Available online at www.sciencedirect.com
ScienceDirect
Information Processing and Management 43 (2007) 930–945
INFORMATION PROCESSING & MANAGEMENT
www.elsevier.com/locate/infoproman

Schema and constraints-based matching and merging of Topic Maps

Jung-Mn Kim [a,*], Hyopil Shin [b], Hyoung-Joo Kim [a]

[a] *IDB Lab. School of Computer Engineering, Seoul National University, San 56-1, Shillim-dong, Kwanak-gu, Seoul 151-742, Republic of Korea*
[b] *CL Lab. Department of Linguistics, Seoul National University, San 56-1, Shillim-dong, Kwanak-gu, Seoul 151-742, Republic of Korea*

Received 25 April 2006; received in revised form 3 August 2006; accepted 16 August 2006
Available online 1 November 2006

Abstract

In this paper, we propose a multi-strategic matching and merging approach to find correspondences between ontologies based on the syntactic or semantic characteristics and constraints of the Topic Maps. Our multi-strategic matching approach consists of a linguistic module and a Topic Map constraints-based module. A linguistic module computes similarities between concepts using morphological analysis, string normalization and tokenization and language-dependent heuristics. A Topic Map constraints-based module takes advantage of several Topic Maps-dependent techniques such as a topic property-based matching, a hierarchy-based matching, and an association-based matching. This is a composite matching procedure and need not generate a cross-pair of all topics from the ontologies because unmatched pairs of topics can be removed by characteristics and constraints of the Topic Maps. Merging between Topic Maps follows the matching operations. We set up the MERGE function to integrate two Topic Maps into a new Topic Map, which satisfies such merge requirements as entity preservation, property preservation, relation preservation, and conflict resolution. For our experiments, we used oriental philosophy ontologies, western philosophy ontologies, Yahoo western philosophy dictionary, and Wikipedia philosophy ontology as input ontologies. Our experiments show that the automatically generated matching results conform to the outputs generated manually by domain experts and can be of great benefit to the following merging operations.

Keywords: Ontology matching; Ontology merging; Topic Maps matching; Topic Maps merging

1. Introduction

Finding semantic correspondences between ontologies is fundamental work for ontology-based applications such as Semantic Web, Knowledge Management System, and E-Commerce, because of the need for

* Corresponding author. Tel.: +82 2 886 1830; fax: +82 2 882 0269.
E-mail addresses: jmkim@idb.snu.ac.kr (J.-M. Kim), hpshin@snu.ac.kr (H. Shin), hjk@idb.snu.ac.kr (H.-J. Kim).

0306-4573/$ - see front matter © 2006 Published by Elsevier Ltd.
doi:10.1016/j.ipm.2006.08.012

the matching, merging, aligning, and integrating between different ontologies. In this paper, we propose a new multi-strategic Topic Map matching and merging methodologies to establish interoperability between services or applications based on Topic Maps. Topic Maps (Biezunski, Bryan, & Newcomb, 2002) as well as RDF (Resource Description Framework) (Lassila & Swick, 1999), and OWL (Web Ontology Language) (McGuinness & Harmeba, 2003) are data models for representing and building machine-understandable ontologies on the computer.

In recent years, many approaches for ontology matching have been proposed. All of these earlier approaches for schema or ontology matching, however, focused on providing various techniques for effective matching and merging of schemas or ontologies (Rahm & Bernstein, 2001). They were far from efficiency considerations and thus are not suitable for practical applications based on ontologies of real world domains (Ehrig & Staab, 2004). Also, earlier approaches convert ontologies or schemas of relational database, object oriented database, and XML, into a graph model with only nodes and edges for supporting different applications and multiple schema types (Bouquet, Serafini, & Zanobini, 2003; Giunchglia & Shvaiko, 2003). This conversion results in low efficiency because the characteristics of ontologies that are useful for similarity computation are overlooked. Another problem with the existing matching methods is that given two ontologies 01 and 02, for each entity in ontology 01, they are compared with all entities in ontology 02. This full scanning on ontology 01 and 02 also ends up with low efficiency.

In this paper, we present an approach that considers features of both Topic Maps to reduce the matching complexity and linguistic analysis to improve the matching performance. Our approach does not require ontologies to be converted into a graph model and the entities to be fully scanned into two ontologies. Furthermore, our approach is a composite combination of four matching techniques from both a linguistic and a Topic Maps module: name matching, internal structure matching, external structure matching, and association matching. This composite matching approach combines the results of four matching techniques that are independently processed to measure the unified similarity of each pair.

To evaluate the quality of our approach, we apply it on three kinds of experimental data ontology group A, B, and C. The ontology group A includes ontologies constructed from one knowledge domain, i.e. philosophy by a group of domain experts, which explains why they have similar structure of knowledge organization. The ontology group B includes ontologies constructed from similar knowledge domains but have different structures of knowledge organization. The ontology group C includes ontologies constructed from different knowledge domains, i.e. philosophy and literature. We built an ontology for philosophy learning domain, hence given the name philosophy ontology (Kim, Choi, & Kim, 2004). We use the philosophy ontology, Wikipedia philosophy ontology which is constructed from philosophy-related contents of Wikipedia, and German literature ontology which is constructed from contents on German literature in the Yahoo encyclopedia as experimental data.

We use three measurements such as precision, recall, and overall, which were derived from the Information retrieval field, to evaluate the quality of our approach. We then evaluated the approach by computing three measurements based on a set of manually determined matches and a set of automatically generated matches by matching operations. Based on the experimental results, we could conclude that automatically generated matches by our matching operation can cover most of the manually determined matches.

2. Related work

Schema matching is a process of finding semantic correspondences between entities of two schemas. It is a critical operation found in many schema and data integration applications, such as data warehouse, electronic consumer, data integration, and so on. Earlier approaches for semi-automatic matching between two schemas are COMA (Do & Rahm, 2002), Cupid (Madhavan, Bernstein, & Rahm, 2001), LSD (Doan, Domingos, & Halevy, 2001), MOMIS (Beneventano, Bergamaschi, Guerra, & Vincini, 2001), SemInt (Li, Clifton, & Liu, 2000), and Similarity Flooding (Melnik, Garcia-Molina, & Rahm, 2002). These approaches propose techniques to solve the problems of matching between entities of a relational database, XML, ER model, and graph.

A recent survey on schema matching showed a classification of schema matching approaches in terms of the scope of entities to compare and matching techniques. Schema matching approaches are classified into

Table 1
Comparison of the matching and merging methods

Methods	L	P	D	R	C
PROMPT	Graph	T/ES	HPKB	Merge	$N * M$
Ctx-Match	Graph	T/E	Toy	Mapping	$N * M$
IF-MAP	Graph	T/I	Toy	Mapping	$N * M$
FCA-Merge	Graph	T/I	Toy	Mapping	$N * M$
QOM	RDF	T/IS/ES/E	Real Onto.	Mapping	$n \log n$
TMRM	Topic Maps	T	–	Merge	$N * M$
SIM	Topic Maps	T/IS	Toy	Mapping	$N * M$
TM-MAP	Topic Maps	T/IS/ES/E	Real Onto.	Merge	$n \log n$

instance-level approaches, schema-level approaches, element-level approaches, and structure-level approaches in terms of the scope of entities to compare. They are also classified into syntactic approaches, structural approaches, and semantic approaches in terms of matching techniques (Rahm & Bernstein, 2001).

