PAPER Special Section on Knowledge-Based Software Engineering

Organizational Knowledge Transfer Using Ontologies and a Rule-Based System

Masao OKABE[†,††], ***Student Member***, **Akiko YOSHIOKA**[†*], **Keido KOBAYASHI**[†], ***Nonmembers***, ***and*** **Takahira YAMAGUCHI**[†a)], ***Member***

SUMMARY In recent automated and integrated manufacturing, so-called intelligence skill is becoming more and more important and its efficient transfer to next-generation engineers is one of the urgent issues. In this paper, we propose a new approach without costly OJT (on-the-job training), that is, combinational usage of a domain ontology, a rule ontology and a rule-based system. Intelligence skill can be decomposed into pieces of simple engineering rules. A rule ontology consists of these engineering rules as primitives and the semantic relations among them. A domain ontology consists of technical terms in the engineering rules and the semantic relations among them. A rule ontology helps novices get the total picture of the intelligence skill and a domain ontology helps them understand the exact meanings of the engineering rules. A rule-based system helps domain experts externalize their tacit intelligence skill to ontologies and also helps novices internalize them. As a case study, we applied our proposal to some actual job at a remote control and maintenance office of hydroelectric power stations in Tokyo Electric Power Co., Inc. We also did an evaluation experiment for this case study and the result supports our proposal.

key words: *knowledge management, knowledge transfer, intelligence skill, domain ontology, rule ontology, rule-based system, scheduling*

1. Introduction

A great number of skilled engineers are now retiring and Japanese industries are facing a problem to lose their skills. It is an urgent issue to transfer their skills to next-generation engineers. However, the conventional transfer by OJT needs a lot of time and money, and moreover, it is becoming difficult because the number of engineers has been heavily downsized. To solve this problem, there are several proposals using information technology, mainly, multimedia and virtual reality technology (see e.g. [1]). But, they focus on so-called "craft skill", which is skill such as to create a complex mold with high precisions. Chuma, however, points out that in manufacturing, there is another crucial skill called "intelligence skill", which is a skill such as to detect expected flaws of products or production processes and to solve them [2]. In a recent manufacturing plant, most of the production processes are automated by computers and these processes are integrated through computerized control. Hence, intelligent skill is becoming more and more important because the extent and impact of a flaw become large, while craft skill is being replaced by computerized numerical control. Intelligence skill requires both integrated knowledge and power of logical thinking. Since intelligence skill is different from craft skill, its efficient transfer to the next generation needs a different approach from the one for craft skill. In this paper, we propose combinational usage of ontologies and a rule-based system for the organizational transfer of intelligence skill, using an ontology repository specialized for knowledge management.

The structure of this paper is as follows. Section 2 reviews related works. Section 3 outlines our proposal. Section 4 introduces an ontology repository called GEN. Section 5 presents a case study from Tokyo Electric Power Co., Inc. (hereafter, TEPCO). Section 6 is its evaluation and discussions. Finally, Sect. 7 summarizes our proposal and points out some future works.

Manuscript received July 4, 2009.

Manuscript revised October 19, 2009.

[†]The authors are with Keio University, Yokohama-shi, 223–8522 Japan.

[††]The author is with Tokyo Electric Power Co., Inc., Tokyo, 100–8560 Japan.

[*]Presently, with Nissan Motor Co., Ltd.

a) E-mail: yamaguti@ae.keio.ac.jp

DOI: 10.1587/transinf.E93.D.763

2. Related Works

2.1 Knowledge Management

In knowledge management, Nonaka proposed SECI model for organizational knowledge creation [3]. It shows how organizational knowledge is created by syntheses of tacit and explicit knowledge, Socialization, Externalization, Combination and Internalization. In Socialization, new organizational tacit knowledge is formed from personal tacit knowledge. For that, the spiral of Externalization, Combination and Internalization is necessary. In Externalization, new explicit knowledge is formed from tacit knowledge, in Combination, combined explicit knowledge is formed from different kinds of explicit knowledge, and in Internalization, new personal tacit knowledge is formed from combined explicit knowledge. Nonaka also proposed "ba" where these syntheses are conducted [4]. Hijikata proposed a computerized "ba" where two domain experts externalize and combine their tacit knowledge efficiently with the help of a computer that points out inconsistency among tacit knowledges of two domain experts and knowledge created by inductive case learning [5].

SECI model focuses on organizational knowledge creation. On the other hand, our proposal mainly focuses on its transfer to next-generation engineers, and emphasizes that they can do jobs using the transferred knowledge at various situations. Therefore, internalization is important so that they can apply the knowledge to various situations.

Davenport discussed more practical aspects of knowledge management, including effective use of information technology and considering cultural aspect of an enterprise [6]. He points out that coding of knowledge is important for its sharing but that coding of complex tacit knowledge is very costly and difficult. Even if coding of knowledge is done with a great effort, without face-to-face exposure, sometimes it can hardly transferred to another group, because of the lack of the relationship of trust based on the face-to-face exposure. Even with such recognition, Davenport also points out importance of explicit knowledge repositories using information technology. For the effective use of them, to develop a thesaurus is important so that a necessary knowledge can be retrieved precisely by semantic query. It is also pointed out that an expert system is also useful for a specific technical domain that is stable and where knowledge is not difficult to be coded and needs to be updated only gradually.

Weber pointed out the similarity between externalization of tacit rules to explicit business rules in knowledge management and the knowledge acquisition in an expert system, and showed that expert system, case-based reasoning, and ontologies could be methodologies for knowledge management [7]. In the above referred Hijikata [5], it is presupposed that the domain experts externalize their tacit knowledge to if-then style rules.

From the point of transfer of intelligence skill to the next generation, its externalization is not difficult compared to that of craft skill, and the externalization is well motivated since the transfer without OJT is an urgent issue. The transfer is to the next generation and not to another group, and hence there is no barrier caused by the lack of mutual trust. The knowledge to be transferred is the one that has been accumulated from generation to generation and needs to be updated only gradually, and, therefore, if it belongs to a specific technical domain, an expert system can be effectively used. Since the transfer is from domain experts to novices, the objectives of an expert system should include both to help domain experts externalize their tacit intelligence skills and to help novices internalize them.

