

Ontology-Based Knowledge Map Enabling Referential Navigation between Knowledge

Keedong Yoo

Dept. of MIS, Dankook Univ., 119 Anseo-dong, Cheonan 330-714,
Republic of Korea

Keywords: Ontology-based knowledge map, Referential navigation, Knowledge network, Protégé-OWL

Abstract. A knowledge map describes the network of related knowledge in the form of a diagram, and therefore underpins the structure of knowledge categories by defining the relationship of the referential navigation between knowledge. The concept of referential navigation means the function of cross-reference between related knowledge, which can be exhibited as one piece of knowledge becomes to be utilized after related the other knowledge has been activated. For this reason, building a knowledge map based on the ontology technology has been emphasized to transplant the features of the cross-referential knowledge network into a knowledge map. This paper suggests a methodology to build an ontology-based knowledge map enabling the referential navigation between knowledge. The ontology-based knowledge map resulted from the proposed methodology can not only express the referential navigation between knowledge but also infer additional network between knowledge based on the dependent relationships. To verify the feasibility of the proposed concepts, a referential navigation-enabled knowledge map is exemplified.

1. Introduction

A knowledge map describes the network of related knowledge in the form of a diagram, and therefore underpins the structure of knowledge categories by defining the relationship of the referential navigation between knowledge [1, 2, 3]. The concept of referential navigation means the function of cross-reference between related knowledge, which can be exhibited as one piece of knowledge becomes to be utilized after related the other knowledge has been activated. This kind of cross-referential relationship between knowledge reveals the basic mechanism of user's knowledge utilization, because a user usually concludes proper knowledge optimally applicable to his/her problem situation by moving, namely navigating, around candidate knowledge related with each other according to the learning dependency[4]. Based on the learning dependencies between knowledge, all pieces of related knowledge can be logically sequenced, and therefore the cause and effect relationships between knowledge can be identified. Using this causal relationship, a user can perform knowledge search with navigation in a more convenient and comprehensible manner.

The cause and effect relationship between knowledge can be more expanded as more effective connection between knowledge is identified, and finally it can be shaped as a network of knowledge. A network display using nodes and links to arrange and represent the relationship can provide a more

complex knowledge structure than a hierarchical display. Moreover, it can facilitate a user to infer through the links shown on the network [5]. For this reason, to transplant the features of the network display into a knowledge map, the ontology technology can be properly applied [6, 7]. The most highlighted benefits that can be delivered by applying the ontology technology to the knowledge map include; formal expression about knowledge and its relationships with others, automatic identification of the knowledge network based on the function of self-inference on the referential relationships, and automatic expansion of the knowledge base designed to categorize and store knowledge according to the network of knowledge.

As the necessity to build a knowledge map based on the structure of the ontology technology increased, not a few researches have been proposed to fulfill the needs [8, 9, 10]. However, most of previous researches to apply the ontology technology in building the knowledge map just focused on formal expression of knowledge and its relationships with others to demonstrate the possibility of knowledge reuse. Although various types of ontology-based knowledge maps were proposed, no researches have successfully designed and implemented the referential navigation-based knowledge map.

To enable the referential navigation between knowledge included in a knowledge map, and therefore to form a knowledge map in a network display, the ontology must describe knowledge with respect to the relation with the process and task, in which a piece of knowledge is associated. A process is composed of component tasks, while a task is executed by required knowledge as an input. Since the relation of cause and effect between knowledge can be inheritably determined by the execution sequence of tasks, the dependent relationship between knowledge can be circuitously designed by knowledge which is modeled as one of inputs or outputs of each task. Since knowledge has been related to tasks, and since a task has been also related to processes, the causal relations between knowledge have been circuitously determined by the flows of processes: all of knowledge can be sequentially linked together, and therefore referential navigation between knowledge can be finally enabled.

To describe the knowledge with respect to related process and task, the Protégé-OWL, an editor that enables users to build ontologies for the Semantic Web, can be used. An OWL ontology-based knowledge map includes descriptions of classes (process, task, and knowledge), properties (relationships between process and task, task and knowledge), and their instances. Given such an ontology, the OWL formal semantics specifies how to derive its logical consequences, i.e. facts not literally present in the ontology, but entailed by the semantics. Therefore a knowledge network can be automatically formulated based on the defined relationships, and the referential navigation between knowledge can be enabled.

