Design of Robot Knowledge Management System: OWL-aware Relational Model Approach

Gi Hyun Lim* Hak Soo Kim** Il Hong Suh*** Sanghoon Lee* Jin Hyun Son**
Hanyang University
{hmetal, shlee}@incorl.hanyang.ac.kr*, {hagsoo, jhson}@cse.hanyang.ac.kr**, ihsuh@hanyang.ac.kr***

Abstract − we propose a robot knowledge management system deal with various data spectrum from sensor level to contextual level and to provide inference mechanisms to cover partial and uncertain sensor data as well as well-formed symbolic information. Robot knowledge is represented with ontology. Moreover, relational model is used not only for data management but also for ontology and its inference.

Keywords − Robot Knowledge, Robot-centered ontology, Relational Model, Knowledge management

1. Introduction

The ubiquitous robot needs high level perceptual tasks - context awareness, SLAM, object recognition and etc for ambient intelligence. Those perceptual tasks utilize sensors such as camera, odometer, sonar and data processing algorithms such as SIFT, Kalman filter and Harris corner. Those data are generated in different time interval [2]. Some perceptual tasks also require fusing those data. In addition, those data are robot-centered, this means that those sensor data are represented as numerical descriptors [1]. In conventional robot system, robot control S/W is optimized for the specific application. And thus, there can be hardly shared control-related knowledge such as data structure, data processing mechanisms and rules. Using ontology makes it easy for intelligent robot to share its knowledge and common concepts.

The ontology in the World Wide Web is used for human-centered ontology in which general taxonomy of objects is given and general concepts described all domains including meta, generic, abstract and philosophical. For the practical application, domain ontology should be applied to the system with upper ontology [3, 8, 9].

OWL is a language for making ontological statements in the semantic web. The knowledge represented with OWL is sharable and growing, and hidden knowledge could be inferred with inference engines such as description logics. In the case of robot, robot has different sensing capacity from human five senses. Therefore, robot needs its own ontology, which is represented by its own sensory organs and data processing mechanisms. For example, recognizing an apple, humans can naturally perceive it by using their own senses such as auditory sense, vision sense, olfactory. But, robot has many difficulties to do so. Object recognition is a big problem in machine vision field. Human-centered ontology is used to deduce some facts with completely symbolized information. However, non-symbolic sensor data in robot domain requires us to develop robot-centered ontology suitable for robot perceptions and activities. By use of robot-centered ontology, it may be possible that robot knowledge can be sharable. In addition, by associating robot-centered ontology with human-centered ontology, robot can provides various services during robot-human interaction.

Robot should handle so many types of data such as task information, context [4] and robot-centered data from various sensor data that data management system. Relation model enable robot to manage those data easily. In addition, it is possible to make use of RDBMS as inference engine, especially when there are so many instances. Thus, the relational model makes it possible to deduce knowledge as well as to manage data using the RDBMS engine. In addition, the Robot Knowledge Management System is blackboard architecture that anybody can access.

In section 2, we discuss various kind of robot knowledge. Section 3 describes OWL-aware relational model. The details of robot knowledge management system are discussed in section 4. And section 5 gives some scenario-based examples for OMRKF. Finally, in section 6, a summary and consideration for future research are provided.

2. Robot Knowledge

Robot knowledge includes all data types used in robot; (1) Ontology instances from symbolic level to sensor level. And, Robot knowledge includes data structures for communication between processors and temporal data for its own usage. Those data are generated at different time intervals and will have different data size

Table I shows various kinds of robot knowledge, their periods of generation, data size and resources that are required. Especially, spatial and temporal context and high level contexts requires all other data, including not only current data but also previous history data.

