Ontology-based User Context Modeling for Ubiquitous Robots

Jung Hwa Choi
Soongsil University, Seoul, Korea
cjh79@ailab.ssu.ac.kr

Il Hong Suh
Hanyang University, Seoul, Korea
ihsuh@hanyang.ac.kr

Young Tack Park
Soongsil University, Seoul, Korea
park@comp.ssu.ac.kr

Abstract − Ubiquitous robots need user modeling for providing personalized services. With a user model, ubiquitous robots can build more accurate context model, therefore, we provide appropriate services to users based on the context model. In this paper, we propose a context model with ontology based on conceptual dependency. This model is supposed to be used for modeling observed user behaviors. In order to model user behavior, we designed context ontology based on conceptual dependency. Context modeler monitors user behavior and generate ontology instances to represent the behavior. We model user behavior using temporal concepts to enable temporal reasoning for context model. After a temporal context model is generated, we employ ontology reasoner to infer more high level contexts for ubiquitous.

Keywords − Ubiquitous Robots, Ontology, Context Modeling, Temporal Reasoning, Conceptual Dependency

1. Introduction

Context awareness is becoming an essential feature of ubiquitous robots that assist everyday lives of user. For qualified services, richer contexts are very important. Contexts are any relevant information for ubiquitous robots to provide better services to users. Therefore, contexts consist of various components to represent environments (i.e., devices, associated persons, spaces) to name a few. Context-aware systems can extract, interpret, and use context information and adapt its functionality to the current context of use. In this paper, we propose a method using context ontology to represent contexts. Context ontology uses hierarchical and property information to represent relevant context information. Many context-aware systems employ context ontology for better representing high level contexts. In ubiquitous environments, it is necessary to model user behavior in a more accurate way [1, 2].

For this purpose, we propose an ontology-based context modeling method for user behavior. Ontology is used to represent a user behavior in ubiquitous environments. We use Web Ontology Language OWL [3] to infer semantic contexts. Ontologies provide richer explicit semantic meanings of observed events. To represent observed events, we build diverse classes such as “Action”, “Actor”, “Object”, “Space”, “Coordinate”, etc. Temporal ontology is used to describe temporal events and temporal relations between events. We develop an ontology based on primitives of conceptual dependency which was developed by Schank [4] for representing the meaning contained in sentences. This technique finds the structure and the meaning of a sentence and is useful to represent sentences when there is not a strict grammar associated with the sentences. We learned that primitives of conceptual dependency can be used for model user behavior in ubiquitous environments. Based on this observation, we build a context ontology using OWL-DL to model conceptual dependency primitives.

In ubiquitous environment, each action is observed by various sensors. Such an observed action can be transformed into ontology instances using predefined ontology schema. Since ontology schema encodes much background information, observed ontology instances can be expanded to represent high level contextual information inherited from context ontology schema. In this paper, these contextual ontology instances are used by ontology inference engine to infer personalized contextual information. For this purpose, ontology reasoner reasons about subsumption relations between contextual events and infers new contextual information using heuristic knowledge using SWRL [5]. We can perform a variety of upper-level knowledge representation based on such core research and sharing the knowledge through ontology predicts the reuse of the result.

The reminder of the paper is organized as follows. In Section 2, we introduce the ontology-based context modeling framework for ubiquitous robots: the robot ontologies and ontology reasoners. In Section 3, we design with the key features of ontology concepts and properties for context awareness. In Section 4, a prototype system design of the ontology-based user context model in ubiquitous environment. Lastly, some concluding comments and future works are given in Section 5.

2. Ontology-based Context Modeling Framework

We propose a framework using ontology for modeling observed user behavior. As shown in Fig. 1, there are three kinds of ontologies; *context ontology*, *temporal ontology*, and *SWRL rule ontology*. *Context ontology* represents conceptual dependency using OWL-DL. The ontology is used to model observed user behavior. In ubiquitous environment, all the behavior has temporal characteristics. In order to represent such features, we employ temporal ontology to describe observed user behavior’s temporal information. Such temporal information is useful for reason about user behavior in temporal way. For better model of user behavior, heuristic rules are used to infer high level user behavior based on *context ontology*. We employ SWRL rules to represent such heuristic information.

