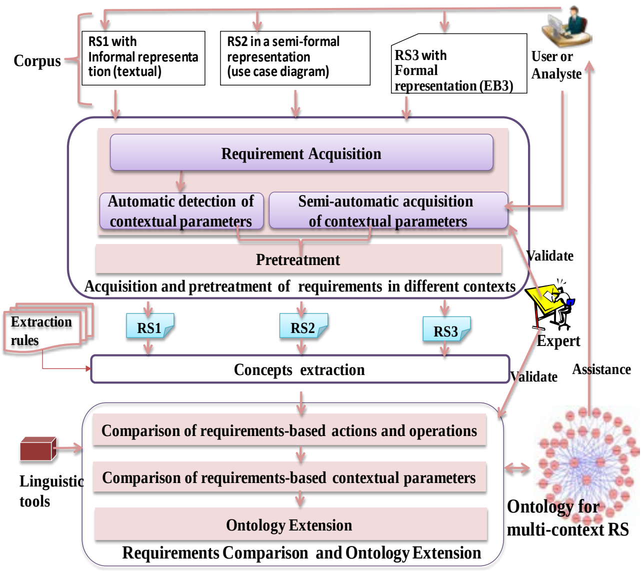

Fig. 3 Approach to establish MRO for RS

This approach is based on an ontology which acts firstly as referential and secondly as models for acquiring maximum information from the user/analyst through easy and understandable interactions. The interdependence of these models with ontology overcomes the multitude of representations and contexts.

DOI: 10.5963/IJCSAI0203002

We have five steps in our approach (cf., Fig. 3). We start by acquiring user requirements of a given domain in different representations (informal, semi-formal or formal). The second step defines the pretreatment of RS to convert them into a unified format. The concepts extraction represents the third step. These concepts will be invested in the fourth step of comparing knowledge that exists in the ontology (supposed initial and constructed from pertinent concepts for a given domain) and new concepts extracted from the new RS. This comparison is based on two criteria, namely, the requirement properties and contextual parameters. Finally, the fifth step is an extension of the ontology by new requirements specified by users.

In the rest of this section, we discuss the steps of this approach to develop an MRO for RS.

A. Acquisition and Pretreatment Steps

The requirements acquisition is a difficult exercise, which must be well prepared. The analyst must have an open mind and listen to the user by preparing a set of specific questions without prejudices. The requirements acquisition and pretreatment involve the user and analyst. The system user expresses his requirements without necessarily knowing all techniques of requirements specification. The analyst specifies the requirements expressed arbitrarily by the user. The domain expert admits a global vision of the system and has more knowledge to model.

We consider that a requirement is characterized by its name, its properties (actions) and its contextual parameters. However, the arbitrary choice of RS techniques may cause some problems for our approach to establish a MRO for RS. Indeed, the techniques are multiple and the users are confronted with forgetfulness and the incompleteness of ideas representing its requirements. Therefore, we propose to develop models for acquiring the specified requirements accorded to various contexts and to define contextual parameters. From these parameters, we can deduce, at first, the context in which the requirement is specified. We pass, in a second step, to the acquisition before beginning the pretreatment step of acquired requirements to translate them into pivot specification.

1) Proposal Models for Requirements Acquisition:

Most analysts and designers use two RS techniques: textual specification and the Use Case Diagram (UCD) of UML. Both techniques can be substituted with two similar models on which to base acquiring of requirements, to simplify it and to further assist users to specify their requirements. For the first type of RS, i.e., the textual RS, we opted for a form in which the user refers to their requirements by answering some questions. This method can guide user to overcome the ambiguity in text format, to make explicit the implicit requirements and facilitate the extraction of concepts. In the textual model, we have underlined, also, these properties through a set of operations to perform from user for each specified requirement.

For the second model, we propose to extend the UCD with certain features because this type of diagram, in its classic version, does not take into account the actions to be taken for each requirement. These actions are prescribed in the scenarios of use cases that match the documentation of UCD. They are therefore not included in the UCD. As a result, we have prescribed for each requirement as specified in a use case, the actions corresponding to the required properties. Through these properties, we can know precisely the user requirement objective.