2.2 Knowledge Modeling

For an expert system, knowledge acquisition and updating are always the bottleneck of its development and maintenance. To solve the problem, knowledge engineering now treats building a knowledge-based system as a modeling activity, instead of simply extracting and transferring the knowledge of domain experts to computer-executable code. This knowledge modeling, such as expertise modeling of CommonKADS [8], improves the maintainability of the system, in addition to its cost-effective development, providing implementation-independent knowledge-level description of the system's problem solving processes. It consists of a reusable PSM (problem solving method) and domain ontologies. The former provides a template for describing knowledge-level problem solving processes and the latter describes the static structure of the domain knowledge that is used to solve the problems.

What we focus on in this paper is transfer of knowledge from domain experts to novices, and not to a knowledge-based system. But, for that, since we will use a kind of knowledge repository and domain experts are expected to externalize their tacit knowledge to the repository, this kind of knowledge modeling is also important for our proposal and its utility for novices to internalize the externalized knowledge should also be paid attention to.

2.3 Ontology

Nowadays, an ontology is widely used also in the area of knowledge management. The famous definition of an ontology is "an explicit specification of a conceptualization" [9], but more intuitively an ontology consists of terms and semantic relations among them that characterize the concepts that terms represent.

Lau presented ontology-based skill management at a large insurance company [10]. There, ontologies of skills, educations, and functions are developed so that description on skill information of the employees can be standardized. They make it possible for semantic query to retrieve necessary information accurately. Morgan presented an ontology used for a case-based reasoning of dimensional management in a vehicle assembly plant [11]. There, an ontology of concepts that are necessary to describe cases is developed and a case description can be generated by selecting appropriate terms in the ontology from pull-down menus. Then, the case descriptions are standardized and a measure of similarity among cases can be introduced and an appropriate case can be easily retrieved. In both cases, the purpose of introducing ontologies is for accurate knowledge retrieval and is similar to the one of a thesaurus. But, the benefits of ontologies are not limited to this. An ontology can also be used to help human understand knowledge and share knowledge with each other.

For human understandability, a primitive that constitutes an ontology is not necessarily a term. For example, MIT Process Handbook [12] is a collection of descriptions of business processes in a natural language. But since they are structured by semantic relations such as generalization-specialization and uses-parts, it is sometimes called an ontology. Since understandability is an important factor of the transfer of intelligence skill from domain experts to novices, in this paper, we also take an ontology in this broad sense.

3. Ontologies and a Rule-Based System for Organizational Transfer of Intelligence Skill

As stated above, some characteristics of the transfer of intelligence skill are as follows:

- The knowledge has been accumulated from generation to generation and needs to be updated only gradually.

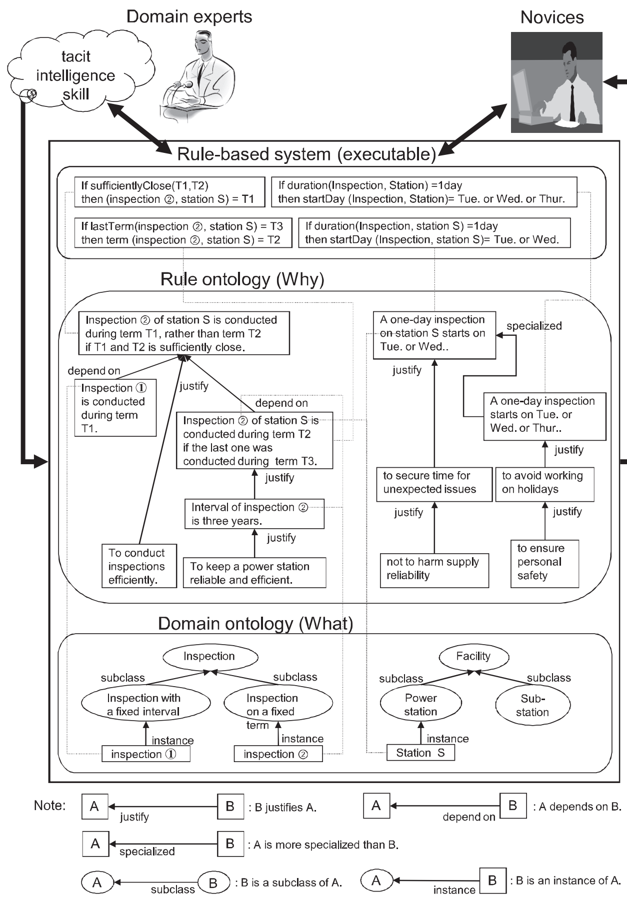

Fig. 1 Overview of the proposal.

- Externalization is relatively easy, compared to craft skill, if a domain is specified.
- Internalization is important so that the transferred knowledge can be applied to various situations.

Based on these characteristics, we propose a domain ontology, a rule ontology and a rule-based system that help domain experts externalize their tacit intelligence skills and also help novices internalize them, for a specific domain that requires intelligence skill. Figure 1 shows the overview of our proposal with a simplified example. Explanations follow in the subsequent sections.

3.1 Domain Ontology and Rule Ontology

Both a domain ontology and a rule ontology are what tacit intelligence skills of domain experts are externalized and combined as. A domain ontology consists of the technical terms that are used in the domain and the semantic relations among them. This is used to standardize the description of intelligence skill externalized by domain experts and to eliminate dependency on each domain expert. It is also used for novices to understand the domain knowledge. So, each term in a domain ontology has a description of its meaning in a natural language.