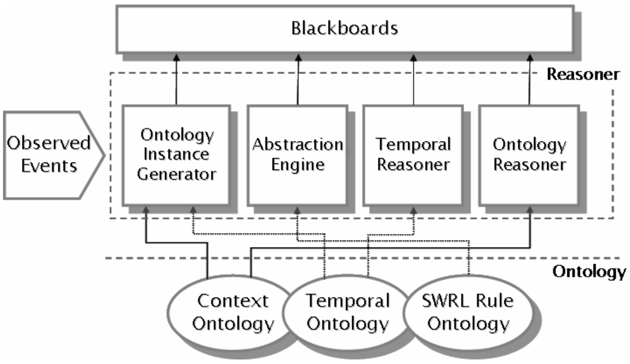

Fig. 1. Ontology-based Context Modeling Framework

Context means a person and the locations, identities and changes of objects near the person. Further, context awareness refers to the time that robots can adapt to the objects around the location of the user as it changes along with time. We define the concept of context awareness as the logical result deductively inferred from the context generated by the instance of pre-defined ontology from what we are aware of in voice and vision modules. We further define context as the information that describes that particular change of the object, when such object of the user's behavior changes along with the change of his location per time change.

When a ubiquitous robot sensors a user behavior, the behavior is map into an ontology instance. An ontology instance is generated by the *ontology instance generator* based on *context ontology*. For example, when the system observes a user to move from a space to another space, it transforms the observed behavior into an ontology instance as follows: location(Alice, Zone-1) location(Alice, Zone-2). All the ontology instances contain temporal information since *context ontology* is a subclass of temporal entity. *Abstraction engine* is used to abstract ontology instances into a high-level user model. We use SWRL rules for abstractions. Heuristic rules can be encoded using SWRL rules. In order to process such SWRL rules, JESS [6] inference engine is now used for processing SWRL rules.

All the observed user behaviors become temporal ontology instances and these are partial ordered by temporal reasoner. We employ James Allen's temporal reasoning method [7, 8]. Each observed user behavior is partial order using temporal reasoner. *Ontology reasoner* is used to identify taxonomy relation between observed user behavior. OWL-DL semantics allow ontology reasoner identify hierarchical relations between ontology instances. We employ OWL-DL reasoners to infer taxonomy relation between observed user behavior.

2.1 The Robot Ontology

This paper suggests three kinds of ontology (i.e., Context ontology, Temporal ontology, SWRL rules ontology) so that we can express robot's knowledge framework per time flow using Web Ontology Language (OWL).

2.1.1 Context Ontology. OWL-DL model of *context ontology* define various components to represent environments. It can be decomposed into core classes: Action, Actor, Object, Space, Coordinate.

Action. Actions that can be observed by actors are defined including the moving objects and time concepts. Actions in "Action" class are defined based on the conceptual dependence theory by Roger Schank [4]. Using the theory, we can infer movements of person and objects and generalized actions of the person through the relation among the objects. The instance of "Action" class is defined by the instances of other contexts (i.e., "Actor", "Object", "Space", "Coordinate" class) and the knowledge by inference engine.

Actor. The "Actor" class defines the features of the actor.

Object. The "Object" class defines every object that exists in ubiquitous environment. For instance, it classifies an object named appliance(i.e., computer, coffee marker, etc), furniture(i.e., chair, cabinet, etc), and food(i.e., rice, ice-cream, etc) in space and defines its features.

Space. The "Space" class represents spatial information. As it is important for inferring the movements of an actor and objects in a particular space, the space must be characterized and classified as much detail as possible with smallest possible zones.