For example, we present in Fig. 4 and Fig. 5, two RS under both models. The user in this example is the Tutor of e-learning domain. This user specifies his first requirement "participate in the forum" according to an extended UCD and the second specifies his requirement "Chat" according to the textual model. The actions of the requirement "Participate in the forum" are graphically reported with use cases. The actions of the requirement "Chat" are given as operations (OP i).

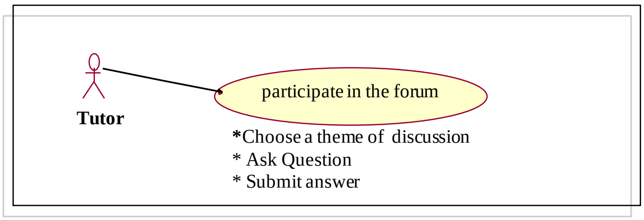

Fig. 4 RS modelled with extended UCD

Textual requirement specification
Localization: Toulouse university
Activity: communication
Time: 2008-2009
User: Tutor
Requirement: chat
Op1: Ask Question
Op2: Submit answer

Fig. 5 RS modelled with textual representation

The requirement properties can be extracted from the proposed models, while the contextual parameters must be explained given explicitly.

2) Requirements Acquisition:

The requirement acquisition must be complied with defined RS models. It will provide the basic elements for RS. Requirements are specified through interaction with forms typically adapted by users. Each requirement is acquired with its properties and contextual parameters.

The requirements properties are extracted from proposed models. They are defined as operations in the textual model and actions in the extended UCD. For the contextual parameters, some of them are acquired through interaction with proposed models, for others are automatically detected from the physical environment (used device, IP address, etc.).

3) Requirement Pretreatment:

The requirement pretreatment step allows transferring information expressed in different representations in a common representation in order to facilitate the extraction step. For that, we choose to translate to XML. In fact, it's a standard for describing the syntax and semantic of data, and documents exchange between applications. The passage to a pivot model admits a double objective. First, it allows the user to visualize its requirements for any corrections.

DOI: 10.5963/IJCSAI0203002

Second, it overcomes the problem of heterogeneity representation requirements models.

For the textual model, the requirements transformation described under this model is in accordance with the DTD (see Fig. 6) which defines the structure of the pivot model. The extended UCD is realized with a CASE tool. This allows, after the acquisition step, to generate a document (file) in format “xmi”. This document contains information describing the requirements and other technical information. In fact, the transformation of this model to pivot model involves a pretreatment step for the purification to keep the relevant information and eliminate technical information. Relevant information corresponds to concepts related to the proposed composition of the MRO.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--DTD generated by XMLSpy v2011 rel. 2
(http://www.altova.com)-->
<!ELEMENT type (#PCDATA)>
<!ELEMENT time ((name))>
<!ELEMENT tb ((name, act-op))>
<!ATTLIST tb numero CDATA >
<!ELEMENT representation ((type))>
<!ELEMENT parameter ((localization, time, activity, IP_adress,
user))>
<!ELEMENT name (#PCDATA)>
<!ELEMENT localization ((nom))>
<!ELEMENT context ((parameter))>
<!ELEMENT IP_adress ((name))>
<!ELEMENT activity ((name))>
<!ELEMENT user ((name))>
<!ATTLIST user number CDATA >
<!ELEMENT act-op (#PCDATA)>
<!ELEMENT RS ((representation, context, tb))>
<!ATTLIST RS  RequirementName CDATA>
```

Fig. 6 DTD of pivot model

As result of the pretreatment step, we have the RS converted in pivot specification in order to solve their heterogeneity. In these RS expressed in a pivot specification, we find all information acquired from various RS. For example, Fig. 7 and Fig. 8 show two RSs in the pivot model, generated from two requirements contained in Fig. 4 and Fig. 5, after the two-step of acquisition and pretreatment. These RSs expressed in this pivot model are the input of the next step.

