Constructing a Legal Ontology from a General Ontology

Masaki Kurematsu Masayoshi Tada Takahira Yamaguchi
Faculty of Engineering Shizuoka University
3-5-1 Johoku, Hamamatsu, 432, Shizuoka JAPAN
{kurematu, tada, yamaguti}@cs.shizuoka.ac.jp

Abstract

This paper discusses how to construct a legal ontology from a general ontology that has already been developed. In the construction process, we must solve two hard issues. The one is to localize legal contexts in a general ontology in order to match a legal ontology with a general ontology. The other is to identify bugs in constructed legal ontology and refine them. Here is presented a new method to match a legal concept with the most similar concept in a general ontology and also two strategies to refine a legal ontology, a static analysis based on the comparison between two ontologies and a dynamic analysis based on Inductive Logic Programming. We have just been implementing a computer environment to help a user construct a legal ontology from a general ontology. The experimental results of matching and a static analysis are described and also a scenario for a dynamic analysis is presented.

1 Introduction

In developing large scale of expert systems, we must build several kinds of knowledge bases and integrate them. If we might build them independently, so much effort would be done in order to maintain the consistency among them. Because a domain-specific ontology would present the definition of concepts that often come up in them, it could be so helpful to support the task. As the scale of expert systems would be larger, a knowledge base development environment based on a domain-specific ontology would be more important.

Recently a large scale of knowledge bases, such as CYC[Guha 90] and EDR[Yokoi 93], have been developed. However, they have just general meaning on general context and are so difficult to be used directly in building domain knowledge bases. We regard them as general ontologies and take them as the aids to construct a domain-specific ontology. For example, in building a legal ontology, we try to identify the concepts in a general ontology, which are most similar to legal concepts, and refine the legal concepts using the general concept.

In the process, we must solve two hard issues. The one is to localize legal contexts in a general ontology. The other is to identify bugs in a legal ontology given by a user and refine them. Here is presented a new method to match a legal concept with the most similar concept in a general ontology and also two strategies to refine a legal ontology, a static analysis based on the comparison between two ontologies and a dynamic analysis based on Inductive Logic Programming. We have just been implementing a computer environment to help a user construct a legal ontology from a general ontology. The experimental results of matching and a static analysis are described and also a scenario for a dynamic analysis is presented.

2 Ontologies

A user, such as a legal expert, must build up an initial legal ontology before using the environment and then tries to refine it, getting the support form the environment. The environment takes EDR Electronic Dictionary[EDR 93] as a general ontology. The following describes the overview for a general ontology and a legal ontology.

2.1 General Ontology

We take as a general ontology EDR Electronic Dictionary including a word dictionary and a concept dictionary. The word dictionary has more than 200,000 words described by the information of grammar, meaning and so on. The concept dictionary has a large scale of concept hierarchy including more than 400,000 nodes. Each node (concept) is represented by a semantic network which has twenty four relation labels shown by Table 1.

Table 1: Concept Relation Labels at EDR

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

2.2 Legal Ontology

Before using the environment, a user must build up an initial ontology, which has a legal concept hierarchy and legal concept definitions. Legal concepts are defined by a semantic network using the relation labels in Table 1 and the relation label values are the nodes on the hierarchy in a general ontology in order to make the matching process described later more convenient.

3 A Legal Ontology Construction Support Environment

The initial ontology from a user might have several bugs. Some important concepts might be missing and a hierarchy structure could be not appropriate there. Our legal ontology construction support environment tries to change it into more refined legal ontology, using the following strategies: a static analysis and a dynamic analysis.

First, the environment tries to find out the concepts in a general ontology, which are most similar to concepts in a legal ontology, using a new matching method described later. Afterward the environment puts the links between two ontologies. Second, in a static analysis, the environment compares two linked nodes from several points and gives the difference back to the user. Third, in a dynamic analysis, when a legal inference system would go wrong using the legal ontology, the environment would try to modify the links between two ontologies and/or the legal ontology. In modifying the legal ontology, ILP techniques would try to invent new legal concepts and legal rules, as described in a scenario later. Figure 1 shows the overview for the environment.