2.1 Robot-centered Data

In short term memory (STM), robot stores various types of temporal data such as; sensor data, cache data for communication between different processors and robot state data such as location, angle, control command, sensor data and so on. Symbolic data that are abstracted or deduced in STM are stored in LTM where the information is permanently stored. New sensor data that is

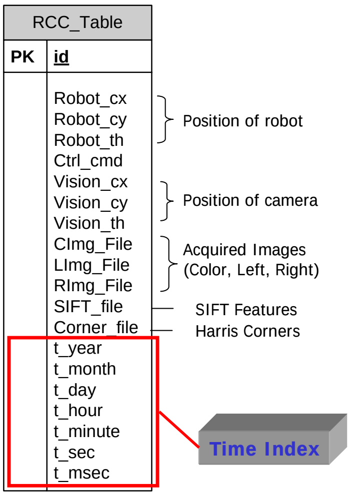


Fig. 1. Robot-centered Data Schema

continuously generated by sensors should be stored in STM, in spite of restrict amount of memory. Therefore, the forgetting mechanism is required to removes STM data not so as to be full. STM is implemented on blackboard architecture that enables any processors to read easily and write new information. Each processor scans the changes to the blackboard, and posts an updated partial solution based on the state of the blackboard whenever its own internal conditions for doing so. These partial solutions cause other processors to update their portions of the solution on the blackboard until eventually an answer is found.

It is required for robot knowledge management system to synchronize not only every sensor data that is generated at each interval but also high level information that uses those sensor data. Moreover, the system should support event-driven methods. If any processor registers an event, the system may invoke the process and pass the data at the time event occurs. Especially event-driven method can be used when some data is missed. Thus, the system can reason those data and make knowledge consistent. Figure 1 shows data schema of robot sensor data and state data, which includes position of robot, control command, camera position, acquired images of stereo camera, SIFT descriptor, Harris corner descriptor. And, it is synchronized with time indices..

2.2 Context

The representative high level data is contextual information. Context is not merely object recognition. Context makes robot perceive entities that is concerned with human, make spatial-temporal context with those entities and derive high level context. Contextual information enable robot to perform its mission properly and again perceive objects and context easily. In case of table at dinner, if there are some object with red color, it may be food or tableware. So it is not required to match all objects that is included in object database but just match task relevant objects; such as "apple," "dish" and something that is possible to be on table. Context gives some clues that make easily perceive. In addition, associating low level data with high level information, robot can be more intelligent. Context also synchronized with time indices and temporal and high level context is generated on demand or at specific event

3. OWL-aware Relational Model

3.1 Web Ontology Language (OWL) & RDBMS

OWL is based on Description logic (DL) and Frame logic (FL) to represent and reason knowledge. Although OWL is initially designed to describe web resources with ontology in Semantic web environment, OWL is used to represent their knowledge in various fields. Especially, OWL is used to represent robot knowledge []. OWL includes concepts that are usually designed as concept hierarchy, and various relationships between concepts. Consequently, OWL enables to represent the knowledge formally.

There are some points that should be considered to use OWL in robot environments. Intelligent robot that resides with human may use enormous knowledge. So, it is required to use not only OWL documents but also Relational Database Manage System (RDBMS) as the method of knowledge representation. By using the existing RDBMS (such as Oracle, MySQL etc) of which performance is proved, we can build the system which manages efficiently enormous OWL documents. And, it is required to OWL inference to guarantee Quality of Service (QoS) for robot environments. Thus, the efficient reasoning engine is required for the knowledge base, and the engine is also made by RDBMS. In our work, we focus on the design of robot knowledge management system using OWL-aware Relational Model and reasoning engine for intelligent robot.

3.2 OWL-aware Relational Model

Because OWL is based on XML, OWL documents are usually semi-structured. The structural and semantic analysis is required to store OWL in RDBMS.

In previous research [6, 11], database schema is designed through the syntactic analysis of OWL. Especially, this schema has been considered not to loss the implicit semantics of OWL. Figure 2 is the database schema for OWL-aware relational model.

