Artificial Intelligence and Law **5:** 119–137, 1997.

A Legal Ontology Refinement Support Environment Using a Machine-Readable Dictionary

MASAKI KUREMATSU and TAKAHIRA YAMAGUCHI
Department of Computer Science, School of Information, Shizuoka University, Jouhoku 3-5-1, Hamamatsu-shi 432, Japan
E-mail: {*kurematu, yamaguti*}*@cs.inf.shizuoka.ac.jp*

Abstract. This paper discusses how to refine a given initial legal ontology using an existing MRD (Machine-Readable Dictionary). There are two hard issues in the refinement process. One is to find out those MRD concepts most related to given legal concepts. The other is to correct bugs in a given legal ontology, using the concepts extracted from an MRD. In order to resolve the issues, we present a method to find out the best MRD correspondences to given legal concepts, using two match algorithms. Moreover, another method called a static analysis is given to refine a given legal ontology, based on the comparison between the initial legal ontology and the best MRD correspondences to given legal concepts. We have implemented a software environment to help a user refine a given legal ontology based on these methods. The empirical results have shown that the environment works well in the field of Contracts for the International Sale of Goods.

Key words: ontology refinement, legal ontologies, machine-readable dictionaries, spell match, definition match, static analysis, Contracts for the International Sale of Goods

1. Introduction

The work in the field of knowledge engineering moves from interview systems such as MORE (Kahn et al., 1985) to modeling domains and tasks (problem-solving methods) using knowledge; subsequently, ontologies engineering has been emerging as a new field in the nineties. In the new field, much attention has first been paid to representation issues for ontologies, such as KIF (Genesereth and Fikes, 1992) and Ontolingua (Gruber, 1992). Recently the attention seems to shift from representation to content or the methodology of constructing ontologies. According to (Heijst, 1995), there are several distinguished ontologies, such as general or generic ontologies for conceptualizations across many domains, domain ontologies for conceptualizations in specific domains and task ontologies for describing problem-solving methods. Several general ontologies have already been developed as MRDs (Machine-Readable Dictionaries), such as CYC (Guha and Lenat, 1994), EDR (Japan Electronic Dictionary Research Institute Ltd., 1993; Yokoi, 1993) and WordNet (Miller, 1990). Task ontologies have also been developed from abstract models of methods, such as Generic Tasks (Bylander and Chandrasekaran, 1986), PROTÉGÉ-II (Musen et al., 1994) and CommonKADS (Breuker and Van de Velde, 1994). However, we have made few attempts to build up domain ontologies compared to general and task ontologies constructions due to hugeness and interaction problems (Heijst, 1995).

On the other hand, in developing large scale expert systems, we need to build several kinds of knowledge bases and integrate them. If we would build knowledge bases independently, high costs would be required in order to maintain consistency among the knowledge bases. Because a domain ontology presents the definitions of concepts shared across the knowledge bases, it can provide the facilities to support the maintenance of the knowledge bases. Considering the background described above, this paper focuses on a construction (refinement) support environment of a legal ontology in the field of Contracts for the International Sale of Goods, using an existing MRD. MRDs have been developed for the tasks of natural language understanding and thus the concepts tend to have general meanings. On the other hand, it logically follows that legal ontologies have legal contexts. However, since judicial precedents have been described by many general concepts, legal ontologies also include general contexts. So MRDs are partly sharable with legal ontologies and could be helpful in changing incomplete legal ontologies into better ones. Because of context difference between a legal ontology and MRDs, when using MRDs to refine a given legal ontology, we must solve two issues to bridge the context gap between the two knowledge resources. One is to find out MRD concepts most related with given legal concepts. The other is to correct bugs in a given legal ontology, using the concepts extracted from an MRD. In the remainder of this paper, we explain how to represent concepts in both EDR Electronic Dictionary taken as an MRD and a legal ontology given from a user. Next, in order to find out the best MRD correspondences to given legal concepts, match algorithms are described, including spell match and definition match. Furthermore, a method called static analysis is explained to correct bugs in a given legal ontology, comparing the legal ontology and the best MRD correspondences from several syntactical features. After implementing the legal ontology refinement environment based on these methods, empirical results emerge in the field of Contracts for the International Sale of Goods.

2. Ontologies representation

A user, such as a legal expert, tries to build up an initial legal ontology and then inputs it into the environment. The environment takes EDR Electronic Dictionary (Japan Electronic Dictionary Research Institute Ltd., 1993) as an MRD. EDR Electronic Dictionary and a legal ontology are summarized here.

2.1. EDR Electronic Dictionary

EDR Electronic Dictionary consists of a word dictionary and a concept dictionary. The word dictionary contains 200,000 words and consists of word entries. A word