3.1 Links between Two Ontologies

In order to find out the concepts in a general ontology, which are most similar to concepts in a legal ontology, the environment takes two kinds of matching: text level matching and conceptual level matching. The former tries to match legal nodes with general nodes directly by spelling. The matching results are taken as lower limits of the space mapped from the legal nodes onto a general ontology. The latter tries to match legal nodes with general nodes by the similarity between two conceptual description structures represented by a semantic network. The matching results are taken as upper limits of the space as well. Now the user can get the space mapped from the legal nodes onto a general ontology and select the best-matched general nodes to legal nodes.

3.2 Static Analysis

In a static analysis, the environment compares two linked nodes from the following points: the number of sub-nodes, the depth of a node in each hierarchy, topology between two nodes in each hierarchy and conceptual descriptions. When the serious difference would come up, the environment would give it back to the user. The explanation about the four points follows:

1. The number of sub-nodes

   When the number of sub-nodes of a legal node is larger than that of the best-matched general node, the legal node could have too much information and so the sub-nodes migth be removed. In the case of smaller, it could have too less information and so the sub-nodes might be added.

2. The depth of a node in each hierarchy

   When the depth of a legal node in a legal hierarchy is larger than that of a general node in a general hierarchy, the legal node could be over-specialized and change into more general concept (super-concept). In the case of smaller, it could be over-generalized and change into more specific concept (sub-node).

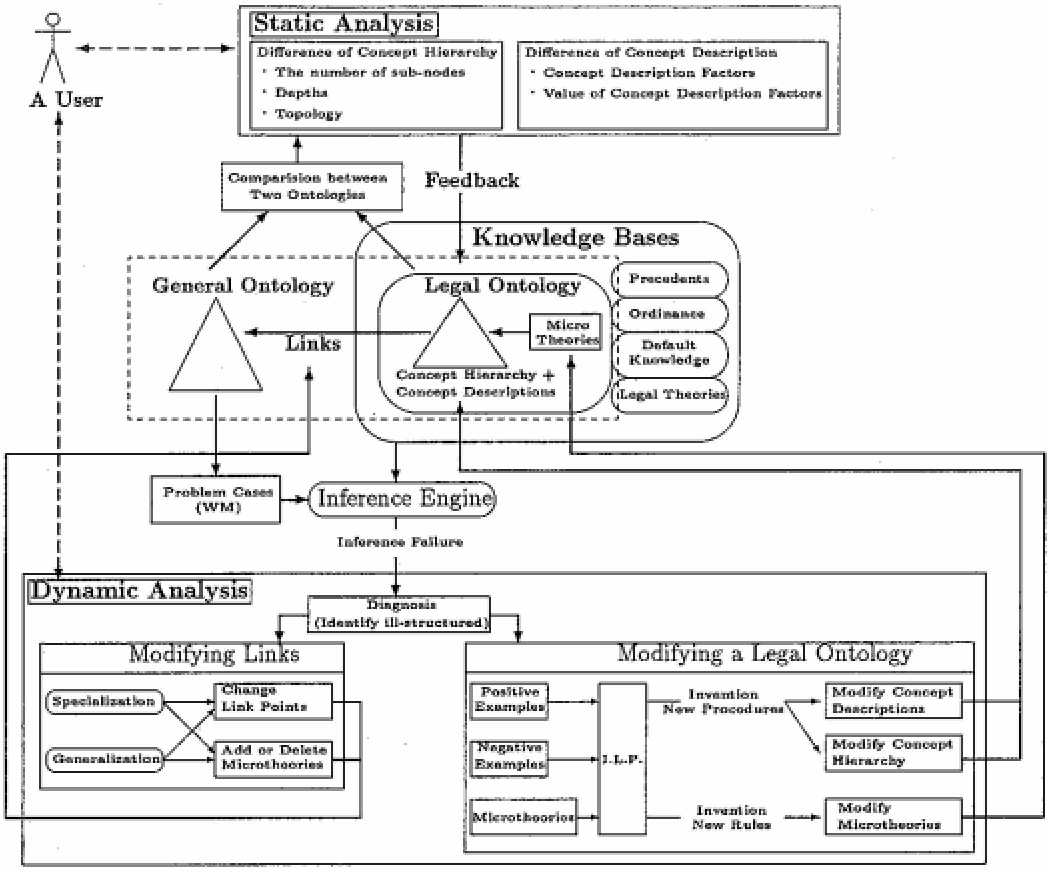

Figure 1: An Overview for a Legal Ontology Construction Support Environment

3. Topology between two concepts in each hierarchy

   The topology between two legal nodes is compared with that between two corresponding best-matched general nodes. It might be parent-child topology, sibling topology and so on. When the topology would differ, the difference would given to the user.

