Available online at www.sciencedirect.com

ScienceDirect

Expert Systems with Applications 35 (2008) 1899–1908

Expert Systems with Applications

www.elsevier.com/locate/eswa

Representation of context-dependant knowledge in ontologies: A model and an application

Fernando Bobillo, Miguel Delgado, Juan Gómez-Romero *

Department of Computer Science and Artificial Intelligence, E.T.S.I. Informática, University of Granada, C. Periodista Daniel Saucedo Aranda, 18071 Granada, Spain

Abstract

Most of current Information and Knowledge Based Systems manage impressive amounts of information, ranging from local databases to resources imported from the web. In addition to widely pointed-out integration and maintenance difficulties, other common problem is overwhelming of users with much more information than the strictly necessary for fulfilling a task, forcing them to dig in a list of results to find valuable answers. This issue is especially critical in mobile decision support systems, since neither the capabilities of the handheld devices nor the users' situation are likely to ease or even permit carrying out this manual post-processing.

Use of context knowledge has been envisioned as an appropriate solution to deal with this information overload matter: system responses can be summarized and customized depending on the situation and the preferences of the user, which results in presenting him only relevant information.

In this work we propose a formal model for representing in ontologies relevance relations between context descriptions and domain-knowledge subsets. Besides the formulation of the model, we describe an algorithm to reason within it. We demonstrate the contributions of our approach with the implementation of the IASO application, a system which provide doctors in nomadic healthcare with brief context-dependant pieces of advice about patients' electronic health records.

Keywords: Knowledge representation; Ontologies; Mobile systems; Context-aware systems; Medical informatics

1. Introduction

If business managers were asked for defining current information systems using one word, they would probably try to zip their answer in order to use two: connected and massive. Storages populated with Gbytes or even Tbytes of data are available across corporate networks. The situation gets even more complicated if the Internet is considered, as huge amount of valuable data can be harvested from it. Consequently, corporative Knowledge Based Systems (KBSs) – i.e. software systems which use massively expert knowledge in order to solve problems in specific application fields – are expected to incorporate these several information sources to provide complete, accurate, and up-to-date pieces of advice to decision makers.

As a result, functional KBSs usually manage so many resources to solve most of the requests that it is common that users accessing big-scale systems are supplied "excessive" information, in such a way that the time to filter it manually is too long or simply it cannot be processed. This issue has been pointed out in the literature with the name of "information overload" (Eppler & Mengis, 2004) and is a frequent cause of Knowledge Management (KM) failure, since it decreases individual's performance and leads to unproductive and ineffective management procedures. Thus, the challenge for KBS technologies is to support tailoring and summarizing of information collected from

* Corresponding author. Tel.: +34 958243194; fax: +34 958243317.
E-mail addresses: fbobillo@decsai.ugr.es (F. Bobillo), mdelgado@ugr.es (M. Delgado), jgomez@decsai.ugr.es (J. Gómez-Romero).

0957-4174/$ - see front matter
doi:10.1016/j.eswa.2007.08.090

massive, heterogeneous and distributed sources depending on user needs (Farhoomand & Drury, 2002).

Though it has not been so widely studied, information overload is especially dramatic in mobile systems, as handheld devices have reduced data transmission and presentation capabilities. All Decision Support Systems (DSSs) are expected to realize what people really need and to act consequently, but this requirement clearly becomes more critical when nomadic users are involved:

- Although new wireless technologies provide Mb/s bandwidth, screen size remains small given that this is mandatory for keeping proper size, weight and battery life of portable devices. Presentation of large volumes of data on a PDA display (not talking about a cell phone, where the state of affairs is even worse) is a critical factor for the success of mobile applications, since it is too easy to annoy users even with the littlest pieces of non-significant data – which could be assumable in a desktop monitor.
- On the other hand, it must be kept in mind that the scenario from where a nomadic user requests system support is completely different from those which only have stationary users. Mobile workforce has to face dynamical decision processes happening in the playing field, which almost always means real-time constrained choices with immediate consequences. Moreover, sometimes they will be interacting with a customer, so all their attention cannot be put on the device. Accordingly, ubiquitous users must not be overwhelmed by a bunch of irrelevant data nor it is acceptable to make them manually filter results.

Hence it is widely accepted that the silver bullet for mobile knowledge delivery is smart result filtering: to summarize available data to provide nomadic users with the smallest amount of information which is significant for the decision process. What is "significant" for a user will depend on his circumstances, which can be regarded as a mix of desires, needs and environment facts, i.e. (in a wide sense) the context. Managing context information – that is, being aware of the context – in mobile applications is both interesting and challenging, since situation changes frequently as the holder of the handheld device moves from one scenario to another. Likewise, context varies if the user's activity does, which will happen often if the device is expected to be carried the most of the time by him.

In this work we will present a proposal to tackle the problem of information overload, paying special attention to mobile KBSs, by using context knowledge. The core of our approach is the Context-Domain Relevance (CDR) model, a formal pattern for representing relevance of information depending on use scenarios in ontological knowledge bases (KBs). Besides the formal semantics of the model, we also provide an algorithm to extract context-dependant summaries by reasoning within the ontology.

We will demonstrate the contributions of our approach describing the Intelligent ASsistant for Outdoors healthcare (IASO), a prototype KBS whose KB is based on this pattern. IASO behaves as an extension to the current information system used in the Clinical Hospital "San Cecilio" of Granada which allows nomadic physicians to get in their mobile devices compact summaries about patients' health records and recommendations about further tests.

The remaining of this document is structured as follows. Section 2 presents a use case of a healthcare support KBS which illustrates the motivation of this work. In Section 3, we give an overview of the literature about environment awareness in mobile and pervasive computing, besides some current proposals concerning reasoning with contexts and micro-theories in ontological representations. Section 4 is devoted to the formalization of our model and the reasoning procedure, and an example is also depicted. In Section 5, we present the IASO prototype; application features, as architecture, implementation and functionalities, are detailed. Finally, in Section 6 some conclusions and directions for future work are pointed out.