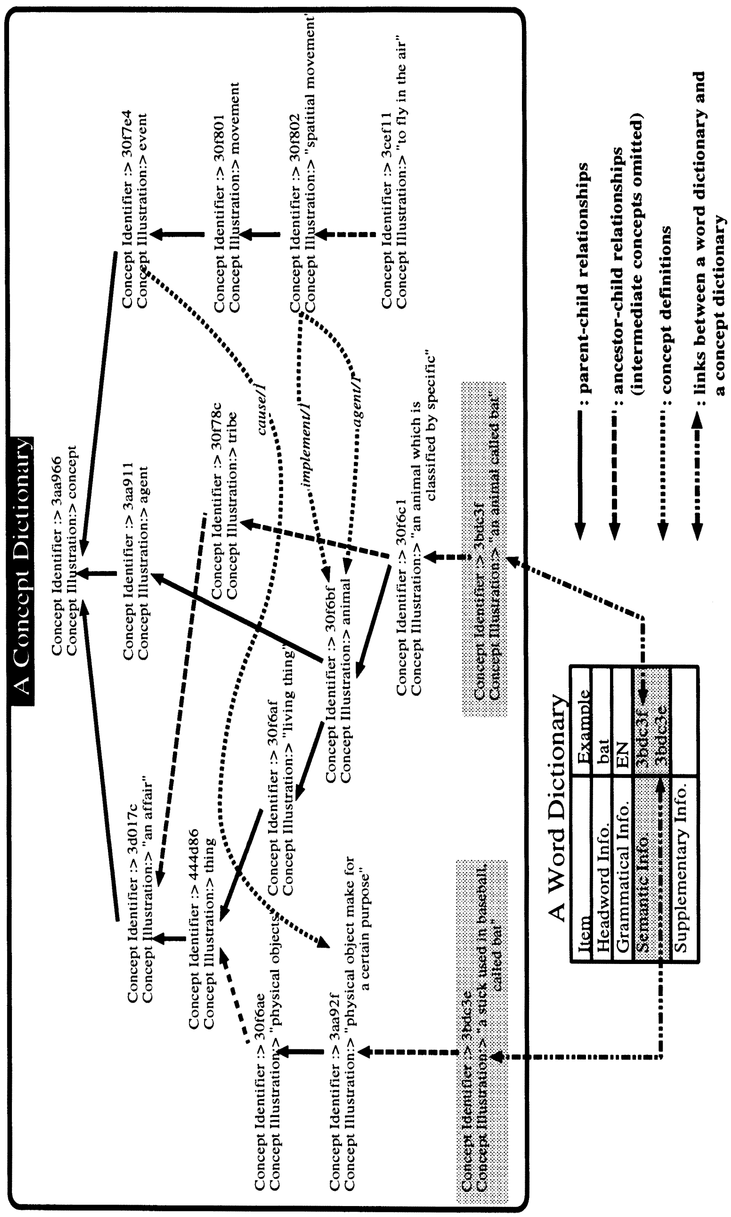

Figure 1. EDR Electronic Dictionary.

Format	→	ID1	Illust1	Direction	Label/CF	ID2	Illust2
		⇓	⇓	⇓	⇓	⇓	⇓
Example	→	3aa92f	physical object make for a certain purpose	←	cause/1	30f7e4	event

Note:	ID1	.. concept identifier(self)
	Illust1	.. concept illustration (self)
	Direction	.. the direction of concept relation label(← or →)
	Label	.. concept relation label
	CF	.. the certainty factor value of concept relation label (1 or 0)
	ID2	.. the other concept identifier
	Illust2	.. the other concept illustration

Figure 2. Concept definition.

entry is composed of headword information, grammatical information, semantic information, and supplementary information, as shown in Figure 1. The semantic information connects a word dictionary to a concept dictionary, and becomes a link to a concept entry that is the essential constituent of the concept dictionary. The link has been represented as a concept identifier in the concept entry. On the other hand, the concept dictionary contains the 400,000 concepts (concept entries) that are composed of the leaf-concepts connected to the word dictionary and super-concepts of the leaf-concepts. A concept entry is composed of a concept identifier, a concept illustration and concept definitions. The concept illustration is a word, phrase or sentence that explains the concept and distinguish the concept from other concepts. The concept definition is denoted as Figure 2. The concept relation labels associate the concept itself with other concepts through semantic relations. They include both super-sub relation labels to represent hierarchical relationships and other relation labels such as agent, object, goal etc., to represent semantic relationships between the concept and other concepts, as shown in Table I. It also has direction and a certainty factor value. A certainty factor indicates the degree of certainty of a relationship. Currently, since only 0 and 1 are being used, this value may be regarded as truth value. Taking "3aa92f: 'physical object make for a certain purpose' ← cause/1 30f7e4: event" as a concept definition, it means that a event "causes" physical objects made for a certain purpose.

2.2. A LEGAL ONTOLOGY

When a user builds up an initial legal ontology and then inputs it into the environment, it includes a legal concept hierarchy and legal concept definitions. Legal concepts are defined in Figure 2, with the exception of giving the concept identifier of a legal concept. In order to make the matching process described in Section 3 convenient, the concept relation labels also come from Table I and the other concept identifier (illustration) comes from EDR concept dictionary. For example, the legal concept of "person" is defined as follows:

Table I. Concept relation labels in EDR concept dictionary

Label	Definition	Label	Definition
agent	an agent of intentional movements	quantity	a quantity of things and movements and changes
a-object	an object of attribute	number	number
object	an object of movements and changes	condition	a conditional relation between two events or facts
cause	cause	cooccurrence	a concurrent relation between two events or facts
implement	an implement or a method of intentional movements	purpose	purpose
material	a material or a constitutional element	sequence	a sequential relation between two events or facts
source	a first position or situation for an agent or an object of events	basis	the basis of a comparison
goal	a final position or situation for an agent or an object of events	and	a connective relation between two concepts
place	place where events happen	or	a selective relation between two concepts
scene	a scene in which events happen	modifier	the other relation
manner	a way of movements and changes	possessor	a possessive relation
time	the time when events happen	beneficiary	a beneficiary of events and situations
time-from	the time when events start	from-to	range
time-to	the time when events are over	unit	unit

"person ← agent/1 30f826: move ownership",
"person ← beneficiary/1 30f794: right" and
"person ← beneficiary/1 30f795: duty".

The definitions means that "ownership" moves from person to person and a person has rights and duties.

3. A legal ontology refinement support environment

A user builds up a rough initial legal ontology and gives it to a legal ontology refinement support environment. The initial legal ontology might have bugs such as some important concepts are missing and/or the concept hierarchy has a flawed substructure. The environment tries to support the user in fixing the bugs and refining the initial legal ontology, using EDR dictionary. In this section, typical ontological bugs are explained and then two methods for helping the user correct bugs are described. The legal ontology refinement support environment is summarized in Figure 3.

