# DODDLE-OWL: OWL-based Semi-Automatic Ontology Development Environment

Takeshi Morita[1], Yoshihiro Shigeta[1], Naoki Sugiura[1], Naoki Fukuta[1], Noriaki Izumi[2], and Takahira Yamaguchi[3]

[1] Shizuoka University, 3-5-1 Johoku, Hamamatsu, Shizuoka 432-8011, Japan, morita@ks.cs.inf.shizuoka.ac.jp, http://mmm.semanticweb.org
[2] National Institute of AIST, 2-41-6, Aomi, Koto-ku, Tokyo, Japan
[3] Keio University, 4-1-1 Hiyoshi, Kohoku-ku, Yokohama-shi, Kanagawa, Japan

**Abstract.** In this paper, we propose an ontology development support environment for the Semantic Web. The advantage of our environment is focusing the quality refinement phase of ontology construction. Through interactive support for refining the initial ontology, OWL-Lite level ontology, which consists of taxonomic relationships (class - sub class relationship) and non-taxonomic relationships (defined as property), is constructed effectively. The environment also provides semi-automatic generation of the initial ontology.

## 1 Introduction

The Semantic Web [1] is now gathering attentions from researchers in wide area. Adding semantics (meta-data) to the Web contents, software agents are able to understand and even infer Web resources. To realize such paradigm, the role of ontologies [2] is important in terms of sharing common understanding among both people and software agents [3]. In knowledge engineering field ontologies have been developed for particular knowledge system mainly to reuse domain knowledge. On the other hand, for the Semantic Web, ontologies are constructed in distributed places or domain, and then mapped each other. For this purpose, it is an important task to realize a software environment for rapid construction of ontologies for each domain. Towards the on-the-fly ontology construction, many researches are focusing on automatic ontology construction from existing Web resources, such as dictionaries, by machine processing with concept extraction algorithms. However, depending on domains (a law domain etc.), the important concepts which doesn't occur frequently in the resources may be required to be added by hand for ontology construction. In such a domain, if a user doesn't intervene, constructing ontologies cannot readily be done. Considering such situation, we believe that the most important aspect of the on-the-fly ontology construction is that how efficiently the user is able to complete making the ontology for the Semantic Web contents available to the public. For this reason, ontologies should be constructed not fully automatically, but through interactive support by software environment from the early stage of ontology construction.

Although it may seem to be contradiction in terms of efficiency, the total cost of ontology construction would become less than automatic construction since if the ontology is constructed with careful interaction between the system and the user, less miss-construction will be happened. It also means that high-quality ontology would be constructed.

In this paper, we propose a software environment for user-centered on-the-fly ontology construction named DODDLE-OWL (Domain Ontology rapiD DeveLopment Environment - OWL [4] extension). The architecture of DODDLE-OWL is re-designed based on DODDLE-II [5], the former version of DODDLE-OWL. DODDLE-OWL has the following five modules: Input Module, Construction Module, Refinement Module, Visualization Module, and Translation Module. Especially, to realize the user-centered environment, DODDLE-OWL dedicates to the Refinement Module. It enables us to develop ontologies with interactive indication of which part of ontology should be refined. DODDLE-OWL supports the construction of both taxonomic relationships and non-taxonomic relationships in ontologies. Since DODDLE-II has been built for ontology construction not for the Semantic Web but for typical knowledge systems, it needs some extensions for the Semantic Web such as OWL (Web Ontology Language) [4] export facility. DODDLE-OWL contributes the evolution of ontology construction and the Semantic Web.

## 2 The DODDLE-OWL Architecture

Figure 1 shows the overview of DODDLE-OWL. The main feature of DODDLE-OWL has the following five modules: Input Module, Construction Module, Refinement Module, Visualization Module, and Translation Module. The Input Module, the Hierarchy Construction Module, and the Hierarchy Refinement Module are included in DODDLE-I to support a user to construct taxonomic relationship. The Relationship Construction Module and the Relationship Refinement Module were added on DODDLE-II development that supports constructing both taxonomic and non-taxonomic relationships. The Visualization Module and the Translation Module are newly developed in DODDLE-OWL to extend functionalities for the Semantic Web such as exporting constructed ontology in OWL format. Here, we assume that there are one or more domain specific documents, and we also assume that the user can select important terms that are needed to construct a domain ontology. First, the user selects input concepts (important terms) in the Input Module. In the Construction Module, DODDLE-OWL generates the basis of the ontology, an initial concept hierarchy and set of concept pairs, based on input concepts by referring to WordNet [6] as an MRD (Machine Readable Dictionary) and documents. An initial concept hierarchy is constructed as an IS-A hierarchy of terms. Set of concept pairs are extracted by using co-occurrency based statistic methods. These pairs are considered to be closely related and that will be used as candidates to refine and add non-taxonomic relations. The user identifies some relationship between concepts in the pairs. In the Refinement Module, the initial ontology produced

