Reasoning with Semantic Web Technologies in Ubiquitous Computing Environment

WenYing Guo
ZheJiang GongShang University, Hangzhou, China
gwy@mail.zjgsu.edu.cn

***Abstract* —The vast amounts of data about people, things and the environment will require new ways of handling, searching and presenting information. New applications will increasingly be able to respond to data coming from the real world and take appropriate action without human intervention. Ubiquitous computing technologies are believed to be the third wave in computing by building a global computing environment where seamless and invisible access to computing resources. The Semantic Web is specially a web of machinereadable information whose meaning is well defined by standards so that both people and computers can understand. This paper has present a attempt to apply Semantic Web technologies to ubiquitous computing, It takes family ontology as an example to show how Semantic Reasoning System (SRS) can make the system automatic by improving the interoperability between systems, applications, and information, It uses OWL for defining a domain family ontology, then set up rules in JESS engine, finally run reasoning by Racer.**

***Index Terms*—Ubiquitous computing, Semantic Web, ontology, reasoning,**

I. INTRODUCTION

The real world network of data will allow humans to be better informed and make better decisions, it will also mean that machines can make better decisions too. However, with the rapid development of the Internet and web technologies, the power of the network increases exponentially by the number of computers connected to it. Every computer added to the network both uses it as a resource while adding resources in a spiral of increasing value and choice. The vast amounts of data about people, things and the environment will require new ways of handling, searching and presenting information. New applications will increasingly be able to respond to data coming from the real world and take appropriate action without human intervention. Increasingly computers will be making decisions on our behalf.

Ubiquitous computing technologies are believed will become more useful if they could undertake tasks on behalf of the user, rather than forcing the user to do essentially everything himself. It is about a shift to human centered computing, where technology is no longer a barrier, but works for us, adapting to our needs and preferences and remaining in the background until required. This implies a change in our daily life to a much more natural way of interacting and using the power of networked computing systems which will be connected not just to the Internet or other computers, but to places, people, everyday objects and things in the world around us.

Shareing information is key to enabling the ubiquitous computing. The development of a semantic web is one solution. The semantic web makes communication between machines possible that means machines and systems are able to interrogate other machines and systems. It uses ontologies and schemas to separate data from how it is presented and give it a structure that enables information on the web to be retrieved, interpreted and shared by machines intelligent agents rather than just humans.

This paper discusses the possible application of Semantic Web technologies to ubiquitous computing. It is motivated by the need for better automation of user's tasks, In particular, It demonstrates that Semantic Web technologies are particularly well suited to rich, flexible representation of various policies. The paper is organized as follows. Section 2 describes the motivation and focuses on related works. Section 3 describes our Semantic Reasoning System (SRS) framework. Section 4 shows the implementation of SRS. Section 5 concludes the paper.

II. Motivation and Related works

Existing methodologies for implementing a Ubiquitous computing environment is by using smart devices, which have some processing power and are specialized in a set of specific tasks. The advantage of this is their ability to communicate with each other by building and storing contextual information used by the Pervasive environment to offer services based on the stored information. However, current devices are costly and thus it is difficult to replace all current devices with smart devices to implement pervasive computing environments. Also, smart devices need to have functionality beyond what they are expected to do because they are integral to the environment.

Our solution eliminates the need for smart devices is by using the Semantic Web to build dynamic context models and reasoning models, as a user moves from one environment to another. He/She can achieve dynamic building of contexts by sharing knowledge and context information between local Pervasive environments through the Semantic Web. The user also can get the implicit information by the reasoning modules. This approach will be helpful to quickly implement Ubiquitous computing since the user can use currently available resources and do not need specialized devices.