```
<?xml version="1.0" encoding="UTF-8" ?>
<tb>
< RequirementName > participate in the forum </
RequirementName >
<representation> UCD </representation >
<act-op> Choose a theme of discussion </ act-op>
< act-op > Ask Question </ act-op >
< act-op > Submit answer </ act-op >
<localization> Sfax university </localization>
<user> Tutor </ user >
<activity> Communication </activity>
<time> 2008-2009 </time>
<IP_adress> 192.68.2.2 </IP_adress>
</paramter>
</tb>
```

Fig. 7 First RS under pivot model

```
<?xml version="1.0" encoding="UTF-8" ?>
< tb>
< RequirementName >chat </ RequirementName >
 <representation > Textual </representation >
 < act-op > Ask Question </ act-op >
< act-op > Submit answer </ act-op >
 <parameter>
<localization> Toulouse university </localization>
<user> Tutor </user>
<activity> Communication </activity>
<Time> 2008-2009 </Time>
< IP_adress > 192.68.0.6 </ IP_adress >
</paramater>
</tb>
```

Fig. 8 Second RS under pivot model

B. *Concepts Extraction*

In this step, we take advantage of all information acquired during the first two steps of our approach. Indeed, the input of this step is a requirement described in accordance to pivot model.

TABLE IV REQUIREMENT EXTRACTION IN TABULAR FORMAT

	Requiremen t Name	**Properties**	**Contextual Parameters**
Tb1	participate in the forum	- Choose a theme of discussion - Ask Question - Submit answer	User: Tutor Activity: Communication Localization: Sfax university Time: 2008-2009 IP_adress: 192.68.2.2
Tb2	chat	-Ask Question - Submit answer	User: Tutor Activity: Communication Localization: Toulouse university Time: 2008-2009 IP_adress: 192.68.0.6

From this model, we identify the requirement concepts namely the requirement name, its properties and its contextual parameters. In this step, we determine the set of concepts that can extend the ontology.

For example, Table IV presents the two requirements acquired and processed in two steps by extracting all relevant information.

C. *Comparison and Extension Steps*

The comparison step compares the information extracted from the new RS with those in ontology. This step allows on the one hand determining the relationship between the new requirements with those existing in the ontology. On the other hand, it allows extending the ontology with new concepts. The step of upgrading is realized in the ontology after the comparison. The latter is subdivided into two phases. The first is based on properties (operations and actions of requirements). The second is based on contextual information [14]. This information includes contextual parameters and particular properties or local context. We admit that the requirement properties are classified into two types. Those are global or common to all contexts and others are local or particular to a given context.

DOI: 10.5963/IJCSAI0203002

The requirements upgrade step is based on the comparison of concepts extracted from a new RS with the concepts existing in the MRO. However, this ontology is built in the OWL ontology language, while the new requirement is identified in pivot model (XML). Therefore, a correspondence of this requirement in OWL is intended to compare two homogeneous structures and minimize the complexity of the comparison.

To ensure the extension of ontology, three phases are involved [14]. We begin by transforming the concepts of the pivot model in OWL, followed by to phase-based comparison of properties to finally arrive at the comparison-based contextual parameters.

To establish a set of rules designed to simplifying and rationalizing the MRO, we propose formalization and some visualizations of this ontology.

VII. FORMALIZATION AND VISUALIZATION OF MRO

