Constructing Domain Ontologies Based on Concept Drift Analysis

Takahira YAMAGUCHI
Dept. Computer Science, Shizuoka University
3–5–1 Johoku Hamamatsu Shizuoka 432–8011 JAPAN
yamaguti@cs.inf.shizuoka.ac.jp
Tel: +81–53–478–1473 Fax: +81–53–473–6421

Abstract

This paper focuses on how to construct domain ontologies, in particular, a hierarchically structured set of domain concepts without concept definitions, reusing a machine readable dictionary (MRD) and making it adjusted to specific domains. In doing so, we must deal with concept drift, which means that the senses of concepts change depending on application domains. So here are presented the following two strategies : match result analysis and trimmed result analysis. The strategies try to identify which part may stay or should be moved, analyzing spell match results between given input domain terms and a MRD. We have done case studies in the filed of some law. The empirical results show us that our system can support a user in constructing a domain ontology.

1 Introduction

In the field of ontology engineering, much attention has first been paid to representation issues for ontologies, such as KIF[Gen92] and Ontolingua[1]. Recently the attention seems to shift from representation to contents or the methodology of constructing ontologies. According to [Hei95], there are several distinguished ontologies, such as generic ontologies for conceptualizations across many domains, domain ontologies to put constraints on the structure and contents of domain knowledge in a particular-field and task ontologies for describing problem-solving methods. Several natural language ontologies (including generic ontologies) have already been developed as MRDs (machine-readable dictionaries), such as CYC[Guh90], WordNet[Mil90] and EDR[EDR93]. Task ontologies have also been developed from abstract models of methods, such as Generic Tasks[Byl88], PROTEGE-II[Mus94] and CommonKADS[Bre88]. Because domain ontologies have large number of specified concepts, they make less progress than generic ontologies and task ontologies that have just a few specified concepts. Thus this paper focuses on how to construct domain ontologies, in particular, a hierarchically structured set of domain concepts without concept definitions, reusing existing MRDs and making them adjusted to specific domains. Actually, from the same motivation, we have already presented a domain ontology refinement support environment called LODE[Kur97] . A user gives an initial domain ontology with a hierarchically structured set of domain concepts and the relationships between them to LODE. LODE does match between the initial domain ontology and EDR. The match results have been analyzed from several syntactical features in order to refine the initial domain ontology into better one. Applying LODE to the field of particular law, we find that LODE can support a legal expert in refining an initial legal ontology into better one. However, it took costs to prepare an initial legal ontology and legal experts did not like it. We must reduce the costs to set up the input to LODE. To do so, the technical issue of "concept drift" comes up to us. Because the senses of concepts in a MRD

Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5) Stockholm, Sweden, August 2, 1999

(V.R. Benjamins, B. Chandrasekaran, A. Gomez-Perez, N. Guarino, M. Uschold, eds.)

http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-18/

come from a common domain and so not good for some specific domain, we must deal with the change of concept's senses caused by the change of domains, called concept drift. Our domain ontology rapid development environment (called DODDLE) tries to manage concept drift, analyzing match results by several strategies for concept drift. In order to evaluate DODDLE, case studies have been done in the particular law called Contracts for the International Sale of Goods (CISG). The empirical results have shown us that DODDLE can support a user in constructing a domain ontology.

2 Ontology Capture

Various approaches of ontology design have been proposed by many researchers. According to [Usc96],ontology capture consists of identifying and defining the important concepts and terms. They propose the following approach to capture ontologies; 1) Have a brainstorming session to produce all potentially relevant terms and phrases. 2) Structure the terms loosely into categories corresponding to naturally arising sub-groups. 3) Commit to the basic terms that will be used to specify the ontology. 4) Address each category in tern and define each term in the category. 5) Commit to the ontology.

In using DODDLE, domain terms are supposed to be already identified and given to DODDLE. Since DODDLE just generates a hierarchically structured set of domain terms, it support a user in structuring terms into categories and giving names to the categories in the second phase in the above-mentioned ontology design. Furthermore, DODDLE may contribute to identifying basic terms and defining each term in the third and fourth phases, through the process of adjusting concept hierarchies from DODDLE to specific domains.

3 Ontological Bugs and Concept Drift

Suppose that we could extract information relevant to given input domain terms from a MRD. We call it an initial model in this paper. The initial model is not sufficient for a domain ontology. It might have bugs such that some important domain-specific concepts are missing and/or the concept hierarchy has flawed part from the point of domain specificity. Which type of bug could emerge in the initial model ? The following typical bugs could appear: missing concepts, existing unnecessary concepts, flawed hierarchical relationships such as confusion of super-sub relationship and parent-child relationship, missing concept definitions and existing unnecessary concept definitions.

