Extending a Lexicon Ontology for Intelligent Information Integration

Roberta Benassi and **Sonia Bergamaschi** and **Alain Fergnani** and **Daniele Miselli**[1]

Abstract. One of the current research on the Semantic Web area is semantic annotation of information sources. On-line lexical ontologies can be exploited as a-priori common knowledge to provide easily understandable, machine-readable metadata. Nevertheless, the absence of terms related to specific domains causes a loss of semantics. In this paper we present *WNEditor*, a tool that aims at guiding the annotation designer during the creation of a domain lexicon ontology, extending the pre-existing WordNet ontology. New terms, meanings and relations between terms are virtually added and managed by preserving the WordNet's internal organization.

1 INTRODUCTION

The *Semantic Web* [12] will provide intelligent access to heterogeneous and distributed information sources, enabling software agents to mediate between users' needs and the ever-expanding number of on-line data. The heterogeneity of sources involves terminology (different names indicating the same real-word concept), structure (different models/primitives to represent the same concept) and domain [2, 14]. In addition, the Internet-based environment introduces complicating factors such as organizational and functional aspects of the information use [17]. Thus, developing intelligent tools for automatic information integration and data reconciliation is a challenging but crucial step to carry out the Berners-Lee's vision of the future Web.

Many *Information Integration Systems* usually manage data sources coming from a particular domain of interest (e.g. medical, biological, geographical...), so the semantic heterogeneity reconciliation is performed on the basis of a lexicon-specialized reference ontology. On the contrary, systems able to integrate data from different domains, adopt an *upper level ontology*, which represents a set of general and well-known concepts, shared by the most part of the human knowledge.

Current systems usually need human supervisors (*integration designers*) to handle the complexity of the whole *integration process* and therefore achieve meaningful results. In particular, the mapping among the lexicon of each data source and the reference ontology (the so-called *annotation phase*) is one of the most critical step because it deeply compromises the subsequent phases. For example, if the reference ontology does not contain a satisfactory meaning for a concept expressed in a given data source, the designer may annotate it with a similar one, generating a *partial* loss of knowledge. On the other hand, the designer might not annotate the concept at all, with a *total* loss of semantics.

In this work, we propose *WNEditor*, a tool that aids the integration designer during the creation of (additional) specific-domain lexica: the reference ontology is physically extended by introducing new terms, new meanings and relations between them and/or the existing ones.

The paper is organized as follows: Section 2 investigates the interactions between an information integration system and the reference ontology. In order to clarify our approach we adopt MOMIS [4, 3] as the representative integration system and WordNet[19] as the common lexicon ontology. Section 3 describes how to confer the designer the ability to physically extend WordNet, while Section 4 investigates several issues about reconciliation between independently-developed extensions.

[1] Dipartimento di Ingegneria dell'Informazione, Universita' di Modena e Reggio Emilia, Italy. Sonia Bergamaschi also at IEIIT-CNR Bologna, Italy.

2 THE MOMIS/WORDNET INTERACTION

To provide information sources with easily understandable metadata, current Semantic Web approaches rely on the a-priori existence of ontologies. Nevertheless, most ontologies in common use are too generic and they do not adequately describe the domain of interest. On the contrary, the MOMIS's approach aims at synthesizing a more accurate ontology just involving the sources themselves. In fact, MOMIS (acronym of **M**ediator Envir**O**nment for **M**ultiple **I**nformation **S**ources [4, 3]) performs information extraction and integration from structured and semistructured data sources, both in static and dynamic environments[4, 3]. By means of MOMIS, a user can query the integrated sources, without knowing exactly where they are located and which heterogeneity data degree they present. The information integration process (Fig. 1) is performed in 5 steps and produces a conceptualization of the underlying domain, i.e. a reconciled, integrated *Global Virtual View* (GVV) of (local) data sources.