We claim that in most cases, tacit intelligence skill can be decomposed into pieces of simple knowledge, and that each of them can be externalized as one or a few sentences in a natural language with several tens of words. Hereafter, we call it an engineering rule. A domain ontology plays an important role for novices to understand each engineering rule. But, there are some additional points for them to know to get the high-level picture of the intelligent skill and to do the jobs properly. First, to apply a rule properly, they need to know why it is adequate to apply it. Second, they also need to know that some rules can be applied only after some other rules are applied. Finally, to understand the rules well, they need to know that some rules are specialized from more general rules for more specific situations, in some cases, with some overrides. To provide novices these kinds of knowledge, a rule ontology is introduced. A rule ontology consists of engineering rules as primitives and semantic relations among them. From the objectives of a rule ontology, its semantic relations are mainly as follows:

- relation "justify", which is a relation between a rule and another rule that is justified by the rule
- relation "depend on", which is a relation between a rule and another rule whose application is prerequisite for its application
- relation "specialized", which is a relation between a specialized rule and its generalized rule
- relation "override", which is a relation between a specialized rule with some overrides and its generalized rule that it overrides

If a rule justifies another rule, we say the former is deeper than the latter or the latter is shallower than the former. We also call the former a deep rule and the latter a shallow rule. The deepest rules are the rules that no rules are deeper than and the shallowest rules are the rules that no rules are shallower than. In most cases, the shallowest rules are applied directly to perform the jobs, because otherwise the shallowest rules are not necessary since they justify no rules. The deepest rules mainly express the objectives and basic constraints of the jobs and have a similar role of the top-level task description by PSM. The reason why we introduce a rule ontology, rather than PSM, is that a rule ontology is more flexible to be applied to various situations and also easier for novices to understand.

A rule ontology is a kind of extension of explanation facilities based on deep knowledge in an expert system [13] and has some similarity to coarse-grain intelligent content [14]. A coarse-grain intelligent content consists of mostly single sentences because a single sentence is a chunk for human to understand easily. But, a rule ontology consists of engineering rules, which are usually more than a single sentence to keep an engineering meaning but smaller than a description of a business process in MIT Process Handbook [12].

3.2 Rule-Based System

These ontologies are expected to be developed and maintained by domain experts, but it is not easy for them because they themselves got the intelligence skills by OJT and have no experience of externalizing and combining them to ontologies. Especially, it is difficult for them to recognize all the rules explicitly and precisely since they do the job based on their tacit intelligence skills and not based on the ontologies.

One of the objectives of a rule-based system is to help them refine rules in a rule ontology. A rule-based system has executable rules that correspond to the shallowest rules that are directly used to do the job. Domain experts can refine rules in a rule ontology, checking the outputs from the rule-based system. A rule ontology, giving domain experts a high-level picture, reduces the difficulty in maintaining the rule-based system, and conversely the rule-based system motivates domain experts to develop and maintain the ontologies, giving them the output of the job semi-automatically on behalf of them.

A rule-based system also helps novices internalize the intelligence skills of the domain experts. Even though the intelligence skills are externalized and combined as ontologies that novices can understand, it is still difficult for novices to internalize them. To internalize them, novices need experience to do the job using them. Using a rule-based system, novices can compare the outputs by themselves with the ones from the rule-based system as many times as necessary. Then, novices can sufficiently internalize them.

4. GEN (General knowlEdge Navigator)

To support and examine our proposal, we have also developed an experimental ontology repository, specialized for knowledge management, called GEN (General knowlEdge Navigator). GEN provides a "ba" where domain experts externalize and combine their tacit intelligence skills to ontologies and also novices internalize them.

To develop GEN, we have used Squeak, a kind of Smalltalk, which is suitable for agile development of an experimental system. GEN is similar to Protégé [15] but is more end-user oriented and suitable for structured textual information like a rule ontology. Figure 2 shows how a semantic relation of a rule ontology is represented in GEN. When a slot, for example, "override", is defined and its link is added in GEN, GEN automatically defines its inverse slot "overriddenBy" and also adds the inverse link. GEN also has a rule-based system as its subsystem. This rule-based system is developed based on Backtalk. Backtalk is a library of Smalltalk for finite domain constraint programming, with object-oriented features [16].

GEN has a client-server architecture and stores ontologies in omniBase, a database management system for Smalltalk, on a server, to support concurrent use. One of the features of GEN is that it is almost independent of computer environments although it has a client-server architecture. GEN client can run on any client environment so far as Squeak virtual machine runs. Also to minimize the dependency on server environments, GEN invokes no process on a server and omniBase can be installed on any file server as shared files.

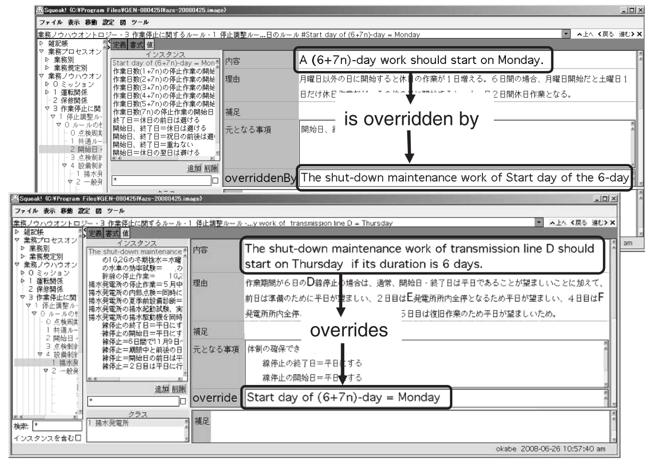

Fig. 2 Snapshot of GEN.

5. Case Study from TEPCO

Having described our proposal, let us turn to a case study and examine our proposal in details. In this case study, we have focused on some specific job on hydroelectric power stations at some remote control and maintenance office in TEPCO. It is so-called "inspection and maintenance work scheduling job" [17].

What we did first was similar to OJT. We, as novices, tried to do the job, referring to the previous outputs, under the direct supervision of the domain experts of this job. Second, we externalized what we learned, and the domain experts reviewed and refined them. Then, the domain experts and we collaboratively reorganized and combined them as ontologies in GEN. We also developed a rule-based system specific to this job, that is, a scheduling subsystem. It was first developed in Prolog [18], and transferred to GEN. Using this rule-based system, the domain experts and we refined the ontologies repeatedly.

5.1 Case Description

The system of hydroelectric power stations in TEPCO has a long history and is now highly automated and integrated. All the hydroelectric power stations are unmanned and a remote control and maintenance office is responsible to remote-operate and maintain all the hydroelectric power stations along a river system, which vary from very old small ones to state-of-the art large-scale pumped storage ones.

The "inspection and maintenance work scheduling job" is mainly to make out yearly inspection and maintenance work schedules of generators and other devices of all power stations controlled by a remote control and maintenance office. The schedule is made out so that it minimizes discharged water, which is water not used for power generation, under various constraints such as:

- statutory inspection interval of each device
- agreements with agricultural unions and other outside associations
- natural environment conditions
- operational conditions among interrelated facilities etc.