4. Conceptual descriptions

   When the conceptual description of a legal node would be different from that of a general node, the difference would be given to the user.

3.3 Dynamic Analysis

Although a static analysis helps a user find out bugs in an initial legal ontology and repair them, there might still remain bugs there. When a legal expert system would solve a problem using the legal ontology, it might go wrong and have provable negative examples and unprovable positive examples. The wrongness would come from the links between two ontologies and/or the legal ontology itself and in a dynamic analysis the environment would try to modify the links and the legal ontology, based on inductive learning methods.

1. Modifying links

   To a provable negative example, the environment tries to identify the link relevant with it and put the link from the legal node to a general sub-node. So the link is specialized in order not to prove the negative example. To an unprovable positive example, the environment tries to put the link from the legal node to a general super-node. So the link is generalized in order to prove the positive example.

2. Modifying a legal ontology

   To unprovable positive examples, the environment tries to modify a legal ontology based on ILP techniques, such as CIGOL[Muggelton 88]. CIGOL is a kind of model inference system from a set of ground input clauses. It has three inverse resolution operators, Truncation, W-operator and V-operator. Truncation is almost similar to generalization, such as

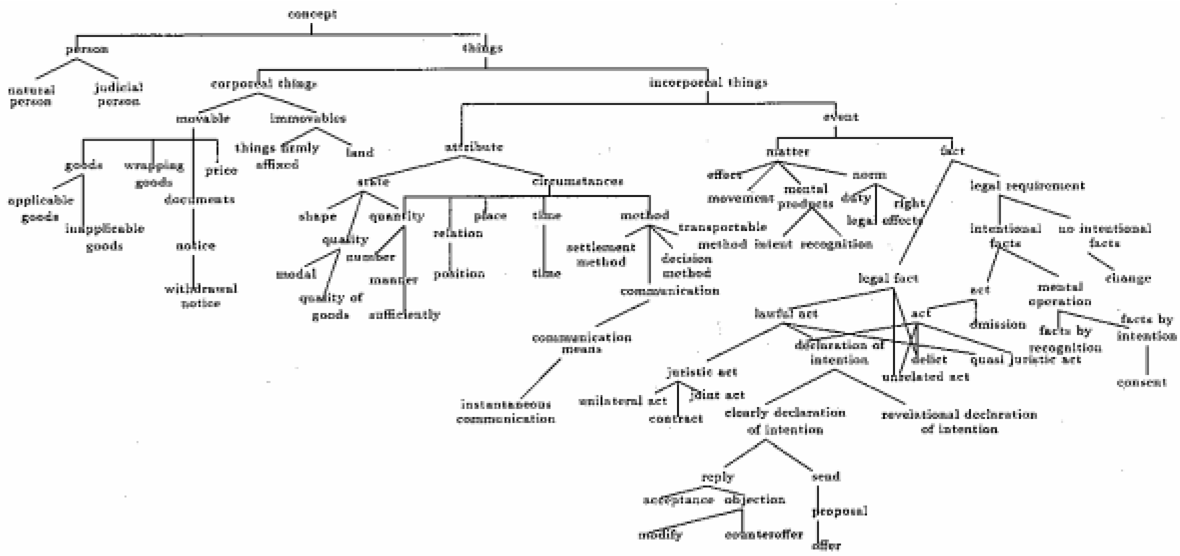

Figure 2: A Hierarchy in an Initial Legal Ontology

people	← agent/1	2022− − [move ownership(30f820)
people	← beneficiary/1	131615[right(30f794)
people	← beneficiary/1	131616[duty(30f795)
natural person	← agent/1	2013− − [physiological phenomenon(30f7f4)
natural person	← agent/1	201361[live(30f7f8)
judicial person	← agent/0	2013− − [physiological phenomenon(30f7f4)
judicial person	← agent/0	201361[live(30f7f8)
things	← object/1	2022− − [move ownership(30f820)
things	← object/1	131615[right(30f794)
things	← object/1	131616[duty(30f795)
corporeal things	← object/1	2051− − [attribute of space(3f9872)
corporeal things	← possessor/1	2051− − [attribute of space(3f9872)
incorporeal things	← object/0	2051− − [attribute of space(3f9872)
incorporeal things	← possessor/0	2051− − [attribute of space(3f9872)