Figure 1 shows an example of an initial model and a legal ontology (a hierarchically structured set of legal concepts without the relationships between them). There are two types of bugs in Figure 1. "A more than three wheeled vehicle" marked with a rectangle in the legal ontology is an example of missing concepts. The other bug is an example of a flawed hierarchical relationship, the parent-child relationship of "vehicle" and "a motor-cycle under 50cc" in an initial model. It should be corrected into the ancestor-child relationship, illustrated by a dotted line in the legal ontology. Judging from the field of Traffic Law, it is better to correct these bugs as described above.

When we change an initial model into a domain ontology, the part infected with domain specificity is regarded as ontological bugs in the initial model. Because DODDLE just constructs a hierarchically structured set of domain concepts without concept definitions, flawed hierarchical structures and existing unnecessary concepts seem to come up frequently as the part drifted by domain specificity. DODDLE takes the strategies based on match result analysis and trim result analysis to do so, as described in section 4.3.

4 DODDLE Design

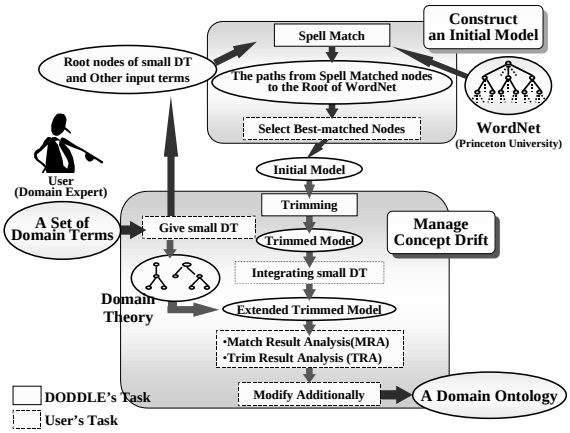

Figure 2: DODDLE Overview

After giving an overview of DODDLE, we present detailed descriptions about WordNet taken as a MRD and strategies for concept drift.

4.1 An Overview of DODDLE

Figure 2 shows an overview of DODDLE. In order to analyze concept drift between a MRD and a domain ontology (a hierarchically structured set of domain concepts), here are two basic activities: constructing an initial model from a MRD (extracting information relevant to given domain terms from a MRD) and managing concept drift (making an initial model adjusted to the domain).

A user gives a (not structured) set of domain terms to DODDLE. The user can also give small size of trees including domain concepts, which are called small DT