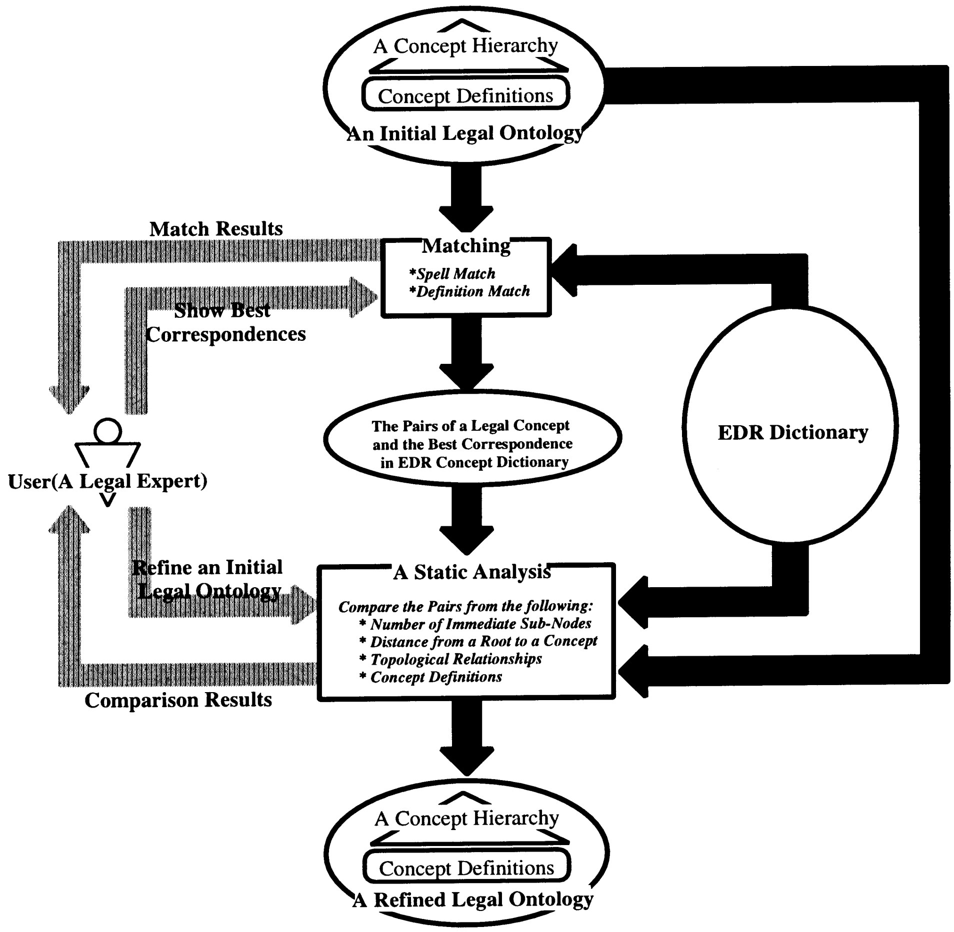

Figure 3. Legal ontology refinement support environment.

3.1. ONTOLOGICAL BUGS

Which types of bugs could emerge in the initial legal ontology? The following typical bugs could appear: missing concepts, existing unnecessary concepts, flawed hierarchical relations such as confusion of super-sub relations and one of parent-child relations, missing descriptions of existing concepts and then existing unnecessary descriptions of existing concepts. Suppose that we have an initial legal ontology and a revised legal ontology as shown in Figure 4. There are two types of bugs in Figure 4. “A more than three wheeled vehicle” marked with a rectangle in the revised legal ontology is an example of missing concepts. The other bug is an example of a flawed hierarchical relationship in that the parent-child relationship of “vehicle” and “a motor-cycle under 50cc” should be corrected into the ancestor-

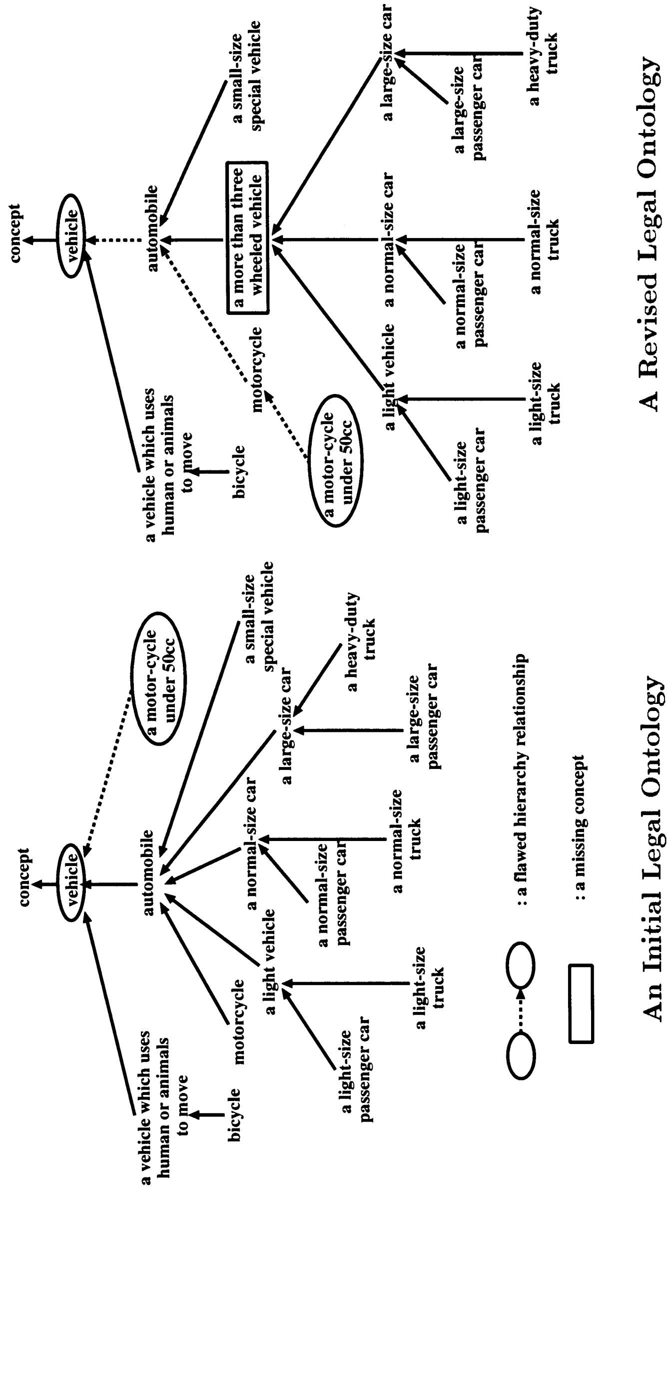

Figure 4. Ontological bugs.

child relationship, illustrated by a dotted line in the revised legal ontology. Judging from Traffic Law, it is better to correct these bugs as described above.