Coordinate. The "Coordinate" class shows the coordinates of an actor and objects within the classified smallest space. The coordinates among objects are needed to express the upper-level knowledge the person is thinking of. For example, using the coordinates of a glass and a bowl, we can inter the context, 'The glass is on the right of the bowl."

2.1.2 Temporal Ontology. Time information can be obtained regardless of user, equipment or location and we defined time ontology referring to *OWL-time ontology* [9, 10]. The *OWL -time ontology* formalizes time expressing it with concepts and properties. For this ontology, thirteen interval relations for defining relations among actions proposed by James Allen to formalize time were referenced. *Temporal ontology* makes it possible to obtain knowledge not defined in ontology by inferring relations among actions using interval relation defined by Allen (Fig. 2).

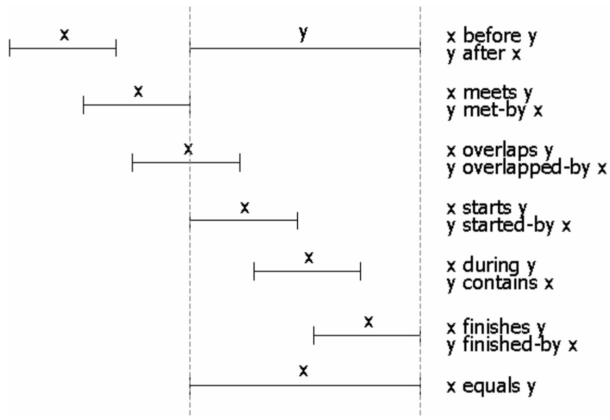

Fig. 2. The Definition of Interval Relation by Allen

2.1.3 SWRL Rules Ontology. It is relations among logically reasonable contexts and infers new contexts. This ontology defines the rule as the concepts defined in context ontology and temporal ontology, which are relations between class and properties. The following is an example of SWRL rules. Heuristic rules can be encoded using SWRL rules.

$$above(?x, ?y) \wedge differentFrom(?x, ?y) \rightarrow on(?x, ?y)$$

2.2 Ontology Reasoner

2.2.1 Ontology Instance Generator. As time information can be obtained regardless of user, object or location, we add time information when defining an observed event as an instance. (i.e., begins, ends). For example, if we define the beginning time with the sensor information of a particular actor as the value of begins and the coordinates of the actor stay unchanged for a given time, the finishing time of the actor's action is stored as the value of ends. At this time, the relations of time among actions are not considered.

2.2.2 Abstraction Engine. This engine defines relations between classes and properties defined in ontologies as a rule and infers new class, properties and instances, For example, it includes a SWRL rule like followings:

$$right(?x, ?y) \wedge left(?y, ?x) \wedge differentFrom(?x, ?y)$$
$$\rightarrow beside(?x, ?y)$$

Above rule lets you infer new knowledge that "Two objects are side by side." when you learn of context information that the right and the left between two concepts and two entities are different.

2.2.3 Temporal Reasoner. This infers the value of temporal ontology properties. In other words, this engine defines relations per time among actions. Knowledge drawn by *Temporal Reasoner* infers Interval relation defined by Allen using *Temporal reasoner*. For example, an action before the other ($x\ before\ y$), an action after the other ($x\ after\ y$), and actions whose starts and ends overlap ($x\ overlaps\ y$) can be inferred.

2.2.4 Ontology Reasoner. This infers new knowledge using Axiom provided by ontology. For instance, if we define that the domain of the property named "hasAction" is "Actor" class and that "Human" class must have "hasAction" property with at least one "Action" class as a range, the relation of "Human" class's inclusion in "Actor" class is defined ($Human \subseteq Actor$). We therefore can infer that the instance defined in "Human" class is the Actor.

3. Context Awareness based on Robot Ontology

This system recognizes semantic contexts based on proposed *temporal ontology*. It enables us to infer actions occurred in the given contexts through the location and relation of the actor and the object per time and the relation among objects.