1. Heterogeneous data sources are presented to MOMIS in a standard way: local source schemata are extracted and translated into ODL_{I3}, a modified version of the *Object Definition Language*[2].
2. Each item of a local source's schema (class or attribute name) is manually annotated, i.e. associated to one or more meanings according to the WordNet lexicon ontology.
3. MOMIS generates a *Common Thesaurus* of the involved local sources, by incrementally building a set of intensional and extensional relationships, such as *schema-derived relationships* (automatic extraction of intra schema relationships from each schema separately), *lexicon-derived relationships* (inter-schema lexical relationships derived by the annotated sources), *designer-supplied relationships* (specific domain knowledge capture) and *inferred relationships* (via equivalence and subsumption computation).

[2] www.service-architecture.com/database/articles/odmg_3_0.html

4. Starting from the Common Thesaurus and the local schemata descriptions, MOMIS generates a global reconciled schema (GVV) plus sets of mappings on local sources[3].
5. The GVV is annotated by semi-automatically assigning a meaning to each element of the global schema[3].

During the annotation of local schemata and global schema, the integration designer is asked to explicitly declare lexical relations between the items' names of each schema and proper WordNet meanings[3]. More precisely, the WordNet-designer interaction is firstly concerned about the choice of the *word form*: the WordNet morphological processor stems each name and eliminates suffixes due to declination/conjugation. Then, given a word form, the designer has to manually solve possible ambiguities by mapping the given name on zero, one or more WordNet senses. Thus, when WordNet does not contain satisfactory word forms and/or meanings (or does not contain word forms and/or meanings at all), the item name is considered unknown and the semantic richness of data sources is lost because no inter-schema lexicon relationships can be derived. In order to clarify how WNEditor can be exploited to preserve the semantics of sources, we will refer to the running example in Tab. 1, where *Source#1* is the Document Type Definition of a XML file about professors and students of a University, while *Source#2* represents a relational database storing data about students and their activities during apprenticeship. In the annotation phase, the designer finds the most adequate word forms/meanings for the items of *Source#1*, while two different problems arise from *Source#2*. The first is that the designer does not find satisfactory the meaning associated to `tutor#1: a person who gives private instructions (as in singing or acting)`; the second is about the term `tirocinium`, which is completely absent from WordNet.

Table 1. Sources to be integrated. Underlined attributes in Source #2 are primary keys. AK and FK indicate alternative keys and foreign keys.

Source#1

```
<!ELEMENT University(People*)>
<!ELEMENT People(Professor*|Undergraduate*)>
<!ELEMENT Professor(name, e-mail, rank)>
<!ELEMENT Undergraduate (name, year)>
```

Source#2

```
Student(code,name,phone)
Tutor(id,name,rank)
Enterprise(name,address)
Tirocinium(code, name, year, id)
 AK: year, code
 FK: code references Student, id references Tutor,
 name references Enterprise
```

3 EXTENDING A REFERENCE ONTOLOGY WITH *WNEDITOR*

WordNet represents a huge lexicon ontology written by lexicographers and inspired to current psycholinguistic theories of human lexical memory. *Extending WordNet* has the aim to maintain the semantics of the local schemata as much as possible and make the GVV fully exploitable by external users and other applications.

[3] WordNet recognizes that there is a conventional association between word forms and the concept or meaning they express. Currently, WordNet 2.0 is available from http://www. cogsci.princeton.edu/˜wn.

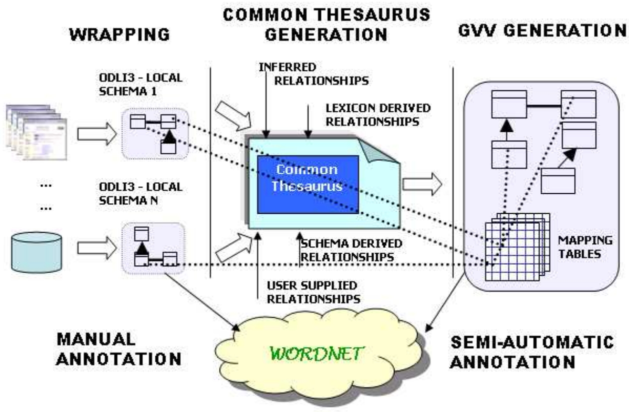

Figure 1. Global Virtual View Generation Process in MOMIS [3]

More precisely, we refer to the possibility to introduce new terms and senses[4]. Unfortunately, WordNet is distributed *as-it-is* and external applications are not allowed to directly modify its data files. Then, the first problem to be faced is how to physically extend the lexicon ontology and keep sound relations among new concepts and the already existing ones. After extrapolating the WordNet internal organization, a relational DBMS (DataBase Management System) can be used to store data (Tab. 2).