Figure 3: A Part of Concept Description in an Initial Legal Ontology

changing constants into variables. W-operator tries to invent a new concept (predicate) that is a sub-concept to a given concept (A user gives the environment the name of invented concept). V-operator regards two clauses as the one parent clause and a resolvent and then tries to generate the other parent clause. Here in the environment, provable and unprovable positive examples are given to CIGOL and then CIGOL tries to change a legal ontology into more refined one, putting a invented concept there by W-operator, and generate a new legal rule by V-operator.

4 Experimental Results and a Scenario

Here is described experimental results of the matching between two ontologies and a static analysis and also a scenario of a dynamic analysis. Before the following experiment, we have built an initial legal ontology composed of 75 legal concepts from United Nations Convention on Contracts for the International Sale of Goods (CISG) and Japanese Civil Code related with CISG. Figure 2 illustrates the legal hierarchy and Figure 3 shows a part of concept description in the initial legal ontology. The environment has been implemented by C and Perl languages on S-4/ model 51 engineering workstation and also the initial legal ontology by SICStus-Prolog.

4.1 Matching

Figure 4 shows an example of matching process between two ontologies. Twenty two legal nodes have been matched with general nodes and the matching rate was about thirty per cent. Other twenty four legal nodes have failed in text level matching and other twenty nine legal nodes in conceptual level matching. Although too legal specific concepts would fail in matching, the matching rate would go up by the improvement of matching method.

```
List of items nominated for ' reply ': (display by tree)

Meaning of marking
0 reply ;;sequence: 3f96e7  2 0 2 3 3 0 | receive information
1 reply ;;sequence: 3f96e7  2 0 2 3 3 0 | receive information
        ;;object  : 30f799  1 3 1 7 1 2 | contents of communication
*3aa960 (null) 0 0 0 0 0 0 | concept
*  | 30f77e (null) 1 3 - - - - | pseudo still life
0  |  | 30f7e4 (null) 2 - - - - - | event
1  |  |  | 30f801 (null) 2 0 2 - - - | move
1  |  |  |  | 30f832 (null) 2 0 2 3 - - | move information
1  |  |  |  |  | 3f96e6 (null) 2 0 2 3 2 0 | send information
1  |  |  |  |  |  | 30f86f (null) 2 0 3 3 1 1 | express
1  |  |  |  |  |  |  | 3f9729 (null) 2 0 3 3 1 2 | say
1  |  |  |  |  |  |  |  | 1f7aff -answer[answer]" answer
1  |  |  |  |  |  |  |  |  | 1088b2 -reply" reply
0  |  |  | 30f83e (null) 2 0 3 - - - | activity
1  |  |  |  | 30f869 (null) 2 0 3 3 - - | thinking activity
1  |  |  |  |  | 30f86f (null) 2 0 3 3 1 1 | express
1  |  |  |  |  |  | 3f9729 (null) 2 0 3 3 1 2 | say
1  |  |  |  |  |  |  | 1f7aff -answer[answer]" answer
1  |  |  |  |  |  |  |  | 1088b2 -reply" reply
```

Figure 4: An Example of Matching Process

4.2 Static Analysis

Using the above matching result, a static analysis has been done. Table 2 shows the number of sub-nodes of legal nodes and the matched general nodes. Now we take the difference of more than three as the advice to a user and it is marked with a circle in Table 2. All circles In Table 2 show that the number of sub-nodes of legal nodes is more than three smaller than that of the matched general nodes. So the environment gives a user such an advice that some sub-nodes of legal nodes (marked with a circle) could be added. Actually the advice let the user do the following change: adding "positional relation" and "mutual relation" as the sub-nodes of "relation", "area" and "part" as the sub-nodes of "place", and "period" and "moment" as the sub-nodes of "time".