3.1 The Function of "Action" Class

Based on "Action" class, the ontology of this system infers an action through relations among relevant instances. The final result inferred by the proposed system turns into instances of "Action" class through four inference engines. Referring to conceptual dependency theory by Roger Schank, "Action" class is classified as "ATRANS", "GRASP", "PROPEL", "INGEST" and "PTRANS" class. Such actions can be inferred through relations among context information like actor, object, location and time. An action occurs with on actor in one location, occurs with more than one object or actor and has begins and ends. This condition is shown as a clear restriction of "Action" class.

Table 1 Actions using Conceptual Dependency Theory

Action	Description
ATRANS	Actor *A* gives object *x* to actor *B*
GRASP	Actor *A* grasps object *x*.
PROPEL	Actor *A* forces physical pressure upon object *x*.
INGEST	Actor *A* eats object *x*
PTRANS	The physical location of actor or object changes.

3.2 The Function of Context Awareness

For the robots to provide intelligent service, it is important for him to analyze the information on ubiquitous environment and on the user and understand the situation implied in the user's order. The robots becomes aware of the user context and provides intelligent service for the user's order using context information like the status of the user, physical environment and analyses of existing information.

We defines 9 relations for context awareness (Table 1). They are inferred using coordinates of "coordinate" class. They are "right", "left", "front", "rear", "above", "below" properties. Each property defines its opposing property using *inverse* Axiom (Table 2). This Axiom makes it possible to infer the opposing relation even when only one relation exists. The other relations like on, beside and enclose are inferred using the *Abstraction Engine* for instance generation (Table 3). This relation enables us to be aware of the environment for the existence of objects using relations in Table 2 and 3.

Table 2 Relation between Objects using Axiom Inference

Relation	Description / Axiom
right	Object *x* is on the right of object *y*
	$inverseOf(right, left)$
left	Object *y* is on the left of object *x*.
	$inverseOf(right, left)$
front	Object *x* is in front of object *y*
	$inverseOf(front, rear)$

rear	Object *y* is on the rear of object *x*.	$inverseOf\ (front, rear)$
above	Object *x* is right above object *y*	$inverseOf\ (above, below)$
below	Object *y* is right below object *x*.	$inverseOf\ (above, below)$

Table 3 Relation between objects using SWRL Rules

Relation	Description / Rule
on	Object *x* is on object *y* $above(?x, ?y) \wedge differentFrom(?x, ?y) \rightarrow on(?x, ?y)$
beside	Object *x* is beside object *y* $right(?x, ?y) \wedge left(?y, ?x) \wedge differentFrom(?x, ?y) \rightarrow beside(?x, ?y)$
encloses	Object *x* surrounds object *y* $on(?x, ?y) \wedge differentFrom(?x, ?y) \wedge beside(?x, ?a) \wedge differentFrom(?x, ?a) \rightarrow encloses(?x, ?y) \wedge encloses(?x, ?a)$

3.3 Context Modeling for Action Awareness

The robot must explore solutions within the knowledge base to understand order of the user's behavior and context and solve problems. The movements of actor or object can be detected if time information is added to awareness information in section 3.2. This information is stored as instances in "Action" class of ontology, which means it is represented as a piece of context information, that is, consented knowledge. This knowledge inferred is stored in ontology, shared and reused.