The distinction between original and extended data is achieved by introducing the table WN_EXTENDER, which contains information about the owner of specific modifications on the reference ontology.

The second problem is concerned to the criticalness of the extension process due to the complexity of the lexical ontology. Thus, the designer should have the possibility to perform step-by-step operations, such as providing definitions for new concepts (glosses) and *building relations* between added concepts and the pre-existing ones.

Every WordNet relation holds between two members, a *source synset* and a *target synset*. Given a new concept X as the (fixed) source synset, the designer should be helped in searching for the most appropriate target, i.e. in retrieving a list of candidate synsets $\{S_1, S_2 \ldots, S_n\}$ that share somewhat similarities with X.

Under the assumption that *similar enough natural language definitions should also provide some evidence of concept similarity*, we can obtain the target candidates by exploiting an heuristic known in literature as **definition match**[1] and applying it to the senses' definitions (WordNet glosses). We firstly implemented the definition match technique suggested in [13], adapting it to our specific case:

Definition 1 *Let D_1 and D_2 be the English definitions of two concepts, where D_1 is the current sense gloss and D_2 is any other definition of meaning in the reference ontology. Both definitions are separated into individual words which are compared to an English stop words list and then stemmed. Remaining words are used to compute **reliability** (number of shared words) and **strength** (ratio of reliability to number of words in the shorter definition).*

However, our tests showed that *strength similarity function* was

[4] The example in Tab. 1 is quite simple, since it presents only two terms/meanings missed in WordNet, but when the designer has to handle complex integration tasks within highly-specialized domains, there could the need to define a new vocabulary from scratch. In this case, the designer can exploit WNEditor for a faster development of specialized lexica.

Table 2. Extrapolation of the WordNet internal organization and its representation in a relational DBMS.

```
WN_EXTENDER (wn_extender_id, name, description)
 AK: name

WN_SYNSET (wn_synset_id, offset,syntactic_category,
 word_cnt, gloss, wn_extender_id)
 FK: wn_extender_id references wn_extender

WN_LEMMA (wn_lemma_id, lemma, syntactic_category,
 sense_cnt, wn_extender_id)
 AK: (lemma, syntactic_category)
 FK: wn_extender_id references wn_extender

WN_LEMMA_SYNSET (wn_lemma_synset_id, wn_synset_id,
wn_lemma_id, lemma_number, sense_number, wn_extender_id)
 AK: (wn_lemma_id, sense_number ),
 (wn_synset_id, lemma_number)
  FK: wn_extender_id references wn_extender
 wn_synset_id references wn_synset
 wn_lemma_id references wn_lemma

WN_RELATIONSHIP (wn_relationship_id, wn_source_synset_id,
wn_target_synset_id, wn_source_lemma_number,
wn_target_lemma_number, wn_relationship_type_id,
wn_extender_id)
 FK: wn_extender_id references wn_extender
 FK: wn_source_lemma_number references wn_lemma
 FK: wn_target_lemma_number references wn_lemma
 FK: wn_source_synset_id references wn_synset
 FK: wn_target_synset_id references wn_synset
 FK: wn_relationship_type_id references
 wn_relationship_type

WN_RELATIONSHIP_TYPE (wn_relationship_type_id, symbol,
description, reflex)
 AK: symbol

WN_REVERSE_INDEX (wn_reverse_index_id,
term, wn_synset_id_list)
 AK: term
```

not as sensible as we expected[11]. We modified the strength function in order to achieve a more quickly growth with the increasing number of matching words (and a more fine granularity in assigning similarity scores, too). Several other similarity measures were tested, and, at last, we chose the function defined as:

$$Sim(x, y) = 1 - \exp\left(-\frac{y^2}{x}\right) \tag{1}$$

with the following constraints:

- both y (number of shared words) and x (number of words in the shorter definition) assume only discrete values.
- y can never be greater than x: the maximum match case happens when the whole shorter definition is included in the longer one;
- x can never assume the value 0. The demorphing algorithm we implemented replaces all the original terms of a gloss in the case that all of them are stop words. This choice is due to the impossibility, on the contrary case, to find out in a similarity search meanings with an only *common words definition*. y can assume the value 0 when no match between the two definitions is found.

