DODDLE : A Domain Ontology Rapid Development Environment

Chizuru Aoki, Rieko Sekiuichi, Masaki Kurematsu and Takahira Yamaguchi
School of Information, Shizuoka University,
3-5-1 Jouhoku Hamamatsu, 432 JAPAN
Phone: +81-53-478-1473
FAX: +81-53-473-6421
E-mail Address :{aoki, s3026, kurematsu, yamaguti}@cs.inf.shizuoka.ac.jp

Abstract

In order to reduce the costs to construct large scale knowledge based systems, many researchers in the field of knowledge engineering pay attention to ontologies. An ontology is an explicit specification of a conceptualization and so must be represented explicitly. Ontologies can be classified into the following ontologies: natural language ontologies such as lexicon and thesaurus, information ontologies such as conceptual scheme in database design and knowledge modeling ontologies. This paper focuses on knowledge modeling ontologies, in particular, domain ontologies that put constraints on the structure and contents of domain knowledge in a particular field. Because domain ontologies have large number of specified concepts, they make less progress than other knowledge modeling ontologies. So this paper focuses on how to construct domain ontologies, in particular, a hierarchically structured set of domain concepts without concept definitions, reusing existing information resources and making them adjusted to specific domains. We use a machine readable dictionary (MRD) that has already been developed. The concept's senses from a MRD are not good for some specific domain but for a common domain. We must deal with the change of concept's senses caused by the change of domains, called concept drift. So here are two technical issues to construct domain ontologies using a MRD: 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). In order to construct an initial model, our domain ontology rapid development environment (DODDLE) does spell match between the input domain terms given by a user(such as a domain expert) and a MRD. DODDLE gets a trimmed model by removing unnecessary internal terms from the initial model. In order to manage concept drift, DODDLE tries to identify which part should be drifted in the trimmed model based on match result analysis, and moves the part to proper location in the initial model interacting with the user, and gets a final domain ontology (a hierarchically structured set of domain concepts). We have been done experiments in the following three domains: transportation, a particular law and software engineering processes. The empirical results have shown us that DODDLE can support a user in constructing a domain ontology.
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) at knowledge levels. 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 contents or the methodology of constructing ontologies. According to [Heijst 1995], 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[Guha and Lenat, 1994], WordNet[Miller, 1990] and EDR[EDR 1993]. Task ontologies have also been developed from abstract models of methods, such as Generic Tasks[Bylander and Chandrasekaran, 1986], PROTEGE-II[Munsen et al., 1994] and CommonKADS[Breuker and Van de Velde,1994]. 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[Aoki et al., 1996] . 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[EDR 1993]. 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 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, experiments have been done in the following three domains: transportation, the particular law called Contracts for the International Sale of Goods (CISG) and software (engineering) processes. The empirical results have shown us that DODDLE can support a user in constructing a domain ontology.
2 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: Ontological Bugs

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 to do so, as described in section 3.
3 DODDLE Design

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

3.1 An Overview of DODDLE

Figure 2 shows an overview for 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. DODDLE does spell match between the input and a MRD. The input domain terms are linked to WordNet by the spell match. The spell match results are a hierarchically structured set of all the nodes on the path from the input terms to the root of WordNet. 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
**Table 1: WordNet**

<table>
<thead>
<tr>
<th>Dictionary Name</th>
<th>word synsets</th>
<th>word senses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun dictionary</td>
<td>60557</td>
<td>107424</td>
</tr>
<tr>
<td>Verb dictionary</td>
<td>11363</td>
<td>25761</td>
</tr>
<tr>
<td>Adjective dictionary</td>
<td>16328</td>
<td>28749</td>
</tr>
<tr>
<td>Adverb dictionary</td>
<td>3243</td>
<td>6201</td>
</tr>
<tr>
<td>Index dictionary</td>
<td>91519</td>
<td>119217</td>
</tr>
</tbody>
</table>

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 two basic processes to do so: removing unnecessary internal terms in the initial model (called ‘a trimmed model’ later) and finding out which part should be drifted in the trimmed model. 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.

### 3.2 WordNet

DODDLE takes WordNet [Miller, 1990] 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.

### 3.3 Match Result Analysis for 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 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.

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

---

5
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 4 illustrates examples of PABs and STMs in a trimmed model.

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 5 shows a typical screen of DODDLE.
Table 2: DODDLE Specifications

<table>
<thead>
<tr>
<th>Module</th>
<th>Language</th>
<th>Size (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Construct an Initial Model</td>
<td>Perl &amp; Tcl-Tk</td>
<td>50.8</td>
</tr>
<tr>
<td>Identify PABs and STMs</td>
<td>Perl &amp; Tcl-Tk</td>
<td>31.7</td>
</tr>
<tr>
<td>GUI</td>
<td>Tcl-Tk</td>
<td>48.5</td>
</tr>
</tbody>
</table>

Figure 5: DODDLE Browser

4 Experimental Results

In order to evaluate how DODDLE is doing in practical fields, experiments have been done in the following three domains: transportation, a particular law called Contracts for the International Sale of Goods (CISG) and software (engineering) processes.

In the first experiment, DODDLE has been applied to a small size of transportation domain that has just 12 terms as shown in Table 3. After doing spell match completely and trimming, DODDLE got a trimmed model composed of 5 SINs and 12 best-matched nodes, as shown in Figure 6. A user moved 3 STMs and modified just 5 nodes by hand later. Figure 7 shows us a final transportation ontology. DODDLE works so well in this small size of transportation domain.