by the Construction Module is refined by the user through interactive support by DODDLE-OWL. In order to refine the initial ontology, we manage concept drift and evaluate set of concept pairs. Since the initial concept hierarchy is constructed from a general ontology, we need to adjust the initial concept hierarchy to the specific domain considering an issue called Concept Drift. It means that the position of particular concepts changes depending on the domain. For concept drift management, DODDLE-OWL applies two strategies: Matched Result Analysis and Trimmed Result Analysis. These strategies are described in our former study [5]. At the construction phase of concept specification template from set of concept pairs generated by the Construction Module, DODDLE-OWL needs a criterion to evaluate significant concept pairs. In [5], two statistics based methods are investigated: the value of context similarity by WordSpace method [7] and the value of confidence by the association rule learner [8]. Those methods and values based on co-occurrency of concepts work well in terms of wide use (do not depend on some particular domains), its cost of preparation. The ontology constructed by DODDLE-OWL can be exported with the representation of OWL. Finally, $MR^3$ (Meta-Model Management based on RDF(S) [9] Revision Reflection) [10] is connected with DODDLE-OWL and works with an RDF(S) graphical editor.

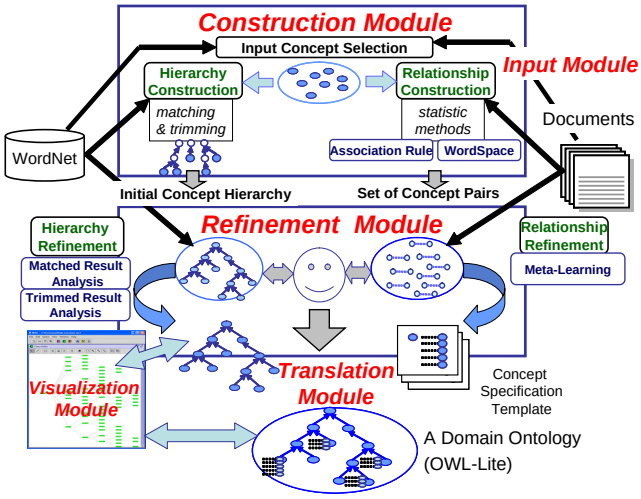


**Fig. 1.** DODDLE-OWL overview

## 3 Implementation

In this section, we describe the system architecture from the aspect of system implementation. DODDLE-OWL is realized in conjunction with $MR^3$ [10].

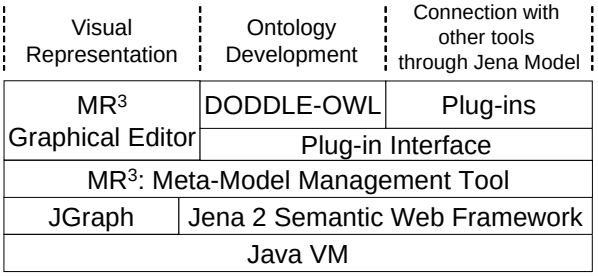


**Fig. 2.** DODDLE-OWL architecture

$MR^3$ is an RDF(S) graphical editor with meta-model management facility such as consistency checking of classes and a model in which these classes are used as the type of instances. Figure 2 shows the relationship between DODDLE-OWL and $MR^3$ in terms of system implementation. Both $MR^3$ and DODDLE-OWL are implemented in Java language. $MR^3$ is implemented using JGraph [11] for RDF(S) graph visualization, and Jena 2 Semantic Web Framework [12] for enabling the use of Semantic Web standards such as RDF, RDFS, N-triple and OWL. By using these libraries, $MR^3$ is implemented as an environment for graphical representation of the Semantic Web contents. Additionally, $MR^3$ also has plug-in facility to extend its functionality.