3.2. Matching a legal ontology with EDR dictionary

When there is something wrong with the given legal ontology from a user, the environment tries to support the user in fixing the bugs and refine the initial legal ontology, using EDR dictionary. In order to perform the support task, the environment should first help to find out the best EDR correspondences to given legal concepts. The following is the algorithm to perform the first task, using spell match and definition match. To give a brief sketch of the algorithm, it tries to find out the small space where the best correspondence to a given legal concept can be included in the large space with EDR concept dictionary.

Input: The name and the definitions of a legal concept

Output: The small space that can have the best correspondence in EDR concept dictionary

Algorithm:

1. Find the lower boundary of the small space by spell match:
   Do spell match between a legal concept name and EDR word dictionary. Spell match succeeds when a legal concept name is the same as the headword information of a word entry in EDR word dictionary. Get concept entries in EDR concept dictionary, which have been shown by the semantic information with the matched word entry. The user selects a good concept entry from them. Because concept entries linked to EDR word dictionary frequently become leaf nodes in EDR concept dictionary, the user's selection can be taken as the lower boundary of the small space that can have the best correspondence in EDR concept dictionary.
2. Find the upper boundary of the small space by definition match:
   Do definition match between legal concept definitions and EDR concept dictionary, moving up from the lower boundary to the root node. The climb up stops when the definition match fails. Definition match succeeds when one or more legal concept definitions match with concept definitions of a concept entry in EDR concept dictionary. The match is such a loose match that the concept relation label, its direction and its certainty factor are the same but its value is the same or subordinated to the concept entry in EDR concept dictionary. Taking concept drift into account, concept entries with at least one matched concept definition can be upper concept entries. So the last and highest definition match results make up the upper boundary of the small space that can have the best correspondence in EDR concept dictionary.

3. Extract the small space between the lower boundary and the upper boundary, from the whole space of EDR concept dictionary. Given the small space available to the user, (s)he selects the best correspondence to the given legal concept.

The above-mentioned algorithm has been applied to all legal concepts in the initial legal ontology. Thus the user can see the whole space mapped from the initial legal ontology onto EDR concept dictionary and then select all the best correspondences in EDR concept dictionary to the initial legal ontology.

3.3. A STATIC ANALYSIS

After getting the pairs of a legal concept and the best correspondence in EDR concept dictionary, the environment does a static analysis of each pair, comparing it to the following aspects: the number of immediate sub-nodes, the distance from a root to a concept, the topological relationship between two concepts and concept definitions. The four kinds of comparison are explained using Figure 5.

- The comparison of the number of immediate sub-nodes:
  Compare the number of immediate sub-nodes of a pair component. When the number of a legal concept is much larger than that of the best correspondence, some sub-node(s) of the legal concept might be removed. In the case of a much smaller number, some sub-node(s) of the legal concept might be added. It is decided empirically whether the difference between the two numbers is large or not. Considering the pair of “automobile” in Figure 5, the number of immediate sub-nodes of the legal automobile (five) is larger than that of the EDR automobile (two). So, some sub-node(s) of the legal automobile could be removed. In this case, as shown in Figure 4, the super-concept of “a more than three wheeled vehicle” for “a light vehicle”, “a normal-size car” and “a large-size car” is invented by a user and becomes a new immediate sub-concept of the legal automobile. Thus the number of immediate sub-nodes of the legal automobile decreases from five to three.
- The comparison of the distance from a root to a concept:
  Compare the distance from a root to a pair component in each concept hierarchy. When the distance from the root to a legal concept in an initial legal hierarchy is much larger than that of the best correspondence, the legal concept might be moved up. In the case of smaller, it might be moved down. It is also decided empirically whether the difference of two distances is large or not. Considering the pair of “a motor-cycle under 50cc” in Figure 5, the distance from a root to the legal “a motor-cycle under 50cc” (two) is much smaller that of the EDR “a motor-cycle under 50cc” (five). So the legal “a motor-cycle under 50cc” might be moved down. In this case, as shown in Figure 4, it has been moved down to the sub-node of “motorcycle” by a user. Thus the distance from a root to the legal “a motor-cycle under 50cc” increases from two to four.

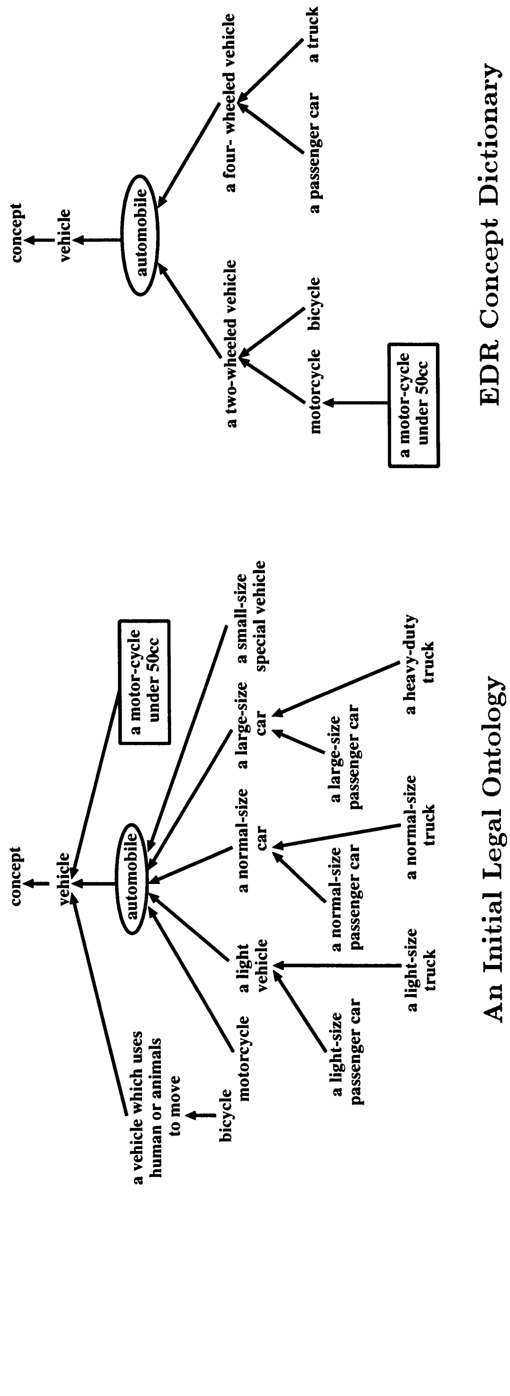

Figure 5. The comparison between an initial legal ontology and EDR concept dictionary.