Table 3 shows the depth of each node in each hierarchy. The hierarchy has multiple inheritance and so the node has more than one values of the depth. As well as the number of sub-nodes, we take the difference of more than three as the advice to a user and it is marked with a circle in Table 3. All circles In Table 3 show that the depth of legal nodes is more than three larger than that of the matched general nodes. So the environment gives a user such an advice that the legal nodes could change into more general concept (super-concept). However, the advice did not let the user change them.

The topology between two legal nodes is compared with that between two corresponding best-matched general nodes. There is a sibling topology about "reply" and "send" in a legal ontology and a parent-child topology in a general ontology. So the environment gives a user such an advice that the node of "reply" might be a sub-node of the node of "send". Actually the advice let the user change so. Another advice let the user change the parent-child topology between "position" and "relation" into another topology.

Table 2: The Number of Sub-Nodes in Two Ontologies

Node name	Legal ontology	General ontology	
person	2	2	
things	2	4	
goods	0	2	
price	0	1	
quantity	0	1	
relation	0	4	○
place	0	3	○
time	0	4	○
method	4	2	
position	0	0	
communication means	1	0	
norm	3	2	
movement	0	5	○
recognition	0	5	○
delict	0	1	
declaration of intention	2	2	
send	1	4	○
replay	3	1	
offer	0	2	
acceptance	0	1	
modify	1	1	
change	0	1	

Table 3: The Depth of Each Node in Each Hierarchy

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

The advice from the comparison between conceptual descriptions let the user change the concept description of "reply" and "place". Thus the environment helps a user modify an initial legal ontology into better one. Figure 5 shows the hierarchy of a refined legal ontology.

4.3 A Scenario of a Dynamic Analysis

Suppose that the problem represented as Figure 6 is given to a conventional Prolog-like inference system that has a rule base to represent CISG and a legal ontology, based on a matching mechanism such as all LHSs of a rule would subsume the nodes in problem representation (be super-nodes of the corresponding nodes in problem representation). The conclusion to derive is whether the proposal constitutes an offer or not. The following shows a rule to define the legal concept of "offer".

[1] offer(X):-proposal(X),sufficiently_definite(X).

[2] sufficiently_definite(X):-

goods(object(X)),has_quantity(object(X)),has_price(object(X)).