The ontology formalization aims to respect the standards of ontology languages (eg OWL) and being treated in an ontology editor. Recall that description logics are logical formalisms of knowledge representation. A formal ontology uses description logic to describe the concepts of a domain through concepts and atomic roles [17]. Although the atomic concepts correspond to unary predicates specifying the domain objects, atomic roles correspond to binary predicates and describing the relationships between objects. These roles are specified using constructors provided by the formal language of description logic.

We choose the description logic (DL) as a formalization language for the following reasons.

DLs are commonly used by standards such as OWL. In addition, they have inference algorithms whose complexity is often lower than the complexities of demonstrators' evidence of first order logic [18].

The DL uses an ontological approach. This approach requires, to describe individuals (ABox), the definition of general categories of individuals (TBox) and the logical relationships that individuals or categories can maintain between them. This approach is natural for the ontological argument because even if the majority of interaction takes place at the individual level, much of the reasoning occurs in categories [19].

The DL and their inference engines provide a logical foundation and efficient algorithms for reasoning, while enjoying the benefits of an ontological approach.

To formalize our ontology, we rely on the DL extended by the projection adding contextual specificities of our contribution.

A. DL extended by the Contextual Projection

It should be noted that the DL has passed through several steps and each time, there are new constructors that are added. Some works such as [4], are based on *ALCN* (*ALCNR* without the roles conjunction) to propose an extension to the DL by giving a new constructor to consider the notion of context. They added to the formalization of a not contextual concept a constructor called projection (in a context). We present the Syntax (1) and Semantics (2) of this new DL (knowing that C is a concept).

(1) $C \rightarrow (C)\,[S]$ (contextual restriction)

$S \rightarrow$ List of contexts names

(2) $\perp^{Ij} = \varnothing$

$I^{j} = \Delta^{I}$

$(C \cap D)^{Ij} = C^{Ij} \cap D^{Ij}$

$(C \cup D)^{Ij} = C^{Ij} \cup D^{Ij}$

$(\exists R.C)^{Ij} = \{x \in \Delta^{I} \mid \exists y : (x, y) \in R^{Ij} \cap y \in C^{Ij}\}$

$(\forall R.C)^{Ij} = \{x \in \Delta^{I} \mid \forall y : (x, y) \in R^{Ij} \rightarrow y \in C^{Ij}\}$

$(\leq nR)^{Ij} = \{x \in \Delta^{I} \mid \ \| \{y \mid (x, y) \in R^{Ij}\} \| \leq n\}$

$(\geq nR)^{Ij} = \{x \in \Delta^{I} \mid \ \| \{y \mid (x, y) \in R^{Ij}\} \| \geq n\}$

$((C)[S])^{Ij} = C^{Ij}$ si $j \in S$ else $= \varnothing$

We adopt the *ALCN DL* extended by the contextual projection. Indeed, it aligns with our proposal for a multi-contextual ontology to model the specificities of multi-context RS. This ontology contains contextual projections of requirements concepts. The definition of non-contextual concepts is always possible. Such concepts exist in all contexts (if concept is not projected (defined) in a particular context then it is defined by all contexts).

B. MRO Formalized by the Extended DL

We propose the formalization of the requirement concept designated by RC in (3).

(3) RC = C1 ∩ C2 ∩ C3 ∩ C4

C1 = (≤1 name_RC) ∩ (≥1 name_RC) ∩ (∀ name_RC.string)

C2 = (≤1 representation) ∩ (≥n representation) ∩ (∀ representation.string)

C3 = (≤1 common_property) ∩ (≥n common_property) ∩ (∀ common_property.string)

C4 = (≤0 particular_property)[Si] ∩ (≥n particular_property)[Si]∩(∀particular_property.string)