The change of the distance frequently causes that of topological relationship described next.

- The comparison of the topological relationship between two concepts:
  Compare the topological relationship between two legal concepts and that between two of the best EDR correspondences. We use topological relationship such as parent-child, ancestor-child and sibling relationships. When two topological relationships are different, the topological relationship between two legal concepts might be replaced with that between two of the best EDR correspondences. Taking two topological relationships between "automobile" and "a motor-cycle under 50cc" in Figure 5, the topological relationship in an initial legal ontology is a sibling relationship but that of the EDR concept dictionary is an ancestor-child relationship. So the topological relationship between "automobile" and "a motor-cycle under 50cc" in an initial legal ontology might change into an ancestor-child relationship. Actually, it changes into the ancestor-child relationship where "motorcycle" is between "automobile" and "a motor-cycle under 50cc", as shown in Figure 4.
- The comparison of the concept definitions:
  Compare the concept definitions of a pair component. When the legal concept definitions are different from those of the best EDR correspondence, the differences are shown to a user. Looking at the differences, if necessary, the user might add new concept definitions to and remove unnecessary ones from the legal concept.

When the differences are larger than the threshold decided empirically in advance regarding the number of immediate sub-nodes and the distance from a root to a concept, and differences emerge about the topological relationship between two concepts and concept definitions, the environment shows the differences to a user and ask the user to revise an initial legal ontology. Some (or No) revision is completed, depending on the user's decision.

4. Experimental results

The following describes experimental results of the match between an initial legal ontology and EDR dictionary and then a static analysis in the field of Contracts for the International Sale of Goods (CISG) (Sono, 1993). Although the first author of this paper is not in the field of CISG, after studying CISG and law fields related to CISG for a year, he is able to play the role of the user in this experiment. He builds an initial legal ontology composed of 79 legal concepts from Part II of CISG and Japanese Civil Code (Takanashi, 1990) related to CISG. Figures 6 and 7 show the hierarchy structure and part of the concept definitions in the initial legal ontology respectively.

4.1. Matching a legal ontology with EDR dictionary

We ran the spell match and the definition match between the initial legal ontology and EDR word/concept dictionaries. Figure 8 shows us the match results to the legal concept of "reply". In the spell match, the legal concept of "reply" has been matched with a word entry in EDR word dictionary. Because the word entry has the concept identifier of "1088b2" as semantic information, the definition match starts, climbing up from the concept (entry) with the concept identifier of "1088b2" in EDR concept dictionary. The concept entry has the concept illustration of "to give an answer". The legal concept of "reply" has the following three concept definitions: "reply $\leftarrow$ cause/1 3f96e7: receive information", "reply $\leftarrow$ sequence/1 3f96e7: receive information" and "reply $\rightarrow$ object/1 30f799: contents of communication". The tree (except the top node of 3aa966: concept) in Figure 8 shows the small space in EDR concept dictionary which can have the best correspondence to the legal concept of "reply". The concept with no rectangle (3aa966: concept) means no definition match with the concept definitions of "reply". The concepts with one rectangle means that only one concept definition ("$\leftarrow$ sequence/1 3f96e7: receive information") has been matched. The concepts with double rectangles mean that two concept definitions ("$\leftarrow$ sequence/1 3f96e7: receive information" and "$\rightarrow$ object/1 30f799: contents of communication") have been matched. There was no concept that matched with all three concept definitions. Given the tree that includes concepts with single or double rectangles available to the user, he selected the best correspondences of "1f7aff: the action of answering" to the legal concept of "reply". The complete results are as follows: The spell match results include 51 matched legal concepts and 28 other unmatched legal concepts. The definition match results include 22 matched legal concepts and 29 other unmatched legal concepts. In the end, given the 22 small spaces available to the user that have been extracted from EDR concept dictionary, the user has selected the best correspondences in EDR concept dictionary to the 22 legal concepts.

4.2. A static analysis

After getting 22 pairs of a legal concept and the best correspondence in EDR concept dictionary based on the above-mentioned match results, we ran the static analysis over the pairs. Table II shows the comparison of the number of immediate sub-nodes of the pairs. When the difference is more than three, the environment ask for user interaction. All the legal concepts with a check in Table II mean that the number of immediate sub-nodes of a legal concept is three or smaller than that of the best correspondence in EDR concept dictionary. So the environment asks the user if some new immediate sub-concepts of the legal concepts with a check can be added or not. Actually the advice made the user add the new immediate sub-concepts of "positional relation" and "mutual relation" to "relation", "area" and "part" to "place", and then "period" and "moment" to "time". Figure 9 shows the modification of the number of immediate sub-nodes of "relation".