In the second experiment, DODDLE has been applied to CISG. Our third author have studied CISG for two years and gave 46 legal terms from CISG Part-II to DODDLE. Table 4 shows the input legal terms. DODDLE did spell match between 46 input legal
terms and WordNet. Legal terms, "offerer" and "offeree", have not been matched with WordNet and so added later by hand to the legal ontology. DODDLE identified the best-matched nodes interacting with the user and got an initial model. After trimming the initial model, DODDLE left 13 SINs and 44 best-matched nodes and thus got a trimmed model as shown in Figure 8. After dividing 11 PABs and 13 STMs in the trimmed model, the user moved 7 SINs in the trimmed model. For example, STM in which the root is artifact was moved to the sub-node of address (But artifact node have been deleted later by the user). Because the meaning of "place of business" have more relation with place rather than thing in a legal domain. Afterwards, the user has done
Table 4: Input terms (law)

<table>
<thead>
<tr>
<th>acceptance</th>
<th>delay</th>
<th>modification</th>
<th>quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>act</td>
<td>delivery</td>
<td>offer</td>
<td>quantity</td>
</tr>
<tr>
<td>addition</td>
<td>dispatch</td>
<td>offerer</td>
<td>rejection</td>
</tr>
<tr>
<td>address</td>
<td>discrepancy</td>
<td>offeree</td>
<td>reply</td>
</tr>
<tr>
<td>assent</td>
<td>effect</td>
<td>party</td>
<td>residence</td>
</tr>
<tr>
<td>circumstance</td>
<td>envelope</td>
<td>payment</td>
<td>revocation</td>
</tr>
<tr>
<td>communication system</td>
<td>goods</td>
<td>person</td>
<td>silence</td>
</tr>
<tr>
<td>conduct</td>
<td>holiday</td>
<td>place of business</td>
<td>speech act</td>
</tr>
<tr>
<td>contract</td>
<td>indication</td>
<td>price</td>
<td>telephone</td>
</tr>
<tr>
<td>counteroffer</td>
<td>intention</td>
<td>proposal</td>
<td>telex</td>
</tr>
<tr>
<td>day</td>
<td>invitation</td>
<td>proposal</td>
<td>time</td>
</tr>
<tr>
<td></td>
<td>letter</td>
<td>withdrawal</td>
<td>transmission</td>
</tr>
</tbody>
</table>

Figure 8: A Trimmed Model (Law)
additional modifications to 27 nodes by hand and got a final legal ontology, as shown in Figure 9. The user had more or less costs to do additional modifications.

In the third experiment, DOODLE has been applied to a small size of software process domain, composed of 21 technical terms from Articulator (Peiwei et al., 1990), as shown in Table 5. In doing spell match, 6 terms have been not matched with WordNet because of high domain specificity and so added later by hand. Figure 10 and 11 show us a trimmed model and a final software process ontology. 6 out of 10 STMs have been moved by the user and 13 nodes have been modified additionally by the user. The concept of “resource” changes into a root node in a software process ontology and so the change have infected other nodes so much. However, if root node exchange would come first, DOODLE worked well.

Table 5: Input Terms (Software Process)

<table>
<thead>
<tr>
<th>resource</th>
<th>agent</th>
<th>budget</th>
<th>setting</th>
<th>role</th>
<th>document</th>
<th>task</th>
<th>primary_task</th>
<th>articulation_task</th>
<th>people</th>
<th>organization</th>
<th>team</th>
<th>working_group</th>
<th>individual_primary_task</th>
<th>software</th>
<th>collective_primary_task</th>
<th>hardware</th>
<th>accommodation</th>
<th>individual_agent</th>
<th>negotiation</th>
<th>collective_agent</th>
</tr>
</thead>
</table>

Suppose that a final domain ontology has X nodes and user’s additional modifications have been done to Y nodes. DOODLE’s support rate has been computed by (X-Y)/X. Using this definition, support rates were 87%, 64% and 0% to transportation, CISG and software process domains. However, if DOODLE would be applied after root node
exchange in software process domain, the support rate became 80%. So these evaluation results show us that DODDLE can work well over several different domains.

5 Related Work

Because domain ontologies have a 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 [Heijst 1995]. When similar domain ontologies are missing in constructing a new domain ontology, it
is hard to construct it.

On the other hand, Ontosaurus [Swartout et. al. 1996] has points similar to DODDLE. Ontosaurus constructs a domain ontology using SENSUS [Knight and Luk, 1994] 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 [Aoki et al., 1996] 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.

6 Conclusions and Future Work

This paper discusses how to construct a domain ontology using existing MRDs. To do so, concept drift came up as an important technical issue. In order to make concept drift operational, match result analysis has been proposed and empirical results show us that it works well. However, there still are concept drift not to be operationalized. We need more analysis for concept drift. We can use trim result analysis as well as match result analysis. For example, there may be some difference between one part with many trimmed nodes and another part with less trimmed nodes. The difference analysis between them may be effective for managing concept drift. Then human experts have much objection against lower part of initial or trimmed models. Because the nodes on lower part have much relation with domain specificity, it may be a better way for a user to give a small size of domain theory. Then it can be integrated with upper part from a MRD. Furthermore, it is possible to reuse and adjust other ontologies besides a MRD. For example, data base ontologies can be built automatically by extension subsumption relationships between fields. It is another interesting issue to integrate heterogeneous ontologies. Finally, after constructing domain ontologies, we will get into how to use domain ontologies in order to build and validate existing knowledge bases.

Acknowledgments

We would like to express our thanks to Professor H.Yoshino and the members in the study of legal expert systems development. Mr. S.Komori worked together with us in doing experiments.

References