This concept is defined by a combination of four elements C1, C2, C3 and C4. C1 is the name of RC which is unique but defined in the ontology in several interpretations depending on the context in which it is specified. The unicity of the name translates with DL as follows: (1 ≤ name_RC) ∩ (≥ 1 name_RC). The name of a RC is a string (∀ name_RC. string). C2 refers to the representation with which the RC is defined. A RC can be specified in one or more representations. Each one can concern one context. This is formalized by (≤ 1 representation) ∩ (n ≥ representation). Representation is of type string: (∀ representation. string). C3 refers to the common property. A RC can have one or more common properties: (1 ≤ common_property) ∩ (n ≥ common_property). Common

DOI: 10.5963/IJCSAI0203002

property is a string ($\forall$ common_property. String). C4 refers to a particular property. A RC can accept zero or more particular property. Each one is projected in a specific context designated by [Si]. This is formalized as follows ($0 \leq$ particular_property) [Si] $\cap$ ($n \geq$ particular_property) [Si]. A particular property is a string ($\forall$ particular_property. String).

1) TBOX of OMR:

Recall that TBOX is composed of concepts and primitive and defined roles (relationships). A primitive concept is an atomic concept subsumed by a Top concept (most general concept). The latter denotes the most general concept known as the root of concepts hierarchy. A primary role is subsumed by Top role which is the root of roles hierarchy. Primitive concepts and roles can be combined using constructors to form respectively the concepts and roles. TBOX of the contextual knowledge of our ontology is described in (4) and (5). Our ontology is composed essentially with *Requirement* concept and *Context* concept that are primitive concepts which are subsumed by Top concept. The parameter concept is subsumed by the concept Context. The property concept is subsumed by the concept *Requirement*.

Concepts:

(4)
Requirement $\sqsubseteq$ Top
Context $\sqsubseteq$ Top
Property $\sqsubseteq$ Requirement
Parameter $\sqsubseteq$ Context
Localization $\sqsubseteq$ Parameter
User $\sqsubseteq$ Parameter
Activity $\sqsubseteq$ Parameter
Time $\sqsubseteq$ Parameter
Physical_environment $\sqsubseteq$ Parameter
IP_Adress $\sqsubseteq$ Physical_environment

Roles:

(5) (Requirement, Context): accorded_contexte
(Context, Parameter): composed_parameter
(Requirement, Property): composed_of
(Requirement, Requirement): have_relation

We admit five contextual parameters for our study domain. This constraint is defined in (6).

(6) Context $\sqsubseteq \exists$ composed_parameter. Parameter
Contexte $\sqsubseteq \forall$ =5 composed_parameter. Parameter

Contextual parameters are disjoint. This disjunction is formalized as shown in (7) or (8).

(7) Localization $\cap$ User $\cap$ Activity $\cap$ Time $\cap$ Physical_environment = {}

(8) Activity $\sqsubseteq \neg$ Time
Activity $\sqsubseteq \neg$ Localization
Activity $\sqsubseteq \neg$ Physical_environment
Activity $\sqsubseteq \neg$ User

The requirement concept is given in a context and admits at least one property (9).

(9) Requirement $\sqsubseteq \exists$ accorded_context.Context $\cap \forall$ composed_of.Property $\cap \geq 1$ composed_of.property

Once the T-BOX is defined, we proceed to define the ABOX.

2) ABOX of OMR:

Recall that the ABOX represents the instantiation. In fact, ABOX contains the ontology assertions. We present in (10), a part of the ABOX used in the instantiation of our ontology.

(10) tut_ratf_ins : Context
2010/2011 : Time
192.16.0.33 : IP_Adress
inscription : Activity
rmc : Localization
tuteur : User
composed_parameter (tut_ratf_ins, 2010/2011)
composed_parameter (tut_ratf_ins, 192.16.0.33)
composed_parameter (tut_ratf_ins, connexion)
composed_parameter (tut_ratf_ins, rmc)
composed_parameter (tut_ratf_ins, tutor)
Subscribe: Requirement
accorded_context (Subscribe, tut_rmc_ins)
composed_of (Subscribe, confirme_subscription)
composed_of (Subscribe, feel_forms)