This is a typical job that needs intelligence skill since it requires a variety of integrated knowledge and logical thinking. This is, however, not a well-defined optimization problem. Some of the constraints are not mandatory but desirable and in most cases there is no feasible solution that satisfies all the constraints. In case that there is no strictly feasible solution, sophisticated intelligence skill is required to determine what constraints should be loosened, depending on a situation. Most of the knowledge has not been externalized and the skill for the job has been transferred by costly OJT.

5.2 Rule Ontology

What we learned from the domain experts was decomposed into 134 engineering rules. Figure 3 shows the distribution of character counts of the rules in Japanese. Most of the rules have brief supplementary notes so that a novice can understand them well and the character count is the one that includes the supplementary note.

Among the 134 rules, 90 rules were the shallowest rules that are used directly to make out the schedule, and the rest are deep rules that justify other rules. As a rule becomes shallower, its character count tends to be longer, since a shallower rule includes more specific conditions under which it can be applied. The deepest rules were classified into two categories. One is the objectives and basic constraints of the job and the other is the objective facts. A rule in the objective facts expresses an objective fact such as the phenomena or the laws of natural science that needs to be recognized for the job.

Figure 4 shows a simplified fragment of the rule ontology. This is about a scale removal work of a conduit pipe of a particular power station. "To keep facilities efficient" and "Not to harm electric supply reliability" are the objectives and basic constraints of the job and "Scale grows during the summer season" is an objective fact. If we see its deepest rules, we can easily understand that it is done not only for efficiency of facilities but also for electric supply reliability. And even though scale grows during the summer season, the 3rd removal work is scheduled after the 3rd week of September because no shutdown maintenance work is done during the summer peak season not to harm electric supply reliability.

We also developed a class hierarchy of the rule ontology as shown at Fig. 5, where each rule is treated as an instance. There are two reasons why a rule is treated as an instance and a class hierarchy is introduced on it. One is to define semantic relations among rules uniformly in a schema and the other is to make it easy for novices to apply neces-

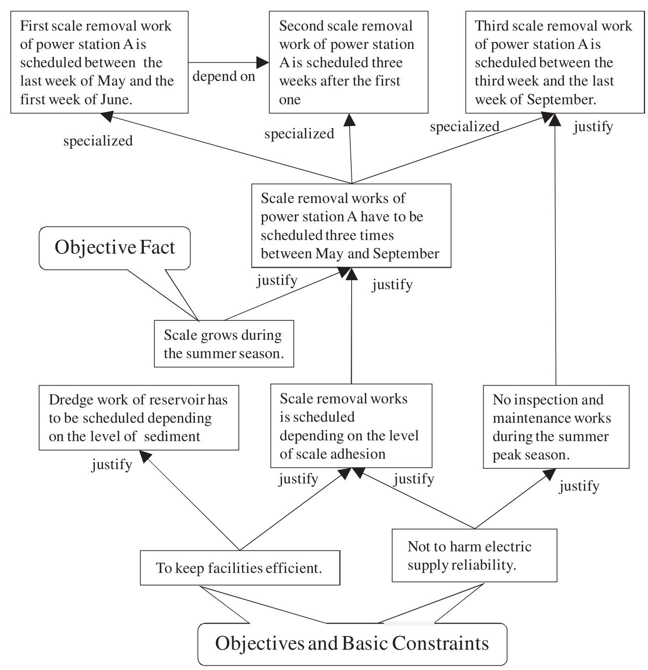

Fig. 4 Fragment of the rule ontology.

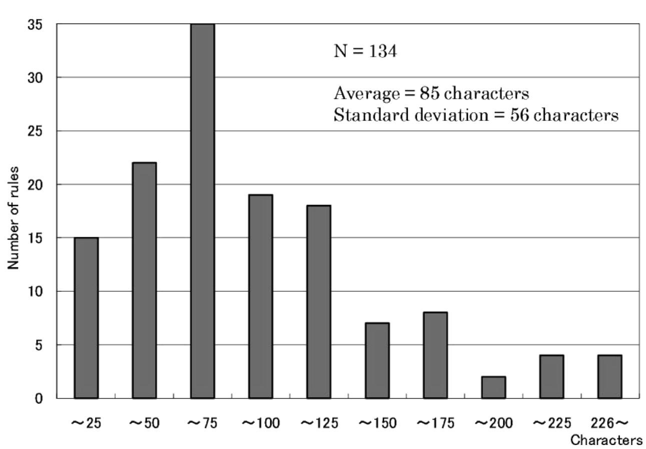

Fig. 3 Distribution of length of the rules.

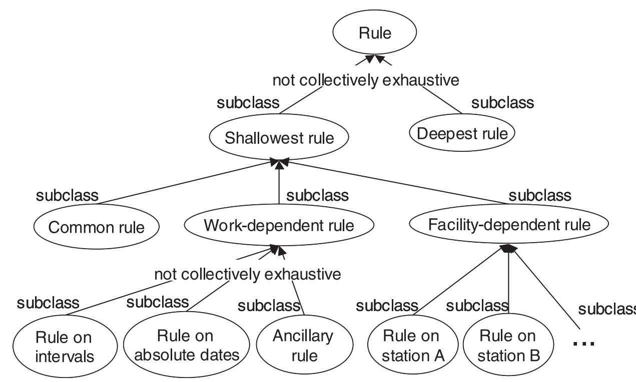

Fig. 5 Class hierarchy of the rule ontology.

Table 1 Number of classes and instances.

	Classes	Instances	Note
Rule ontology	20	134	90 instances among 134 intances were converted to a rule base.
Domain ontology	55	292	

sary rules to proper works. Hence, a subclass does not necessarily have additional structure to its superclass. Table 1 shows number of classes and instances of the rule ontology.

5.3 Domain Ontology

Domain ontologies were created from the technical terms in the rules. Table 1 shows number of classes and instances of them. Since the domain is scheduling of inspection and maintenance works of hydroelectric power station facilities, main domain ontologies are a facility ontology and a inspection and maintenance work ontology (hereafter, work ontology) [19].