Semlnt is an element-level and structural approach as it determines matches between attributes of tables on relational database schema and processes matching operation on data types, length, and key information, whereas Cupid is a hybrid matching technique which combines results of syntactic, structural, and semantic techniques.

Ontology matching approaches are influenced by the schema matching approaches, and thus, have similarities with schema matching approaches in matching operations (Shvaiko & Euzenat, 2004). With respect to matching and merging ontologies, there have been a few approaches, such as PROMPT (Noy & Musen, 2000), Anchor-PROMPT (Noy & Musen, 2001), information flow (Kalfoglou & Schorlemmer, 2002), FCA-Merge (Stumme & Madche, 2001), QOM (Ehrig & Staab, 2004), and so on.

Topic Maps Reference Model (Durusau, Newcomb, & Barta, 2006) defines a generic merging function based on the equivalence rules to determine if two or more topic items can be merged. The equivalence rules include topic items, topic name item, variant name, occurrence item, association items, and association role item equivalence conditions. All of these conditions, however, evaluate only the equality of entities of Topic Maps. They do not consider the similarity of entities and composite match results of each entitie types.

To overcome this weakness, Subject Identity Measure (SIM) (Maicher & Witschel, 2004) was used to measure the similarity between topics based on their name similarity and occurrence similarity. In the SIM, the processes were only string comparison of the name of topics and resource data of occurrences. The hierarchical structure and association in Topic Maps are not considered.

Table 1 represents characteristics of the methods at a glance. Abbreviated column names mean that Language (L), Patterns (P), Experimental Data (D), Results (R), and Complexity (C). Patterns column indicates matching approaches such as terminological (T), internal structure (IS), external structure (ES), extensional (E), and instance (I). Our approach called TM-MAP is similar with QOM in terms of the use of features of a data model for an ontology to reduce the complexity of matching operation. Our approach is different from the previous ones in that it treats the matching problem of distributed Topic Maps.

3. Matching problem and process definition

3.1. Overview of topic maps data model

Topic Map is a technology for encoding knowledge and connecting this encoded knowledge to relevant information resources, It is used as a formal syntax for representing and implementing ontologies. Topic maps are organized around topics, which represent subjects of discourse; associations, which represent relationships between the subjects; and occurrences, which connect the subjects to pertinent information resources.

Definition 1. We define a Topic Map model as following seven tuples:

$$TM := (T_C, T_O, T_A, T_R, T_I, R_H, R_A)$$

- T_C *denotes a set of topic types*
- T_O *denotes a set of occurrence types*
- T_A *denotes a set of association types*
- T_R *denotes a set of role types*
- T_I *denotes a set of instance topics*
- R_H *denotes a set of subsumption hierarchy relations*
- R_A *denotes a set of associative relations*

These entities have different meaning and usage, and so we measure the similarity between same entity types only.

3.2. Topic Map matching

Matching function f takes two source Topic Maps and domain-specific terminology dictionary $Dict_i$ as input and generates a set of match results M and increased dictionary $Dict_{i+1}$. It is depicted in Fig. 1.

Definition 2. A matching function *map* is defined as following expression:

$$\begin{aligned}&\boldsymbol{map}(\mathrm{A},\ \mathrm{B},\ \mathrm{D}) = \boldsymbol{map}(AT_C, BT_C, D)\cup\\ &\boldsymbol{map}(AT_C, BT_I, D)\cup\\ &\boldsymbol{map}(AT_I, BT_C, D)\cup\\ &\boldsymbol{map}(AT_O, BT_C, D)\cup\\ &\boldsymbol{map}(AT_A, BT_A, D)\cup\\ &\boldsymbol{map}(AT_R, BT_R, D)\end{aligned}$$

A and B are source Topic Maps and D is term dictionary. A matching function map (A, B, D) is processed by matching functions of different entity types.

3.3. Topic map matching process

The Topic Map matching process takes two Topic Maps as input and determines semantic correspondences between entities of the input Topic Maps. Our Topic Map matching process is composed of seven steps as depicted in Fig. 2.

Initialization. Step takes two serialized Topic Map documents, so-called XTM (XML Topic Maps) (Pepper & Moore, 2001), as input and interprets them to build Topic Maps in memory. During interpretation, PSI (Published Subject Indicator), which is used to share common description of topics between Topic Maps, and TopicWord indexes are generated for each Topic Map. Figs. 3 and 4 show the structure of PSI and TopicWord indexes respectively.

Entity pairs generation. Step creates the reduced number of entity pairs rather than whole entity pairs of two Topic Maps.

Entity pair selection. Step selects a pair of entities to be measured from a set of pairs of entities. We, first, select isolated entities which does not have any links from or to other entities. Then we select leaves from the hierarchy of entities. Using a bottom-up approach we select entities to be measured from leaves to root. Selected pair of entities is given to the next step to generate the similarity value between two entities.

Similarity computation. Step applies composite combination of matching techniques to measure similarity between entities based on the linguistic analysis. Our composite matching approach combines the results of

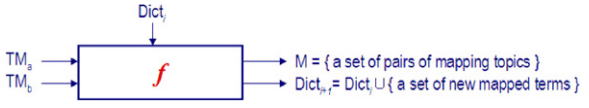


Fig. 1. Topic Maps matching function.