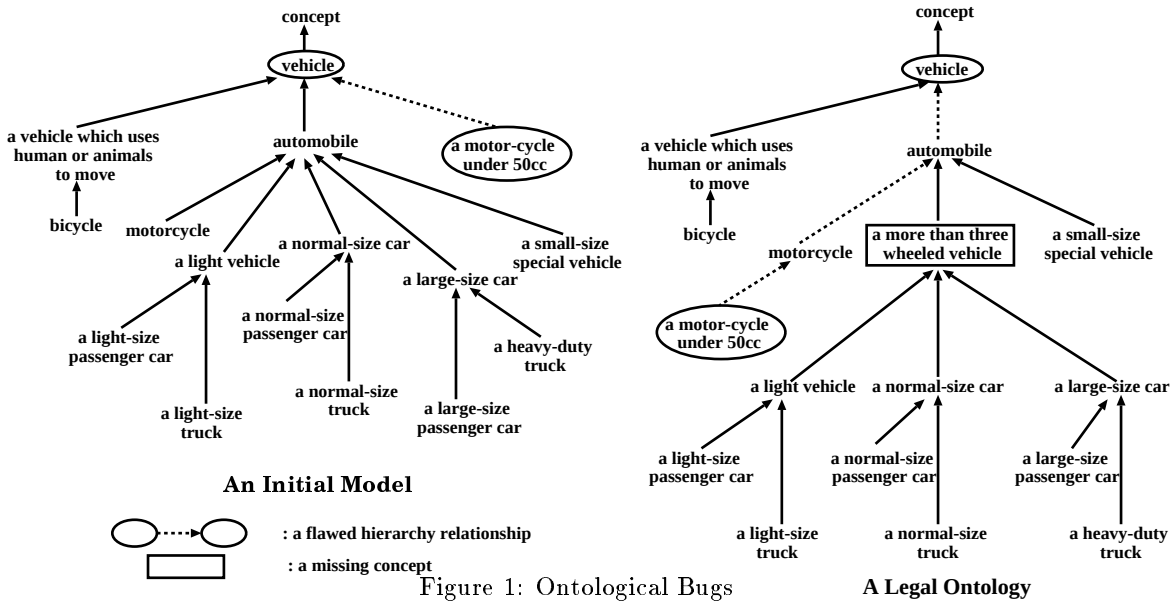

Figure 1: Ontological Bugs

(Domain Theory) later. Because the input terms in small DT have already been structured, it is no necessary to do spell match between them and a MRD. However, in order to integrate small DT into a trimmed model, spell match must be done between just root nodes of small DT and a MRD. So DODDLE do spell match between other all input terms except inner and leaf nodes of small DT and a MRD. These terms are linked to a MRD by the spell match. The spell match results are a hierarchically structured set of all the nodes on the path from these terms to the root of a MRD. Because a matched node (concept) from a MRD sometimes has one or more senses, it must be selected which sense is best. DODDLE supports the user in doing the selection by showing the user the following information: detailed descriptions on each sense and where each sense is put in the concept hierarchy structure from a MRD. We call the selected nodes "best-matched nodes" and the hierarchy structure composed of paths from best-matched nodes to the root in a MRD "an initial model".

Because an initial model has been extracted from a MRD, DODDLE tries to manage the infection (concept drift), analyzing match results by several strategies for concept drift. Here are three basic processes to do so: removing unnecessary internal terms in the initial model (called 'a trimmed model' later), integrating small DT into the trimmed model and finding out which part should be drifted in the trimmed model. the latter process has the following two strategies: match result analysis and trimmed result analysis. After moving the part infected with domain specificity and doing additional modifications, the user finally gets a hierarchically structured set of domain concepts as a domain ontology.

Table 1: WordNet

Dictionary Name	word synsets	word senses
Noun dictionary	60557	107424
Verb dictionary	11363	25761
Adjective dictionary	16328	28749
Adverb dictionary	3243	6201
Index dictionary	91519	119217

4.2 WordNet

DODDLE takes WordNet[Mil90] as a MRD. WordNet is an on-line lexical reference system and is developed by a group of psychologists and linguists at Princeton University. WordNet contains English nouns, verbs, adjectives and adverbs. Table 1 shows WordNet specification. We use a noun dictionary and an index dictionary for DODDLE.

4.3 Managing Concept Drift

In order to remove unnecessary internal nodes in an initial model based on match result analysis, internal nodes are divided into important internal nodes called SINs (Salient Internal Nodes) and other internal nodes. If internal nodes branch subordinate best-matched nodes, they work for keeping structural relationships among best-matched nodes, such as parent-child relationship and sibling relationship. So SINs are regarded as internal nodes that branch sub-ordinate best-matched nodes and other SINs. Thus DODDLE leaves a root, best-matched nodes and SINs in an initial model. The process looks like a trimming. Thus DODDLE gets a trimmed model.

Figure 3 illustrates the trimming process. Because the structural relationships among best-matched

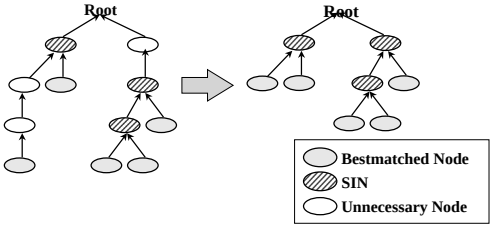

Figure 3: Trimming Process

nodes are kept even by removing white nodes, the original tree is reduced. Because the cross-hatched nodes work for branching best-matched nodes, they become SINs. Thus DODDLE gets a trimmed model that includes only best-matched nodes and SINs.

Because the nodes on lower part have much domain specificity, it may be a better way for a user to give another structure to them. It comes from DT given by a user. So then it can be integrated into trimmed models.

Figure 4 illustrates extending a trimmed model. After matching the root node of small DT with a trimmed model, matched part in the trimmed model is replaced with the small DT. After the sub nodes the replaced part are reconfigurated by the user. DODDLE gets an extended trimmed model.