OWL-aware relational model is designed to enable any processors to access the knowledge. Accordingly, the system has to provide QoS for any processors using the robot knowledge such as environment, object, activity,

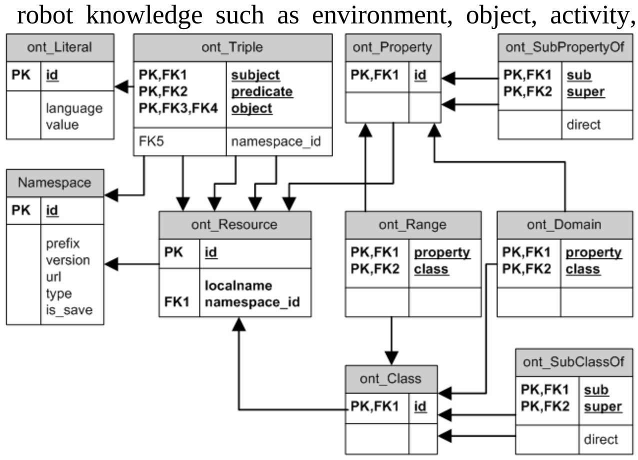
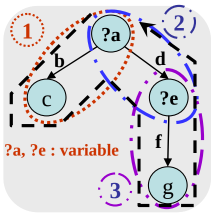
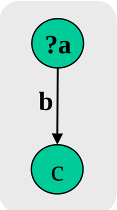


Fig.2. OWL-based Database Schema



1. Generate SQL(1)
2. Generate SQL(3)
3. Generate SQL(2)
4. By the pattern 2 of Definition 2
SQL(3) JOIN SQL(2) ON SQL(2).object=SQL(3).subject
5. By the pattern 1 of Definition 2
SQL(1) JOIN SQL(2,3) ON SQL(1).subject=SQL(2,3).subject

Fig. 3. An example for RDQL2SQL algorithm




```
SELECT OWL_Triple.subject
FROM   OWL_Triple
WHERE OWL_Triple.predicate = 'b' AND
      OWL_Triple.object = 'c'
```

Fig. 4. An example of conversion from RDF graph model to an SQL query

query is run.

i) There are definitions of semantics for all rules of OWL in W3C document; "OWL Web Ontology Language Semantics and Abstract Syntax" [10] that also include formal definitions for OWL entailment. When OWL documents are inserted in RDBMS, all facts should be deduced on the inference engine based on all rules of OWL.

ii) It is required that ontology instance stored in RDBMS are always consistent for concepts and relationships among them. The inserting, updating and deleting ontology instances may cause the inconsistency of knowledge base. OWL-aware relational model preserves the version, date and author information for all OWL documents, therefore, that make knowledge base consistent. In addition, in the case of inconsistency and system failure, knowledge base may be safely recovered with recovery mechanism of RDBMS. Lastly, it is also considered that the there may be some hidden rules when ontology instance is inserted, so inference engine should deduce hidden knowledge for data consistency. For example, if "A is B" is stored in knowledge base and the new knowledge instance; "F is A" occurs this time, then, inference engine should deduce hidden knowledge "F is B" and also store it in the RDBMS.

iii) Query languages for OWL documents includes RDQL[12], SPARQL[13], RQL, SeRQL and etc. However, those languages still dose not support the query level inference. The current researches of query languages are focused on triple pattern search algorithms of ontological data satisfying query. Therefore, we introduce the fast search mechanism to find he patterns of query. The target language is RDQL, and this language is SQL-like syntax.

The RDQL query process is three steps.

Step 1: RDQL Parser generates the RDF graph model from RDQL query. We developed the RDQL parser based on the well-known toolkit: Jena, which is a Java framework for Semantic Web applications.

Step 2: RDQL2SQL converter generates SQL queries from RDF graph model generated by RDQL parser.

Step 3: Lastly, SQL query is processed by the existing RDBMS query processing engine.

For the efficient converting process, RDQL2SQL is the most important module to generate SQL queries. This module has been implemented through the following algorithm;

The conversion algorithm is performed by depth-first traverse of RDF graph model. In the below, we formally set up Definitions 1 and 2 to convert RDF query into an SQL query.

Definition 1 : If the triple set for RDF graph is given as T = {t1, t2, ⋯, tn}, each element of T is converted as an SQL query. That is, there are conversion given as t1→s1, t2→s2, ⋯ tn→sn. Here, the set of SQL queries is defined as S = {s1, s2, ⋯, sn}.