The facility ontology should have information on facilities such as power stations, transmission lines etc. TEPCO already has a kind of class hierarchy of facilities with a long history although TEPCO does not call it a class hierarchy. Basically we adopted this class hierarchy for the facility ontology because the facility ontology should be commonly applicable to a broader domain. Based on this class hierarchy, information and semantic relations that are necessary to “inspection and maintenance work scheduling job” are introduced. Between the facility ontology and the rule ontology, there are two semantic relations. One is an instance-instance relation between an instance of the facility ontology and the rules that apply to it. The other is a class-instance relation between a class of the facility ontology and the rules that commonly apply to all instances of the class. Figure 6 shows how the facility ontology is represented and related to the rule ontology in GEN. An arrow marked with (1) shows a link from an instance of the facility ontology to the rule that applies to it. An arrow marked with (2) shows its inverse link automatically generated. An arrow marked with (3) shows a link from a class of the facility ontology to the rule that commonly applies to all instances of the class. An arrow marked with (4) shows its inverse link automatically generated.

The work ontology should have information that is necessary for inspection and maintenance works. In real situations, it should have the information how works can be done and the information for scheduling them is a small part of it. But, since to develop the work ontology for real situations is far beyond the scope of the case study, the work ontology we developed for this case study had only a relation between a work and the rules that are applied to it. Each work here is not a particular work done at some particular date and time and is more natural to be treated as a class, but for the same reason as the rule ontology, it was treated as an instance and the class hierarchy is basically based on classi-

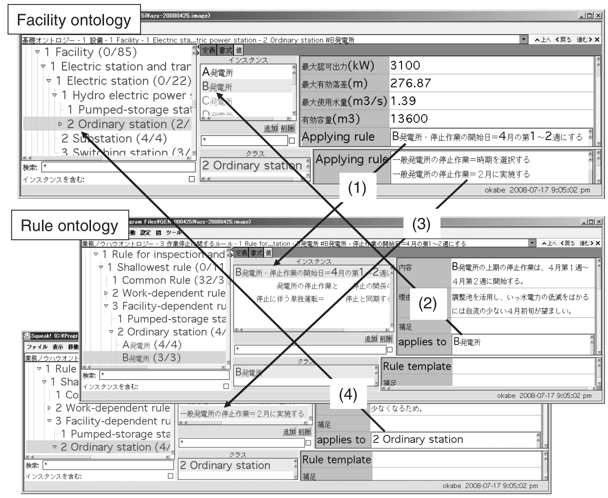

Fig. 6 Examples of the facility ontology and the rule ontology.

fication by power stations. Similar to the facility ontology, the work ontology has an instance-instance relation and an class-instance relation to the rule ontology.

5.4 Scheduling Subsystem

The shallowest rules (90 rules among 134 rules, see Table 1.) were converted to the executable format on a reasoner based on Backtalk in GEN. Taking advantage of an object-oriented language, the reasoner can interpret the semantics of super-subclass relations of the facility ontology and the rule ontology, regarding what rule are applied. Hence, the structure of the rules that are suitable for human understanding could be transferred as it was to the rule base of the reasoner.

However, it was not easy to get a proper solution. As mentioned before, this is not a well-defined problem and most of the rules are not mandatory and may contradict. Here is a simplified example to illustrate how they contradict.

Rule 1. day(startDate(inspection1))=Wednesday
Rule 2. startDate(inspection1)=March 1st
where inspection1 is an individual name to be scheduled for some specific inspection.

Then, a question is which of Rule 1 or Rule 2 should be excluded or loosened if March 1st is not Wednesday in the target year. This kind of situation occurs very often usually in much more complex manners. To treat these situations properly is an important part of the intelligence skill of this job. First, we tried to enumerate all such cases and give a specific solution for each case. But, there are a lot of cases and it is difficult to investigate all the cases. Moreover, even if all the cases are investigated at one time with a great effort, a lot of new cases may appear when a rule is added or updated and it is almost impossible to maintain them. Hence, we changed the strategy. Giving up enumerating all the infeasible cases, we decided to give each rule a priority num-

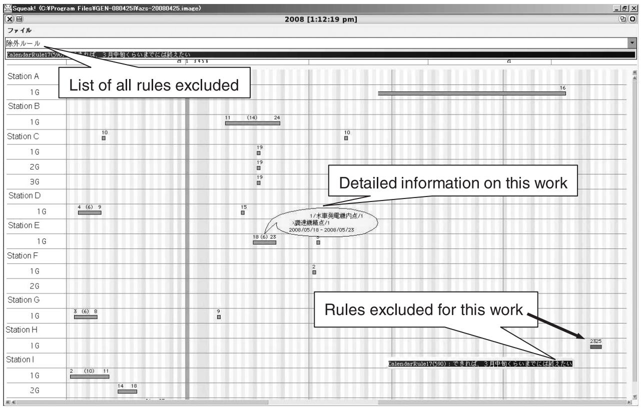

Fig. 7 Output of the scheduling subsystem.

ber. If there is no feasible solution, then rules that have the least priority are simply ignored. Now, the way to externalize the tacit knowledge to treat infeasible cases became dramatically simple. It now is to give priority numbers to all the rules by trial and error using the scheduling subsystem so that they can work well in any cases. Practically, we can assume that if they work well in many cases, they presumably work well in all cases. By trial and error, we were successful in giving a suitable priority number to each rule. Figure 7 shows a part of the output based on them [20], [21].

6. Evaluation and Discussion

In this section, we evaluate the case study in the previous section, from the point of internalization by novices and also from the point of externalization and combination by domain experts. For the former, we have done an experiment for the evaluation [22], and the latter was evaluated by how the domain experts could developed the ontologies with us.

6.1 Internalization by Novices

6.1.1 Hypothesis and Experiment Design

To evaluate the internalization by novices, first, we made a hypothesis, next have done an experiment, and then examine and discuss how well the result supports the hypothesis.

We do not claim that using GEN a novice can get the exactly same level of skill for the job as OJT. But, since this job is a typical job that needs an intelligent skill as stated in Sect. 5.1, the rule ontology as well as the domain ontologies were expected to help a novice get the knowledge of the job efficiently, and also the scheduling subsystem could help a novice internalize it, instead of OJT. Hence, the hypothesis we made is that using the ontologies and the scheduling subsystem in GEN, a novice can get the intelligence skill for the job close to the level of the one by OJT in a shorter time compared to OJT.