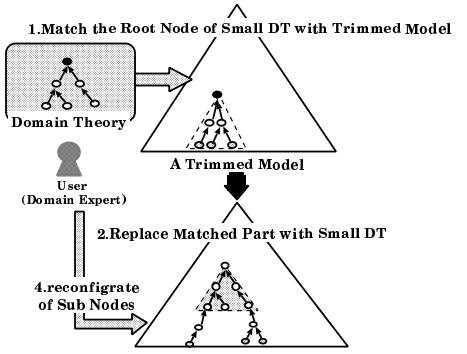

Figure 4: Integrating Small DT into a Trimmed Model

Here are two strategies for managing concept drift.

Strategy1:Match Result Analysis for Concept Drift

In order to find out which part should be drifted in the trimmed model, DODDLE takes a look at the distribution of best-matched results. Thus the following strategies come to DODDLE: A trimmed model is divided between a PAB (PAth including only Best-matched nodes) and a STM (SubTree Moved) based on the distribution of best-matched nodes. On one hand, a PAB is a path that include only best-matched nodes that have the senses good for given domain specificity. Because all nodes have already been adjusted to the domain in PABs, PABs can stay there in the trimmed model. On the other hand, STM is such a subtree that a SIN is a root and the subordinates are only best-matched nodes. Because SINs have not been confirmed to have the senses good for a given domain and so STMs can be infected with domain specificity, STMs can be moved somewhere in the trimmed model. Thus DODDLE identifies PABs and STMs in the trimmed model automatically and then supports a user in constructing a domain ontology by moving STMs. Figure 5 illustrates examples of PABs and STMs in a trimmed m

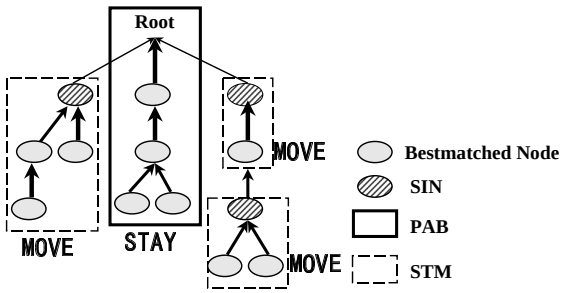

Figure 5: PABs and STMs

Strategy2:Trimmed Result Analysis for Concept Drift

In order to manage concept drift in the trimmed model, DODDLE uses trim result analysis as well as match result analysis. Taking some sibling nodes with the same parent node, there may be many difference about the number of trimmed nodes between them and the parent node. When such a big differences comes up on a subtree in the trimmed model, it may be to change the structure of the subtree. DODDLE suggests the user in changing the structure of the subtree. Based on empirical analysis, DODDLE takes reconstructed part as the subtree which has two and more difference about the number of trimmed nodes. Figure 6 illustrates examples of reconstructing a subtree in a trimmed model.

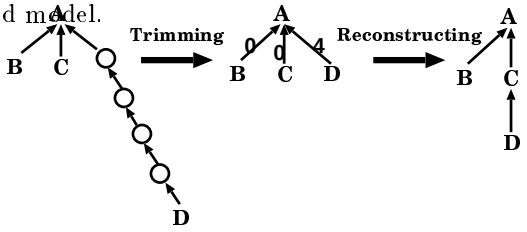

Figure 6: Reconstructing a Subtree in a Trimmed Model

After managing concept drift with two strategies, the user do additionally modification just node by node without support from DODDLE. Finally, the user gets a hierarchically structured set of domain concepts as a domain ontology.

Based on the above-mentioned design, DODDLE has been implemented by Perl language and Tcl-tk on UNIX platforms. Table 2 shows the specification of

DODDLE. Figure 7 shows a typical screen of DODDLE.

Table 2: DODDLE Specifications

Module	Language	Size(KB)
Construct an Initial Model	Perl&Tcl-Tk	62.5
Manage Concept Drift	Perl&Tcl-Tk	86.2
GUI	Tcl-Tk	46.5