Definition 2 : There are two patterns such as RDF triples have the same subject or same object
Pattern 1 : In the case of a same subject
SQL(1) JOIN SQL(2)
ON SQL(1).subject=SQL(2).subject
Pattern 2 : In the case of ①.object=②.subject
SQL(1) JOIN SQL(2)
ON SQL(1).object=SQL(2).subject

For announcement, "JOIN" and "ON" is SQL keywords, which mean inner join process. An inner join finds the intersection between the two tables in RDBMS. Especially, "ON" keyword indicates join conditions [14].

In the case of a same subject (Pattern 1), The SQL query of the triple ① and ② is generated as inner join (join condition: ①.subject = ②.subject). By Definition 1, the triple ① and ②is converted SQL queries; SQL(1) and SQL(2), it is described in Definition 2 of pattern 1.

In the case of same object (Pattern 2), the SQL query of the triple ① and ② is generated as inner join (join condition: ①.object = ②.subject), it is described in Definition 2 of pattern 2

The next step-the conversion algorithm of RDQL2SQL follows;

Algorithm: Conversion of RDQL2SQL
Input: RDQL Graph Model
Output: SQL Query
(1) Traverse nodes of RDF graph model by depth-first traverse.
(2) If the current node is leaf node, convert a triple into an SQL query following Definition 1.
(3) If the current node is one of patterns in Definition 2, convert triples into a joined SQL using the converting method of pattern 1 or pattern 2 in Definition 2.
(4) Until all nodes of RDF graph are visited, repeat (1), (2), and (3).

Figure 3 shows an example for RDQL2SQL algorithm. If one traverses RDF graph model with depth-first, part ① is converted to SQL(1) following Definition 1. Next, part ③ and ② are converted to SQL(3) and SQL(2). By the pattern 2 of Definition 2, SQL(2) and SQL(3) is joined on SQL(2).object=SQL(3).subject, and the result query is SQL(2,3). Lastly, SQL(1) and SQL(2,3) is joined on

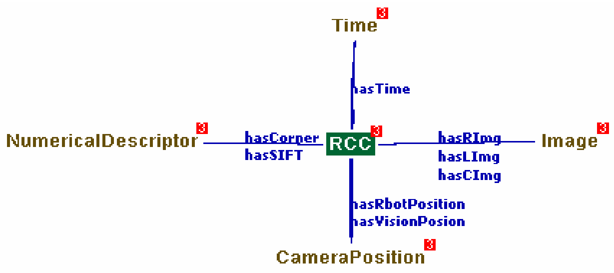


Fig. 5. RCC ontology schema converted by RCC_Table

SQL(1).subject=SQL(2,3).subject by the pattern 1. Figure 4 shows an example that RDF graph model (?a, b, c) is converted to an SQL query.

The SQL query may depend on the table structure. To make the conversion algorithm simple and efficient, OWL-aware relational model is designed with single table referring to OWL_Triple(subject, predicate, object).

With OWL-aware relational model and OWL inference engine of RDQL on RDBMS, robot may be able to manage his data consistently. Moreover, it is possible to implement efficient RDQL query engine.

4. Robot Knowledge Management System

In robot environment, robot-centered ontology is essential. The ontology makes robot fine hidden knowledge in spite of uncertain and partial data. So we convert robot-centered data to OWL-aware relational model and build Ontology-based Multi-layered Robot Knowledge Framework (OMRKF) which is consisted of robot knowledge boards and rules

4.1 Conversion of robot-centered data into OWL-aware relation model

In our experiments, we convert robot-centered data into OWL-aware relational model that enable robot to use inference mechanism on RDBMS.

Robot-centered data includes the positions of robot and camera, stereo camera images, and numerical descriptors; SIFT Harris corners with time index. Figure 5 shows the RCC ontology schema converted by RCC_Table in Fig. 1.

4.2 Ontology-based Multi-layered Robot Knowledge Framework (OMRKF)

OMRFK is robot knowledge framework for intelligent robot to perform the household service. Figure 6 show OMRKF which consists of robot-centered ontology and rules. Robot-centered ontology consists of four levels of knowledge such as perception, model, activity and context level. And, rules and inference engine makes robot find hidden knowledge from uncertain and partial data, link between sub-symbolic data and symbolic information, associate each level of knowledge with others.