For visualization of the MRO concepts, we present visualization in three levels of abstraction.

C. Ontology Visualization

The visualization of ontology is an important step in the process of knowledge representation. It allows the designer and the domain expert to see the structure and components of the ontology. This allows the visualization of the ontology expert to check the MRO components and make some adjustments if necessary. Several plugins like *OWLviz*, available in the *Protégé* editor, allow the visualization according to several levels of abstraction. Fig. 9 shows a visualization of our MRO according *OWLviz*.

Fig. 9 Ontology with OWLviz

DOI: 10.5963/IJCSAI0203002

Fig. 10 shows the visualization of the MRO as the plugin Ontoviz [20]. This visualization helps to explain the relationship between instances on the one hand and among organizations and individuals on the other.

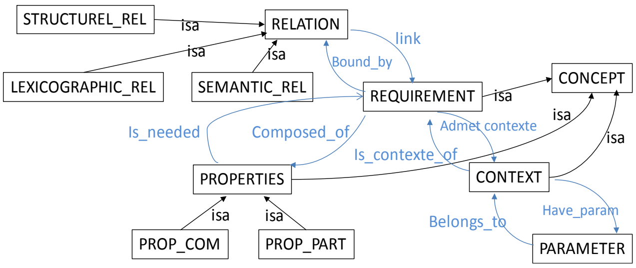

Fig. 10 MRO Visualization (Level 0)

Fig. 11 shows another level of abstraction. Each concept is interpreted by its instances. These are determined through relationships with other concepts.

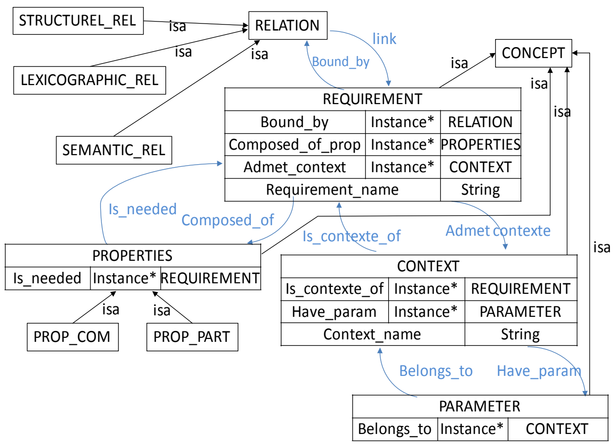

Fig. 11 MRO Visualization (Level 1)

To validate our proposal, we present the *ContextOntoMR* prototype.

VIII. FUNCTIONAL ARCHITECTURE OF CONTEXTONTOMR

ContextOntoMR (*Context & Ontology Multi-representation*) is a prototype developed with the Java language in *NetBeans* environment. It represents an extension of the CASE tool *ArgoUML*. It's an open source application developed with the Java language supports adding new features for the implementation of our RS models, ontology integration and assistance during the RS. The ontology is created by *Protégé* editor. The treatments on the ontology use the *DOM* (*Document Object Model*) and *SAX* (*Simple API for XML*) parsers. For the pretreatment of RS, we used XML. The choice of XML is justified by the possibility of exchange data between applications. XML file manipulation was carried out by the two parsers *DOM* and *SAX*. For inference, we used *Jena*.

We choose the e-learning as application domain for its interest and its richness. In addition, several studies have shown the existence of relationships between pedagogical approaches and socio-cultural contexts of e-learning. For highlighting our contribution, we select a case study coming from this area to illustrate the different modules of *ContextOntoMR* prototype.

The *ContexOntoMR* functional architecture is based on four modules (see Fig. 12). We start from a set of RS specified using the two proposed models (textual and extended UCD). These RSs represent the corpus of MRO. The first module concerns the requirements acquisition and relevant information. These RSs pass through a pretreatment to convert them into a pivot model to overcome the heterogeneities of RS representations.

This pretreatment uses *XQuery* queries to transform the requirements models to pivot model. Comparison and upgrade of MRO is the subject of the third module. The latter refers on the one hand, to linguistic tools and natural language processing (*Wordnet*, *TreeTagger*, *SynoTerme*, etc.). On the other hand, a reasoner is triggered to infer the context from contextual parameters. This module is pursued by an expert for validation and possible interventions. Finally, the assistance module is proposed to facilitate the expression phase of user requirements.

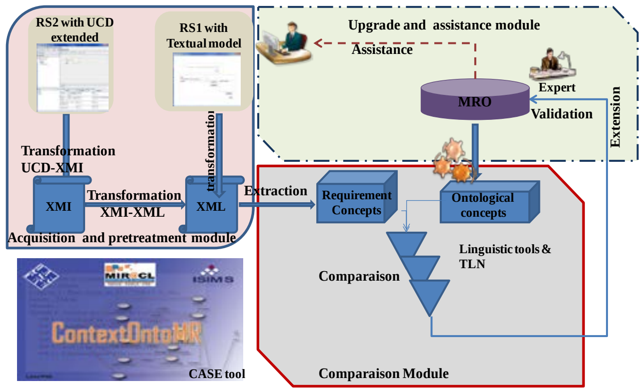

Fig. 12 *ContextOntoMR* functional architecture

IX. MRO CONTRIBUTIONS IN RS VIA CONTEXTONTOMR