2. Motivation

Let us suppose a physician who needs to consult a patient's clinical data in order to set a proper treatment for him. If the healthcare act is taking place inside the hospital, the doctor will be allowed to access the Hospital Information System (HIS) and to retrieve all the patient's Electronic Health Records (EHRs). Having enough time and knowledge – and depending on the usability of the software system – the specialist will rule out all the useless pieces of information and will get the ones he is interested in.

This example depicts a typical use case of a classical Information System. We can consider now another physician in an emergency-assistance unit which is caring at the road for a patient injured in an accident. Knowing some data about his clinical history will be as well helpful in this situation; for instance, some data about patient's adverse drug events (ADEs) may have been recorded. Nevertheless it is not probable that the HIS can be accessed from outside the hospital (even less using a portable device as the one which will be likely used in emergency healthcare) and, if possible, the doctor would not have enough time to review all the stored electronic records.

In the latter situation, a brief report including those pieces of the patient's clinical data which ought to be considered would be very valuable. The clinical procedure which is going to be carried out would determine which data should be part of this summary. For example, is the patient is slightly unconscious and has an hemorrhagic laceration, information about if he has been diagnosed of bad reactions to procaine (an anesthetic drug which reduces bleeding but is also often badly metabolized and triggers allergic reactions) should be taken into account, among others.

Consequently, two different kinds of knowledge are to be managed by such mobile system:

- Domain knowledge about the problem which must be resolved; this is made up by the patients' electronic health records.
- Context knowledge about the scenarios where the domain knowledge will be used; for our doctor, this would be a vocabulary to briefly describe the situation of the patient he is going to attend.

To state which knowledge from the domain must be considered in each scenario, links between both submodels can be defined. Continuing our example, a link asserting that 'data about previous anesthetic drugs reactions' should be considered when 'the patient has a penetrating wound' should be created. Other links can be similarly included following recommendations of clinical and ADE guidelines.

Building these links is the aim of the Context-Domain Relevance model, a formal pattern for designing knowledge bases where these different kinds of knowledge are shown. Given a representation of the domain knowledge and a vocabulary for describing contexts, the Context-Domain Relevance model defines the semantics of relationships between both and establishes some rules to encode and reason with them. The formal definition of the model is presented in Section 4, whereas some related work is reviewed next.

3. Related work

Nowadays, advent of new portable devices and wireless communication technologies has resulted in putting more emphasis in mobile applications, given raise to the so-called area of Pervasive Computing. Pervasive Computing aims to develop technologies that support every day routines unobtrusively using a swarm of reduced wireless-connected computing devices (Weiser, 1999). Possible applications of Pervasive Computing areas range from smart rooms to recommendation systems, teleassistance, marketing and advertisement, monitoring and alerting in healthcare, etc. Some early research projects, as Interactive Workspaces (Johanson, Fox, & Winograd, 2002) (Stanford University), EasyLiving (Brumitt, Meyers, Krumm, Kern, & Shafer, 2000) (Microsoft), Aura (Garlan, Siewiorek, Smailagic, & Steenkiste, 2002) (Carnegie Mellon) and Oxygen (Dertouzos, 1999) (MIT), were aimed to provide platforms for pervasive systems where a considerable amount of different devices must be coordinated.

Context awareness, defined as the capacity to acquire, represent and process context information to change application behavior, has been traditionally remarked as a key requisite for pervasive systems (Satyanarayanan, 2001). In a wide sense, context is "any information that can be used to characterize the situation of an entity" (Dey & Abowd, 2000). More specifically, context knowledge is usually considered to gather a mixture of geo-spatial data, sensor inputs, user intentions and desires, and service descriptions (Schmidt, Beigl, & Gellersen, 1999). In that regard, the Context Toolkit is one of the first systematical approaches to a generic framework for context-aware systems (Dey, Abowd, & Salber, 2001).

A highly-descriptive formalism with reasoning features is convenient for context-knowledge representation and management. Not surprisingly, ontologies have been proposed to be used for modeling context knowledge, as they provide some advantages over other formalisms: reusability, sharing, reasoning, standardization, supporting tools, etc. (Strang & Linnhoff-Popien, 2004). Some early approaches (and common failures) are presented in Brézillon (1999). More recent developments are for instance these in Chen, Finin, and Joshi (2005), Gu, Pung, and Zhang (2005), Khedr and Karmouch (2005) and Kwon et al. (2006).

It is worth to note that most of these works borrow technologies from Semantic Web (Berners-Lee, Hendler, & Lassila, 2001). Actually, skyrocketing research activity in Semantic Web during last years has contributed with several theories and tools which have been assimilated by Pervasive Computing. In Ranganathan, McGrath, Campbell, and Mickunas (2003) some issues which can be addressed using Semantic Web technologies are explained. Masuoka, Labrou, Parsia, and Sirin (2003) tackles one of these problems, the automatic coordination of actors in mobile interaction, and suggests attaching formal descriptions to services in order to discover, communicate, and integrate clients and providers in pervasive environments. The work in Lassila and Khushraj (2005) also points out this difficulty and proposes to use Semantic Web Services (an in-progress specification from the W3C) to achieve serendipitous device coalitions.