Figure 3 shows a typical usage of DODDLE-OWL. DODDLE-OWL's user interface consists of Input Module, Construction & Refinement Modules for Hierarchy, Construction & Refinement Modules for Relationships, Visualization Module $MR^3$ , and Translation Module into OWL-Lite. First, the user opens a document ((1) in Figure 3). Then, in the Input Module ((2) in Figure 3), the user can see noun terms that come up sorted frequently in the document. The user selects some terms as the input from the noun terms. As input of DODDLE-OWL, the user associates those terms with concepts by referring "the WordNet concepts" in (2) of Figure 3. For example, the user decide which "concept" (i.e. synset in WordNet) is suitable for the term "party". By referring to the synset and term's definition, the user selects an appropriate concept for the word "party". After mapping terms and their synsets, an initial concept hierarchy is produced. Also set of concept pairs are extracted by statistic methods such as WordSpace method and the association rule learner by default parameters. (3) of Figure 3 shows the Construction & Refinement Modules for Hierarchy. This module indicates some groups of concepts in the taxonomy so that the user can decide which part should be refined. (4) of Figure 3 shows the display of concept drift management in the Visualization Module $MR^3$ . (5) of Figure 3 shows the Construction & Refinement Modules for Relationships. This module is used for setting parameters used in the WordSpace method and the association rule learner to apply to documents in order to generate significantly related concept pairs. In WordSpace method, there are parameters such as the gram number (de-

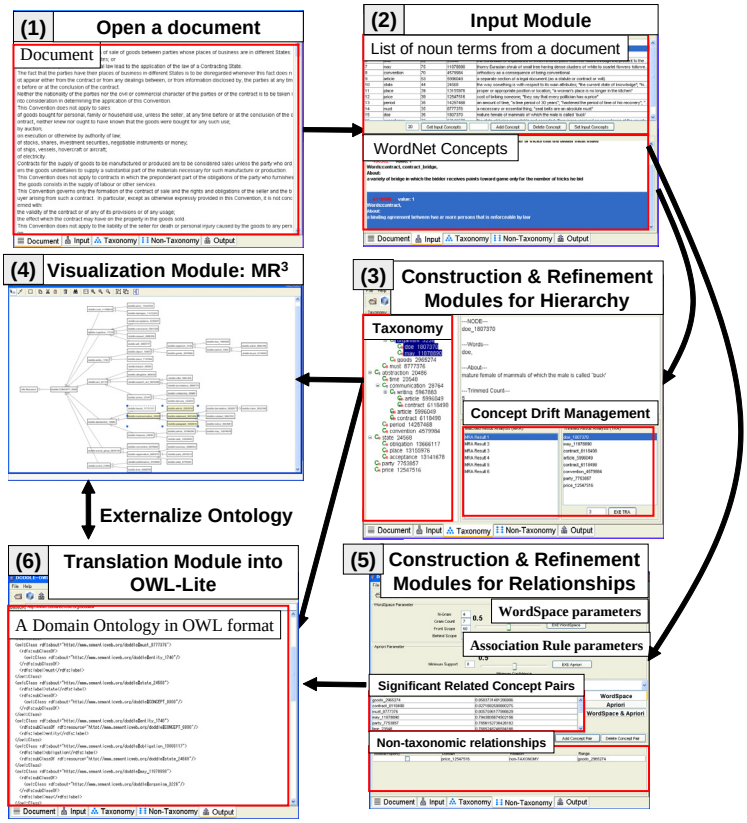


**Fig. 3.** A typical usage of DODDLE-OWL

fault gram number is four), minimum N-gram count (to extract high-frequency grams only), front scope and behind scope in the text. In the association rule learner, minimum confidence and minimum support are set by the user. As a result, the user got a domain ontology as (6) in Figure 3.

## 4 Case Studies

In order to evaluate how DODDLE-OWL is doing in a practical field, case studies have been done in particular field of business. The particular field of business is called "XML Common Business Library"(xCBL) [13].

### 4.1 A Case Study in the Business Field

**Input terms in the Case Study with xCBL** Table 1 shows input terms in this case study. They are 57 business terms extracted by a user from xCBL Document Reference. The user is not a expert but has business knowledge.