This paper suggests a methodology to build an ontology-based knowledge map enabling the referential navigation between knowledge. The ontology-based knowledge map resulted from the proposed methodology can not only express the referential navigation between knowledge but also infer additional network between knowledge based on the dependent relationships. To verify the feasibility of the proposed concepts, a real business process-based knowledge map is exemplified: a knowledge map for the process of 'Business Trip Application'. By applying the 'DL-Query', a plug-in module provided by the Protégé-OWL, the performance of the implemented ontology-based knowledge map is to be examined.

2. Methodology

2.1 Overview

From the viewpoint of the knowledge engineering, knowledge to be managed must be formally defined and expressed not only to enhance the understandability about knowledge but also to increase reusability of knowledge by expanding the domain of knowledge application. Knowledge must be

formally defined and stored, because any knowledge included in one knowledge map can be reused (or exhibited) in the other map. To formally define knowledge, ontology languages such as RDF/OWL can be used: To formally express the relationships between knowledge in the form of a knowledge map, the tool of Protégé can be deployed.

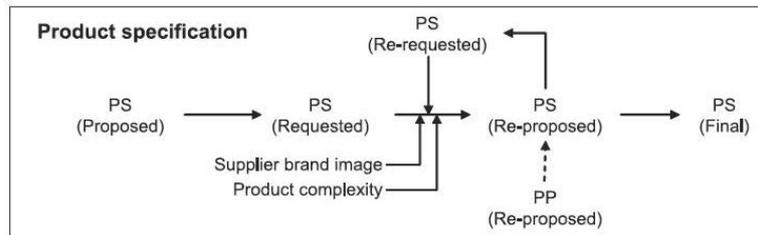


Fig. 1. An example of the process-based knowledge map [3]

The network between knowledge can be automatically formulated by the self-inference-based relation expansion, one of the representative capabilities of an ontology. In a network-shaped knowledge map, as seen in Fig 1, a new piece of relation between knowledge can be further expanded based on the already-defined relations. The property between classes defined in an ontology can be understood as the relation between knowledge, because knowledge is defined as a class while the relation is defined as the property. New pieces of relations between knowledge can be automatically identified by using ontology’s function of self-reasoning, and therefore the referential navigation, the navigation between something that have effects on each other, between knowledge can be concluded. While conventional way to form a knowledge network must manually update the relation between knowledge whenever a piece of knowledge is newly exhibited, the ontology-based approach does not have to manually form the network with the aid of self-reasoning.

Since the knowledge network can be automatically expanded by ontology’s function of self-reasoning, the knowledge map can be also automatically expanded accordingly. The knowledge-base, therefore, can be naturally expanded in itself as the knowledge network expands.

2.2 Procedures

The procedure for building an ontology-based knowledge map is similar that of a general ontology rather than that of a knowledge map in order to relate knowledge with tasks and processes. Based on the relationship between knowledge and tasks, and also the relationship between tasks and processes, knowledge can be mapped with respect to processes associated. The procedure for build an ontology is usually composed of following steps;

2.2.1 Step 1: Conceptualization

The relation between process and task, and the relation between task and knowledge can be expressed as follows.

$$P_i = f(T_{ij}) \quad T_{ij} = g(K_{ijk})$$

$$P_i = f(g(K_{ijk})) = h(K_{ijk})$$

P_i : Process i $(i = 1, 2, 3, \dots, n)$
 T_{ij} : Task j composing Process i $(j = 1, 2, 3, \dots, n)$
 K_{ijk} : Knowledge k used in Task j composing Process i $(k = 1, 2, 3, \dots, n)$
 f, g, h : Arbitrary function

A process can be expressed as the function of related knowledge used in the process, for this reason any flows of information or knowledge can be used in checking the status of a process [3]. This fact also means a knowledge flow-based knowledge map which has the form of a network must be

expressed based on related processes and tasks, because knowledge has its value within related processes and tasks.

Therefore, to apply the concept of an ontology to a knowledge map, a piece of knowledge must be defined with respect to related tasks and processes in a roundabout way. Previous researches [11, 12] defined the relation among ‘process-task-knowledge’ using ERD (Entity Relationship Diagram), these trials also expressed knowledge with respect to related tasks and process.