The vision of Semantic Web proposed by Tim Berners-Lee et al. (2001) is "The Semantic Web is an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation". The Semantic Web can be seen as a huge repository of Web data, like database as a repository of structured data. The Semantic Web techniques have proven useful in providing richer descriptions for Web resources, and consequently they can also applied to describing functionality: Semantic Web Services appear to be an appropriate paradigm to be applied in representing the functionality of ubiquitous computing devices. Virtual and physical functions can be abstracted as services, providing a uniform view of all different kinds of functionality. Realization of this is contingent on the continuing emergence of suitable ontologies for modeling ubiquitous computing environments.

Semantic Web technologies represent a potential for this qualitatively stronger interoperability as compared to the traditional standards-based approach. With the Semantic Web approach, it is possible for agents to “learn” new vocabularies and via reasoning make meaningful use of them. Furthermore, in addition to current notions of device and application interoperability, the Semantic Web represents interoperability at the level of the information itself.

Ontologies usually are used as a formalism to describe knowledge and information in a way that can be shared on the web is becoming common. Adoption of the standard for the ontology web language (OWL) is propelling this trend toward large scale application in different domains. However, the utility of the ontologies is limited by the processing mechanisms that are smoothly integrated with this form of representation. Therefore there is an effort on the way to formalize the logic layer for ontologies. The semantic web rule language (SWRL) is proposed as an important step in this direction, building on the experience of the previous work on RuleML. Eventually the availability of standardized rule language for the semantic web will make it possible to use both ontologies and rules as a basis for innovative applications that are connected to the semantic web. The understanding of capabilities and implications of this combination will be essential for successful deployment and adoption of these technologies.

III. SRS Framework

Our Semantic Reasoning System (SRS) consists of the following four parts:

[1] ontologies architecture: representing OWL concepts as Jess Knowledge,

[2] SWRL rule: establishing the rules to represent the dependencies between the relationships

[3] Jess Engine: Importing the SWRL rules into Jess engine and executing Jess Rules and Updating an OWL Knowledge Base

[4] reasoning module implement

A. Ontology Architecture

A vocabulary of concepts for an information system described in the scenario requires definitions about the relationships between objects of discourse and their attributes. The W3C defines two standards that can be used to design an ontology:

RDF Schema: RDF Schema (RDFS) allows the engineer of an ontology to create hierarchies of concepts (classes) and also hierarchies of attributes which specify a class.

Web Ontology Language (OWL): Because of the upward compatability of the Semantic Web Architecture, all constructs of RDFS can also be used in an OWL Ontology.

The W3C divides OWL into three syntax classes: OWL Lite, OWL DL and OWL Full. OWL DL is suited to be read by description logic reasoner.

Because of the larger complexity of constructs, We choose OWL to implement the needed ontologies. OWL allows to set property restrictions and to indicate whether a property is transitive, symmetric, functional or inverse to another property. Contrary to RDFS, the value of a property can be allocated with an instance. This is a benefit in terms of defining properties that describe relationships between classes.

The family ontology (FO) in our SRS is based on the standard conceptual reference model developed by Christine Golbrelch[11]. It provides definitions and a formal structure for describing the implicit and explicit concepts and relationships used in family heritage documentation. We draw out some classification and relevant properties from its models, then re-establish some core concepts, instances and relationship, set up our own family ontology FO，In our FO, We include an OWL ontology representing the family usual relationships and a SWRL rule base representing the dependencies between those relationships.

We choose Protege as our pntology editor, which supports knowledge acquisition and knowledge base development. It is a powerful development and knowledge-modeling tool with an open architecture.

The FO architecture is shown in Fig. 1.

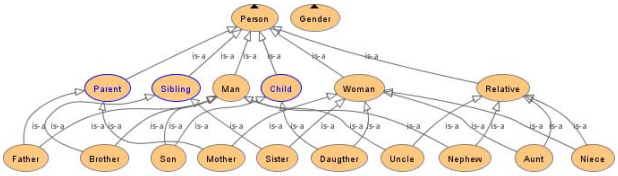

Figure 1. FO describs the core concepts and attributes in the family heritage documents

In FO, We have following definitions:

Core concepts ＝{Thing, Person, Gender, Parent, Sibling, Woman, Child, Man, Relative, Mother, Father, Sister, Brother, Daughter, Aunt, Niece, Son, Uncle, Nephew}。

Attributes＝{hasAunt, hasChild{hasSon,hasDaughter}, hasConsort, hasNephew, hasNiece, hasParent{hasMother,hasFather}, hasSex, hasSibling{hasBrother,hasSister}, hasUncle}

Members ＝{Smith, Alice, Betty, Charles, Doris, Eve, Anna, George, Michael}

OWL for representing the family ontology object is shown below:

```
…
<owl：Class rdf：about="#Relative">
    <rdfs：subClassOf rdf：resource="#Person"/>
    <owl：equivalentClass>
      <owl：Class>
        <owl：unionOf rdf：parseType="Collection">
          <owl：Class rdf：ID="Child"/>
          <owl：Class rdf：about="#Parent"/>
          <owl：Class rdf：about="#Aunt"/>
          <owl：Class rdf：ID="Nephew"/>
          <owl：Class rdf：ID="Niece"/>
          <owl：Class rdf：about="#Uncle"/>
          <owl：Class rdf：ID="Sibling"/>
        </owl：unionOf>
      </owl：Class>
    </owl：equivalentClass>
  </owl：Class>
```

B. Semantic Web Rule Language

Being a combination of OWL and Horn Logic, SWRL can be used to define rules. We choose SWRL as our rule

language of our Semantic Web and write rules expressed in terms of OWL concepts to reason about OWL individuals. The rules can be used to infer new knowledge from existing OWL knowledge bases.

SWRL extends OWL DL abstract syntax by a further axiom:

"axiom ::= rule where:
rule::='Implies('[URIreference]{annotation}
antecedent consequent ')'
antecedent ::='Antecedent(' { atom}')'
consequent ::='Consequent('{atom}')'
atom ::= description '(' i-object ')'
|dataRange '(' d-object i-object ')'
|individualvaluedPropertyID'('i-object i-object ')'
|sameAs '(' i-object i-object ')'
|differentFrom '(' i-object i-object ')'
|builtIn'(' builtIn builtinID{ d-object} ')'
builtinID ::=URIreference

A rule is a combination of an antecedent and a consequent. A rule typically claims that if the antecedent is true then the consequent has to be true. An antecedent/consequent is an assertion e.g. parent(x,y) which means x is a parent of y.

(Antecedent(parent(x,y)),Consequent(older(x,y)))

means, if x is a parent of y, then x is older than y. The combination of the hasSon and hasSister properties implies the hasDaugher property.

In SWRL the rule would be written like:

Implies

(Antecedent(hasSon(I-variable(x1) I-variable(x2))
hasSister(I-variable(x2) I-variable(x3)))
Consequent(hasDaughter(I-variable(x1)
I-variable(x3))))

Fig. 2 shows the Jess representation of SWRL rules in the Protege SWRL Editor.

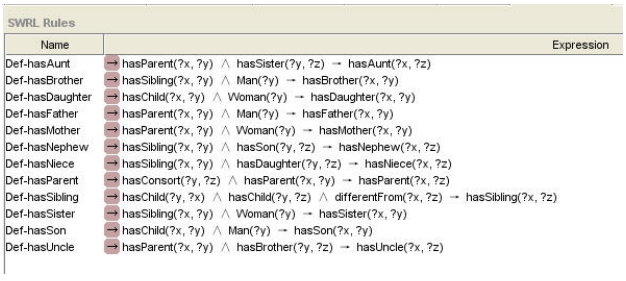

Figure 2. Jess Rules Tab in the Protege SWRL Editor.

C. *Abox and Tbox*

Knowledge base declarations include concept axioms and role declarations for the TBox and the assertions for the ABox. The TBox object and the ABox object must exist before the functions for knowledge base declarations can be used. A knowledge base is just a tuple consisting of a TBox and an associated ABox.

The Tbox in our SRS Framework is shown in Fig. 3.