Communication between user and analyst is extremely difficult. The analyst is confronted with the users' reluctance to express their requirements in detail. These actors do not use the same terminology that leads to a divergence of meaning given to the requirements expressed by users. Some of them cannot make their requirements and their points of view. The MRO is designed to ensure better communication between the actors. This communication offers a better requirements understanding to achieve a system that meets the fixed objectives. This ontology assists the user and analyst during the RS step taking into account the collaborative aspect in a rather complex system. Requirements must be well expressed and well studied due to the complexity of computer systems, the continuous increase of the shared information volume and the users' contexts divergence. This divergence needs to understand and model the information system functional process in order to obtain a global and coherent vision of user requirements which insured by the proposed ontology.

In fact, *ContextOntoMR* helps user express his requirements and analyst to specify these requirements by supporting the multitude of contexts and representations. *ContextOntoMR* ensures the requirements acquisition in different representations using the proposed models. The prototype ensures the requirements pretreatment specified

DOI: 10.5963/IJCSAI0203002

by saving them as a pivot model. The tool allows the information extraction from the acquired requirements. This information is stored as concepts to make them the source of the MRO. *ContextOntoMR* ensures interaction with a domain expert to validate certain actions. Once the ontology is equipped, the main objective is carried out, namely, the assistance of the user to specify its requirements by taking into account contexts and representations diversity used. It allows users to specify requirements in two models (textual and UCD) making it easy for user / analyst by overcoming the problem of multitude of contexts and representations. This assistance is done by taking into account the context in the proposed ontology. *ContextOntoMR* can assist users through MRO giving the possibility of reusing identified requirements in other contexts.

X. CONCLUSION

In this paper we have investigated the use of ontology in RSs. For that we have proposed an approach implementing a MRO for multi-representation and multi-context RS. This ontology aims to assist users during the acquisition requirements. For this purpose we have proposed two models to acquire requirements. These models are simple, familiar, easy and understandable for both users and analysts.

We have proposed a formalization of ontology concepts with extended DL assuming that this ontology admits a multilayer (core layer and contextual layer). Our ontology is operationally defined in *ContextOntoMR* prototype.

We plan to extend our approach to support other forms in requirement acquisition (in particular the formal representation). Also, we will expand the spectre of relationships formalization (semantic, structural and lexical) between the requirements concepts in the extended DL. These relationships help to further clarify the semantics of requirements concepts according to its contexts.

For the contextual aspect, we will take into account other contextual parameters to support other features to improve ontology flexibility. Furthermore, we intend to overcome other problems relating to the RS, ie, uncertainty and fuzzy requirements using possibilistic ontology.

REFERENCES

[1] Claramunt C., "Un modèle de vues spatiales pour une représentation flexible de données géographiques", Thèse de doctorat de l'Université de Bourgogne, 1998.

[2] Papazoglou M.P., Krämer B.J., "A Database Model for Object Dynamics", VLDB Journal, Vol. 6, 1997.

[3] Gentile M., "An object-oriented approach to manage the multiple representations of real entities", Thèse de doctorat en informatique, Lausanne, EPFL, N° 1490, 1996.