Finally, we must remark that contexts have been widely studied from a theoretical perspective in Knowledge Representation, in contrast to the more pragmatic aforementioned approaches. Whereas the latter intend just to provide means for managing and using environment data, the former studies non-monotonic knowledge representation formalisms, i.e. models which are satisfiable or not depending on some circumstances. In this sense, Guha, McCool, and Fikes (2004) examines some classical works about contexts and micro-theories in Artificial Intelligence, and extends some of these ideas to solve context-dependant aggregation problems in the Semantic Web. Similarly, Bouquet, Giunchiglia, van Harmelen, Serafini, and Stuckenschmidt (2004) proposes C-OWL, an extension to OWL to define mappings between locally-interpreted and globally-valid ontologies. To end up, we shall mention that the idea underlying our model is quite similar to the multi-viewpoint reasoning in Stuckenschmidt (2006), though it concentrates on the conditional interpretation of a model (how to reduce an ontology depending on the viewpoint submodel), whereas we focus on their relevance (in which circumstances a submodel should be considered).

4. The Context-Domain Relevance model

4.1. Antecedents

In this work we will use ontologies to materialize our model since, as mentioned in Section 3, they have been remarked to be a suitable formalism to build a KB including context knowledge. Ontologies, defined as "formal, explicit specifications of a shared conceptualization" (Studer, Benjamins, & Fensel, 1998), encode machine-interpretable descriptions of the concepts and the relations in a domain using abstractions as class, role or instance, which are qualified using logical axioms.

Properties and semantics of ontology constructs are determined by Description Logics (DLs) (Baader, Calvanese, McGuinness, Nardi, & Patel-Schneider, 2003), a family of logics for representing structured knowledge which have proved to be very useful as ontology languages. DLs are structured in levels, having each of them a computational complexity that depends on the expressivity of the allowed constructs. DL levels are named with capital letters that denote these constructs. In the remaining of this work we will consider the minimal subset proposed in $\mathcal{ALC}$ (attributive concept description language with complements), since it is expressive enough to encode our model. Other commonly-used logic is $\mathcal{SHOIN}(\mathcal{D})$ (almost equivalent to the 'DL' level of the standard ontology language OWL (McGuiness & van Harmelen, 2004)), which extends the logic $\mathcal{SH}$ ($\mathcal{ALC}$ plus transitive roles and role hierarchy) with nominals ($\mathcal{O}$), inverse roles ($\mathcal{I}$), cardinality restrictions ($\mathcal{SHOIN}(\mathcal{N})$) and datatypes ($\mathcal{D}$).

Formally, an ontology is a triple $O = \langle K_R, K_T, K_A \rangle$, where K_R (the Role Box or RBox) and K_T (the Terminological Box or TBox) comprise the intensional knowledge, i.e. general knowledge about the world to be described (statements about roles and concepts, respectively), and K_A (the Assertional Box or ABox) the extensional knowledge, i.e. particular knowledge about a specific instantiation of this world (statements about individuals in terms of concepts and roles).

In $\mathcal{ALC}$ there is no RBox, since no axioms involving roles are allowed. In more expressive logics, K_R consists of a finite set of role axioms stating restrictions as subsumption, transitivity, cardinality, etc.

An $\mathcal{ALC}$ TBox K_T consists of a finite set of general concept inclusion (GCI) axioms of the form $C_1 \sqsubseteq C_2$, which means that concept C_1 is more specific than C_2, i.e. C_2 subsumes C_1. A concept definition $C_1 \equiv C_2$ (C_1 and C_2 are equivalent) is an abbreviation of the pair of axioms $C_1 \sqsubseteq C_2$ and $C_2 \sqsubseteq C_1$. Concept expressions for C_1, C_2 can be derived inductively starting from atomic primitives. Valid constructs for $\mathcal{ALC}$ are shown below:

$$
\begin{array}{lll}
C_1, C_2 \rightarrow & A & \mid \text{(atomic concept)} \\
& \top & \mid \text{(top concept)} \\
& \bot & \mid \text{(bottom concept)} \\
& C_1 \sqcap C_2 & \mid \text{(concept conjunction)} \\
& C_1 \sqcup C_2 & \mid \text{(concept disjunction)} \\
& \neg C_1 & \mid \text{(concept negation)} \\
& \forall R.C & \mid \text{(universal quantification)} \\
& \exists R.C & \mid \text{(full existential quantification)}
\end{array}
$$

An $\mathcal{ALC}$ ABox consists of a finite set of assertions about individuals (denoted a and b). An assertion is either a concept assertion $a : C$ (meaning that a is an instance of C) or a role assertion $(a,b) : R$ (meaning that (a,b) is an instance of R).

A DL ontology not only stores axioms and assertions, but also offers some reasoning services, such as KB satisfiability (or consistency), concept satisfiability, subsumption or instance checking. In $\mathcal{ALC}$ most inference services are mutually reducible, so only some of them are usually considered.

4.2. Definition

Before defining the Context-Domain Relevance (CDR) model, let us remind the two knowledge submodels mentioned in Section 2: the domain ontology and the context ontology.

The domain ontology $O^D = \langle K_R^D, K_T^D, K_A^D \rangle$ contains the knowledge required to solve the concrete problem that the system is facing. As expected, concepts of this ontology represent entities with associated semantics, roles establish connections among them, and instances represent individuals of this world. This ontology can be arbitrarily complex and is closely related to the problem. We will use the notation $D_j \mathring{\in} O^D$ to name complex concepts expressions D_j built using elements in O^D and ontology constructs. Note that, in principle, these D_j are not part of the domain ontology.

The context ontology $O^C = \langle K_R^C, K_T^C, K_A^C \rangle$ contains the knowledge required to express the circumstances or the *surroundings* under which the domain knowledge will be used. The context ontology can be seen as a (formal) vocabulary or *lingo* with which these situations can be described. Being strict, context knowledge is not part of the original problem, though it can be indispensable to solve it; in fact, it would be possible to reuse the same context model in completely different areas. Context knowledge can range from low-level sensor data (like location, time or humidity) to abstract information (like preferences, desires or mental state). We will use the notation $C_i \mathring{\in} O^C$ to name complex concepts expressions C_i built using elements in O^C and ontology constructs. Like in the previous case, these C_i are not necessarily part of the context ontology.