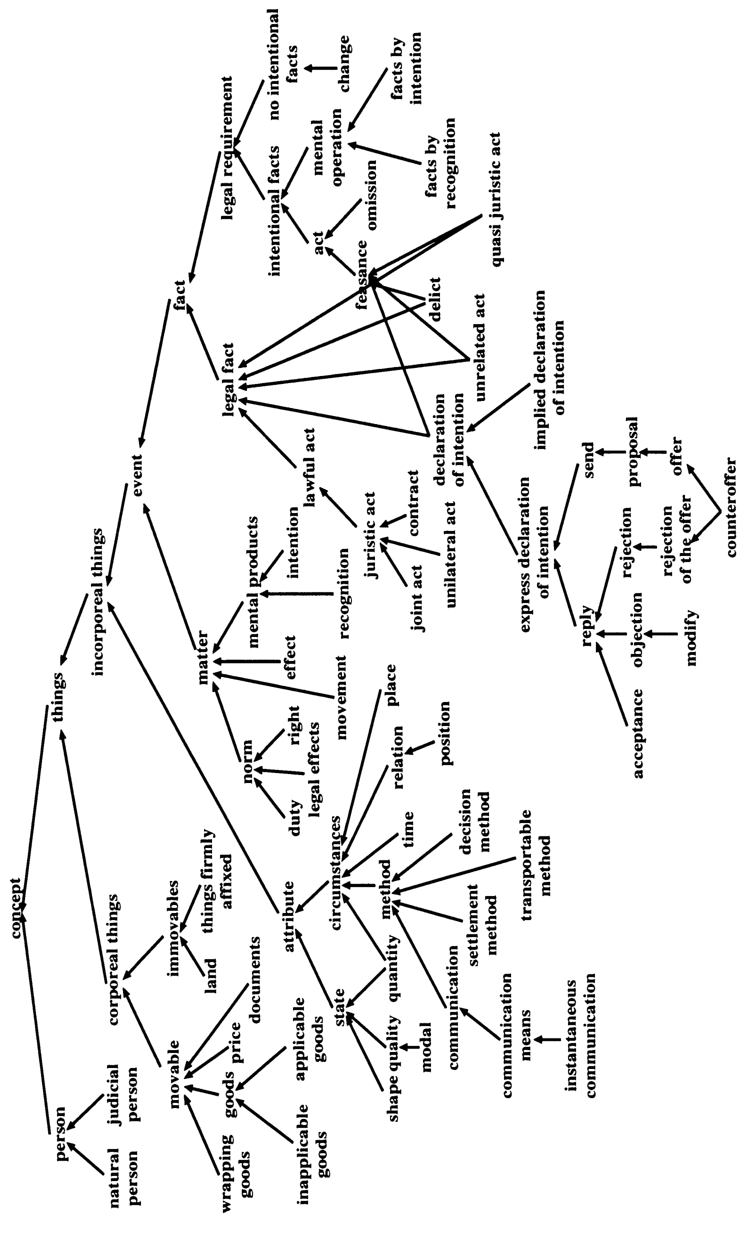

Figure 6. The hierarchy structure in an initial legal ontology.

Legal Term	Concept Relation Label			Concept Identifier	Concept Illustration
	Direction	Relation Label	CF		
person	←	agent	1	30f826	move ownership
person	←	beneficiary	1	30f794	right
person	←	beneficiary	1	30f795	duty
natural person	←	agent	1	30f7f4	physiological phenomenon
natural person	←	agent	1	30f7f8	live
judicial person	←	agent	0	30f7f4	physiological phenomenon
judicial person	←	agent	0	30f7f8	live
things	←	object	1	30f826	move ownership
things	←	object	1	30f794	right
things	←	object	1	30f795	duty

Figure 7. Part of concept definitions in an initial legal ontology.

Table II. The comparison of the number of immediate sub-nodes

Legal concepts	Legal ontology	EDR	Ask a user	Legal concepts concepts	Legal ontology	EDR	Ask a user
person	2	2		norm	3	2	
things	2	4		movement	0	5	✓
goods	2	2		recognition	0	5	✓
price	0	1		delict	0	1	
quantity	0	1		declaration of intention	2	2	
relation	1	4	✓				
place	0	3	✓	send	1	4	✓
time	0	4	✓	replay	3	1	
method	4	2		offer	1	2	
position	0	0		acceptance	0	1	
communication means	1	0		modify	0	1	
				change	0	1	

Table III shows the comparison of the distance from a root to a pair component in each conceptual hierarchy. The hierarchy has multiple inheritance, and so the concept has more than one distance value. When the difference is more than three, the environment asks for user interaction as well as the number of immediate sub-concepts. All the legal concepts with a check in Table III mean that the distance from the root to the legal concept in the initial legal concept hierarchy is more than three larger than that of the best correspondence in EDR concept dictionary. So the environment asks the user if the legal concepts with a check can be moved up or not. However, the advice made the user move no legal concepts up.

The topological relationships between two legal concepts and those between two of the best EDR correspondences have been compared. When they are different, the environment asks the user whether or not the topological relationship between two legal concepts can be replaced with that between two of the best EDR