OMRKF includes four levels of knowledge (KLevel); perception, model, context and activity level. And, each level of knowledge has three knowledge layers (KLayer), Perception level is knowledge of visual concepts that are anchored with numerical descriptors comes from robot own sensors and visual feature mechanisms. Model level is knowledge for world model consists of space, objects and object feature. Context level is for knowledge from low level contexts; spatial context and temporal context to high level context. And, activity level is knowledge for robot action from atomic robot behavior and task to high level service. Moreover, each knowledge layer has three ontology layers (OLayer); meta ontology, ontology and ontology instance layer. Meta ontology layer is for ontology schema to be shareable and growing, ontology layer is applicable to real environment and its instance is included in instance layer.

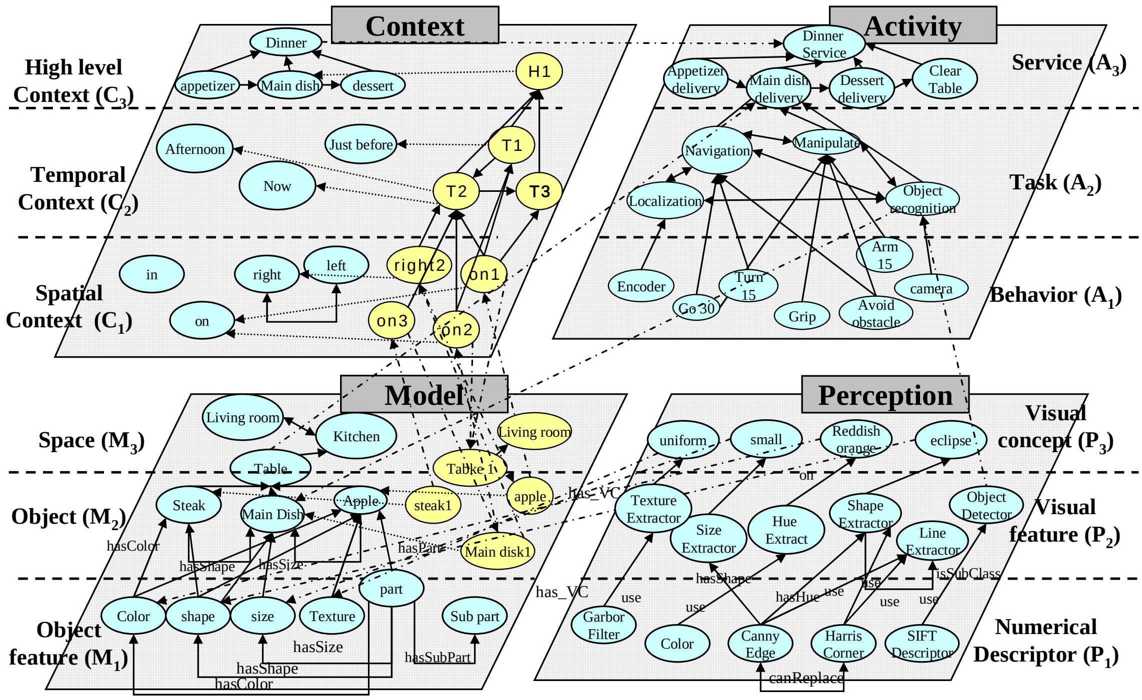


Fig. 6. OMRKF

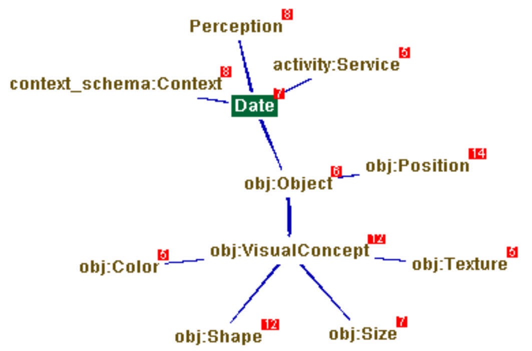


Fig. 7. Robot-centered ontology schema with time index

In addition, OMRKF contains axioms and rules. Axioms are for the relationship of concepts in the same ontology layer and rules are for the relationship between different ontology layer or different knowledge layers, and association rules among several level of knowledge.