Intuitively, we can guess that a CDR ontology will be made of new classes (the so-called profiles) which will relate C_i context concepts with D_j domain concepts through quantified roles. We must note that, accordingly, our proposal only considers the intensional component of the knowledge base.

Regarded this we define constructively a CDR ontology as follows:

Definition 1. Let O^D and O^C be, respectively, the domain ontology and the context ontology, $C_i \mathring{\in} O^C$ a context concept built with K_T^C classes, and $D_j \mathring{\in} O^D$ a domain concept built with K_T^D classes.

The CDR ontology which relates the set of pairs of concepts $\{(C_i, D_j)\}$ (i.e. states that D_j is interesting when C_i happens) is an ontology $O^P = \langle K_R^P K_T^P, K_A^P \rangle$ which satisfies:

(1) $K_A^P = \emptyset$.
(2) K_T^P include definitions for the concepts $P_\top, C_\top, D_\top, P_{i,j}, C_i, D_j$, where:
  (a) $P_\top, C_\top, D_\top$ are the super-classes Profile, Context and Domain:

$$P_{i,j} \sqsubseteq P_\top \wedge P_\top \equiv \bigcup_{i,j} P_{i,j}$$

$$C_i \sqsubseteq C_\top \wedge C_\top \equiv \bigcup_{i} C_i$$

$$D_j \sqsubseteq D_\top \wedge D_\top \equiv \bigcup_{j} D_j$$

  (b) R_1 is the bridge property linking profiles and context concepts:

$$P_\top \sqsubseteq \forall R_1.C_\top$$

  (c) R_2 is the bridge property linking profiles and domain concepts:

$$P_\top \sqsubseteq \forall R_2.D_\top$$

  (d) $P_{i,j}$ is the profile linking named context C_i and named domain D_j:

$$P_{i,j} \equiv \exists R_1.C_i \sqcap \exists R_2.D_j$$

$$K_R^P = \{R_1, R_2\}$$

(3) O^P is consistent.

Fig. 1 depicts the meaning of this definition. It shows how $P_{i,j}$ concepts are a reification of the notion of "relevance" relationship between context and domain concepts. Representing relevance as a concept and not as a role presents some advantages, e.g. possibility of reusing previously-defined profiles or defining new properties (with associated semantics) for them.

The main reasoning task within the CDR ontology will be to find the domain restricted by a context, that is, to find all the classes of the domain ontology which are associated using aprofiles (i.e. are relevant) with a given concept built with the context vocabulary. This can be expressed as follows:

Definition 2. Given O^C, O^D and O^P, the restricted domain of the scenario $S \mathring{\in} O^C$ (being S a complex concept expressed in K_T^C vocabulary) considering O^P comprises all the classes I such as

$$\{I \in K_T^D | (S \sqsubseteq C_n) \wedge (P_{n,m} \in K_T^P) \wedge (I \sqsubseteq D_m)\}$$

Algorithm 1. The restricted domain of a scenario S considering O^P can be computed in practice as follows:

(1) Retrieve all the named contexts C_n which subsumes S:

$$\{C_n \sqsubseteq C_\top | S \sqsubseteq C_n\}$$

(2) Retrieve all the named profiles $P_{k,l}$ which include C_n contexts (via R_1):

$$\{P_{k,l} \sqsubseteq P_\top | (P_{k,l} \sqsubseteq \exists R_1.C_k) \wedge (C_n \sqsubseteq C_k)\}$$

(2) Retrieve all the named domains D_m which are related to $P_{k,l}$ profiles (via R_2):

$$\{D_m \sqsubseteq D_\top | P_{k,l} \sqsubseteq \exists R_2.D_m\}$$

(3) Retrieve all the classes I from K_T^D which are subsumed by D_m:

$$\{I \in K_C^D | I \sqsubseteq D_m\}.$$

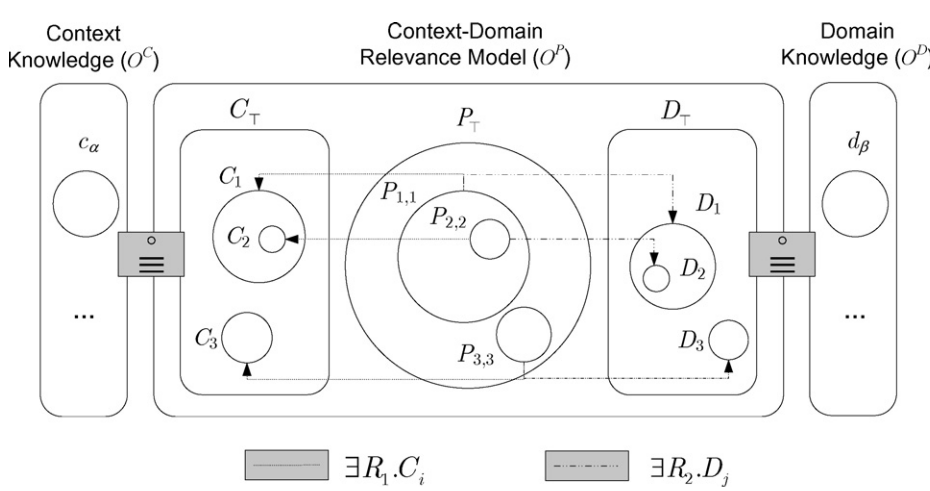

Fig. 1. Sample of a Context-Domain Relevance ontology.

4.3. Complexity

Computational complexity of the inference within the CDR model is conditioned by complexity of context and domain expressions ($C_i \mathring{\in} K_T^C$ and $D_j \mathring{\in} K_T^D$), since $P_{i,j}$ definitions are included in $\mathcal{ALC}$ level. In the simplest case, that is O^C, O^D and O^P ontologies are in $\mathcal{ALC}$, reasoning within the CDR model is asymptotically bounded by ontology classification complexity, which is ExpTime for $\mathcal{ALC}$ with General Concept Inclusions (GCIs) according to Table 1 (Calvanese, 1996).