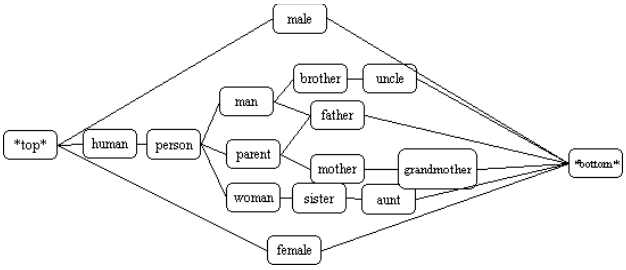

Figure 3. Family T-box

TBOX coding:

;;; initialize the T-box "family"

(signature:atomic-concepts (person human female male woman man parent mother father grandmother aunt uncle sister brother)

(signature

: atomic-concepts (person human female male woman man parent mother father grandmother aunt uncle sister brother)

: roles ((has-child : parent has-descendant)

(has-descendant : transitive t)

(has-sibling)

(has-sister : parent has-sibling)

(has-brother : parent has-sibling)

(has-gender : feature t)))

;;; Domain & range restrictions for roles

(implies *top* (all has-child person))

(implies (some has-child *top*) parent)

(implies (some has-sibling *top*) (or sister brother))

(implies *top* (all has-sibling (or sister brother)))

(implies *top* (all has-sister (some has-gender female)))

(implies *top* (all has-brother (some has-gender male)))

;;; Axioms for relating concept names

(implies person (and human (some has-gender (or female male))))

(disjoint female male)

(implies woman (and person (some has-gender female)))

(implies man (and person (some has-gender male)))

(equivalent parent (and person (some has-child person)))

(equivalent mother (and woman parent))

(equivalent father (and man parent))

(equivalent grandmother (and mother (some has-child (some has- child person))))

(equivalent aunt (and woman (some has-sibling parent)))

(equivalent uncle (and man (some has-sibling parent)))

(equivalent brother (and man (some has-sibling person)))

(equivalent sister (and woman (some has-sibling person)))

In the Abox. We have two family: (Alice's family and Charles's family) shown in Fig. 4 and Fig. 5 respectively.

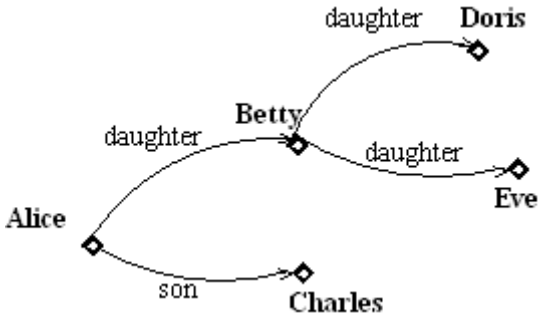

Figure 4. Alice's Family

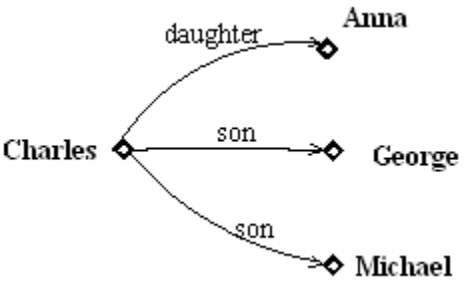

Figure 5. Charles's Family

D. Integrating SWRL Editor and the Jess Rule Engine

Integration with external systems, such as problem solvers, is becoming increasingly important for ontology development and knowledge-modeling tools. We manage our FO Protege ontologies and knowledge by Integrating SWRL Editor and the Jess Rule Engine. The Jess engine running inside the Protégé framework is the basis for the JessTab integration model. Because Protégé and Jess are implemented in Java, we can run them together in a single Java virtual machine. This approach lets us use Jess as an interactive tool for manipulating Protégé ontologies and knowledge bases. Furthermore, we can propagate changes in Protégé to Jess.