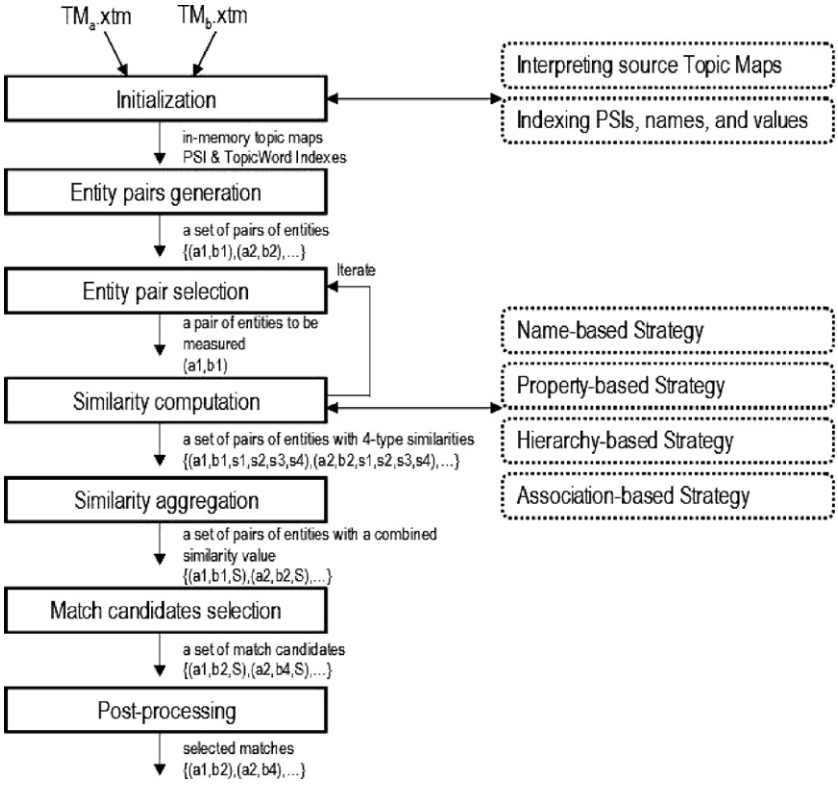


Fig. 2. Topic Maps matching process.

PSI Index
(TopicMap, TopicID, PSI, TreeLevel, Type/Instance)
TopicMap - Topic Map's ID. **TopicID** - Topic's ID which has one or more PSIs. **PSI** - String value or URI address for a PSI. **TreeLevel** - Level of hierarchy in which topic is located. **Type/Instance** - Whether class topic or instance topic.

Fig. 3. Structure of the PSI index.

TopicWord Index
(TopicMap, Word, TopicID, TreeLevel, Type/Instance, Scope, Type)
TopicMap - Topic Map's ID **Word** - A word is included in names or occurrences of the topic. **Name/Occ** - Name if a word is included in topic names or Occ if a word is included in occurrences. **TopicID** - Topic's ID. **TreeLevel** - Level of hierarchy in which topic is located. **Type/Instance** - Whether class topic or instance topic. **Scope** - Scope of a topic name. **Type** - TT for topic type, OT for occurrence type, AT for association type, or RT for role type.

Fig. 4. Structure of the TopicWord index.

independently executed four matching operations: name matching operation, property matching operation, hierarchy matching operation, and association matching operation.

Similarity aggregation. Step aggregates similarity values of four matching operations to generate a combined similarity value for each entity pair.

Match candidates selection. Step automatically chooses match candidates for an entity by selecting the entities of the other Topic Map with the best similarity value exceeding a certain threshold.

Post-processing. Step manually corrects the errors of automatically generated match results by domain experts.

4. Multi-strategies for match operations

4.1. Indexes for match operations

During interpretation of source Topic Maps, PSI and TopicWord indexes are generated for each Topic Map. The PSI index classifies topics by their subjects called Subject Identifier or Subject Locator in Topic Maps. According to Topic Maps standards, two topics will only be merged if their subjects are completely identical, regardless of their names or properties. This means that if two topics have exactly identical subjects, we can include them into the match results without processing any matching operation.

The TopicWord index is an inverted index of words in base names, variant names, and internal occurrences of topics and it is used to measure similarity between names or occurrences of topics. We extract words from topic names and internal occurrences and remove special characters, numbers, and stop words. For each word, we attach ID of topics which include the word in their names or occurrences. The TopicWord index is used to compute the similarity.

4.2. Name-based strategy

Name-based strategy compares strings of base names and variant names of topics. In the field terminology, a single term can refer to more than one concept and multiple terms can be related to a single concept. Name based strategy finds multiple terms referring to a same concept by applying two main categories of methods to the comparing terms. The methods are domain dictionary-based methods, string-based methods (simple token-based methods and token and substring-based method) and linguistic knowledge-based methods.

$$SIM_{\text{name}}(t_1, t_2) = (SIM_{\text{dict}}(t_1.names, t_2.names) + SIM_{\text{string}}(t_1.names, t_2.name))/2$$

Name-based similarity value, $SIM_{\text{name}}(t_1, t_2)$, between two topics t_1 and t_2 is a sum of a domain dictionary-based similarity value, $SIM_{\text{dict}}(t_1.names, t_2.names)$, and a string comparison-based similarity value, $SIM_{\text{string}}(t_1.names, t_2.name)$.

Our string comparison methods are token-based distance functions because most of all ontologies including our experimental ontologies use noun words or phrases for concept naming, e.g. ''Terms of Ancient Philosophy'' is used for a concept name instead of ''AncPhilosophy_Terms''.

4.2.1. Domain dictionary-based string matching

We developed a domain-specific dictionary to represent syntactic, synonymous, and antonymous relationships between terms using one or more philosophy thesauri written in Korean and English, i.e. Library of Congress Subject Headings in Philosophy: A Thesaurus (Berman, 2001). Our dictionary is a table with 4-tuples like (*term, related term, similarity score, scope*). A term and its related term have a domain-specific relation which is represented by their similarity score and scope. A similarity score is a value of range from 0 (*inequality*) to 1 (*equality*). The similarity scores between term and its related term are dependent to the scopes of the terms. For example, in the scope of terms of western modern philosophy, *reason* and *wisdom* have a similarity score 1 but in the scope of terms of oriental philosophy, a similarity score of the terms may be below 1.

Currently our dictionary has about 2100 terms of western and oriental philosophy which are defined as topics in Philosophy Topic Maps and will be increased by domain experts to include the popular terms of philosophy. Our domain-specific dictionary is useful to compute similarity values between two terms which

have different strings but similar meaning because string matching can produce a very low similarity value between them.

4.2.2. Simple token-based string matching

In token-based string matching methods, a string is a set of tokens (or words) rather than characters. These methods compare sets of tokens instead of strings. We use the Jaccard similarity method to compute similarity value between two strings.

$$SIM_{\text{string}}(a,b) = |a \cap b|/|a \cup b|$$

a and b are sets of tokens of two strings to be compared and $SIM_{\text{string}}(a,b)$ computes the ratio of common tokens to all tokens.

4.2.3. Token and substring-based string matching

A space between words is an obstacle to simple token matching. If two strings s and t have identical words except for spaces of string s, then these two strings have possibility to be matched.

$$SIM_{\text{string}}(s,t) = 0, \quad \text{for } s = w_1 +'' + w_2, \quad t = w_1 + w_2$$

To solve this mismatching problem we adopt a substring-based match operation between tokens. We use the following expression to measure the similarity between tokens:

$$SIM_{\text{token}}(x,y) = 2|c|/|x| + |y|$$

Both x and y are tokens and c is the largest common substring of them. The similarity value between two strings based on the token and substring-based method is computed by following expression. In this expression x_i is the ith token of string a and y_j is the jth token of string b.

$$TS - SIM_{\text{string}}(a,b) = \sum SIM_{\text{token}}(x_i, y_j)/|a \cup b|$$

4.2.4. Linguistic knowledge-based string matching

Given two strings '*Western Philosopher*' and '*Western Philosopher*', a SIM_{string} and $TS - SIM_{\text{string}}$ value of them is 0 and 0.54 respectively. However, we expect the similarity value of them is 1 because they are exactly matched. To improve the quality of string matching we use morphological and syntactic analysis to perform term normalization. From the above example '*Western Philosopher*' is splitted into two tokens, '*Western*' and '*Philosopher*' from a morphological analyzer. In many cases, a concept name can be a phrase or even a sentence in ontologies, to represent more specific semantics. For example, in the philosophy ontology, many concepts have noun phrases, such as '*Significance of free will*', '*raw and inevitability*', '*new requirement of question about being*', and so on.