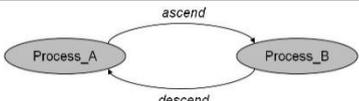
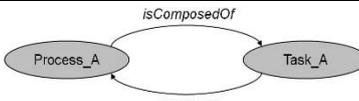
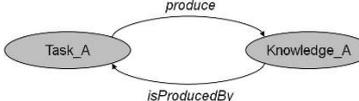
2.2.2 Step 2: Modeling

Once the conceptualization about the target classes and relations among them has been completed, formal notation-based modeling must be done.

Formalized or standardized modeling enhances understandability about the ontology built previously, clarifies the information structure managed by the ontology. One very famous modeling tool is the Protégé-OWL, a free, open-source platform that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontologies. At its core, Protégé implements a rich set of knowledge-modeling structures and actions that support the creation, visualization, and manipulation of ontologies in various representation formats (<http://protege.stanford.edu/overview>). Protégé-OWL basically uses the concept of ‘Individual’, ‘Property’, and ‘Class’ to model the targets. ‘Individuals’ represent objects in the domain in which we are interested, and generally mean the value of Instance of a class. ‘Properties’ are binary relations on individuals - i.e. properties link two individuals together. ‘Classes’ are interpreted as sets that contain individuals. They are described using formal (mathematical) descriptions that state precisely the requirements for membership of the class.

To relate and model ‘Process’, ‘Task’, and ‘Knowledge’ together, Protégé-OWL defines these concepts as the Classes. Since ‘Process’ is composed of more than one ‘Task’, the notion of ‘compose(property)-isComposedOf(inverse property)’ can be assigned as the property linking these two classes. In the same vein, since ‘Task’ uses ‘Knowledge’ to produce outputs, the notion of ‘use-isUsedIn’ can be assigned as the property linking them.

Table 1. Class & Object Property in a knowledge map

Object Property (Inverse)	Related Class	Meaning
<i>ascend</i> (<i>descend</i>)	Process → Process (Task → Task)	
<i>compose</i> (<i>isComposedOf</i>)	Task → Process	
<i>activate</i> (<i>isActivatedBy</i>)	Knowledge → Task	
<i>produce</i> (<i>isProducedBy</i>)	Task → Knowledge	

2.2.3 Step 3: Implementation and Application

Concepts defined by Protégé-OWL can be diversely used as data models according to application situations. Since in this research the ontology model is designed to build a knowledge map, converting conventional process-oriented knowledge map into ontology-based knowledge map is

performed in this step. Therefore, the sequential procedures of processes and tasks are defined, and also the knowledge flows are accordingly defined, as follows;

Processes and tasks are basically sequenced. In other words, ascending and descending processes and tasks can be defined, and therefore the sequential relations must be defined in an ontology. To model the sequential relation, proper property must be selected at first, however alternatively the ‘property characteristics’ can be also effectively used. OWL allows the meaning of properties to be enriched through the use of property characteristics. In this research, the property characteristics ‘transitive’ is very effective to describe the sequential and chained classes such as processes and tasks.

Knowledge flows can be defined by the sequence of tasks in a roundabout way. However, based on one task, the input knowledge and output knowledge must be separately defined, and therefore, deploying differentiated properties is recommended. A piece of knowledge activates a task, while a task produces a piece of knowledge: therefore the properties of ‘activate’ and ‘produce’ must be defined split, which might yield misunderstanding that the inverse property of ‘activate’ is ‘produce’. If we understood in this way, however, a contradiction could be exhibited because a piece of knowledge cannot simultaneously activate and be produced by one task. Table 1 summarizes classes and properties to describe ‘process-task-knowledge’ in the form of knowledge map using Protégé-OWL.

3. Real Case: A Knowledge Map for the Process of ‘Business Trip Application’

To verify the feasibility of the proposed methodology, a knowledge map of a real business process is to be reformed based on the concept of ontology. The exemplified business process is about ‘business trip application’ as the knowledge map for this process, Fig 2, shows.