**Table 1.** Significant 57 Concepts in xCBL

| | | | | |
|---|---|---|---|---|
| acceptance | agreement | auction | availability | business |
| buyer | change | contract | customer | data |
| date | delivery | document | exchange rate | financial institution |
| foreign exchange | goods | information | invoice | item |
| line item | location | marketplace | message | money |
| order | organization | partner | party | payee |
| payer | payment | period of time | price | process |
| product | purchase | purcahse agreement | purchase order | quantity |
| quotation | quote | receipt | rejection | request |
| resource | response | schedule | seller | service |
| shipper | status | supplier | system | third party |
| transaction | user | | | |

**Table 2.** The Change of the Number of Concepts under Taxonomic Relationship Acquisision

| Model | Input Terms | Initial Model | Triimed Model | Concept Hierarchy |
|---|---|---|---|---|
| # Concept | 57 | 152 | 83 | 82 |

**Taxonomic Relationship Acquisition** Table 2 shows the number of concepts in each model under taxonomic relationship acquisition and table 3 shows the evaluation of two strategies by the user. The recall per subtree is more than 0.5 and is good. The precision and the recall per path are less than 0.3 and are not so good, but about 80 % portion of taxonomic relationships were constructed with Hierarchy Construction Module and Hierarchy Refinement Module support.

**Non-Taxonomic Relationship Learning**

*1. Construction of WordSpace*

High-frequency 4-grams (sets of four words that co-occur term-by-term) were extracted from xCBL Document Description (about 2,500 words) standard form conversion removed duplication, and 1240 kinds of 4-grams were obtained. In order to keep density of a collocation matrix high, the extraction frequency of 4-grams must be adjusted according to the scale of text corpus. In order to construct a context vector, a sum of 4-gram vectors around appearance place circumference of each of 57 concepts was calculated. In order to construct a context scope from some 4-grams, it consists of putting together 10 4-grams before the 4-gram and 10 4-grams after the 4-grams independently of length of a sentence. For each of 57 concepts, the sum of context vectors in all the appearance places of the concept in xCBL was calculated, and the vector representations of the concepts were obtained. The set of these vectors is used as WordSpace to extract concept pairs with context similarity. Having calculated the similarity from the inner product for concept pairs which is all the combination of 57 concepts, 40 concept pairs were extracted.

**Table 3.** Precision and Recall in the Case Study with xCBL

| | Precision | Recall per Path | Recall per Subtree |
|---|---|---|---|
| Matched Result | 0.2 (5/25) | 0.29 (5/17) | 0.71 (5/7) |
| Trimmed Result | 0.22 (2/9) | 0.13 (2/15) | 0.5 (2/4) |

**Table 4.** Evaluation by the User with xCBL definition

| | WordSPace (WS) | Association Rules (AR) | The Join of WS and AR |
|---|---|---|---|
| # Extracted concept pairs | 40 | 39 | 66 |
| # Accepted concept pairs | 30 | 20 | 39 |
| # Rejected concept pairs | 10 | 19 | 27 |
| Precision | 0.75 (30/40) | 0.51 (20/30) | 0.59 (39/66) |

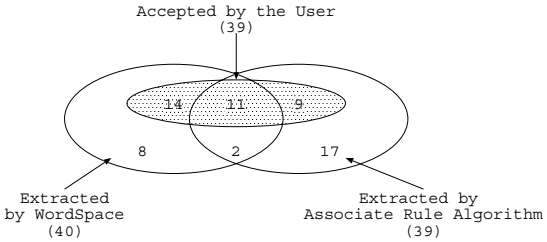


**Fig. 4.** Two Difference Sets of Concept Pairs from WS and AR and Concept Sets have Relationships

*2. Relationship Construction Module*
DODDLE-OWL extracted 39 pairs of terms from text corpus using the above-mentioned association rule algorithm. There are 13 pairs out of them in a set of similar concept pairs extracted using WordSpace. Then, DODDLE-OWL constructed concept specification templates from two sets of concept pairs extracted by WordSpace and Associated Rule algorithm.
*3. Evaluation of Results of Relationship Refinement Module*
The user evaluated the following two sets of concept pairs: one is extracted by WS (WordSpace) and the other is extracted by AR (Association Rule algorithm). Figure 4 shows two different sets of concept pairs from WS and AR. It also shows portion of extracted concept pairs that were accepted by the user. Table 4 shows the details of evaluation by the user, computing precision only. Since the user didn't define concept definition in advance, we can not compute recall. Looking at the field of precision in Table 4, the precision from WS is higher than others. Most of concept pairs which have relationships were extracted by WS. The percentage is about 77% (30/39). But there are some concept pairs which were not extracted by WS. Therefore taking the join of WS and AR is the best method to support a user to construct non-taxonomic relationships.