In our morphological analysis for Korean and English, these phrases or sentences are divided into a several stems and inflectional endings, which attached to stems and represent various inflections or derivations in Korean. We process string matching between words and analyze their orders in the concept names. A word has different meaning in a phrase or a sentence according to word order or inflectional endings. Thus, in order to improve the quality of string match results between words, we use word order and ending information, which classify corresponding ending groups according to their meaning and usage.

4.3. Property-based strategy

If two topics have m occurrences and n occurrences each other, property-based strategy computes similarity values of m by n pairs of occurrences to measure the similarity between topics. An occurrence is denned by an occurrence type and an occurrence value which is a textual description or URI address. For example, a topic, *Immanuel Kant*, has a occurrence which type is '*figure*' and value is 'http://www.encyphilosophy.net/kant/figure.jpg'. Thus, the similarity values of occurrence types and occurrence values need to be combined to determine the occurrence-based similarity value of the paired topics.

$$SIM_{\text{occ}}(t_1, t_2) = \sum (SIM_{\text{occtype}})(t_1.occurence_i, t_2.occurence_j) \times SIM_{\text{occvalue}}(t_1.occurence_i, t_2.occurence_j)/|m| \times |n|, \quad \text{for } 1 \leqslant i \leqslant m \quad \text{and} \quad 1 \leqslant j \leqslant n$$

For the property-based similarity between two topics t_1 and t_2, two similarity values, $SIM_{\text{occtypr}}(t_1.occurrence_i, t_2.occurrence_j)$ and $SIM_{\text{occvalue}}(t_1.occurrence_i, t_2.occurrence_j)$ of occurrence types and occurrence values are computed for each occurrence. According to Topic Maps Data Model (Garshol & Moore, 2005), an occurrence type is defined as a topic. For example, '*figure*', '*description*', and '*biography*' are defined as topics and used as occurrence types of *philosopher* topics. Thus, if an occurrence type of $occurrence_i$ of t_1 is another topic t_p and an occurrence type of $occurrence_j$ of topic t_2 is another topic t_q, the similarity value of two occurrence types is determined by the following expression in which $SIM(t_p, t_q)$ means the combined similarity between t_p and t_q.

$$SIM_{\text{occtype}}(t_1.occurrence_i t_2.occurrence_j) = SIM(t_p, t_g)$$

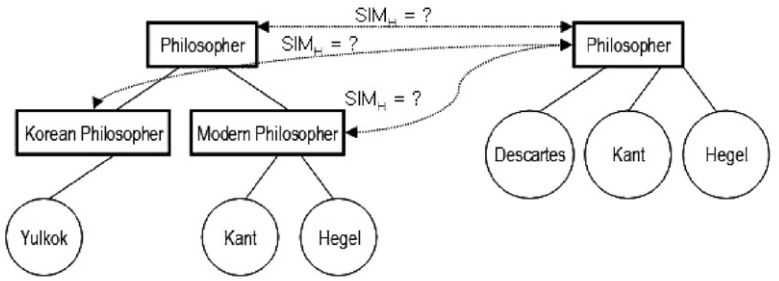
The similarity value between occurrence values is computed by string comparing methods because occurrence values are textual descriptions or URI addresses. If occurrence values are textual descriptions, the similarity values range from 0 to 1. If occurrence values are URI addresses, the similarity values are 0 or 1.

4.4. Hierarchy-based strategy

Hierarchy matching measures the similarity between two class topics based on the combined similarity between their child topics as well as their parent topics. For instance, we know that *philosopher* of topic map B is closer to *modern philosopher* rather than *philosopher* of topic map A from an example of hierarchy depicted in Fig. 5. Name matching operation determines higher similarity value between *philosopher* topics than the one between *philosopher* and *modern philosopher*. Whereas, hierarchy matching operation determines that *philosopher* of topic map B matches well with *modern philosopher* of topic map A because both topics have similar child topics.

The following expression computes the similarity value between two topics based on the similarity of their hierarchical structure. In this expression, t_1 and t_2 are topics that have m and n parent topics and x and y child topics respectively. And $t_1.parent_i$ is ith parent topic of t_1 and $t_2.parent_j$ is jth parent topic of t_2. We average SIM_{name} and SIM_{occ} of $t_1.parent_i$ and $t_2.parent_j$ to determine a combined similarity value between parent topics of t_1 and t_2. Likewise, $t_1.child_k$ is kth child topic of t_1 and $t_2.child_1$ is lth child topic of t_2. We average SIM_{name}, SIM_{occ}, and of $t_1.child_k$ and $t_2.child_i$ to produce the combined similarity value SIM between them. In the expression, w is a weight ranging from 0 to 1. We set a different value to w in order to emphasize the similarity of parent topics or child topics.

$$SIM_H(t_1, t_2) = (1 - w)\left(\sum (SIM_{\text{name+occ}}(T_1.parent_i, t_2.parent_j))/|m| \times |n|\right) + w\left(\sum (SIM(t_1.child_k t_2.child_j)/|x| \times |y|)\right)$$



(a) Hierarchical structure of philosopher in the Topic Map A (b) Hierarchical structure of philosopher in the Topic Map B

Fig. 5. An example of Topic Maps need hierarchy based strategy.

4.5. Association-based strategy

Association matching operation determines the similarity between association types. For example, Topic Map A has an "*author of*" association between *Kant* with *author* role and *Critique of Practical Reason* with *book* role. Topic Map B has a "*written by*" association between *Critique of Practical Reason* with *philosophical text* role and *Immanuel Kant* with *writer* role. Association matching operation measures the similarity between the "*author of*" of Topic Map A and the "*written by*" of Topic Map B.

An association type is composed of a set of members, which have their roles in the relation. Thus, the similarity between association types is determined by similarities between members of them. Following expression measures the similarity between association types. Given two association types, t_1 and t_2, for a set of pairs of members, the similarity value between paired members is computed.

$$SIM_{\text{assoc}}(t_1, t_2) = \sum SIM(m_i, m_j) \cdot SIM(r_i, r_j)/|m| \times |n|, \quad \text{for } 1 \leqslant i \leqslant M, \ 1 \leqslant j \leqslant N$$

M and N is the number of members of two association types each other. m_i is the ith member of t_1 and m_j is the jth member of t_2.r_i is role of m_i and r_j is role of m_j.