Supposing that O^C and O^D do not add further complexity, it is possible to reduce the complexity of the CDR model by restricting the allowed constructors for C_i and D_j, moving consequently to a less expressive logic. Restricting negation to atomic concepts and disallowing union concepts would enclose the CDR ontology to $\mathcal{ALE}$, which has PSpace complexity for general reasoning. Other alternative consists on using only acyclic TBoxes, which would give complexities of PSpace for $\mathcal{ALC}$ and coNP for $\mathcal{ALE}$.

Other choices are not appropriate, however. Moving from $\mathcal{ALC}$ to $\mathcal{ALU}$ does not reduce the complexity, neither in the general case nor with acyclic TBoxes. Moving to $\mathcal{AL}$ is not possible, because existential quantification cannot be restricted. Similarly, expressivity of $\mathcal{FL}^-$ is too limited.

According to formulation in Section 4.2, role hierarchies are not necessary in the CDR model. Nevertheless, they may be considered for convenience, in such a way that sub-roles of R_1 and R_2 can be defined with particular semantics and handled consequently. This will increase the complexity to $\mathcal{ALCH}$, but with the advantage that reasoning for the general case still remains ExpTime.

In any case, $\mathcal{ALCH}$ is less expressive than $\mathcal{SHIF}(\mathcal{D})$ (equivalent to OWL-Lite), so reasoning in practice with available DL engines (e.g. Racer (Haarslev & Moller, 2001) or Pellet (Sirin, Parsia, Cuenca-Grau, Kalyanpur, & Katz, 2005)) will be quite efficient, as they are highly optimized and to that worst-case inferences are infrequent. Hence, more complex logics with extended semantics could be as well considered to extend the basic formulation without significant performance impact.

4.4. Example

Continuing with the medical example we have sketched in Section 2, let us suppose the following sample ontologies:

Table 1
Complexity of reasoning in basic DLs

DL\TBox	Acyclic	General
$\mathcal{FL}^-$	PTime	PTime
$\mathcal{AL}$	coNP	PSpace
$\mathcal{ALE}$	coNP	PSpace
$\mathcal{ALU}$	PSpace	ExpTime
$\mathcal{ALC}$	PSpace	ExpTime

- A domain ontology O^D abstracting the information units managed by the hospital information system. Among others, it includes concepts as *Patient*, *ElectronicDocument* or *ElectronicRegisterCoagulationDisorder*, and properties as *relatedToPatient* (with domain equals to *Patient* and range equals to *ElectronicDocument*). Instances of this ontology are the concrete values of patient's electronic health records.
- A context ontology O^C defining a suitable vocabulary to describe patient situations. Therefore, it contains concepts as *Hemorrhage*, *Unconsciousness*, *Trunk* or *High*, and properties like *hasSeriousness* (from *ClinicalFact* to *Seriousness*).

Using the definition of the CDR model, an ontology O^P can be built to reflect which information from the information system must be considered when facing each clinical case. With *hasClinicalFact* and *hasElectronicRegister* being the bridge properties R_1 and R_2, respectively, the profiles in Table 2 will be valid. It can be observed that $C_3 \sqsubseteq C_2 \sqsubseteq C_1$ and $D_1 \sqsubseteq D_3$, $D_2 \sqsubseteq D_3$.

Given this profile set, if the doctor is attending to a "hemorrhagic and unconscious patient with a penetration wound", the system will answer that electronic records about "drug intolerances" should be checked. This is achieved by using the previous algorithm to calculate the restricted domain of a context concept. The process is shown in Table 3.

The final information to be delivered to the doctor's mobile device will be the instances of the *ElectronicRegister* classes in I filtered by patient ID, which must mirror the data stored in the hospital information system. In Section 5, we explain how these instances are recovered in our application without having to import the entire database into the ontology.

Note that in Algorithm 1 descendants of S are not inferred during the reasoning process, since these concepts correspond to more specific context situations – which will probably drive to more specialized domain information. However, it may be interesting to calculate the profiles involving these subcontexts and to provide them as feedback information to the user, in order to recommend him to describe further details of the current scenario. For instance, in this example, C_3 in $P_{3,3}$ is subsumed by S:

$$\begin{aligned}&\mathit{Unconsciousness} \sqcap (\mathit{Hemorrhage} \sqcap \exists \mathit{hasSeriousness.High})\\&\quad \sqcap \mathit{PenetrationWound} \sqsubseteq \mathit{Unconsciousness} \sqcap \mathit{Hemorrhage}\\&\quad \sqcap \mathit{PenetrationWound}\end{aligned}$$

Consequently, the doctor could be advised to carry out other clinical trials to see if the specific part of this restriction ($\exists$ *hasSeriousness.High* qualifier of *Hemorrhage*) is present but has not been diagnosed yet. If this knowledge is supplied afterwards, more information about the patient (information unit *ElectronicRegisterCoagulationDisorder*) will be provided.