3.3.1 An Example of PTRANS Knowledge Representation. "PTRANS" shows a person or an object moved. For example, if the person and soft drink coordinates move from the "refrigerator" area to "sofa" area, respectively, and the beginning and ending times of them are same, we can infer new knowledge that "The person moved." (PTRANS) and "The person grasped the soft drink." (GRASP) and additional knowledge that "The person moved, grasping the soft drink." (PTRANS, GRASP) based on the first two pieces of knowledge (Fig. 3)

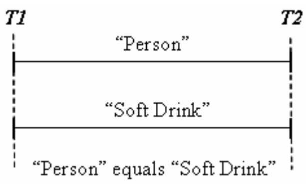

Fig. 3. Relation per Time of Two PTRANS Actions

4. Ontology-based User Context Modeling

4.1 Contexts with the Addition of Time Information

As we can't be aware of invisibility like a movement of person using visional or voice sensor, we become aware of the relations of interacting objects adding time information to the context information stored in ontology. Looking at the movements of entities in the kitchen in Fig. 4 as an example, we find that being aware of the movements of the person is prioritized.

4.1.1 Context. When a person comes in the kitchen in the scenario, instances are generated with his movements, hence continuously changing the coordinate instance values of the person. We divided zones according to the coordinate values of the given space. The context information according to the scenario needed for context awareness of Fig. 3 is as follows:

These predicates are posted on the blackboard in the *semantic instance generator*. The generator applies *abstraction engine* for generating new contexts that may be used by ubiquitous robots.

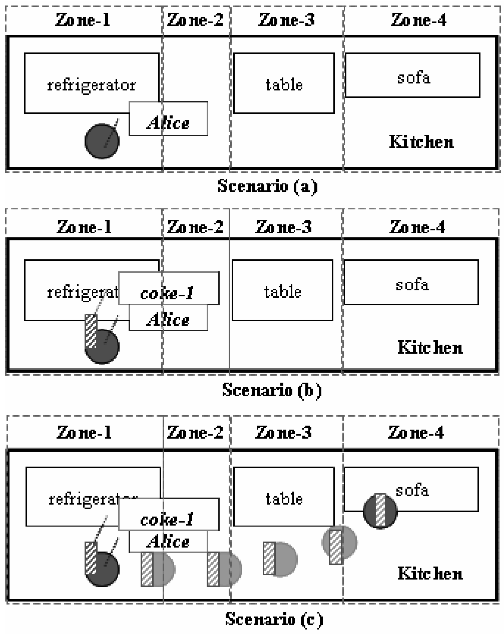

Fig. 4. Scenario

- Context of Scenario (a)
  space(*Kitchen_01*). human(*Alice*).
  hasZone(*Kitchen_01*, *Zone-1*). location(*Alice*, *Zone-1*).
  Appliance(*refrigerator_01*).
  location(*refrigerator_01*, *Zone-1*).

- Context added from Scenario (b)
  ConsumableObject(*Coke-1*). location(*Coke-1*, *Zone-1*).

- Context added from Scenario (c)
  location(*Alice*, *Zone-2*). location(*Alice*, *Zone-3*).
  location(*Alice*, *Zone-4*). location(*Coke-1*, *Zone-2*).
  location(*Coke-1*, Zone-3). location(*Coke-1*, *Zone-4*).

4.1.2 Contexts with the addition of time information. In this paper, context awareness becomes aware of the context adding time information to the real-time generated context information. In the context information generated as such, time information is generated.

- Scenario (a)
  CalendarClockDescription(*T1*). time(*Alice*, *T1*).
- Scenario (b)
  CalendarClockDescription(*T2*).
  time(*Alice*, *T2*). time(*Coke-1*, *T2*).
- The time when Alice of Scenario (c) is zone 4.
  CalendarClockDescription(*T5*).
  time(*Alice*, *T5*). time(*Coke-1*, *T5*).

Property CalendarClockDescription is a class defined in *temporal ontology*. The instance of the class includes date, time day of the week. If instance *T1* of Scenario (a) is defined as “2006-10-22 18:00:00”, we can see from the context that Alice was near the refrigerator in the kitchen at 18, October 22, 2006.

4.2 User Context Modeling

The context information generated in 4.1.is analyzed through *Reasoners* and we become aware of it as context. This information is generated as an action instance via four inference engines. First, we find a group of instances related with instances generated per time through time instance generating engines and infer the beginning and ending times.