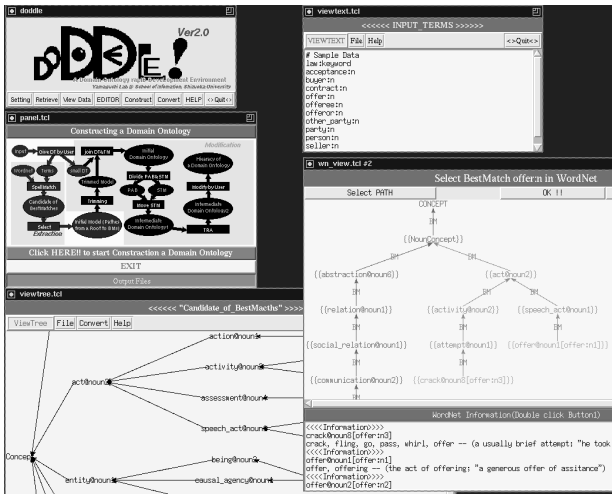

Figure 7: DODDLE Browser

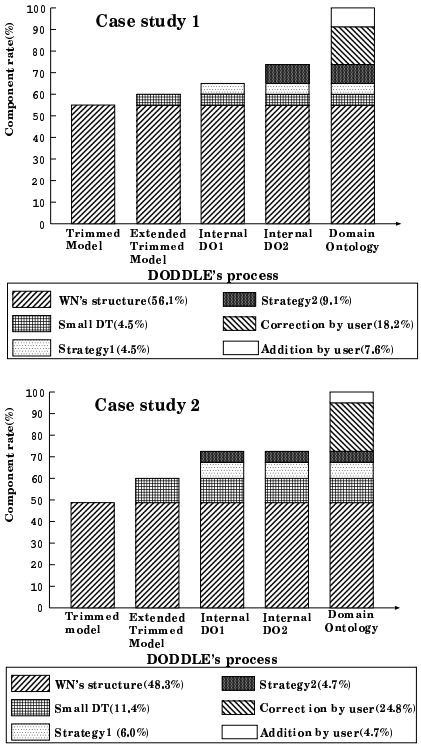

Figure 8: The Component Rate of the Final Domain Ontology

5 Case Studies in a Legal Domain

In order to evaluate how DODDLE is doing in practical fields, case studies have been done in a particular law called Contracts for the International Sale of Goods (CISG). Two lawyers joined the case studies. In the first case study, input terms are 46 legal terms from CISG Part-II. In the second case study, they are 103 terms including general terms in an example case and legal terms from CISG articles related with the case. One lawyer did the first case study and the other lawyer did the second.

Table 3 shows the case studies results. Figure 8 shows how much is included in final domain ontology the intermediate products at each DODDLE activity.

Generally speaking, in constructing legal ontologies, 70 % or more support comes from DODDLE. About half part of the final legal ontology results in the information extracted form WordNet. Because the two strategies just imply the part where concept drift may come up, the part generated by them has just low component rates and about 30 % hit rates . So one out of three indication based on the two strategies work well in order to manage concept drift. Because the two strategies use just such syntactical feature as matched and trimmed results, the hit rates are not so bad. In order to manage concept drift smartly, we maybe need the strategies using more semantic information that is not easy to come up in advance.

6 Related Work

Because domain ontologies have large number of specified concepts, we need existing useful information resources in designing domain ontology environments. Here are two information resources for the purpose: existing similar domain ontologies and natural language ontologies such as MRDs.

On one hand, Gertjan van Heijst et. al. try to reuse existing a similar medical domain ontologies, extending it with domain specificity and method specificity [Hei95]. When similar domain ontologies are missing in constructing a new domain ontology, it is hard to construct it.

On the other hand, Ontosaurus [Swa96] has points similar to DODDLE. Ontosaurus constructs a domain ontology using SENSUS [Kni94] as a MRD semi-automatically. A user has only to input some "seed" terms that (s)he identified. However, Ontosaurus supports a user in constructing a domain ontology just by giving spell match results between seed terms and SENSUS. The idea of managing concept drift in DODDLE is missing in Ontosaurus. Furthermore, LODE [Kur97] already came up as our first approach. However, it took costs for a user to give an initial domain ontology with a hierarchically structured set of domain concepts and concept definitions. Although it is hard to scale up the approach, the integration of LODE and DODDLE is pro-missing.

Table 3: The Case Studies Results

The number of X	The first case study	The second case study
Input terms	46	103
Small DT(Component terms)	2(6)	6(25)
Nodes matched with WordNet(Unmatched)*	42(0)	71(4)
Salient Internal Nodes(Trimmed nodes)	13(58)	27(83)
Small DT integrated into a trimmed model(Unintegrated)	2(0)	5(1)
Modification by the user(Addition)	17(5)	44(7)
Evaluation of strategy1**	4/16(25.0%)	9/29(31.0%)
Evaluation of strategy2**	3/10(30.0%)	4/12(33.3%)

* "Nodes matched with WordNet" is the number of input terms which have be selected proper senses in WordNet and "Unmatched" is not the case.

** The number of suggestions accepted by a user/The number of suggestions generated by DODDLE

From the point of knowledge representation, the KL-ONE family, such as CLASSIC [Bra92] and LOOM[MGr91], is good for representing conceptual definitions and hierarchies. If DODDLE would be re-implemented by he KL-ONE family, DODDLE could get the facilities for property inheritance and subsumption of two descriptions and so on.