### 4.2 Results and Evaluation of Case Studies

In regards to support in constructing taxonomic relationships, the precision and recall are less than 0.3 in the case study. Generally, 70 % or more support comes from Hierarchy Construction Module and Hierarchy Refinement Module. About more than half portion of the final domain ontology results in the information extracted from WordNet. Since the two strategies just imply the part where concept drift may come up, the part generated by them has low component rates and about 30 % hit rates. So one out of three indications based on the two strategies work well in order to manage concept drift. The two strategies use matched and trimmed results, therefore based on structural information of an MRD only, the hit rates are not so bad. In order to manage concept drift

smartly, we may need to use more semantic information that is not easy to come up in advance in the strategies, and we also may need to use domain specific text corpus and other information resource to improve supporting a user in constructing taxonomic relationships.

In regards to construction of non-taxonomic relationships, the precision in the case study with xCBL is good. Generating non-taxonomic relationships of concepts is harder than modifying and deleting them. Therefore, DODDLE-OWL supports the user in constructing non-taxonomic relationships.

After analyzing results of case studies, we have the following problems.

**1. Determination of a Threshold:** Threshold of the context similarity changes in effective value with each domain. It is hard to set up the most effective value in advance.

**2. Specification of a Concept Relation:** Concept specification templates have only concept pairs based on the context similarity, it requires still high cost to specify relationships between them. It is needed to support specification of concept relationships on this system in the future work.

**3. Ambiguity of Multiple Terminology:** For example, the term "transmission" is used in two meanings, "transmission (of goods)" and "transmission (of communication)", in a document, but DODDLE-OWL considers these terms as the same and creates WordSpace as it is. Therefore constructed vector expression may not be exact. In order to extract more useful concept pairs, semantic specialization of a multisense word is necessary, and it should be considered that the 4-grams with same appearance and different meaning are different 4-grams.

## 5 Related Work

Navigli et,al. proposed OntoLearn [14], that supports domain ontology construction by using existing ontologies and natural language processing techniques. In their approach, existing concepts from WordNet are enriched and pruned to fit the domain concepts by using NLP (Natural Language Processing) techniques. They argue that the automatically constructed ontologies are practically usable in the case study of a terminology translation application. However, they did not show any evaluations of the generated ontologies themselves that might be done by domain experts. Although a lot of useful information is in the machine readable dictionaries and documents in the application domain, some essential concepts and knowledge are still in the minds of domain experts. We did not generate the ontologies themselves automatically, but suggests relevant alternatives to the human experts interactively while the experts' construction of domain ontologies. In another case study [15], we had an experience that even if the concepts are in the MRD (Machine Readable Dictionary), they are not sufficient to use. In the case study, some parts of hierarchical relations are counterchanged between the generic ontology (WordNet) and the domain ontology, which are called "Concept Drift". In that case, presenting automatically generated ontology that contains concept drifts may cause confusion of domain experts. We argue that the initiative should be kept not on the machine, but on the hand

of the domain experts at the domain ontology construction phase. This is the difference between our approach and Navigli's. Our human-centered approach enabled us to cooperate with human experts tightly.

From the technological viewpoint, there are two different related research areas. In the research using verb-oriented method, the relation of a verb and nouns modified with it is described, and the concept definition is constructed from this information (e.g. [16]). In [17], taxonomic relationships and Subcategorization Frame of verbs (SF) are extracted from technical texts using a machine learning method. The nouns in two or more kinds of different SF with the same frame-name and slot-name are gathered as one concept, base class. And ontology with only taxonomic relationships is built by carrying out clustering of the base class further. Moreover, in parallel, Restriction of Selection (RS) which is slot-value in SF is also replaced with the concept with which it is satisfied instantiated SF. However, proper evaluation is not yet done. Since SF represents the syntactic relationships between verb and noun, the step for the conversion to non-taxonomic relationships is necessary.

On the other hand, in ontology learning using data-mining method, discovering non-taxonomic relationships using an association rule algorithm is proposed by [18]. They extract concept pairs based on the modification information between terms selected with parsing, and made the concept pairs a transaction.