The WNEditor's philosophy is based on the awareness that the designer knows the organization in synsets of the WordNet lexicon ontology and, as shown in Fig. 2, (s)he interacts with it by *creating a new synset*, by *searching for a synset* or by *by writing a synonym in the synonyms's meaning(s) panel*. For instance, the designer creates a new synset for the noun tirocinium (Fig. 3) by introducing the gloss `the period when a student gets practice and learns about a field or activity`. Then, the designer, through the *Synset Relationship Editor*, is asked to pick up some words that should appear in the target gloss. In our case, we want to relate tirocinium with already existing meanings about `learning from practice`. Another possible solution is to relate it to the existing sense of `apprenticeship#1: the position of apprentice`. As

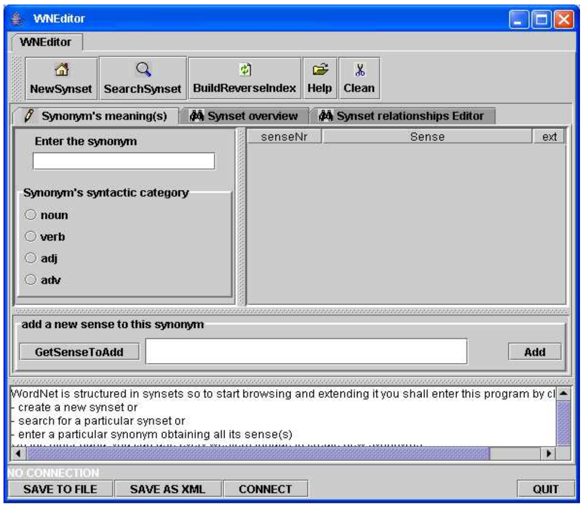

Figure 2. WNEditor masks.

far as `tutor` is concerned, the existing meaning is not completely satisfactory, then the more specific sense "`a professor that guides a student during his apprendiceship/tirocinium`" (Fig. 4) is added.

In both cases, according to the precision and selectivity of the inserted keywords, the result will be more or less satisfactory and of different size. We implemented similarity search techniques that can help the designer during the connection of new terms/meanings with the already existing ones. More precisely, we developed an *approximate string match* algorithm to perform the similarity search on the whole ontology's network. In this way, we are able to suggest possible meanings/lemmas to the designer who is not expected to exactly know the words used in glosses' definitions. The Levenshtein distance (or edit distance), as the number of deletions, insertions or substitutions required to transform a string S (source) into a string T (target), is the implemented similarity measure [16]. The length of the source string is assumed as the similarity threshold, then only most similar targets are displayed. The similarity search task is performed by introducing a well-known indexing technique in IR, a *reverse index* that corresponds to the table WN_REVERSE_INDEX in the E/R schema (Tab. 2). Each entry in this table is composed by a term and a list of all the synset identifiers whose definitions contain a given term[5]. The state of the reverse index identifies, at every moment, the set of terms used within the reference ontology to build senses' definitions. It is worth noting that every time a designer introduces/deletes a (new) synset's gloss, the reverse index is updated in order to guarantee consistency according to the pre-existing state.

4 RECONCILIATION OF EXTENSIONS

Wandering from a local perspective to a wider distributed environment, new and different problems about data integration and ontology extensions' exploitation arise. Fig. 5 represents three systems (A, B and C), each of these exploits MOMIS to perform data integration. Let us suppose that the designer of system A wants to integrate

[5] Each synset gloss length and the position(s) of the terms within the gloss are also stored to reduce computation time.