7 Conclusions and Future Work

This paper discusses how to construct a domain ontology using existing MRDs. T o do so, concept drift came up as an important technical issue. In order to m ake concept drift operational, two strategies have been proposed and empirical results show us that they work well. However, we have just syntactical strategies to identify concept drift. In order to manage concept drift so well , we need to operationalize semantic strategies using more information with domain specificity. Furthermore, we need to extend DODDLE into getting more facilities such as learning from case data and referring to other ontology descriptions, in order to facilitate DODDLE in large scale application domains.

Acknowledgments

I would like to have many thanks to Prof. H. Yoshino and Prof. S. Kagayama for the cooperation of case studies and Dr. M. Kurematsu, Miss C. Aoki and Miss R.Sekichi for the implementations of DODDLE.

References

[Kur97] M.Kurematsu, C.Aoki and T.Yamaguchi (1997). LODE : A Legal Ontology Development Environment Using Natural Language and Case Ontologies: Proc. IJCAI-97, Poster Session Abstract, p.62.

[Bra92] R.J.Brachman, A.Borgida, D.L.McGuinness, P.F.Patel-Schneider, and L.A.Resnick (1992) The CLASSIC knowledge representation system, or, KL-ONE, the next generation : Proc. the International Conference on Fifth Generation Systems, pp.1036-1043.

[Bre88] J.Breuker and W.Van de Velde.(1994). Common KADS Library for Expertise Modeling : IOS Press.

[Byl88] Bylander,T. & Chandrasekaran,B. (1988). Generic Tasks in Knowledge-based Reasoning: The "Right" Level of Abstraction for Knowledge Acquisition. In *Knowledge Acquisition for Knowledge Based Systems Vol.1*, ed. B.Gaines & J. Boose, 65-77. London : Academic Press.

[EDR93] Japan Electronic Dictionary Research Institute LTD.(1993). EDR Electronic Dictionary Technical Guide : Japan Electronic Dictionary Research Institute LTD.

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

[Usc96] M.Uschold and M.Gruninger(1996). Ontologies: Principles, Methods and Applications: AAAI-96 tutorial syllabus. SA1, 1996.

[Hei95] Heijst,G.(1995). The Role of Ontologies in Knowledge Engineering : Ph.D. diss., University of Amsterdam.

[Gen92] Genesereth,M.R. and Fikes, R.(1992). Knowledge Interchange Format Version 3.0 Reference Manual : Technical Report, Logic-92-1, Computer Science Department, Stanford University, San Francisco, CA.

[Guh90] R.V.Guha and D.B.Lenat(1990). Cyc : A Mid-term Report : AI Magazine, Vol.11,No.3,pp.32-59.

[Kni94] K.Knight, S. Luk (1994). Building a Large Knowledge Base for Machine Translation : Proc. AAAI-94. Seattle, WA. 1994.

[Mil90] Miller,G. (1990). WordNet An on-line lexical database : International Journal of Lexicographer 3 (4) (Special Issue).

[MGr91] Robert M.MacGregor(1991). The evolving technology of classification - based knowledge representation systems : Principles of semantic networks: explorations in the representation of knowledge, John F. Sowa, ed., The Morgan Kaufmann series in representation and reasoning, Morgan Kaufmann Publishers, Inc., San Mateo, CA, 385-400.

[Mus94] Musen,M.A., Gennari,J.H., Eriksson,H., Tu,S.W., and Puerta,A.R. (1994). PROTÉGÉ-II: Computer Support For Development Of Intelligent Systems. Genesereth,M.R. & Fikes, R. 1992. Knowledge Interchange Format Version 3.0 Reference Manual : Technical Report. KSL-94-60, Computer Science Department, Stanford University, San Francisco, CA.

[Pei90] Peiwei Mi and Walt Scacchi (1990). A Knowledge-Based Environment for Modeling and Simulating Software Engineering Processes : IEEE Transactions on Knowledge and Data Engineering,Vol.2,No.3:283–294.

[Son93] K.Sono and M.Yamate (1993). United Nations Convention on Contracts for The International Sales of Goods : Seirin Shoin.

[Swa96] Bill Swartout, Ramesh Patil, Kevin Knight and Tom Russ (1996). Toward Distributed Use of Large-Scale Ontologies : Proc. of the 10th Knowledge Acquisition Workshop (KAW'96).