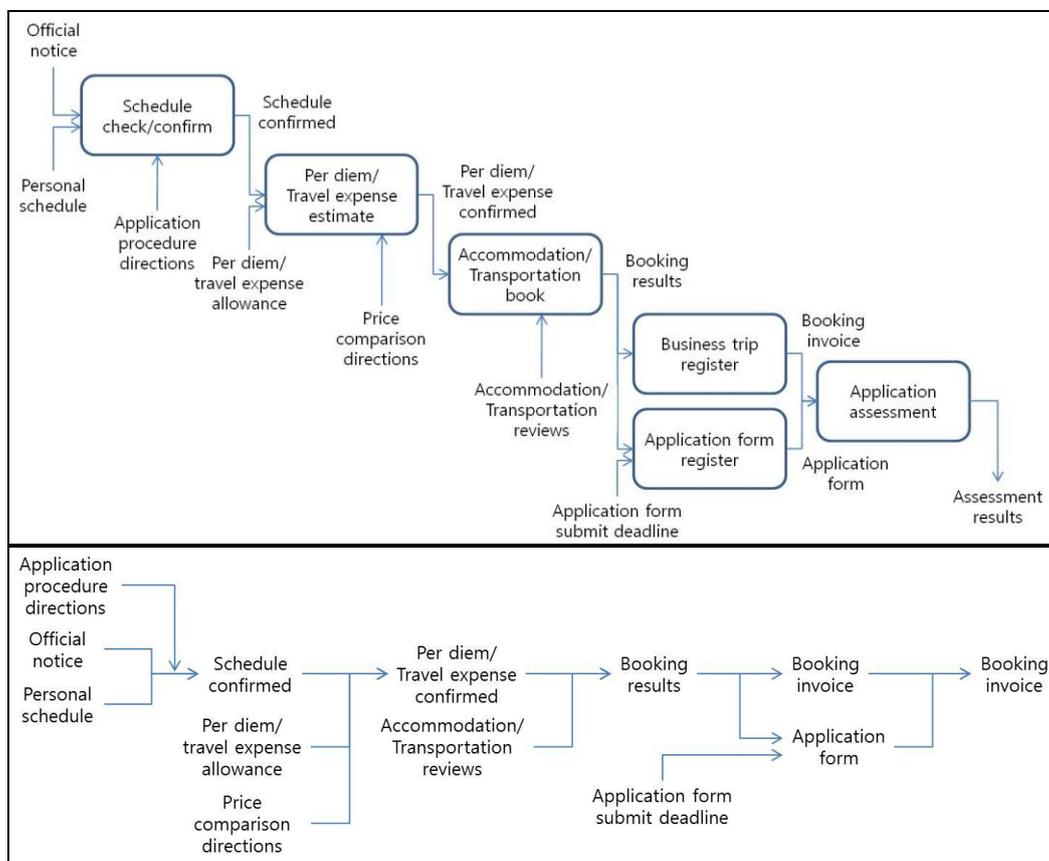


Fig. 2. Process of Business Trip Application (above); Knowledge map of the Business Trip Application (below)

3.1 Conceptualization

A process is composed of more than one task. A task is executed by more than one piece of knowledge as an input, and produces more than one piece of knowledge as an output. Consequently, a process can be executed by more than one piece of knowledge as an input, and can produce more than one piece of knowledge as an output: the process and knowledge can be related via the task.

Since a process which produces certain knowledge as its final output basically can have a relation with knowledge via a task, the object of ‘Process’, ‘Task’, and ‘Knowledge’ must be defined to be mutually exclusive (disjoint). The object property between ‘Process’ and ‘Task’ must be defined as ‘(Task) compose (Process)’; the object property between ‘Task’ and ‘Knowledge’ must be defined as ‘(Task) produce (Knowledge)’, while in turn ‘(Knowledge) activate (Task)’. Fig 3 shows the result of conceptualization using Protégé-OWL.