4.6. Match candidates selection

Similarity values computed by four matching operations are collected to generate a combined similarity value for each pair of topics. There are several approaches to generating a combined value, such as Max, Weighted, Average, and Min (Do & Rahm, 2002). The Max approach determines a combined value to the maximal value of similarity values. The Weighted approach computes a weighted sum of similarity values of four matching operations for a combined value. Weight, as used herein, means the importance of the matching operations. The Average approach computes an average similarity of all matching operations. Average means that weights of four matching operations are equal. The Min approach chooses the lowest similarity value of any matching operation. In this paper, we use the average approach to aggregate similarity values because it is difficult to determine adequate weights for combining independent single operations. For example, the combined similarity value of a pair, (a_1, b_1), is computed by the following expression.

$$SIM(a_1, b_1) = (SIM_{\text{name}} + SIM_{\text{occ}} + SIM_H + SIM_{\text{assoc}})/4$$

To determine the match candidate from input Topic Map T_2 for a topic t_1 of the other Topic Map T_1 we rank all pairs including t_1 in descending order of their similarity values and choose the match candidates. The methods for selecting the match candidates from ranked pairs are MaxN, MaxDelta, and Threshold (Do & Rahm, 2002). The MaxN method selects N pairs from top rank of ordered list of pairs. The MaxDelta method selects all pairs located in a range from the highest similarity value to a particular offset value d. The threshold method selects all pairs with similarity values exceeding a given threshold value t. The threshold method with $t = 0.8$ and *displacement* $(d) = 0.07$ is used to select match candidates from a list of pairs.

5. Merging between topic maps

Merging between topic maps describes the process of integrating two topic maps into a new topic map. For creating an integrated topic map, we remove duplicates and union entities from two source topic maps. We define a merge operation for topic maps as the following expression.

Definition 3. Given a set of topic maps S a merge operation is defined as the following expression:

$$\textbf{\textit{merge}} : (S \times S) \rightarrow S$$

If two topic maps TM_{A} and TM_{B} are elements of S, merging between them is defined as the following expression:

$$
\begin{aligned}
MERGE(TM_A, TM_B, TM_{AB}) \rightarrow TM_C \Longleftrightarrow\ & \{\forall_{c_1} \in T_c/T_c \subseteq TM_A\} \cup \{\forall_{c_2} \in T_c/T_c \subseteq TM_B\} \wedge \\
& \{\forall_{o_1} \in T_o/T_o \subseteq TM_A\} \cup \{\forall_{o_2} \in T_o/T_o \subseteq TM_B\} \wedge \\
& \{\forall_{a_1} \in T_a/T_a \subseteq TM_A\} \cup \{\forall_{a_2} \in T_a/T_a \subseteq TM_B\} \wedge \\
& \{\forall_{r_1} \in T_r/T_r \subseteq TM_A\} \cup \{\forall_{r_2} \in T_r/T_r \subseteq TM_B\} \wedge \\
& \{\forall_{i_1} \in T_i/T_i \subseteq TM_A\} \cup \{\forall_{i_2} \in T_i/T_i \subseteq TM_B\} \wedge \\
& \{\forall_{h_1} \in R_h/R_h \subseteq TM_A\} \cup \{\forall_{h_2} \in R_h/R_h \subseteq TM_B\} \wedge \\
& \{\forall_{a_1} \in R_a/R_a \subseteq TM_A\} \cup \{\forall_{a_2} \in T_a/T_a \subseteq TM_B\}
\end{aligned}
$$

The merge function, MERGE, takes three input values such as two source topic maps and a mapping matrix M_{AB}, in which correspondences between two topic maps are stored. Using this mapping information our merge function integrates corresponding entities from two topic maps into a new merged entity. Then it adds remain entities of two topic maps, which do not exist in the mapping matrix, to the merged topic map. Our merge function satisfies the following merge requirements to improve the quality of merge result:

Entity preservation. If an entity a, which has a corresponding entity b, is a entity of $TM_A \cup TM_B \cup M_{AB}$, a new entity c merging a and b must be created in the topic map TM_C. If an entity a' does not have corresponding entities, i.e. $a' \in TM_A$ and $a' \in M_{AB}$ or $a' \in TM_B$ and $a' \in M_{AB}$, the entity must be copied in topic map TM_C.

Property preservation. If an entity a has a property p and its corresponding entity b has a property q, a merged entity c must have a property r, which is union of p and q only if p is corresponding to q. When p is not similar to q a merged entity c must have all of two properties p and q.

Relationship preservation. Relationships belong to an entity a and its corresponding entity b must be preserved as relationships of a new entity c in the merged topic map TM_c. In a topic map TM_A, a relationship, $\mathbf{R}_a(a_1, a_2)$, exists between two entities, a_1 and a_2. In a topic map TM_B, a relationship, $\mathbf{R}_b(b_1, b_2)$, exists between two entities b_1 and b_2. A merged entity c_1 of a_1 and b_1 has two relationships $\mathbf{R}_a(c_1, a_2)$ and $\mathbf{R}_b(c_1, b_2)$.

Conflict resolution. Conflicts are caused by different conceptualization of mapping entities in the process of merging. These conflicts are detected and resolved to improve the quality of merged results adequately according to their types.

We define a taxonomy of merging conflicts which categorize conflicts with element-level, structure-level, and temporary-level. The taxonomy of merging conflicts is depicted in Fig. 6.

Element-level conflicts are caused by different definition of a topic, such as different topic name, different number of properties, and different specification of properties of a topic. Element-level conflicts are classified into naming conflicts and property conflicts. Naming conflicts are caused when two mapping topics have

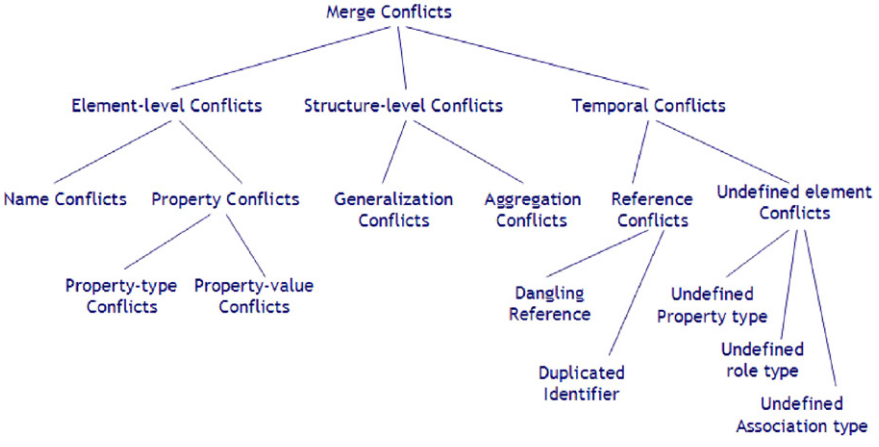


Fig. 6. A taxonomy of merge conflicts.

different or similar topic names. Property conflicts are caused when two mapping topics have same values for different property types or different values for same property types.