Figure 7 shows a robot ontology schema with time index. Each level in OMRKF is synchronized with time index, and is filled completely with inference mechanisms. Thus at one time, OMRFK includes perception level knowledge which is performed data processing algorithms and its numerical descriptors, model level with is recognized by visual concepts, context level which helps robot act properly to the situation, and activity level which robot will performs

4.3 Inference on Robot Knowledge Management System

We represent OMRFK with OWL-aware relational model, which utilizes entailment relation to inference new sentences. Moreover, we convert robot-centered data to OWL-aware relational model. Thus, OMRKF includes sub-symbolic data that are seldom utilized by conventional ontology system. The data generated from robot perception or activities are numerical data, which are partial and incomplete. The probabilistic approach has dominated the solution of that case [2]. However, those systems may be application-specific, which is difficult to reuse and requires verification of the procedures. Sub-symbolic data are applied on OMRKF to ontology-based inference mechanisms, so that OMRKF can deduce hidden knowledge that is generated by a partial observation or an observation error, and make it easy to reuse and verify. Moreover, OMRKF also includes association rules that associate each level of knowledge, those rules enable robot to query not only unidirectional reasoning but also bidirectional reasoning. Table III show rules between each level of knowledge represented by FOL

Time	Perception	Model	Activity	Context
200606261 5000	Image1, Color1, Shape1, Encoder1	Salad_1, Bread_1, Table_1	Dinner service1	Appetizer_1

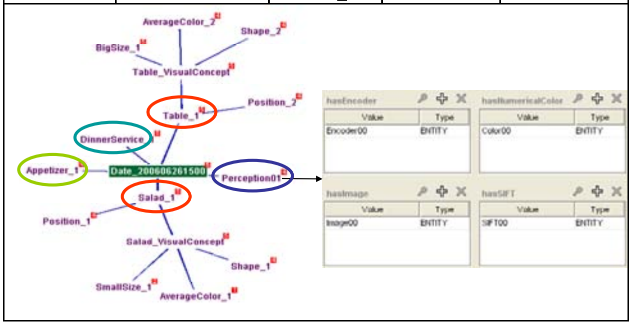


Fig. 8. An example of robot-centered ontology instance for dinner service

5. Experiment

Our proposed system for robot management is verified by the experiment of robot dinner service. Figure 8 shows an instance of OMRKF and its representation for dinner service that uses not only OWL entailment in the same level but also association rules between levels of knowledge as shown in Table III. At on time, robot performs image processing algorithms such as color, shape and measure encoder data, and anchors them with

TABLE I
CONVERSION OF RULES TO RDQL AND SQL FOR DINNER SERVICE

Rules	Model => Context
OWL	<Salad rdf:ID="Salad_1"> <obj:hasPosition> <obj:Position rdf:ID="Position_1"> <obj:hasX rdf:datatype="http://www.w3.org/2001/XMLSchema#float">0.0</obj:hasX> … </obj:Position> </obj:hasPosition> <obj:hasVisualConcept> <obj:VisualConcept rdf:ID="Salad_VisualConcept"> <obj:hasSize> <obj:SmallSize rdf:ID="SmallSize_1"/> </obj:hasSize> … </obj:VisualConcept> </obj:hasVisualConcept> </Salad>
FOL	Salad(?y1) $\wedge$ Table(?y2) $\wedge$ first(?on, ?y1) $\wedge$ second(?on, ?y2) $\wedge$ On(?on) $\wedge$ hasSpace(?x1, ?on) $\Rightarrow$ Appetizer(?x1)
RDQL to SQL	Select ?y1, ?y2, ?on Where (?on type On) (?on first ?y1) (?on second ?y2) (?y1 type Salad) (?y2 type Table)
SQL	SELECT DISTINCT (SELECT concat((SELECT Namespace.url FROM Namespace WHERE Namespace.id = owl_Resources.namespace_id) ,localname) FROM owl_Resources WHERE owl_Resources.id = xxx_12.on1) AS on1 , … WHERE xxx_9.on1=xxx_10.on1 AND xxx_9.on1=xxx_11.on1