Table 2
Example of a Context Restriction Model Ontology

Top concepts
$P_\top \sqsubseteq \top$ $C_\top \sqsubseteq \top$ $D_\top \sqsubseteq \top$
Profile 1,1. When the patient is "unconscious" and "hemorrhagic", registers about "blood pressure disorders" must be checked
$C_1 \equiv Unconsciousness \sqcap Hemorrhage$ $D_1 \equiv ElectronicRegisterBloodPressureDisorder$ $P_{1,1} \equiv \exists hasClinicalFact.C_1 \sqcap \exists hasElectronicRegister.D_1$
Profile 2,2. When the patient is "unconscious", "hemorrhagic" and has a "penetrating wound", registers about "drug intolerances" must be checked
$C_2 \equiv Unconsciousness \sqcap Hemorrhage \sqcap PenetrationWound$ $D_2 \equiv ElectronicRegisterDrugIntolerance$ $P_{2,2} \equiv \exists hasClinicalFact.C_2 \sqcap \exists hasElectronicRegister.D_2$
Profile 3,3. When the patient is "unconscious", with a "highly serious" "hemorrhage" and has a "penetrating wound", registers about "blood pressure disorders", "drug intolerances" and "coagulation disorders" must be checked Unconsciousness $C_3 \equiv \sqcap(Hemorrhage \sqcap \exists hasSeriousness.High) \sqcap PenetrationWound$ $D_3 \equiv ElectronicRegisterBloodPressureDisorder \sqcup ElectronicRegisterDrugIntolerance \sqcup ElectronicRegisterCoagulationDisorder$ $P_{3,3} \equiv \exists hasClinicalFact.C_3 \sqcap \exists hasElectronicRegister.D_3$

Table 3
Resolution of the example

(1)	$S \equiv Hemorrhage \sqcap Unconsciousness \sqcap PenetrationWound$ $C_n = \{C_1, C_2\}$
(2)	$P_{k,l} = \{P_{1,1}, P_{2,2}\}$
(3)	$D_m = \{D_1, D_2\}$
(4)	Let us suppose that *ElectronicRegisterBloodPressureDisorder* is a leaf concept in O^D and *ElectronicRegisterDrugIntolerance* has two subclasses: *ElectronicRegisterProcaineIntolerance* and *ElectronicRegisterPenicillinIntolerance*. Then: $I =$ *ElectronicRegisterBloodPressureDisorder* *ElectronicRegisterDrugIntolerance* *ElectronicRegisterProcaineIntolerance* *ElectronicRegisterPenicillinIntolerance*

5. Intelligent ASsistant for Outdoors healthcare (IASO)

In this section, we present a prototype of the IASO[1] (Intelligent ASsistant for Outdoors healthcare) system, an application which aims to improve specialized attention out of the hospital. We pretend to provide ubiquitous doctors with a tool which increases their knowledge about a patient and consequently, the probability of success of an intervention.

IASO relies on the information system of the Clinical Hospital "San Cecilio" of Granada and uses the CDR model to represent knowledge. IASO is intended to provide brief highly-valuable context-dependant advice about clinical protocols and EHRs to support doctors' decision processes. More precisely, the objective of IASO is to offer answers to physicians' questions such as "what should I know about the history of this patient" and "what tests should I do next to him?" in a given situation. Two tasks are required to build IASO system: (i) to build a suitable knowledge model which describes semantically the information to be managed, and (ii) to implement the software that allows to distribute the (summarized) information about protocols and clinical histories to the doctors.

The knowledge model of the system has been built following the CDR pattern, so we have three OWL ontologies which, respectively, characterize the information about histories stored in the HIS (the *domain*), define a language to describe patients' clinical situation (the *context*), and connect subsets of both of them (the *relevance*). Reuse of ontologies has been borne in mind and, for instance, UMLS (Bodenreider, 2004) could be used to state the context vocabulary, or (formal) medical guidelines could be considered when building the profiles of the CDR ontology (ten Teije et al., 2006).

The software has been implemented following the architecture for mobile KBS proposed in a previous work (Delgado, Gómez-Romero, Magaña, & Pérez-Pérez, 2005). Fig. 2 depicts the materialization of this architecture in the IASO system, where the IASO Client, the IASO Service and the Ontology Management modules can be identified.

[1] Iaso (also, "Iaso Tholus" or "Jaso"; in Ionian Greek, "Ieso") was the Greek goddess of recovery. She was the daughter of Asclepius and had three sisters and one step sister: Panacea, Aceso, Aglæa- Ægle, and Hygeia. She helped the sick and diseased along with Panacea, Aglæa, and Hygeia.

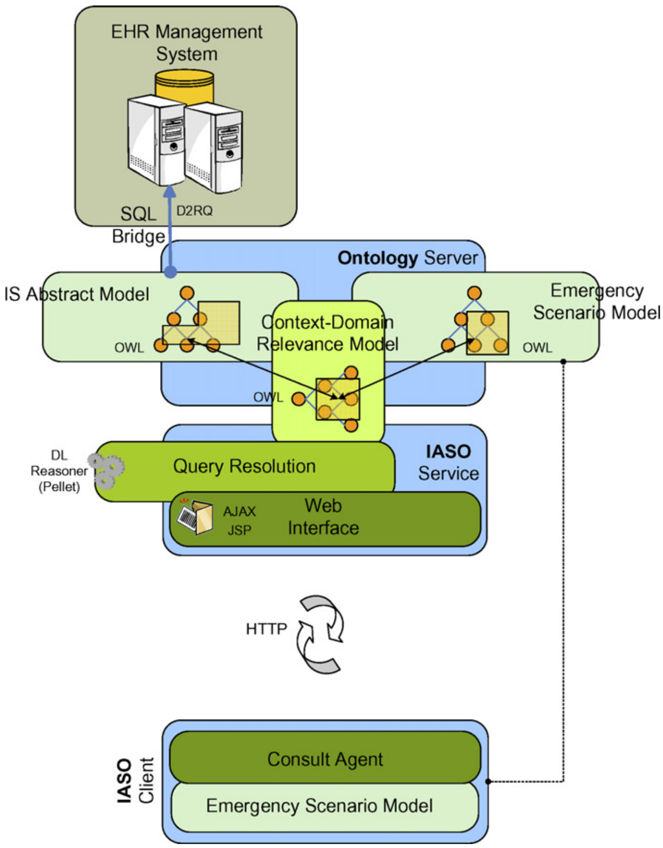

Fig. 2. Schema of the IASO application.