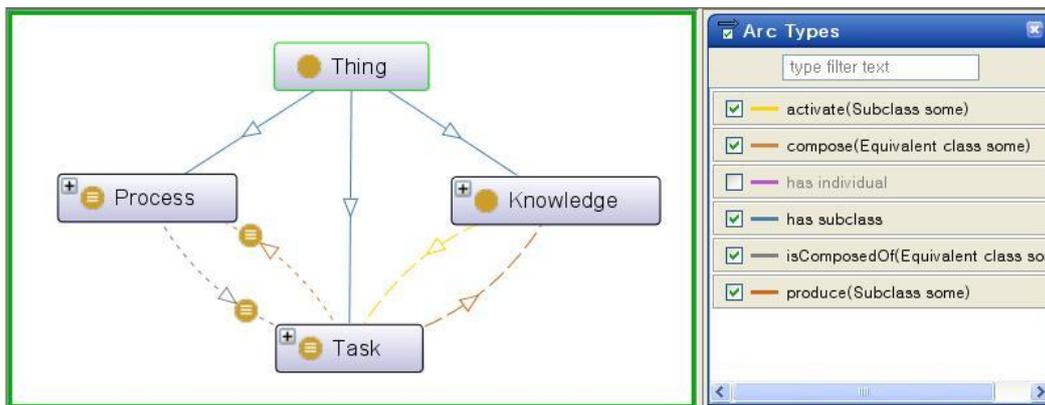


Fig. 3. Class Conceptualization

3.2 Modeling

Based on the result of conceptualized relations between ‘Process’, ‘Task’, and ‘Knowledge’, subclasses of each class must be defined. Especially, the sequential relations (‘ascend’ and ‘descend’) within ‘Task’ must be clearly defined, so that the causal relationships within ‘Knowledge’ can be effectively transplanted.

There exist four kinds of object properties (inverse object properties): ‘compose (isComposedOf)’, ‘ascend (descend)’, ‘activate (isActivatedBy)’, and ‘produce (isProducesBy)’ as illustrated in Table 1 and exemplified in Fig 4, and all of these properties have the characteristics of ‘transitive’ so that syllogism-like chained inference can be made.

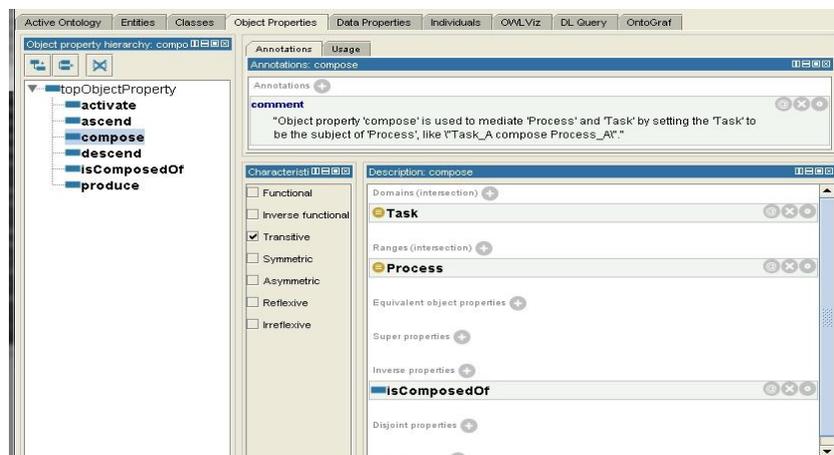


Fig. 4. Object Property Definition (for the case of ‘compose’)

3. Implementation and Application

Based on the defined classes and object properties, knowledge must be defined according to related tasks, while each task must be defined based on related processes in turn. Since the sequential relations ('ascend' and 'descend') within knowledge can be expressed via the sequential relations within tasks, 'Knowledge' as an input and as an output must be strictly separated. Fig 5 shows the resultant ontology-based knowledge map modeled and implemented using Protégé-OWL, and it coincides with the knowledge map in Fig 2 (Lower).

4. Testing the Validity of Referential Navigation among Knowledge

To check whether the resultant knowledge map satisfies basic characteristics of ontology technology and, therefore, whether the referential navigation between knowledge, the key requirement of this research, is successfully established, 'Fact++' and 'DL-Query' can be applied. 'Fact++' is an inference engine provided by the Protégé-OWL as a plug-in module, and checks whether the implemented ontology has logical errors or not. If errors are found, the sources of errors are highlighted red; if not, no changes occur or inferred classes are automatically added in the page of 'Class hierarchy'. 'DL-Query' quests using query statements to check whether the implemented ontology correctly defines the relations between concepts. The query statement has the form of 'frame', and is supported by the Manchester OWL syntax, which is user-friendly syntax for OWL DL.