visual concepts which robot has in perception level. The visual concepts of “Salad_1” is “SmallSize_1,” “AverageColor_1,” and “Shape_1.” Robot matches objects with corresponding visual concepts such as “Salad_1,” “Bread_1” and “Table_1.” With these models, robot generates spatial-temporal contexts such as “Salad_1 on Table_1 at 200606261500” and high level context; “Appetizer_1 now.” In addition, OMRKF expects forthcoming activities such as “Dinner Service1.” Again, task-relevant objects are deduced by activity level of context level.

Table I shows an example of conversion rule from model to context with SQL for dinner service. With perceived objects such as “Salad_1” and “Table_1,” the context, “Appetizer_1” is deduced by the rules which are represented by FOL in Table III. The FOL rule is converted by OWL-aware relational model with RDQL2SQL. Finally, an SQL query is generated.

With proposed system, the ontological data that includes robot-centered ontology and robot-centered data are represented with OWL-aware relational model, So robot can manage all types of data in DBMS, and can deduce hidden knowledge.

6. Conclusion

We proposed a robot knowledge management system based on ontology, and the ontology is represented by OWL-aware relational model that can perform OWL entailment inference. Robot should handle many types of data from low level numerical sensor data to high level symbolic information. We represent these data with OWL-aware relational model, so that the robot knowledge can be sharable and growing. For the future work, OWL-aware relational model should have more representation ability to handle uncertain and partial data, and we apply our proposed system to various domains for more knowledge.

Acknowledgements

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

References

[1] R. Pfeifer and C. Scheier, Understanding Intelligence. MIT Press, 2001, ch 12.

[2] S. Thrun, W. Burgard and D. Fox, Probabilistic Robotics. MIT Press 2005.

[3] E. Wang, Y.S. Kim, H.S. Kim, J.H. Son, S. Lee, I.H. Suh, “Ontology Modeling and Storage System for Robot Context Understanding”, Lecture Notes in Computer Science 3683, Springer , 2005: 922-926

[4] S. Jeong, J. Chung, S. Lee, I.H. Suh, B. Choi, “Design of a simultaneous mobile robot localization and spatial context recognition system” Lecture Notes in Computer Science 3683, Springer, 2005: 945-952

[5] W. Brenner, R. Zarnekow and H. Wittig, Intelligent Software Agents : Foundations and Applications. Springer 1998, ch 4.

[6] H. S. Kim, H. S. Cha, J. Kim, J. H. Son, "Development of the Efficient OWL Document Management System for the Embedded Applications", HSI 2005, Lecture Notes in Computer Science 3597, pp. 75-84, July 27~29, 2005.

[7] F. Baader, D. Calvanese, D. Mcfuinness, D. Nardi and P. Patel-schneider, “The Description Logic Handbook,” Cambridge university press, 2003, ch. 2.

[8] D. Fensel, M.L. Brodie, "Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce", Springer, 2003.

[9] M. Dean, et al., "OWL Web Ontology Language Refernce", W3C Recommendation 10 Feb 2004, See "http://www.w3.org/TR/owl-ref/".

[10] P. F. Patel-Schneider, et al., "OWL Web Ontology Language Semantics and Abstract Syntax", W3C Recommendation 10 Feb 2004, See "http://www.w3.org/TR/owl_semantics/".

[11] A. Reggiori, D. van Gulik, Z. Bjelogrlic, "Indexing and retrieving SemanticWeb resources: the RDFStore model. EuropeWorkshop on Semantic Web Storage and Retrieval", SWAD 2003.

[12] A. Seaborne, "RDQL - A Query Language for RDF : W3C Member Submission", Jan 2004, See "http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/".

[13] E. Prudhommeaux, A. Seaborne, "SPARQL Query Language for RDF", W3C Candidate Recommendation 6 April 2006, See "http://www.w3.org/TR/rdf-sparql-query/".

[14] H. Garcia-Molina, J. D. Ullman, J. Widom, "Database Systems - The Complete Book," Prentice Hall, 2002.