A core component of Protégé integration is the mapping mechanism. In the Protégé frame model, classes, slots, and facets are themselves instances. So, it is sufficient to map Protégé instances to facts. This mapping approach fits well with both the Protégé and Jess models. This approach makes it possible to mark classes in the ontology for mapping to Jess facts. By using the expression (mapclass <class-name>), we can map all instances of the specified class to Jess facts.

Jess is a reimplementation of a subset of Clips in Java. Jess implements some additional functionality not provided by Clips. Like Clips, reasoning in Jess is based on a list of known facts and a set of rules that try to match on these facts in its fact base.

Interaction between the SWRL Editor and the Jess rule engine is user-driven. The user controls when OWL knowledge and SWRL rules are transferred to Jess, when inference is performed using those knowledge and rules, and when the resulting Jess facts are transferred back to the OWL Plugin as OWL knowledge.

E. Executing Jess Rules and Updating OWL Knowledge Base

Once the relevant OWL concepts and SWRL rules have been represented in Jess, the Jess execution engine can perform inference. As rules fire, new Jess facts are inserted into the fact base. Those facts can be used in further inference. When the inference process completes, these facts must be transformed into OWL knowledge,

The Jess system consists of a rule base, a fact base, and an execution engine. The execution engine matches facts in the fact base with rules in the rule base. Fig. 6 shows that the rules can assert new facts and put them in the fact base.

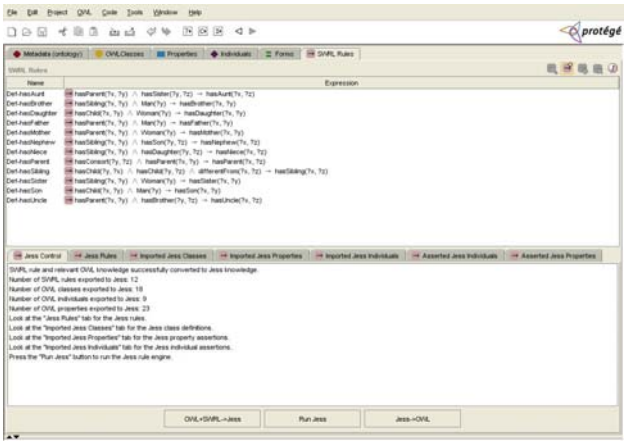

Figure 6. Run JESS engine

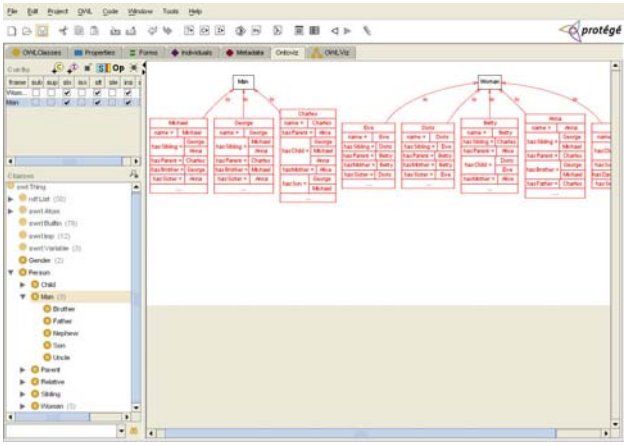

Figure 8. Performance

F. *Reasoning Engine*

The order of axioms and assertions does not matter because forward references can be handled by RACER.

The macros for knowledge base declarations add the concept axioms and role declarations to the (current-tbox) and the assertions to the (current-abox).

A major task of SRS Reasoner is discovering the implicit information that OWL may not define. Each web resource contains metadata which describes the content of the web resource and which is noted in RDF. The Reasoner has to send queries to web resources to detect whether the content is suitable to a given interest or not.

We choose Racer as our reasoning engine. After running Racer, we can discover the implicit relationship between the family members. (shown in Fig. 7).