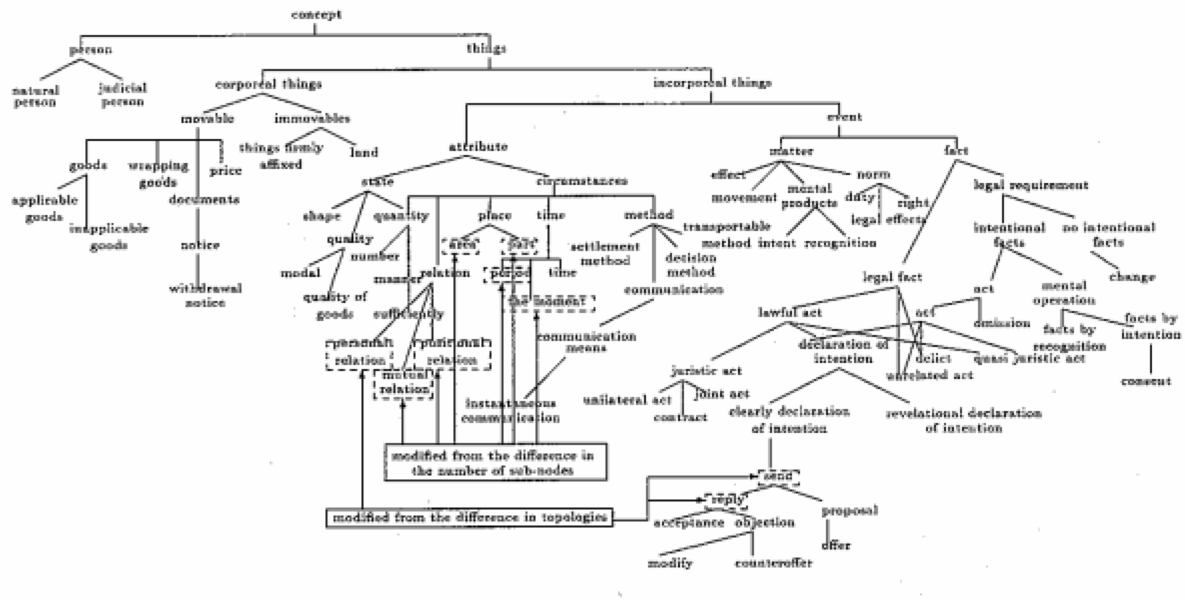

Figure 5: A Hierarchy in a Modified Legal Ontology

```
<id(a, 1), issue(no1), agent, seller>
<id(a, 2), issue(no1), object, order_sheet(no1)>
<id(a, 3), issue(no1), goal, buyer>
<id(a, 4), issue(no1), time, 6/1>
<id(a, 5), issue(no1), purpose, sell>
<id(a, 6), sell, object, sugar(no1)>
<id(a, 7), sugar(no1), quantity, first_class>
<id(a, 8), sugar(no1), quantity, sack>
<id(a, 9), sack, number, 1000>
<id(a, 10), order_sheet(no1), purpose, shipment>
<id(a, 11), shipment, object, sugar(no1)>
<id(a, 12), shipment, time, 7/1>
<id(a, 13), order_sheet(no1), purpose, wrapping>
<id(a, 14), wrapping, object, sugar(no1)>
<id(a, 15), wrapping, implement, sack(no1)>
<id(a, 16), strong, a-object, sack(no1)>
```

Figure 6: Problem Representation

Suppose that a user would guess that the proposal constitutes an offer. However,because the second clause in the above rule would not be probable, the inference system would fail in getting it. So the user would give the following unprovable clause to the inference system:

sufficiently_definite(X):-goods(object(X)),has_quantity(object(X)).

In a dynamic analysis, the above clause would have been given to CIGOL. The truncation would generate the following clause:

sufficiently_definite(X):-goods(object(X)),has_quantity(object(X)).

The above clause says that "price" is not necessary for a proposal to be "sufficiently definite". It is almost the same as a legal theory from K. Sono[Sono 93].

Furthermore, W-operator would generate the following clauses, inventing a new predicate named governed by applicable laws by a user. The above clause says that "price" should be governed by applicable laws. It is almost the same as a legal theory from Honnold[Sono 93].

- *sufficiently_definite(X):-goods(object(X)),has_quantity(object(X)), governed_by_applicable_rules(object(X)).*
- *governed_by_applicable_rules(object(X)):-has_price(object(X)).*
- *governed_by_applicable_rules(object(X)).*

5 Conclution and Future Works

This paper presents a legal ontology construction support environment, including two strategies to refine a legal ontology, a static analysis based on the comparison between two ontologies and a dynamic analysis based on Inductive Logic Programming. Although the implementation has not yet be done, some experimental results shows that the environment would go well. As future works, real legal experts actually use the environment and we will have to improve it from their comments. Furthermore, we will introduce much techniques from machine learning into the environment and find out balanced cooperation between man and machine[Morik 94].

Acknowledgments

We would like to express our thanks to Professor K.Yoshino and the member in the study of legal expert system development and Japan Electronic Dictionary Research Institute LTD.

References

[Guha 90] *Guha, R.V. and Lenat, D.B. : "Cyc : A Midterm Report", AI Magazine, Vol.11 No.3 pp.32-59(1990)*

[Yokoi 93] *T.Yokoi : "Very Large-Scale Knowledge Bases Embodying Intelligence Space", Proc. International Conference on Building and Sharing of Very Large-Scale Knowledge Bases '93,pp.11-20(1993)*

[EDR 93] *EDR Electronic Dictionary Specifications Guide, Japan Electoronic Dictionary Research Institute LTD.,(1993)*

[Muggelton 88] *S.Muggelton, W.Buntine : "Machine Invertion of First-order Predicates by Inverting Resolution", Proc. the Fifth International Conference on Machine Learning, pp.339-352 (1988)*

[Sono 93] *K.Sono and M.Yamate : "The Law for International Sales of Goods", Sieirin Shoin,(1993)*

[Morik 94] *K.Morik : "Balanced Cooperative Modeling", Machine Learning Vol.4 - A Multistaregy Approach -, pp.295-317(1994)*