By using heuristics with shallow text processing, the generation of a transaction more reflects the syntax of texts. Moreover, RLA, which is their original learning accuracy of non-taxonomic relationships using the existing taxonomic relations, is proposed. The concept pair extraction method in our paper does not need parsing, and it can also run off context similarity between the terms appeared apart each other in texts or not mediated by the same verb.

## 6 Conclusion

In this paper, we presented a support environment for ontology construction named DODDLE-OWL, which aims at a total support environment for user-centered on-the-fly ontology construction. Its main principle is that high-level support for users through interaction. First, the user selects some terms as the input in the Input Module. Then, the Construction Module generates the basis of ontology in the forms of an initial concept hierarchy and set of concept pairs, by referring to WordNet as an MRD and a document. The Refinement Module provides management facilities for concept drift in the taxonomy and identifying significant set of concept pairs in extracted related concept pairs. According to case studies, it is important to select combinations of algorithms to get better candidate of set of concept pairs. Finally, the Translation Module produces an OWL-Lite file, which is able to put on public as a Semantic Web ontology.

We think that meta-learning scheme can be applied to the Refinement Module of DODDLE-OWL. CAMLET [19], a constructive meta-learning scheme has been proposed that can reconstruct learning algorithms from method level. This

approach will help to determine which learning algorithm to use on extracting set of concept pairs on each domain.

## References


1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American (2001)
2. Heijst, G.V.: The Role of Ontologies in Knowledge Engineering. Dr.thesis, University of Amsterdam (1995)
3. Ding, Y., Foo, S.: Ontology Research and Development, Part 1 – a Review of Onlotogy. Journal of Information Science (2002) pp.123–136
4. Michael K. Smith, C.W., McGuinness, D.L.: OWL Web Ontology Language Guide (2004) http://www.w3.org/TR/owl-guide/.
5. Sugiura, N., et al.: A Domain Ontology Engineering Tool with General Ontologies and Text Corpus. Proceedings of the 2nd Workshop on Evaluation of Ontology based Tools (2003) pp.71–82
6. G.A.Miller: WordNet: A Lexical Database for English. ACM (1995) pp.39–41
7. Marti A. Hearst, H.S.: Customizing a Lexicon to Better Suit a Computational Task. Corpus Processing for Lexical Acquisition (1996) pp.77–96
8. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. Proceedings of VLDB Conference (1994) pp.487–499
9. Brickley, D., Guha, R.: RDF Vocabulary Description Language 1.0: Rdf Schema. W3C Proposed Recommendation (2003) http://www.w3.org/TR/2004/REC-rdf-schema-20040210/.
10. Takeshi Morita, Noriaki Izumi, Naoki Fukuta and Takahira Yamaguchi: $MR^3$: Meta-Model Management based on RDFs Revision Reflection. Proceedings of the 6th Joint Conference on Knowledge-Based Software Engineering (JCKBSE) (2004) pp.228–236
11. Alder, G.: Jgraph. (2003) http://www.jgraph.com.
12. HP Labs: Jena Semantic Web Framework. (2003) http://jena.sourceforge.net/downloads.html.
13. One, C.: (xcbl:xml common business library) http://www.xcbl.org/.
14. Navigli, R., Paola Velardi: Automatic Adaptation of WordNet to Domains. Proceedings of International Workshop on Ontologies and Lexical Knowledge Bases (2002)
15. Yamaguchi, T.: Constructing domain ontologies based on concept drift analysis. Proceedings of the IJCAI99 Workshop on Ontologies and Problem Solving methods(KRR5) (1999)
16. Hahn, U., Schnattingerg, K.: Toward text knowledge engineering. AAAI-98 proceedings (1998) pp.524–531
17. Faure, D., Nédellec, C.: Knowledge Acquisition of Predicate Argument Structures from Technical Texts. Proceedings of International Conference on Knowledge Engineering and Knowledge Management (1999)
18. Maedche, A., Staab, S.: Discovering Conceptual Relations from Text. Proceedings of 14th European Conference on Artificial Intelligence (2000) pp.321–325
19. Hidenao Abe and Takahira Yamaguchi: Constructive meta-learning with machine learning method repositories. in Proc. of the 17th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expoert Systems (IEA/AIE) (2004) pp.502–511