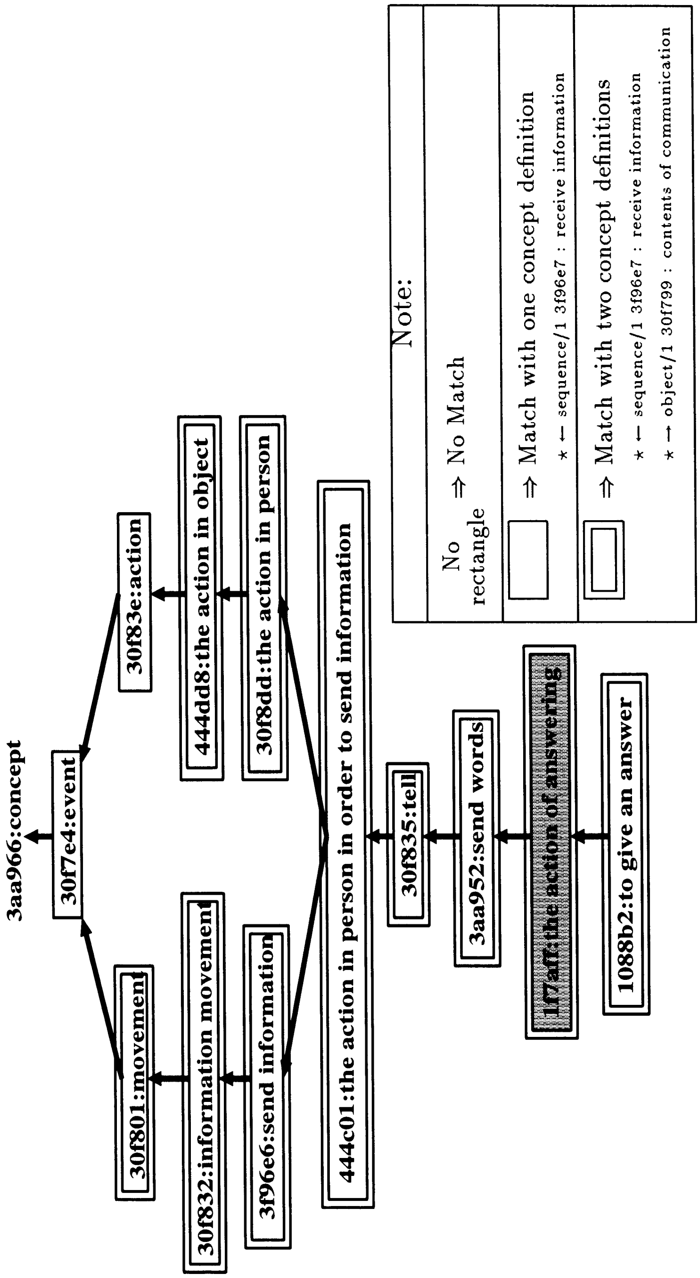

Figure 8. Part of match results

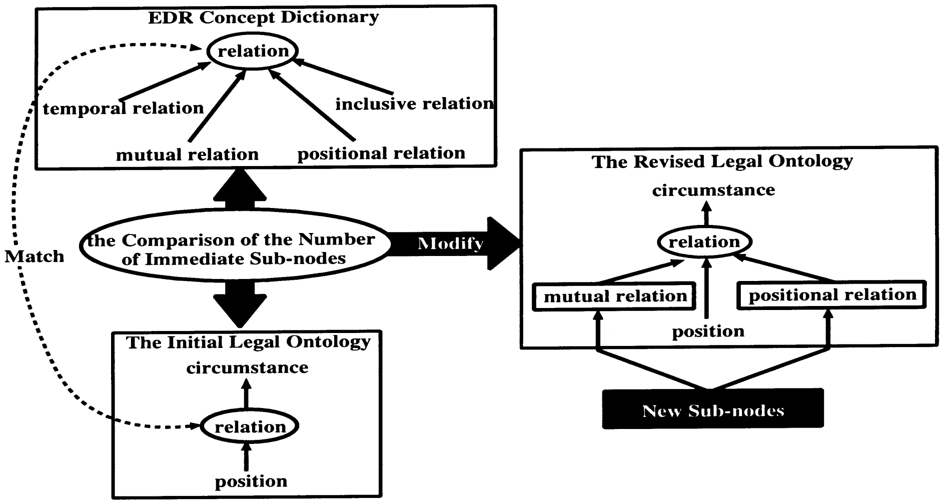

Figure 9. The modification of the number of immediate sub-nodes of "relation".

Table III. The comparison of the distance from a root to a concept

Legal concepts	Legal ontology	EDR	Ask a user	Legal concepts	Legal ontology	EDR	Ask a user
person	[1]	[1]		norm	[5]	[3]	
things	[1]	[1]		movement	[5]	[3]	
goods	[4]	[4]		recognition	[6]	[5, 6]	
price	[4]	[2]		delict	[6, 9]	[7]	
quantity	[5]	[6]		declaration of intention	[6, 9]	[4]	✓
relation	[5]	[4]					
place	[5]	[2]	✓	send	[8, 11]	[5]	✓
time	[5]	[2]	✓	reply	[8, 11]	[5]	✓
method	[5]	[4]		offer	[10, 13]	[6, 7]	✓
position	[6],	[5, 7, 8]		acceptance	[9, 12]	[8, 9]	✓
communication means	[7]	[6]		modify	[10, 12]	[6]	✓
				change	[7]	[4]	✓

correspondences. The example of the user's modification is as follows: the sibling relationship between "reply" and "send" in the initial legal ontology was replaced with the parent-child relationship.

The concept definitions of the legal concept and those of EDR best correspondence have been compared. When they are different, the environment asks the user if the legal concept definitions should be modified or not. The user modified the concept definition of "reply" and "place".

Figure 10 shows the legal conceptual hierarchy refined by the above-mentioned static analysis. Although the user was not a real legal expert in this experiment, some