Structure-level conflicts are generated by different level of conceptualization of two source topic maps. For example, TM_1 has simple conceptualization in which *Philosopher* topic is specialized with *Kant, Hegel, Mencius*, and so on without geographic and periodic classification. Whereas TM_2 has complex conceptualization in which *Philosopher* topic is specialized with *Ancient Philosopher, Medieval Philosopher, Modern Philosopher*, and so on by a periodic viewpoint. These TM_1 and TM_2 have structure-level conflicts.

Temporary-level conflicts arise when the merged topic map has inconsistency during merging of mapping topics. The conflicts are classified into reference conflicts and undefined-entity conflicts. Reference conflicts, caused when a topic has invalid reference, are specialized with dangling reference conflicts and duplicated ID conflicts. A dangling reference means that a topic's reference is invalid because the referenced topic is removed. Duplicated ID conflicts arise when the merged Topic map has topics whose IDs are identical. Undefined identity conflicts are produced when a merged topic has one or more undefined properties, undefined roles, or undefined associations.

6. Experiments

6.1. Experiment setup

We set up three kinds of data groups, which are group A, group B, and group C, for our experiment. Group A includes Topic Maps which were constructed from philosophy knowledge domain and by same group of domain experts. Oriental philosophy ontology, modern western philosophy ontology, and contemporary western philosophy ontology are grouped in group A, because these ontologies are philosophy domain's ontologies and created by the same philosophy experts.

Group B includes Wikipedia philosophy Topic Maps constructed from philosophy-related contents of Wikipedia. Group C includes Topic Maps constructed from literature knowledge domain. We translate some of German literature encyclopedia provided by Yahoo Korea portal into Topic Maps. Table 2 shows the characteristics of our experimental data.

6.2. Measurement and experiment results

In this work, we use performance measurement of information retrieval such as precision, recall, and overall, to measure performance of our ontology matching operations. To evaluate the quality of our matching operations, we need to know the *manually determined match set (M)* and the *automatically generated match set (A)* which can be obtained by matching the processes. By comparing these match results, we get

Table 2
The statistics of experimental Topic Maps

Ontologies	Group A			Group B	Group C
	T_1	T_2	T_3	T_4	T_6
Max level	11	10	9	9	4
# of Topics	1826	983	1266	417	30
# of Topic types	1379	384	603	182	3
# of Occ. types	86	56	62	13	2
# of Ass. types	47	40	43	7	2
# of Role types	22	15	18	4	2
# of PSIs	653	328	345	0	3

T_1 – Oriental Philosophy.
T_2 – Modern Western Philosophy.
T_3 – Contemporary Western Philosophy.
T_4 – Wikipedia Philosophy.
T_5 – Yahoo German Literature.

true-positive set (I) which includes correctly identified matches, *false–positive set (P-I)* includes false matches, and *false–negative set (R- I)* which includes missed matches. We can measure match quality of automatic matching processing by evaluating following expression. Table 3 and Fig. 7 shows the experimental result that represents high recall and precision.

$$\text{precision} = \frac{|I|}{|P|} \quad \text{recall} = \frac{|I|}{|R|} \quad \text{overall} = \text{recall}^* \left(2 - \frac{1}{\text{precision}}\right)$$

Pairs of ontologies in group A are matched based on the ontology schema layer because these ontologies are constructed from the same knowledge domain and a group of experts. These ontologies share a common schema, known as the philosophy reference ontology, for standardizing and validating them. In the philosophy reference ontology, topic types, occurrence types, association type, and subject identifiers are defined and referenced by component ontologies of the philosophy ontology, i.e. oriental ontology, modem western ontology, contemporary western ontology, and so on. Thus, most matches between ontologies in group A include topic types, occurrence types, and association types referenced by ontologies. The pair (T_2, T_3) of group A has maximal matches because both ontologies are components of the philosophy ontology and have some relationships in terms of philosophers, texts, terms, doctrines, and so on.

In $(T_1, T_4), (T_2, T_4)$, and (T_3, T_4) of group A and B, most of all matched topics result from topic name-based matching operation because paired Topic Maps have topics describing same philosophers, i.e. *Kant, Hume,* and *Marx,* same texts of philosophy, i.e. *Philosophy of Right, Critique of Pure Reason,* and *Discourse on the Method*, and same terms of philosophy, i.e. *reason, free will,ideology*, and *moral*. Thus, these pairs of Topic Maps have better accuracy than pairs of Topic Maps of group A, because Topic Maps of group A have many topics whose names are phrases or sentences. But a difference of measures of two groups is an insignificant value because our morphological analysis-based string match evaluates the similarity of phrases or sentences correctly.

Table 3
Match results of pairs of Topic Maps ($t = 0.8$ and $d = 0.07$)

Pairs of ontologies	(T_1, T_2)	(T_1, T_3)	(T_2, T_3)	(T_1, T_4)	(T_2, T_4)	(T_3, T_4)	(T_2, T_6)
R	207	217	275	92	76	81	3
P	222	224	284	96	78	85	7
I	193	199	261	89	74	78	3
Precision	0.87	0.89	0.92	0.93	0.95	0.92	0.42
Recall	0.93	0.92	0.95	0.97	0.97	0.96	1
Overall	0.79	0.80	0.87	0.90	0.92	0.88	−0.38

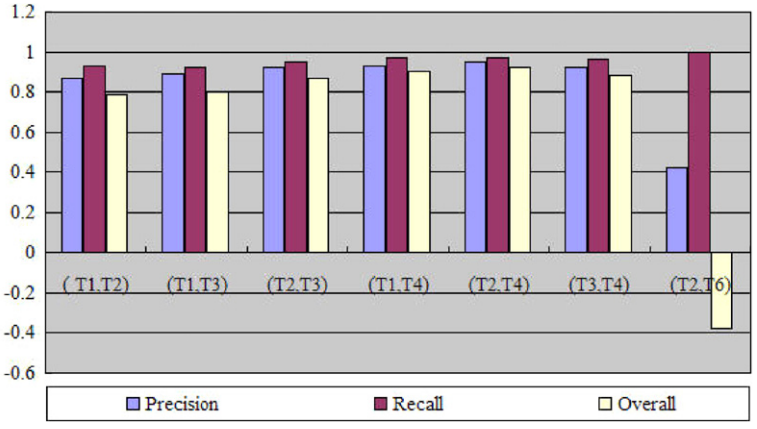


Fig. 7. Experiment results of pairs of Topic Maps.

The recall of a pair of modern western philosophy and German literature, (T_2, T_6), is 1 because the number of matches between different domain's ontologies are very low and matching operations easily find matches based on topic names, such as *Nietzsche, Philosophy of Right*, and so on. This pair has poor overall, −0.38, but in contrast to recall, it is not that low accuracy. This means that domain experts must make more efforts to adopt automatically generated matches than to determine matches in manual. In other words, it seems useless to match ontologies between different knowledge domains.