4.2.1 Temporal Reasoning Phase 1. The *ontology instance generator* infers the moment values stop continuously coming in as an action. An action has an actor and as many actions as actors are generated. The beginning and ending points of each action are calculated and they become important knowledge in becoming aware of the context by grasping the prior, after and inclusive relations among actions. Fig. 5 schematizes knowledge obtained by running the context information of 4.1. through *ontology instance generator*.

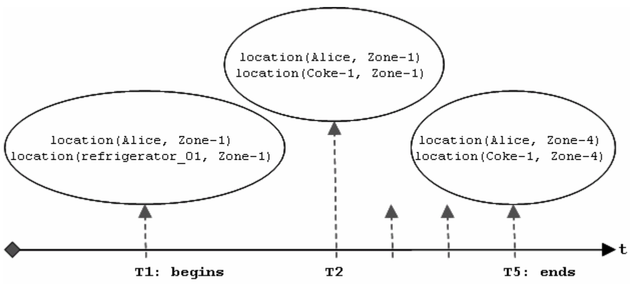

Fig. 5. Temporal Reasoning Phase 1

4.2.2 Temporal Reasoning Phase 2. Instances inferred as an action through *ontology instance generator* extract the characteristics of the action using *abstraction engine*. For example, following rule infers from relation of Action PTRANS. Instances inferred as beginning and ending points have a same actor, but they infer that the actor moved if there locations are different. This example is a rule that infers the first point of movement (from).

$$
\begin{aligned}
& time(?x, ?t1) \wedge location(?x, ?l1) \wedge begins(?t1) \\
& \wedge\, time(?x, ?t2) \wedge location(?x, ?l2) \wedge ends(?t2) \\
& \wedge\, differentFrom(?l1, ?l2) \\
& \rightarrow from(?x, ?l1)
\end{aligned}
$$

4.2.3 Temporal Reasoning Phase 3. *Temporal Reasoner* defines the procedures of actions and relations to be included. It classifies actions, expressing instantly generated time information as interval.

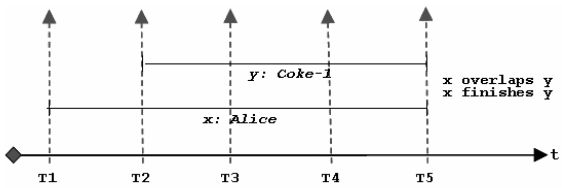

Fig. 6. Temporal Reasoning Phase 3

4.2.4 Temporal Reasoning Phase 4. Finally, *Ontology Reasoner* confirms the kinds of actions through restriction conditions of ontology. The restriction conditions clearly show restrictions of domain classes. This condition id divided into necessary & sufficient and necessary condition. Fig. 7 is a picture of a necessary and sufficient condition of the “PTRANS” class defined in Protégé, an ontology editor [11].