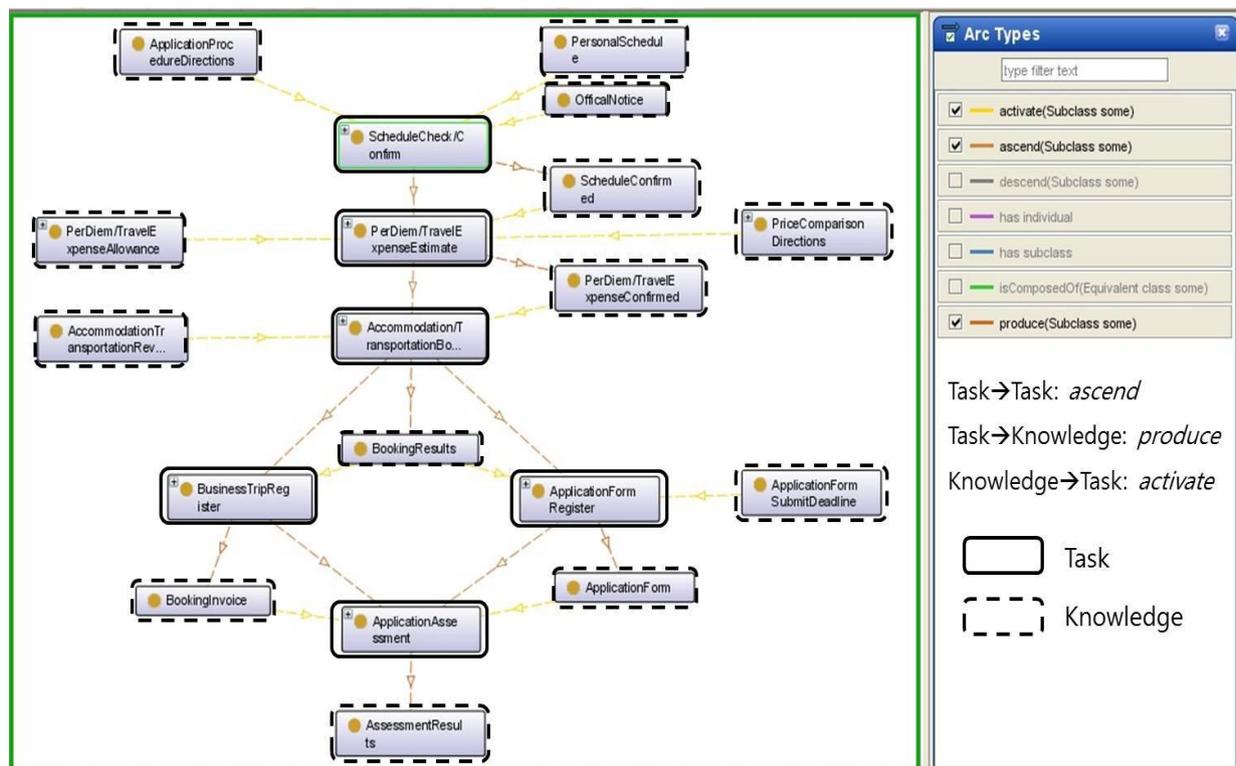


Fig. 5. Resultant ontology-based knowledge map for the process of 'Business Trip Application'

Applying 'Fact++' inference to the implemented ontology, no warning or errors were found. To check the logical correctness using 'DL-Query', five questions were developed as follows. These query statements are to question whether 'Knowledge' has been sequentially linked together according to the execution sequence of 'Task'. If correct answers are outputted, then it proves not only

all of defined knowledge has been networked together, but also the referential navigation between knowledge has been accomplished.

Query#1. Querying Task which produces Knowledge 'AssessmentResults'

DL-Query: Task and produce some AssessmentResults

Answer: ApplicationAssessment (1 task)

Query#2. Querying Knowledge activating Task which produces Knowledge 'AssessmentResults'

DL-Query: Knowledge and activate some (Task and produce some AssessmentResults)

Answer: BookingInvoice, ApplicationForm (2 pieces of knowledge)

Query#3. Querying Task producing Knowledge which activates Task which produces Knowledge 'AssessmentResults'

DL-Query: Task and produce some (Knowledge and activate some (Task and produce some AssessmentResults))

Answer: BusinessTripRegister, ApplicationFormRegister (2 tasks)

Query#4. Querying Knowledge activating Task which produces Knowledge which activates Task which produces Knowledge 'AssessmentResults'

DL-Query: Knowledge and activate some (Task and produce some (Knowledge and activate some (Task and produce some AssessmentResults)))