6.3. Performance evaluation

6.3.1. Comparison of name matching operations

Fig. 8 shows performance evaluation of the string match methods introduced in Section 4. We average precisions, recalls, and overalls of pairs of Topic Maps to represent simplified comparison of string match operations. The simple token-based string match method has the lowest recall because it can not find matches between phrases or sentences. But it has higher precision than *Token + Substring* method because *Token + Substring* finds more true-negative matches in comparison between phrases or sentences. *Morphological analysis* method has the highest precision and recall because it has all benefits of *Token* and *Token + Substring* methods.

6.3.2. Comparison of single matching strategies

Fig. 9 shows the result of performance evaluation of the single matching operations. We average precisions, recalls, and overalls of pairs of Topic Maps to represent simplified comparison of matching strategies. We evaluate the quality of four kinds of combinations of matching operations, (1) *Name*, (2) *Name + Property*, (3) *Name + Hierarchy*, and (4) *Composite*. *Name* is the name matching operation and *Name + Property* is a combination of the name matching operation and internal structure matching operation. *Name + Hierarchy* is a combination of name matching operation and external structure matching operation. *Composite* is a combination of all matching operations.

Values of Fig. 9 show average precision, recall, and overall of pairs of Topic Maps for each match operations. According to the chart depicted in Fig. 8, we know that four combinations have similar recall values but higher precision and overall values.

6.3.3. Comparison of topic maps matching performance

We compare our matching method, which is named TM-MAP, with other Topic Maps matching methods, which are Subject Identity Measure and Topic Maps Reference Model(ISO/IEC JTC1/SC34 2003), to

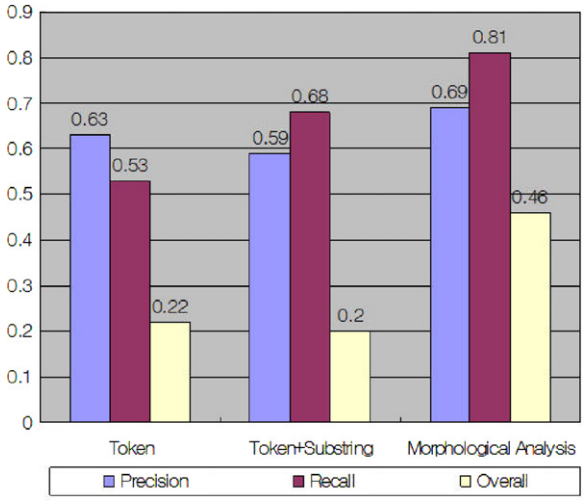


Fig. 8. Performance evaluation of string matching methods.

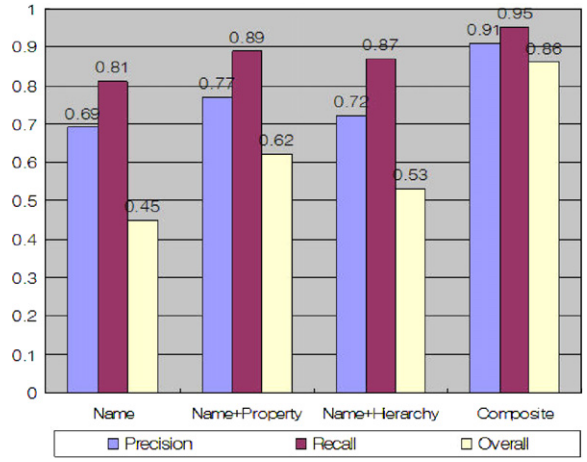


Fig. 9. Performance evaluation of single matching and composite matching.

represent the performance of proposed matching strategies. Fig. 10 shows precisions of matching methods for each pairs of Topic Maps. Figs. 11 and 12 show recalls and overalls of matching methods each other.

SIM measures the similarity between topics based on their names similarity and occurrence similarity. It does not consider external structures of Topic Maps, such as hierarchy and association. TMRM maps only two topics which have the identical names regardless of their occurrences and hierarchies. In Fig. 9, we know

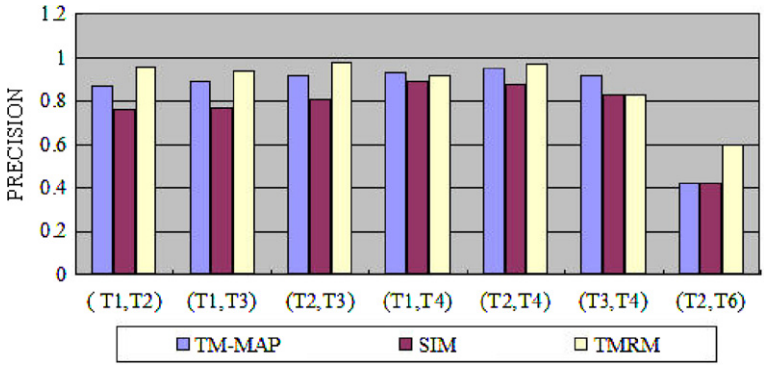


Fig. 10. Precisions of matching methods for each pairs of Topic Maps.

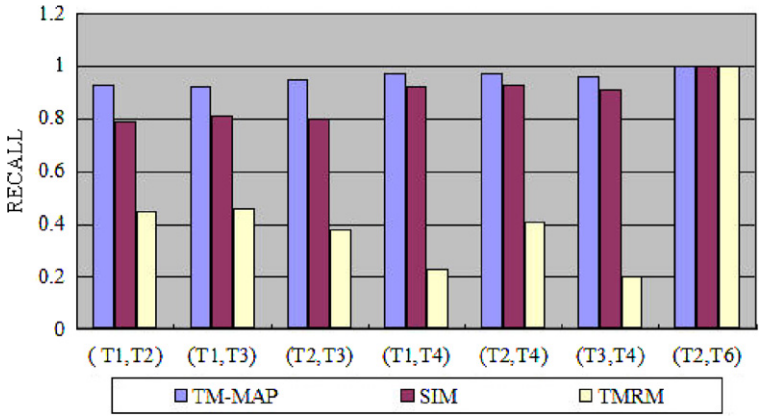


Fig. 11. Recalls of matching methods for each pairs of Topic Maps.