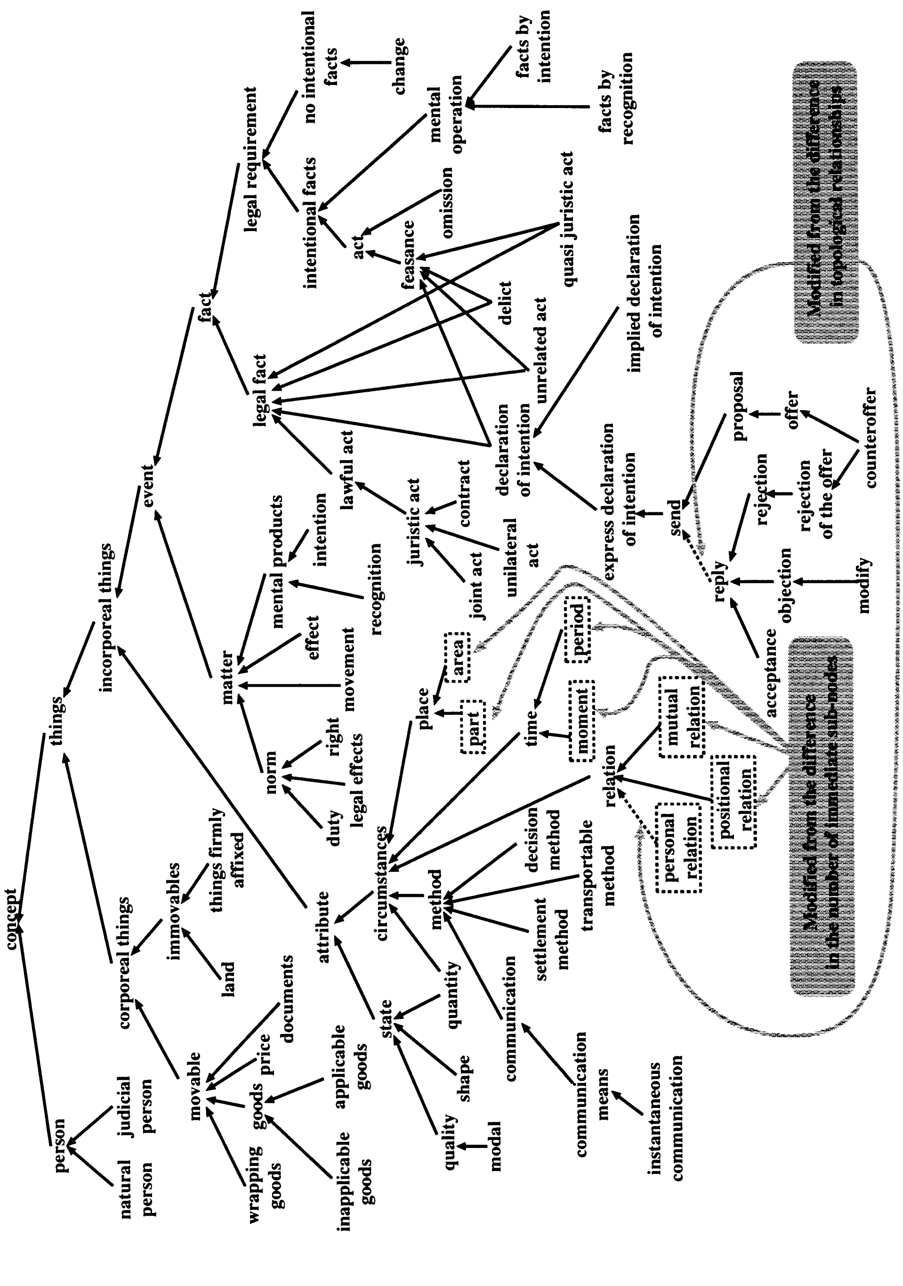

Figure 10. A refined legal conceptual hierarchy.

Table IV. The environment specification

Knowledge bases & modules	Language	Size (Kb)
Initial legal ontology	SICStus-Prolog	24.2
EDR word dictionary	TEXT	159922.5
EDR concept dictionary	TEXT	84624.8
Matching module	Perl & C	28.5
Display module	Tcl-Tk	10.0
Modification module	SICStus-Prolog	16.9

real legal experts found that the refined legal ontology was better than the initial legal ontology and so the environment was useful in supporting legal ontology construction processes.

The environment has been implemented by the integration of SICStus Prolog, C, Perl, Tcl-Tk language on Unix platform. Table IV shows the environment specification at present.

5. Concluding remarks and future work

This paper presents a legal ontology refinement support environment using an MRD, based on the following techniques: the match algorithm that retrieves the best MRD correspondences to given legal concepts and the static analysis including the comparisons of the number of immediate sub-node, the distance from a root to a concept, the topological relationship between two concepts, and concept definitions.

In the future, we will introduce the following new analysis into the environment: a dynamic analysis based on an inductive learning technique, such as Inductive Logic Programming (ILP) (Muggelton and Buntine, 1988). Because a static analysis has been conducted prior to executing legal reasoning using a legal ontology, bugs in the legal ontology might remain. A dynamic analysis will be completed after a legal reasoning. After much legal reasoning, we will get correct legal conclusions (positive examples about legal reasoning) and also incorrect ones (negative examples). So using positive and negative examples, ILP can help to find new fault with the legal ontology. Furthermore, we plan that real legal experts will actually use the environment and give their comments to us. They will also be so helpful in bringing new viewpoints to the environment.

Acknowledgments

We express our thanks for implementations of match algorithms to Mr. Masayoshi Tada, and we are grateful to Professor Hajime Yoshino and other members in the study of legal expert systems development for their significant discussion. We would also like to thank the referees for their constructive comments. This research was supported as part of the research grant referred to by 05208102 from the Japanese Ministry of Education, Science and Culture.

References

Breuker, J. & Van de Velde, W. (1994). *Common KADS Library for Expertise Modeling*. IOS Press: Amsterdam.

Bylander, T. & Chandrasekaran, B. (1988). Generic Tasks in Knowledge-Based Reasoning: The "Right" Level of Abstraction for Knowledge Acquisition. In Gaines B. & Boose, J. (eds.) *Knowledge Acquisition for Knowledge Based Systems Vol. 1*, 65-77. Academic Press: London.

Genesereth, M. R. & Fikes, R. (1992). *Knowledge Interchange Format Version 3.0 Reference Manual.* Technical Report, Logic-92-1, Computer Science Department, Stanford University, San Francisco, CA.

Gruber, T. R. (1992). *Ontolingua: A Mechanism to Support Portable Ontologies.* Technical Report, KSL 91-66, Computer Science Department, Stanford University, San Francisco, CA.

Guha, R. V. & Lenat, D. B. (1994). Enabling Agents to Work Together. *Communications of the ACM* **37**(7): 127–142.

Heijst, G. (1995). *The Role of Ontologies in Knowledge Engineering.* Ph.D. diss., University of Amsterdam.

Japan Electronic Dictionary Research Institute Ltd. (1993). *EDR Electronic Dictionary Technical Guide*. Japan Electronic Dictionary Research Institute Ltd.: Tokyo.

Kahn, G., Nowlan, S. & McDermott, J. (1985). MORE: An Intelligent Knowledge Acquisition Tool. In Proceedings of the *Ninth International Joint Conference on Artificial Intelligence*, 581–584. Morgan Kaufmann: Los Angeles.

Kurematu, M., Tada, M. & Yamaguchi, T. (1994). Constructing a Legal Ontology from a General Ontology. In Proceedings of *Workshop 5 on Fifth Generation Computer Systems*, 73–89. Tokyo: Institute for New Generation Computer Technology.

Kurematu, M., Tada, M. & Yamaguchi, T. (1995). A Legal Ontology Refinement Environment Using a General Ontology. In Proceedings of *Workshop on Basic Ontology Issues in Knowledge Sharing, International Joint Conference on Artificial Intelligence '95.* Montreal: International Joint Conference on Artificial Intelligence.

Miller, G. (1990). WordNet: An On-Line Lexical Database. *International Journal of Lexicographer* **3**(4): (Special Issue).

Muggelton, S. & Buntine W. (1988). Machine Invention of First-order Predicates by Inverting Resolution. In Proceedings of the *Fifth International Conference on Machine Learning*, 339–352. Morgan Kaufman: Michigan.

Musen, M. A., Gennari, J. H., Eriksson, H., Tu, S. W. & Puerta, A. R. (1994). *PROTÉGÉ-II: Computer Support for Development of Intelligent Systems.*

Sono, K. & Yamate, M. (1993). *Contracts for the International Sale of Goods.* Seirin Shoin: Tokyo [in Japanese].

Takanashi, H. (1990). *Examples of Civil Law*. Jiyukokuminsya: Tokyo [in Japanese].

Yokoi, T. (1993). Very Large-Scale Knowledge Bases Embodying Intelligence Space. In Proceedings of *International Conference on Building & Sharing of Very Large-Scale Knowledge Bases '93*, 11–20. Tokyo: Japan Information Processing Development Center.