[4] Benslimane D., Arara A., "The multirepresentation ontologies: a contextual description logics approach", The 15th Conference on Advanced Information Systems Engineering (CAiSE '03), Klagenfurt/Velden, Austria, 16-20 June, 2003.

[5] Strang T., Linnhoff-Popien C. & Frank K. "CoOL: A Context Ontology Language to enable Contextual Interoperability". LNCS 2893, 236–247, 2003.

[6] Bouquet P., Giunchiglia F., Van Harmelen F., Serafini L. & Stuckenschmidt H. "C-OWL: Contextualizing Ontologies", International Semantic Web Conference, pp. 164-179, 2003.

[7] Henricksen, Karen Indulska, Jadwiga "Developing context-aware pervasive computing applications: Models and approach", Pervasive and Mobile Computing, vol. 2, no. 1, pp. 37–64, 2006.

[8] Chen, H. "An Intelligent Broker Architecture for Pervasive Context-Aware Systems", PhD Thesis, University of Maryland, Baltimore County, 2004.

[9] Abhishek Singh, Michael Conway, "Survey of Context aware Frameworks- Analysis and Criticism", UNC-Chapel Hill ITS, The University of North Carolina, 2006.

[10] Korpipää, P., Mantyjarvi, J., Kela, J., Keranen, H. & Malm, E-J, "Managing context information in mobile devices", IEEE Pervasive Computing, Vol. 2, No. 3, July–September, pp. 42–51, 2003.

[11] Dey A. K., Salber D. & Abowd G., "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications", Human Computer Interaction J., vol. 16, p. 97–166, 2001.

[12] Mtibaa A., Gargouri A., "A multi-representation ontology for the specification of multi-context requirements", The second International conference on Signal Image Technology & Internet based System, Hammamet, Tunisia. E. Damiani et al. (Eds.): extended version, LNCS 4879, Springer-Verlag Berlin Heidelberg, pp. 235–246, 2009.

[13] Muller P.A., "Modélisation objet avec UML", Editions Eyrolles ,2002

[14] Mtibaa A, Maalej M. and F. Gargouri, "Contextual dimension in ontology dedicated to requirement specification", IADIS International Conference Information Systems, Berlin, Germany, 10-12, March 2012

[15] Grau B.C., Parsia B., and Evren Sirin. "Working with Multiple Ontologies on the Semantic Web", In Proceedings of the 3thd International Semantic Web Conference, Volume 3298 of Lecture Notes in Computer Science, 2004.

[16] Rifaieh R., Arara A., Benharkat A.N. "MurO: A Multi-representation Ontology as a Foundation of Enterprise Information Systems" International conference on information technology, No. 7, Hyderabad , vol. 3356, pp. 292-301, ISBN 3-540-24126-4, 2004.

[17] Baader F., Calvanese D., Mcguinness D. L., Nardi D., Patelschneider P. F., "The Description Logic Handbook: Theory, Implementation, and Applications", Cambridge University Press Eds, 2003.

[18] Tsarkov D., Horrocks I., "DL reasoner vs. first-order prover", Description Logic Workshop DL 2003, volume. pp. 152-159, 2003.

[19] Russell S.J., Norvig P., "Artificial Intelligence: A Modern Approach", 2ème édition. Prentice Hall, 2002.

[20] Gagandeep S., Prabhakar T.V, Jayanta C. "OntoViz: Visualizing Ontologies and Thesauri Using Layout Algorithms", Th 5th conference of Asian federation for information technology in agriculture, 2006.

Achraf Mtibaa received his Ph.D. in Computer Science from University of Sfax, Tunisia in 2011. He is actually an assistant professor of Computer Science at the *Higher Institute of Electronics and Communication of Sfax* in Tunisia. His research interests include ontology, requirements specification, and knowledge representation.

DOI: 10.5963/IJCSAI0203002

Faiez Gargouri received his Ph.D. in Computer Science from University of Renés Descartes, Paris 5, France in 1995. He is actually a Professor of Computer Science at the *Higher Institute of Computer Science and Multimedia* of Sfax in Tunisia. His research interests include Information system, ontology engineering, semantic web, advanced databases, and data warehouses.

DOI: 10.5963/IJCSAI0203002