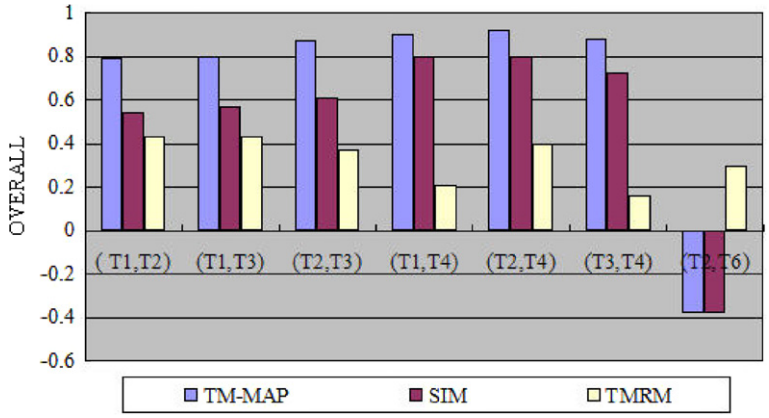


Fig. 12. Overalls of matching methods for each pairs of Topic Maps.

that TMRM has higher precisions than SIM and TM-MAP because false matches of TMRM are lower than other methods.

Unlike precisions in Fig. 10, recalls of TMRM are lower than other methods because it matches only topics having identical names. Our matching method TM-MAP has higher recalls than SIM and TMRM because TM-MAP generates more true matches than other methods through the processing of composite matching operations. SM cannot map between topics having phrases or sentences as their names because it performs a token-based string matching operation only. Thus, SIM has low recalls when it measures the similarity of Philosophy Topic Maps. Because overall is determined by recall and precision, TM-MAP has higher overalls than other methods.

7. Conclusions

In this paper, we propose a multi-strategic matching approach to determine semantic correspondences between Topic Maps. Our multi-strategic matching approach takes advantage of the combination of linguistic module and Topic Maps constraints including name matching, internal structure matching, external structure matching, and association matching. By doing this, the system achieves higher match accuracy than the one of a single match technique.

Our approach simplifies the matching computation but still preserves the quality of ontology matching, because it takes into account the characteristics of Topic Maps which define formal syntax and constraints for representing ontologies. Furthermore, unlike the existing approaches, our approach does not require conversion of ontologies into a graph model and full scanning of entities in two ontologies.

The experiment results shows that precision of automatically generated match set is more than 87%, but the recall of the set is more than 90%. This means that automatically generated match sets include a large portion of all manually determined matches. Matched topics are merged into a new topic or connected by a semantic relationship to enable ontology-based systems to provide knowledge-related services on multiple Topic Maps. However, merging or alignment of Topic Maps is not easy work although we found matches between Topic Maps. The MERGE function for merging Topic Maps has been developed, which takes advantage of mapping information and integrates corresponding entities form two Topic Maps into a new merged entity. We are aware that there are likely to be some hidden complications regarding this approach, but we also believe that well-established matching and merging operations will make Topic Maps easier to use in large- scale applications.

References

Berman, B. L. (2001). *Library of congress subject headings in philosophy: A thesaurus*. ISBN: 0-912632-64-X. Philosophy Document Center.

Beneventano, D., Bergamaschi, S., Guerra, F., & Vincini, M. (2001). The MO MIS approach to information integration. In *IEEE and AAAI International Conference on Enterprise Information Systems* (pp. 194–198), SetubaL Portugal.

Biezunski, M., Bryan, M., & Newcomb, S. (2002). ISO/IEC 13250 TopicMaps. http/www1.y12.doe.gov/capabihties/sgm1/sc34/document/0322.htm.

Bouquet, P., Serafini, L., & Zanobini, S. (2003). Semantic coordination: a new approach and an application. In *Proceedings of ISWC* (pp. 130–145), Sanibel Island, Florida, USA.

Do, H. H., & Rahm, E. (2002). COMA – a system for flexible combination of schema matching approaches, In *Proceedings of VLDB* (pp. 610–621).

Doan, A., Domingos, P., & Halevy, A. (2001). Reconciling schemas of disparate data sources: a machine-learning approach. In *Proceedings ACM SIGMOD Conference*.

Durusau, P., Newcomb, S., & Barta, R. (2006). Topic Maps – Reference Model ISO/IEC JTC1/SC34, Version 6.0, http://www.isotopicmaps.org/tmrm.

Ehrig, M., & Staab, S. (2004). QOM: quick ontology matching. In *Proceedings of ISWC* (pp. 683–697). Hiroshima, Japan.

Garshol, L. M., & Moore, G. (2005). *Topic maps data model.* <http://www.isotopicmaps.org/sam/sam-model>.

Giunchglia, F., & Shvaiko, P. (2003). Semantic matching. *In the Knowledge Engineering Review Journal, 18*(3), 265–280.

Kalfoglou, Y., & Schorlemmer, M. (2002). Information-flow-based ontology matching. In *Proceedings of the 1st international conference on ontologies, database and application of semantics* (pp. 1132–1151). Irvine, California, USA.

Kim, J. M., Choi, B. I., & Kim, H. J. (2004). Building a philosophy ontology based on contents of philosophical texts. *Korea Information Science Society, 11*(4), 275–283.

Lassila, O., & Swick, R. R. (1999). Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation. URL: http://www.w3.org/TR/REV-rdf-syntax.

Li, W., Clifton, C., & Liu, S. (2000). Database integration using neural network: implementation and experiens. *Knowledge Information Systems, 2*(1), 73–96.

Madhavan, J., Bernstein, P., & Rahm, E. (2001). Generic Schema Matching with Cupid. In *Proceedings of VLDB* (pp. 49–58).

Maicher, L., & Witschel, H. F. (2004). Merging of distributed topic maps based on the subject identity measure (SIM) approach. In *Proceedings of Berliner XML tags* (pp. 301–307).

McGuinness, D.L., & Harmeba (2003). OWL Web Ontology Language Overview. W3C Recommendation, http://www.w3.org/TR/ow1-featares/.

Melnik, S., Garcia-Molina, H., & Rahm, E. (2002). Similarity flooding: a versatile graph matching algorithm. In *Proceedings of ICDE* (pp. 117–128). San Jose, California, USA.

Noy, N., & Musen, M. (2000). PROMPT: algorithm and tool for automated ontology merging and alignment. In *Proceedings of the national conference on artificial intelligence (AAAI)* (pp. 450–455). Austin, TX, USA.

Noy, N., & Musen, M. (2001). Anchor-PROMPT: using non-local context for semantic matching. In *Proceedings of the workshop on ontologies and information sharing at the international joint conference on artificial intelligence (IJCAI)* (pp. 63–70). Seattle, WA, USA.

Pepper, S., & Moore, G. (2001). XML Topic Maps (XTM) 1.0, Topic Maps.Org. http://www.topicmaps.org/xtm/.

Rahm, E., & Bernstein, P. (2001). On matching schemas automatically. *VLDB Journal, 10*(4), 334–350.

Shvaiko, P., & Euzenat, J. (2004). A survey of schema-based matching approaches. University of Trento, Technical Report #DIT-04-087 (pp. 1–22).

Stumme, G., & Madche, A. (2001). FCA-merge: bottom-up merging of ontologies. In *Proceedings of 17th international joint conference on artificial intelligence (IJCAI)* (pp. 225–234). Seattle, Washington, USA.