In the current implementation, the IASO Client is simply a web browser which communicates with the service using HTTP protocol. Though this drives to an unbalanced schema with the server supporting nearly all the workload, it ensures compatibility with almost every device no matter how limited it is, as long it is able to browse the Web.

The IASO Service provides an access point to the functionalities of the whole system through HTML forms. It gets user requests and transforms and resolves them using the available models in the Ontology Server. After importing the suitable ontologies, reasoning within the CDR model is carried out with a Java implementation of Algorithm 1 – based on Protégé-OWL API (Knublauch, Fergerson, Noy, & Musen, 2004) – and a DL inference engine – in our case, Pellet (Sirin et al., 2005), which can be executed in other machine. IASO Service runs on an Apache Tomcat web server.

In order to minimize issues due to HTTP interaction, JSP (Java Server Pages) (Dudney, Lehr, Willis, & Mattingly, 2004) and AJAX (Asynchronous JavaScript + XML) (Crane, Pascarello, & James, 2006) technologies have been used to implement the server logic which generates the web forms for the client. Combination of both offers some remarkable advantages in this kind of systems:

- Easy implementation of eye-catching interfaces, comparable to those created with mobile-specific development platforms, which allow to present data properly.
- Reduction of the processing performed in the server, as they promote the execution of complex scripts in the client (which must support Javascript).
- More control of the information flow between the server and the client by means of asynchronous mechanisms, in contrast to the usual synchronous schema in the web.

The Ontology Server manages the underlying knowledge. It runs on an Apache web server, storing the three IASO ontologies and integrating them with other information sources, in this case the HIS. The link between the domain ontology and the patient database, labelled "SQL Bridge" in Fig. 2, is a key part of the application. Once configured, it allows to access transparently the database registers as if they were instances of the ontology, but without importing them. Otherwise all the HIS data should be incorporated to the domain ontology, which brings out two serious problems: data integrity (ontology instances must be synchronized with the database contents, which are likely to change frequently) and efficiency (time and memory spent in tasks with such a huge ontology would be intolerable). The "SQL Bridge" has been implemented using D2RQ (Bizer & Seaborne, 2004), a set of tools to describe associations between relational databases and OWL/RDF(S) ontologies and to deal with resulting knowledge models.

The complete resolving procedure in IASO is the following. A physician, equipped with a handheld device connected to Internet through a wireless network, uses his web browser (IASO Client) to get the query form from the server (IASO Service). Using this query form, the doctor can describe the situation of the patient with the vocabulary stated in the Emergency Situation Model (the context ontology) and send it to the service. The service acts like a filtering agent which takes into account the given description to provide just interesting information related to him, that is, relevant EHRs from the HIS (if available), and further tests which ought to be carried out. Having collected all this information, the service gives it an appropriate format and sends it back to the client, where it can be handily reviewed.

The current prototype (available at http://arai.ugr.es:8084/IasoTest/EntryPoint) implements the schema in Fig. 2. It includes three sample ontologies and a database reproducing the example in Section 4.4. Figs. 3 and 4 show execution of the IASO Client on a Pocket PC (using Windows Mobile 5 emulator and Opera Mobile Web Browser). Fig. 3 presents the entry form and a sample consult for the patient "Juan Gomez", whereas Fig. 4 illustrates the results for this query. 'Clip' icons represent I concepts, i.e. concepts from the domain ontology which have been recovered as interesting. 'Ok' bullets denote instances of I concepts, i.e. the content of the database registers corresponding to this concept of the domain ontology and this patient. 'Exclamation' bullets enumerate named concepts in expressions defining descendants of S, i.e. concepts which are part of more specific defined contexts and that may be considered in subsequent consults.

Fig. 3. IASO prototype query form.

Fig. 4. IASO prototype answer form.

6. Conclusions and future work

In this work, we have presented the CDR model, a formal pattern for the representation and management of context-relevant knowledge in ontologies. This model allows to cope with the problem of information overload in KBSs, which is critical in mobile systems due to their special features. Besides the model we also provide a reasoning procedure to infer which sections of the domain knowledge are interesting or significant in a given situation.

We have demonstrated the contributions of our approach by presenting the IASO system, a KBS which provides nomadic doctors with brief summaries of the patients' EHRs that should be considered when taking care of them and gives recommendations about further test to be carried out. Accordingly, the current prototype of the IASO application, which proves feasibility and advantages of our proposal, has been detailed.

From the description of the IASO prototype it can be deduced that the main task for future work is to extend it to become a fully-operative system and test it in real scenarios. Consequently, we are working to enhance the knowledge models in the system – the clinical vocabulary to describe patients' situation, the HIS abstract semantic layer and the relevance model – in order to make it useful in complex environments. Two more challenges are decentralization of the system (turning clients more intelligent, that is, able to manage more knowledge) and overcoming of security issues (medical information must be kept private).

Fuzzy and probabilistic/possibilistic extensions to the crisp CDR model are being as well considered, as they would allow to define weighted relevance relations between contexts and domains and, which is more interesting, partial matching of similar contexts. We believe that providing mechanisms to represent vagueness and imprecision is crucial for this model if it is really intended to apprehend the "ourselves and our circumstances" in the real world.

Acknowledgements

This research has been partially supported by the project TIN2006-15041-C04-01 (Ministerio de Educación y Ciencia). Fernando Bobillo holds a FPU scholarship from the Spanish Ministerio de Educación y Ciencia. Juan Gómez-Romero holds a FD scholarship from Consejería de Innovación, Ciencia y Empresa (Junta de Andalucía).

References

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., & Patel-Schneider, P. F. (2003). The Description Logic handbook: Theory, implementation, and applications. Cambridge University Press.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific American, 284*(5), 34–43.

Bizer, C., & Seaborne, A. (2004). D2RQ-treating non-RDF databases as virtual RDF graphs. In *Proceedings of the third international semantic web conference (ISWC2004)*.