To examine the hypothesis, we have done an experiment, where five examinees who got the skill in several ways solved same scheduling problems. Two of them (hereafter, examinees A and B) were selected from persons who did not have any experience nor background of TEPCO's work. Examinees A and B self-learned the skill using the ontologies and the scheduling subsystem in GEN. No time limit for learning the skill was given. They learned until they felt that they learned enough. To compare these examinees, we should have had the examinees who got the skill by OJT. But, it was impossible to find a suitable examinee because the organization of the remote control and maintenance office had already been downsized and there was no one who got the skill by OJT in the last few years. Therefore, instead, the rest three (hereafter, examinees C, D, and E) were selected from the members of our projects. Two of them (examinees C and D) were selected because they were the original project members and the way in which they got the skill was similar to OJT. They got the skill under the direct supervision of the domain experts, through interviewing them and developing the ontologies together with them. The last one (examinee E) was also the member of our project, but joined it when the ontologies were almost developed but were not stored in GEN. He was mainly responsible for developing the rule-based system. Examinee E was selected because he learned the skill, using the ontologies not in GEN and the comparison with examinees A and B were expected to suggest the pros and cons of GEN as a tool.

6.1.2 Result of Experiment

Table 2 shows the experiment result for each examinee, with his or her profile. Table 3 shows the summary breakdown by the method of learning at Table 2. The average accuracy rate of examinees A and B was close to that of examinees C and D (88% vs. 91%), although the average learning time of examinees A and B was about a quarter of that of examinees C and D (47 hours vs. 199 hours, excluding time for interview). This result supports the hypothesis that using the ontologies and the scheduling subsystem in GEN, a novice can get the intelligence skill for the job close to the level of the one by OJT in a shorter time compared to OJT. Although not by self-learning by GEN, examinee E got even better accuracy rate as that of the examinees C and D (96% vs. 91%), with about two thirds of the learning time (138 hours vs. 199 hours, excluding time for interview). This is an example that suggests the effectiveness of systematic learning using ontologies, even without GEN. The reason why examinee E got such high accuracy rate is analyzed in the next subsection.

6.1.3 Discussion on Erroneous Answers

Erroneous answers were classified into three categories by cause. The first category is the erroneous answers caused by "lack of description", which mean the ones caused by lack of description of necessary knowledge in GEN. All erroneous answers in this category were made by examinees A and B. Examinee A made 3, examinee B made 5 and totally there were 8 erroneous answers in this category (see

Table 2 Result and profile of each examinee.

Examinee		A	B	C	D	E
Method of learning	For knowledge acquisition	Self-learning by GEN		By interviewing domain experts		Self-learning by ontologies
	For internalization	Self-learning by GEN		Through developing ontologies	Through hand simulations	Through developing a rule-based system
Learning time (in hours)	For interview	-	-	22	22	-
	Other than interview	55	39	198	200	138
Material for solving		GEN			Private documents	
Solving time (in minutes)		320	250	330	255	255
Accuracy rate (%)		88	88	90	92	96
Erroneous answers (among 130 questions)		15	16	13	10	5
Breakdown by cause	Lack of description	3	5	0	0	0
	Ambiguity of description	5	3	5	5	1
	Careless mistake	7	8	8	5	4

Table 3 Result by method of learning.

Method of learning	For knowledge acquisition	Self-learning by GEN	By interviewing domain experts	Self-learning by ontologies
	For internalization	Self-learning by GEN	Other than GEN	
Examinee		A, B	C, D	E
Average learning time (in hours)		47	199	138
Average solving time (in minutes)		285	293	255
Average accuracy rate (%)		88	91	96
Average erroneous answer (among 130 questions)		15.5	11.5	5.0
Breakdown by cause	Lack of description	4.0	0.0	0.0
	Ambiguity of description	4.0	5.0	1.0
	Careless mistake	7.5	6.5	4.0

Note: "Average learning time" does not include time for interview.

row "Lack of description" at Table 2).

There were two kinds of points that lack description. The first kind is the ones that are almost obvious for the person who has the least background in the job. But, examinees A and B had no background in any jobs of TEPCO and made four erroneous answers. For example, examinee A made two erroneous answers that a simple inspection was allocated just after minute one. The fact is that a minute inspection subsumes a simple inspection and that a simple inspection is not necessary to be allocated after a minute one. But, this very basic rule was not described in GEN explicitly since it is almost obvious for all the engineers in the remote control and maintenance office, but examinee A did not know this rule. Examinee B also made two similar erroneous answers because of another almost obvious point that is not described in GEN. This indicates that practically an ontology cannot be independent of how much background its user has since a fully comprehensive ontology can be hardly developed because of its development cost and hence also suggests that novices should also participate in developing ontologies from the early stage.

The other kind is the lack of the procedural information how scheduling should be done efficiently. As stated before, the problems do not necessarily have a strictly feasible solution that satisfies all the rules, and some of the rules may have to be loosened. A domain expert can easily and intuitively find a schedule that satisfies most of the rules and that needs only a few adjustments. But, a novice could hardly find such an initial schedule, the initial schedule tended to need a large amount of adjustments, and consequently he or she loosened the rules that should have not been loosened. All the remaining 4 erroneous answers in this category were such a kind of ones. Examinee A made 1, while examinee B made 3. The difference of the numbers of the erroneous answers made by examinees A and B is explained as follows: Each of examinees A and B established his or her own sequence in which works were scheduled. Examinee B scheduled works each month simply in the order that they appeared in the scheduling problems and this caused more problems of a large amount of adjustments. As stated in Sect. 5.4, each rule in the scheduling subsystem has its own priority number. In the case that a complete feasible solution cannot be found, rules are loosened in order of increasing priority number until a solution can be found. Similarly, if examinees A and B had applied the rules in order of decreasing priority number, it could have reduced the amount of necessary adjustments. But, in GEN, the priority numbers were only for the reasoner and did not have a suitable interface for human.

The second category is the erroneous answers caused by "ambiguity of description", which mean the ones caused by ambiguity of natural language description of the rules in GEN. For the erroneous answers in this category, examinees A and B made 4 erroneous answers on an average, while ex-

Table 4 Result by material for solving.