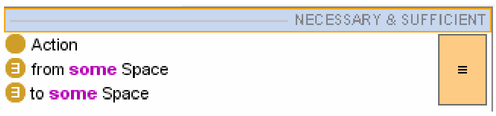

Fig. 7. Restriction of Action

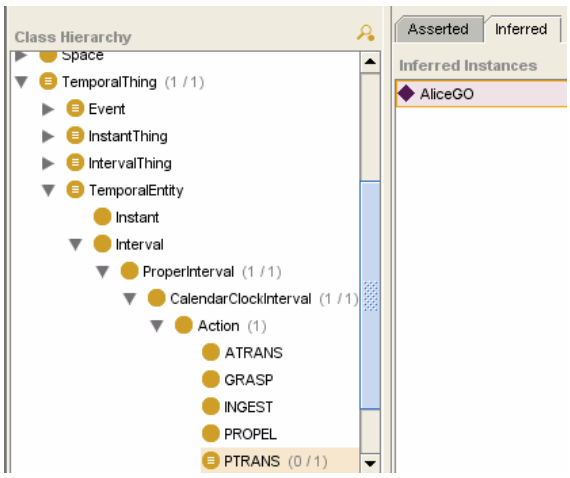

Fig. 8. Temporal Reasoning Phase 4

Fig. 7 is the necessary & sufficient condition of “PTRANS” class. Instances de-fined in “PTRANS” class must meet the condition. Also, instances with “from” and “to” relations are inferred with PTRANS. Fig. 8 shows a picture in which instance “AliceGO” that meets this condition is inferred. Run up to the previous phase of time inference, instance “AliceGO” has been inferred as instances of “Action” class and it infers an action called PTRANS. This information is used as the robot’s knowledge and can be shared and reused as it is generated through ontology.

5. Conclusions and Future Research

We propose a framework using ontology for modeling observed user behavior. The suggested way designs *temporal reasoners* that are aware of behavior of a person after handling and analyzing context information with the addition of time information, referring to *OWL*-time ontology. The suggested system displayed a possibility of generalized representation of knowledge, its sharing and reuse. The System for knowledge representation stores information collected by sensors as ontology-based metadata becomes aware of the context given by the relations of times of generation of stored metadata. This system is efficient in inferring general knowledge within given context. It is also an important technique for an ubiquitous robot to build knowledge frameworks people think of as it satisfies the most fundamental factors the robot must be aware of. This paper performed experiments considering only a few actions. For future study, we will expand this system to make it represent various actions, considering the relation with lower-level information actually collectible by the ubiquitous robots.

Acknowledgements

This paper was performed for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Commerce, Industry and Energy of Korea.

References

[1] Burke, J., Murphy, R., Rogers, E., Lumelsky, V., and Scholtz, j., “Final Report for the DARPA/NSF Interdisciplinary Study on Human-Robot Interaction,” IEEE Transactions on System, Man, and Cybernetics part C: Applications and Reviews, Vol. 34, No. 2, 2004.

[2] J. Hong and J. Landay, “An Infrastructure Approach to Context-Aware Computing,” Human-Computer Interaction, Vol. 16, No. 2-3, pp. 287-303, 2001.

[3] Smith, M., Welty, C., Deborah, L. and McGuinness, D. 2004. OWL Web Ontology Language Guide. W3C Recommendation, http://www.w3.org/TR/2004/REC-owl-guide-20040210/.

[4] R. Schank, and C. Rieger, "Inference and the computer understanding of natural language," Artificial Intelligence, Vol. 5, No. 4, 1974.

[5] Martin J. O'Connor, Holger Knublauch, Samson W. Tu, Benjamin Grossof, Mike Dean, William E. Grosso and Mark A. Musen, “Supporting Rule System Interoperability on the Semantic Web with SWRL,” ISWC, 2005.

[6] Jess. http://herzberg.ca.sandia.gov/jess/.

[7] Allen, J.F. and Ferguson, G.., “Actions and events in interval temporal logic,” In Spatial and Temporal Reasoning, O. Stock, ed., Kluwer, Dordrecht, Netherlands, pp.205-245, 1997.

[8] Allen, J. F. and Kautz, H. A., “A model of naïve temporal reasoning,” In Formal Theories of the Commonsense World. J. R. Hobbs and R. C. Moore, eds., Ablex., pp.251-268, 1985.

[9] Jerry R. Hobbs and Feng Pan. “An Ontology of Time for the Semantic Web,” *ACM Transactions on Asian Language Processing (TALIP): Special issue on Temporal Information Processing*, Vol. 3, No. 1, March 2004, pp. 66-85, 2004.

[10] Hobbs, J.R. and Pustejovsky, J., “Annotating and reasoning about time and events,” *In Proceedings of AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning*, 2003.

[11] schlenoff, C., Washington, R., and Barbera, T., “Experiences in Developing an Intelligent Ground Vehicle (IGV) Ontology in Protégé,” Proceedings of the 7th International Protégé Conference, 2004.