Answer: ApplicationFormSubmitDeadline, BookingResults (2 pieces of knowledge)

Query#5. Querying every piece of Knowledge used in the process 'BusinessTripApplication'

DL-Query: Knowledge and activate some (Task and compose some BusinessTripApplication)

Answer: 12 pieces of knowledge (except for 1 knowledge of 'AssessmentResults' as the final output)

Among these queries, from Query#1 to Query#4 check whether the resultant knowledge map correctly define the dependent relations between knowledge. Especially Query#2 and Query#4 seek ascending knowledge of knowledge 'AssessmentResults'. Results show that 'BookingInvoice', 'ApplicationForm', 'ApplicationFormSubmitDeadline', and 'BookingResults' are linked with 'AssessmentResults', and this fact illustrates that the referential navigation between these 5 pieces of knowledge is enabled. Query#5 confirms the total number of knowledge included in this knowledge map, and proves that every piece of knowledge included in this process is linked together and that the referential navigation between these 12 pieces of knowledge is also enabled. Fig 6 shows the results of queries, and the results demonstrate correct answers are concluded.

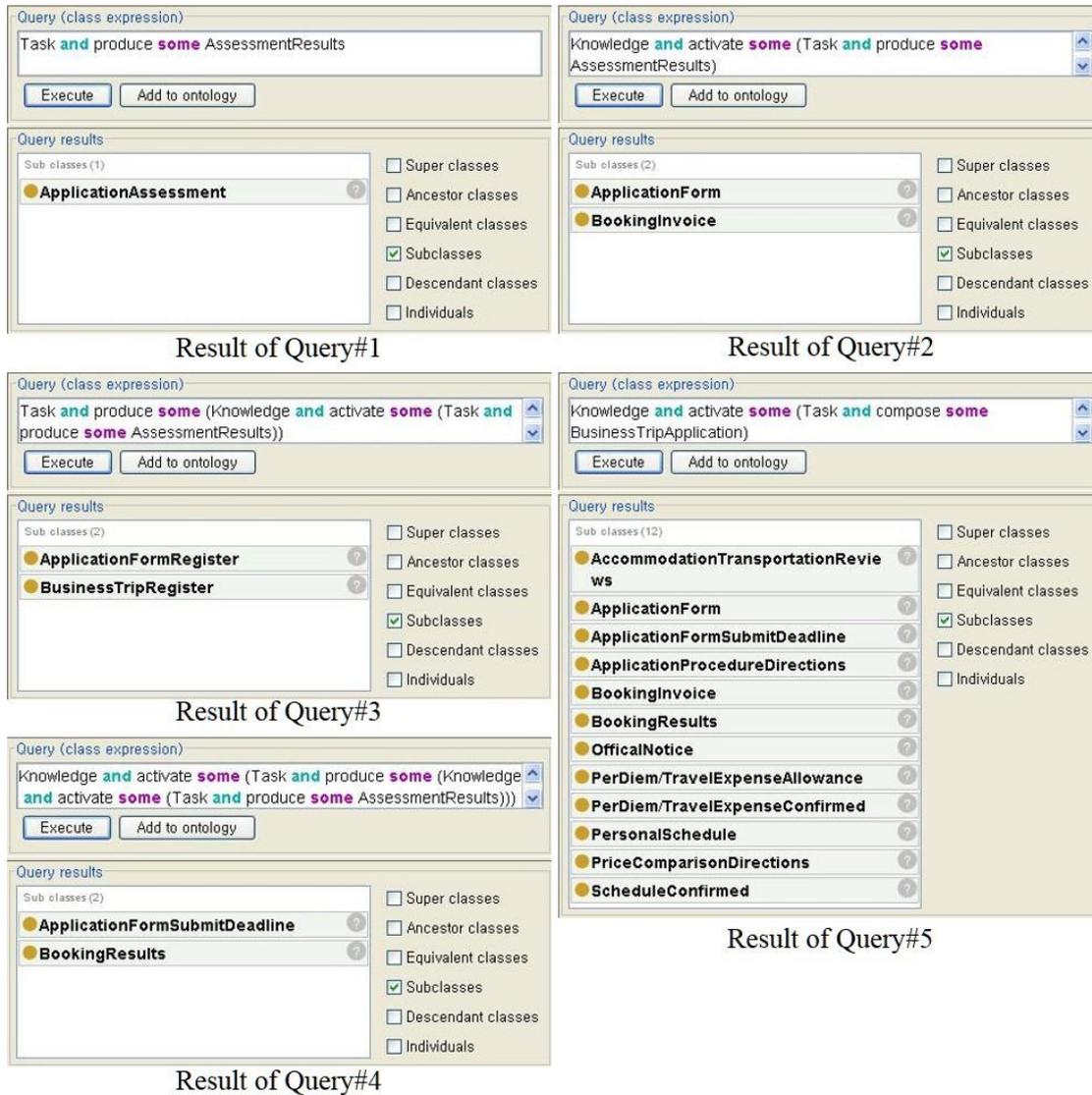


Fig. 6. Results of DL-Queries

5. Summary

This paper addresses a methodology to build an ontology-based knowledge map enabling the referential navigation between knowledge. The ontology-based knowledge map of this research can not only express the referential navigation between knowledge but also infer additional relationships among knowledge based on the dependent relationships.

To verify the validity of the proposed concepts, a real business process-oriented knowledge map is exemplified: the knowledge map of the process of 'Business Trip Application'. By applying the 'DL-Query' provided by the Protégé-OWL as a plug-in module, the performance of the implemented ontology-based knowledge map has been examined. Two kinds of queries to check whether the knowledge is networked with respect to the referential relations as well as the ontology-based knowledge network can infer further facts that are not literally described were tested. The test results show that not only the referential navigation between knowledge has been correctly realized, but also the additional inference has been accurately performed.

6. Acknowledgement

This work was supported by the National Research Foundation of Korea Grant funded by the Korean