Material for solving	GEN	Private documents
Examinee	A, B, C	D, E
Average solving time (in minutes)	300	255
Avarage careless mistakes	7.7	4.5
Avarage ratio of careless mistakes (%)	5.9	3.5

aminees C and D made 5, and there is not a big difference. But, examinee E made only 1 (see row "Ambiguity of description" at Table 3). This is the main reason why examinee E attained the higher accuracy rate than the others. The reason why examinee E made only 1 erroneous answer is that examinee E was responsible for developing the rule-based system and had to disambiguate the rules. This collaterally supports that a rule-based system can contribute higher accuracy of externalization by domain experts and internalization by novices.

The last category is careless mistakes. Examinees A and B made 7.5 careless mistakes on an average and examinee C and D made 6.5 (see row "Careless mistake" at Table 3), and there is not a big difference. But, when we see the summary breakdown by material for solving the problems at Table 4, there is a significant difference. The examinees who used GEN for solving the problems (examinees A, B and C) made 7.7 careless mistakes, but the examinees who used private documents (examinees D and E) made 4.5. In addition, the average solving time of the former was 300 minutes, although the one of the latter was 255 minutes. The reasons are:

- GEN is an experimental system and is not equipped with a good user interface, especially is not good at browsing information at a glance.
- A paper document is better than a computer display for browsing this size of information (134 rules) at a glance.
- For each individual, his or her private documents that reflect his or her background has better understandability than GEN that does not have a facility to personalize its information.

The fact that even with these kind of weaknesses of GEN, examinees A and B gained the accuracy rate close to the one of examinees C and D, only with about quarter of the learning time suggests the effectiveness of our proposal.

6.2 Externalization and Combination by Domain Experts

In this case study, the domain experts and we collaboratively developed the ontologies. The profiles of the domain experts who joined in developing the ontologies are shown at Table 5. Domain expert F was not currently responsible for "inspection and maintenance work scheduling job" but had a lot of experiences on almost on all the jobs in the remote control and maintenance office and had strong concern about the skill transfer to the next generation. Domain expert G was currently responsible for "inspection and maintenance work scheduling job" and had a lot of experience related to the scheduling job both in the remote control and maintenance office and in the related divisions.

Table 5 Profiles of domain experts.

Domain expert	F	G
Currently responsible for "inspection and maintenance work scheduling job"	No	Yes
Experience of "inspection and maintenance work scheduling job" (in years)	(15)	5
Experience in the remote control and maintenance office (in years)	24	15
Experience in other remote control and maintenance offices (in years)	8	-
Experience in other related devisions (in years)	-	15
Total experience in TEPCO (in years)	32	30

Note:
1: The number in parentheses means years of the experience of supervising the "inspection and maintenance work scheduling job".
2: "Experience in the remote control and maintenance office" includes "Experience of inspection and maintenance work scheduling job".

After interviewing the domain experts for the job, first, we developed the basic structure (class hierarchy and semantic relations between classes) of the ontologies and some of their instances. Then, the domain experts reviewed them and made comments for improvement. We discussed the comments and made necessary refinements mainly on the structure and how to write an instance of the rule ontology in a natural language. We went through these refinements three times until the structure of the ontologies was almost fixed.

Then, the domain experts started to add necessary instances under the supervision of us. The domain experts, however, had little idea what extent of detail and carefulness was necessary in externalization so that the novices could understand it. For example, they tended to externalize only the deepest and the shallowest rules because they did not recognize that the novices had difficulty in relating them without in-between rules. A part of the ontologies we had already developed had a major role in solving the problem. First, the description in a natural language of each instances in the rule ontology was a good guide to show what extent of detail and carefulness was necessary. Also the semantic relations among the rule ontology and between the rule ontology and the domain ontologies showed what were necessary to be externalized.

The scheduling subsystem motivated and helped the domain experts, especially domain expert G, who was currently responsible for the job, to refine the rule ontology. Domain expert G highly evaluated the scheduling subsystem because, on behalf of him, it could completely apply simple but tedious rules, such as simple calendar checking rules, which even the domain experts are likely to fail to apply completely. And these rules were the rules that the domain experts tended to forget to externalize, and hence the scheduling subsystem motivated and helped domain expert G to check and refine the rule ontology.

Since the novices could do the job well using the ontologies as shown at the previous sub-section, it can be concluded that the ontolgies were successfully developed by the domain experts and us in this particular case, except that the novices should have joined in developing the ontologies from the early stage to prevent a lack of description depending on their backgrounds. But, it is still not clear what kind of support is necessary for domain experts to develop ontologies and a rule-based system by themselves without us.

7. Summary and Future Work

In this paper, we have proposed ontologies and a rule-based system that enable organizational transfer of intelligence skill without OJT. Intelligence skill is decomposed into pieces of knowledge and they are externalized as engineering rules and technical terms that constitute them. They are combined as a rule ontology and domain ontologies respectively. A rule ontology consists of engineering rules as primitives and semantic relations among them such as "justify", "depend on" etc. The shallowest rules are translated into a rule base so that they can be executed on a rule-based system. The rule-based system motivates domain experts to externalize and combine their tacit intelligence skill to the ontologies and also help domain experts refine them and novices internalize them.

Accompanied with GEN, this proposal was experimentally applied to the actual job at a remote control and maintenance office in TEPCO and the evaluation of this case study, including the result of the experiment, supports the proposal. The lessons learned from the case study include that novices, in addition to domain experts, are encouraged to participate in developing ontologies since ontologies are practically depend on the backgrounds of its users.

In the next step, we will do a more comprehensive and long-term case study using enhanced GEN, to confirm the effectiveness of our proposal and to examine what kind of support is necessary for domain experts:

- to develop ontologies and a rule-based system from scratch by themselves,
- to maintain them for longer term.

In addition, since we have found that novices, as well as domain experts, have an important role to develop and improve ontologies, we will enhance GEN to provide a "ba" where both domain experts and novices collaboratively externalize their tacit intelligence skills and refine them by communicating each other.

Acknowledgments

The authors appreciate the participation of Masahiko Yanagisawa and Hiroshi Yamazaki, Tokyo Electric Power Co., Inc. as domain experts. We also wish to thank Mr. Toshiaki Kurokawa, CSK Holding Cooperation, Masahiro Ohira, Miho Kato, Yurie Takeda and Tatsuya Ishikawa, Keio University for their contributions.