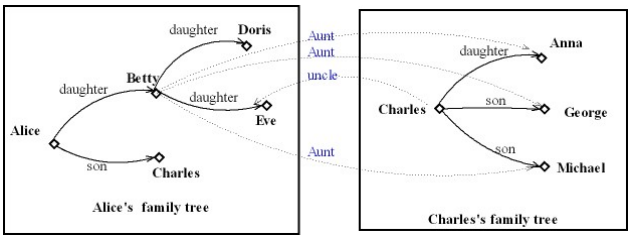

Figure 7. Mapping between two Families

IV. PERFORMANCE

The reasoning module is fully implemented with all features described in the previous section and shown in Fig. 8 below.

V. CONCLUSION AND FUTURE WORK

This paper takes family ontology as an example to show how the SRS can help to discover the implicit relationship between the family members by reasoning. The application will not be limited to family domain. As SRS can “think ”, it can apply to the varous different fields, espcailly ubiquitous computing.

In this paper, we focus more on the Semantic Web technology. In the near term, our research will be how to apply this Semantic Web technology to the ubiquitous computing environment.

VI. REFERENCES

[1] Sachin Singh, Sushil Puradkar, Yugyung Lee, "Ubiquitous computing: connecting Pervasive computing through Semantic Web",Information Systems and E- Business Management,Springer Berlin /Heidelberg ,Volume 4, Number 4 , 2006.10, pp. 421-439

[2] David Martin, Mark Burstein and Drew McDermott etc. "Bringing Semantics to Web Services with OWL-S", World Wide Web, Volume 10, Number 3, 2007.9, pp223-227

[3] Yalin Yarimagan , Asuman Dogac,"Semantics based customization of UBL document schemas", Distrib Parallel Databases, 2007 (22), pp.107-131

[4] Michael J. Shaw, David M. Gardner, Howard Thomas "Research opportunities in electronic commerce", Department of Business Administration, College of Commerce, University', Decision Support Systems, 1997, pp. 149-156

[5] David Ley, Becta, "Ubiquitous Computing", emerging technologie,Volume 2 (2007), pp. 64-79

[6] Ora Lassila, "Applying Semantic Web in Mobile and Ubiquitous Computing: Will Policy-Awareness Help?", http://www.csee.umbc.edu/swpw/papers/lassila.pdf

[7] Marek Hatala, Ron Wakkary, Leila Kalantari,"Rules and ontologies in support of real-time ubiquitous application", Web Semantics: Science, Services and Agents on the World Wide Web, 2005, pp.5-22

[8] Fabian Abel, Jan Brase, "Using Semantic Web Technologies for context- aware Information Providing to Mobile Devices", http://wetice.jpl.nasa,gov/wetice03/presentations/ Ahn.pdf

[9] Martin O'Connor, Holger Knublauch, Samson Tu,etc., "Rule System Interoperability on the Semantic Web with SWRL", ISWC 2005, LNCS 3729,2005, pp. 974 - 986,

[10] Wen-ying Guo, De-ren Chen, and Xiao-lin Zheng "A Semantic Web Approach to 'Request for Quote' in E-Commerce", APWeb Workshops 2006, LNCS 3842, 2006, pp.885-888

[11] Alessandra A. Macedo, Larcio Baldochi Jr. "Automatically linking live experiences captured with a ubiquitous infrastructure", Business Media, LLC 2007

[12] Kalle Lyytinen, Youngjin Yoo, "Designing and implementing effectively high impact ubiquitous computing environments", Springer-Verlag 2006, pp395-397

[13] http://protege.cim3.net/file/pub/ontologies/family.swrl.owl/family.swrl.owl

WenYing Guo is currently an Associate Professor of Computer Information Engineering and Technology at ZheJiang GongShang University, Hangzhou, China. She received her PhD. in the Computer Science from ZheJiang University and got her MSc degree from Northwestern polytechnology university, Xian. Her research over the last ten years has focused mainly on Semantic Web technology, the application of e-Learning and Electronic Commerce.