Government (NRF-2013S1A5A2A01017530).

References

- [1] S. K. Goyal, An integrated inventory model for a single supplier-single customer problem, *International Journal of Production Research*, vol.15, pp. 107-111, 1977.
- [2] Chan, C. and Yu, S., Functional ontology and concept maps for knowledge navigation: an application example for contest robot, *Information Technology Journal*, Vol.10 No.9, 1740-1746, 2011.
- [3] Kim, S., Suh, E., and Hwang, H., Building the knowledge map: an industrial case study, *Journal of Knowledge Management*, Vol.7 No.2, 34-45, 2003.
- [4] Yoo, K., Suh, E., and Kim, K., Knowledge flow-based business process redesign: Applying a knowledge map to redesign a business process, *Journal of Knowledge Management*, Vol.11 No.3, 104-125, 2007.
- [5] Gordon, J.L., Creating knowledge maps by exploiting dependent relationships, *Knowledge-Based Systems*, Vol.13, 71-79, 2000.
- [6] Lin, F. and Yu, J., Visualized cognitive knowledge map integration for P2P networks, *Decision Support Systems*, Vol.46, 774-785, 2009.
- [7] Lee, J., Suh, H., and Han, S.H., Ontology-based Knowledge Framework for Product Development, *Computer-Aided Design and Applications*, Vol.2 No.5, 635-643, 2005.
- [8] Xiong, Q., Wang, Y., Guo, J., and Wu, G., A Searchable knowledge map based on ontology, *Proceedings of the Fourth International Conference on Semantics, Knowledge and Grid*, 457-460, 2008.
- [9] Han, K.H. and Park, J.W., Process-centered knowledge model and enterprise ontology for the development of knowledge management system, *Expert Systems with Applications*, Vol.36 No.4, 7441-7447, 2009.
- [10] Savvas, I. and Bassiliades, N., A process-oriented ontology-based knowledge management system for facilitating operational procedures in public administration, *Expert Systems with Applications*, Vol.36 No.3, 4467-4478, 2009.
- [11] Zhang, J., Zao, W., Xie, G., and Chen, H., Ontology- Based Knowledge Management System and Application, *Procedia Engineering*, Vol.15, 1021-1029, 2011.
- [12] Mansingh, G., Osei-Bryson, K., and Reichgelt, H., Building ontology-based knowledge maps to assist knowledge process outsourcing decisions, *Knowledge Management Research & Practice*, Vol.7, 37-51, 2009.
- [13] Rao, L., Mansingh, G., and Osei-Bryson, K., Building ontology based knowledge maps to assist business process re-engineering, *Decision Support Systems*, Vol.52 No.3, 577-589, 2012.