References

[1] K. Watanuki and K. Kojima, "New approach to handing down of implicit knowledge by analytic simulation," J. Advanced Mechanical design, Systems, and Manufacturing, vol.1, no.3, pp.48–57, 2007.
[2] H. Chuma, "Problem finding and solving skill in manufacturing factories," The Japanese Journal of Labor Studies, vol.510, 2002.
[3] I. Nonaka and H. Takeuchi, The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation, Oxford University Press, 1995.
[4] I. Nonaka and N. Konno, "The concept of "ba": Building a foundation for knowledge creation," California Management Review, vol.40, no.3, pp.40–54, 1998.
[5] Y. Hijikata, T. Takenaka, Y. Kusumura, and S. Nishida, "Interactive knowledge externalization and combination for SECI model," Proc. 4th International Conference on Knowledge Capture, pp.151–158, Whistler, BC, Canada, 2007.
[6] T.H. Davenport and L. Prusak, Working Knowledge: How Organizations Manage What They Know, Harvard Business School Press, 1998.
[7] R. Weber and R. Kaplan, "Knowledge-based knowledge management," in Innovations in Knowledge Engineering, International Series on Advanced Intelligence, vol.4, pp.151–172, 2003.
[8] G. Schreiber, H. Akkermans, A. Anjewierden, R. Hoog, N. Shadbolt, W.V. Velde, and B. Wielinga, Knowledge Engineering and Management The CommonKADS Methodology, MIT Press, 1999.
[9] T.R. Gruber, "A translation approach to portable ontology specifications," Knowledge Acquisition, vol.5, no.2, pp.199–220, 1993.
[10] T. Lau and Y. Sure, "Introducing ontology-based skill management at a large insurance company," Proc. Modellierung 2002, Tutzing, Germany, 2002.
[11] A.P. Morgan, J.A. Cafeo, D.L. Gibbons, R.M. Lesperance, G.H. Sengir, and A.M. Simon, "CBR for dimensional management in a manufacturing plant," Lecture Notes in Artificial Intelligence 2080, pp.597–610, Springer Verlag, 2001.
[12] T.W. Malone, K. Crowston, and G.A. Herman, Organizing Business Knowledge: The MIT Process Handbook, MIT press, 2003.
[13] T. Yamaguchi, R. Mizoguchi, N. Taoka, H. Kodaka, Y. Nomura, and O. Kakusho, "Explanation facilities for novice users based on deep knowledge," IEICE Trans. Inf. & Syst. (Japanese Edition), vol.J70-D, no.11, pp.2083–2088, Nov. 1987.
[14] K. Hashida, "Semantics platform based on ontologies and constraints," J. Japanese Society for Artificial Intelligence, vol.21, no.6, pp.712–717, 2006.
[15] Protégé, http://protege.stanford.edu/
[16] P. Roy and F. Pachet, "Reifying constraint satisfaction in smalltalk," J. Object-Oriented Programming, vol.10, no.4, pp.43–51, 1997.
[17] T. Iwama, H. Tachibana, H. Yamazaki, M. Okabe, T. Kurokawa, K. Kobayashi, M. Kato, A. Yoshioka, and T. Yamaguchi, "A consideration of ontology that supports organized business knowledge accumulation and retrieval," 3rd Annual Conference of Information Systems Society of Japan, Niigata, Japan, 2007.
[18] K. Kobayashi, M. Kato, A. Yoshioka, and T. Yamaguchi, "Constructing business rules and domain ontologies to support externalization of business knowledge in knowledge transfer," 22nd Annual Conference of the Japanese Society for Artificial Intelligence, Asahikawa, Japan, 2008.
[19] A. Yoshioka, M. Ohira, T. Iijima, T. Yamaguchi, H. Yamazaki, M. Yanagisawa, and M. Okabe, "Supporting knowledge transfer and scheduling task with ontologies," 21st Annual Conference of the Japanese Society for Artificial Intelligence, Miyazaki, Japan, 2007.

[20] M. Okabe, M. Yanagisawa, H. Yamazaki, K. Kobayashi, A. Yoshioka, and T. Yamaguchi, "Organizational knowledge transfer of intelligence skill using ontologies and a rule-based system," Practical Aspects of Knowledge Management, 7th International Conference, PAKM 2008 Proc., pp.207–218, 2008.
[21] M. Okabe, M. Yanagisawa, H. Yamazaki, K. Kobayashi, A. Yoshioka, and T. Yamaguchi, "Ontologies that support organizational knowledge transfer of intelligence skill," Interdisciplinary Ontology vol.2, Proc. Second Interdisciplinary Ontology Meeting, pp.147–159, Tokyo, Japan, 2009.
[22] K. Kobayashi, Y. Takeda, A. Yoshioka, M. Okabe, M. Yanagisawa, H. Yamazaki, and T. Yamaguchi, "Organizational knowledge transfer with ontologies and a rule-based system," 23rd Annual Conference of the Japanese Society for Artificial Intelligence, Kagawa, Japan, 2009.

Masao Okabe received the M.S. degree in statistics and in operations research from Stanford University in 1987. He is currently a Ph.D. student at the school of Science and Technology at Keio University and working at Tokyo Electric Power Co., Inc. His research interests include ontology engineering, knowledge management and knowledge-based software engineering. He is a member of JSAI and JASMIN.

Akiko Yoshioka received the B.E. and M.E. degrees from Keio University in 2007 and 2009, respectively. She is currently working at Nissan Motor Co., Ltd. Her research interests include knowledge management and ontology engineering.

Keido Kobayashi received the B.E. degree in Administration Engineering from Keio University in 2008. He is currently a master course student at the school of Science and Technology at Keio University. His research interests include knowledge management and ontology engineering.

Takahira Yamaguchi is a professor at the Faculty of Science and Technology at Keio University. He received his B.E., M.E., and Ph.D. degrees in telecommunication engineering from Osaka University in 1979, 1981, and 1984, respectively. His research interests include Ontology Engineering, KBSE, Advanced Knowledge Systems, and Data Mining. He is a member of IPSJ, JSAI, JSFTS, JCSS, ISSJ, AAAI, IEEE-CS, and ACM.