Bodenreider, O. (2004). The Unified Medical Language system (UMLS): Integrating biomedical terminology. *Nucleic Acids Research, 32*, 267–270.

Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L., & Stuckenschmidt, H. (2004). Contextualizing ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web, 1*(4), 325–343.

Brézillon, P. (1999). Context in problem solving: A survey. *The Knowledge Engineering Review, 14*, 47–48.

Brumitt, B., Meyers, B., Krumm, J., Kern, A., & Shafer, S. (2000). *EasyLiving: Technologies for intelligent environments*. *Lecture Notes in Computer Science 1927*. Springer-Verlag, pp. 19–29.

Calvanese, D. (1996). Reasoning with inclusion axioms in description logics: Algorithms and complexity. In *Proceedings of the 12th European conference of artificial intelligence (ECAI'96)* (pp. 303–307).

Chen, H., Finin, T., & Joshi, A. (2005). The SOUPA ontology for pervasive computing. *Ontologies for agents: Theory and experiences. Whitestein series in software agent technologies*. Springer.

Crane, D., Pascarello, E., & James, D. (2006). AJAX in action. *Manning*.

Delgado, M., Gómez-Romero, J., Magaña, P., & Pérez-Pérez, R. (2005). A flexible architecture for distributed knowledge based systems with nomadic access through handheld devices. *Expert Systems with Applications, 29*(4), 965–975.

Dertouzos, M. (1999). The future of computing. *Scientific American, 281*(2), 36–47.

Dey, A. K., Abowd, G. D., & Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human–Computer Interaction, 16*(2–4), 97–166.

Dey, A., & Abowd, G. (2000). Towards a better understanding of context and context-awareness. In *CHI 2000 workshop on the what, who, where, when, and how of context-awareness*.

Dudney, B., Lehr, J., Willis, B., & Mattingly, L. (2004). Mastering JavaServer faces. John Wiley & Sons.

Eppler, M., & Mengis, J. (2004). The concept of information overload: A review of literature from organization science, accounting, marketing, MIS, and related disciplines. *The Information Society, 20*(5), 325–344.

Farhoomand, A. F., & Drury, D. H. (2002). Managerial information overload. *Communications of the ACM, 45*(10), 127–131.

Garlan, D., Siewiorek, D., Smailagic, A., & Steenkiste, P. (2002). Project Aura: Toward distraction-free pervasive computing. *IEEE Pervasive Computing, 1*(2), 22–31.

Guha, R., McCool, R., & Fikes, R. (2004). *Contexts for the semantic web*. *Lecture Notes in Computer Science 3298*. Springer-Verlag, pp. 32–46.

Gu, T., Pung, H., & Zhang, D. (2005). A service-oriented middleware for building context-aware services. *Journal of Network and Computer Applications, 28*(1), 1–18.

Haarslev, V., & Moller, R. (2001). Description of the Racer system and its applications. In *Proceedings of the international workshop on description logics (DL-2001)* (pp. 131–141).

Johanson, B., Fox, A., & Winograd, T. (2002). The interactive workspaces project: Experiences with ubiquitous computing rooms. *IEEE Pervasive Computing, 1*(2), 67–74.

Khedr, M., & Karmouch, A. (2005). ACAI: Agent-based context-aware infrastructure for spontaneous applications. *Journal of Network and Computer Applications, 28*(1), 19–44.

Knublauch, H., Fergerson, R., Noy, N. & Musen, M. (2004). The Protégé OWL plugin: An open development environment for semantic web applications. In *Proceedings of the Third International Semantic Web Conference (ISWC2004)* (pp. 229–243).

Kwon, O., Shin, J., & Kim, S. (2006). Context-aware multi-agent approach to pervasive negotiation support systems. *Expert Systems with Applications, 31*(2), 275–285.

Lassila, O., & Khushraj, D. (2005). Contextualizing applications via semantic middleware. In *The second annual international conference on mobile and ubiquitous systems: Networking and services, 2005. MobiQuitous 2005* (pp. 183–189).

Masuoka, R., Labrou, Y., Parsia, B., & Sirin, E. (2003). Ontology-enabled pervasive computing applications. *IEEE Intelligent Systems, 18*(5), 68–72.

McGuiness, D. L., & van Harmelen, F. (2004). OWL web ontology language overview. http://www.w3.org/TR/owl-features/.

Ranganathan, A., McGrath, R. E., Campbell, R. H., & Mickunas, M. D. (2003). Use of ontologies in a pervasive computing environment. *Knowledge Engineering Review, 18*(3), 209–220.

Satyanarayanan, M. (2001). Pervasive computing: Vision and challenges. *IEEE Personal Communications, 8*(4), 10–17.

Schmidt, A., Beigl, M., & Gellersen, H. (1999). There is more to context than location. *Computers and Graphics, 23*(6), 893–901.

Sirin, E., Parsia, B., Cuenca-Grau, B., Kalyanpur, A., & Katz, Y. (2005). Pellet: A practical OWL-DL reasoner. Technical Report 2005-68, University of Maryland Institute for Advanced Computer Studies (UMIACS).

Strang, T., & Linnhoff-Popien, C. (2004). A context modeling survey. In *Workshop on advanced context modelling, reasoning and management associated with the sixth international conference on ubiquitous computing (UbiComp 2004)*.

Stuckenschmidt, H. (2006). Toward multi-viewpoint reasoning with OWL ontologies. *The Semantic Web: Research and Applications*, 259–272.

Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data and Knowledge Engineering, 25*(1–2), 161–197.

ten Teije, A., Marcos, M., Balser, M., van Croonenborg, J., Duelli, C., van Harmelen, F., et al. (2006). Improving medical protocols by formal methods. *Artificial Intelligence in Medicine, 36*(3), 193–209.

Weiser, M. (1999). The computer for the 21st century. *Mobile Computing and Communications Review, 3*(3), 3–11.