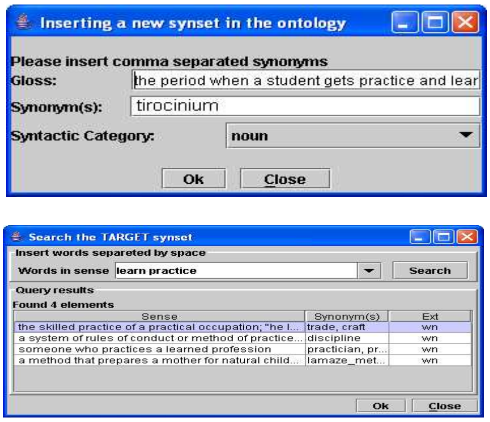

Figure 3. WNEditor: creating a new synset.

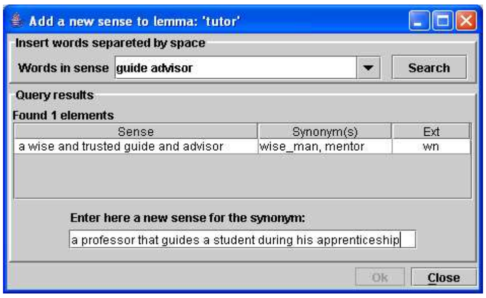

Figure 4. WNEditor: creating a new sense for a lemma

two data sources coming from B and C, that have been already annotated within their local environment. Then, system A requires source data's structures and the WordNet extensions used to annotate them, i.e. *System A temporary loads external WordNet's extensions into its own reference ontology*. Several aspects have to be investigated to guarantee the correct exploitation of external WordNet extensions. Apart from the representation of the lexicon ontology and the issues about exporting extensions between two or more peers, we focalize on the *ontology alignment* tasks. As shown in [18], ontology integration has to:

- find the overlaps between the two ontologies;
- relate concepts that are semantically close via equivalence and subsumption relations;
- check the consistency, coherency and non-redundancy of the result.

In our case, we have to exploit WordNet synsets instead of simple concepts, thus the most critical phase is the concept alignment, since it requires to "understand" the meanings expressed in the glosses' definitions. We need, at least, a similarity metric to be applied on whole synsets, including glosses, synonyms' sets and relationships. More precisely, system A should be able to divide synsets coming from B and C into two categories: the ones that already exist in its own reference ontology and the non-existing ones. Briefly, we have to test if the synset *synX* already exists in the system A reference

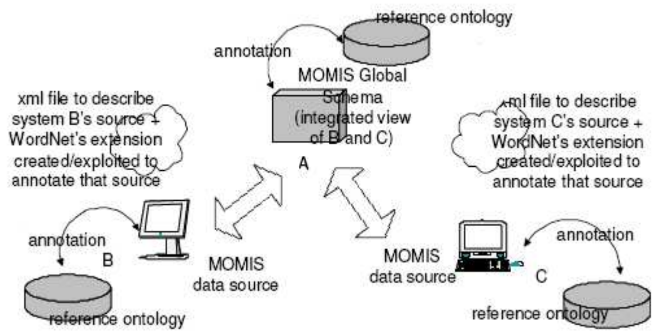

Figure 5. Peer-to-Peer scenario for WordNet extensions reconciliation

ontology[6]. Under the assumption that two definitions of the same concept may share at least one significant word, a first immediate solution is to perform an *exact match* on the gloss' terms and on the set of synonyms. On the other hand, it could be very rare that two synsets belonging to two different reference ontologies are defined in the same manner and/or have a different sense definition but the same set of synonyms. Thus, we propose an *approximate matching technique* that considers both the *syntactic* and *semantic* similarity between two synsets, i.e. the similarity function can be seen a *combination function* f_{SYM}:

Definition 2 *Given a query synset Q_{syn} with a gloss Q_{gloss} whose length is L_{Qgloss}, the set of n synonyms Qlemma$_1$,..., Qlemma$_n$ and the set of m relations (Qrel$_1$,..., Qrel$_m$) involving Q_{syn} as the source synset, it is theoretically possible compare it to any other synset of the reference ontology (candidate target synset T_{syn}, with a gloss T_{gloss} of lenght L_{Tgloss}, a set of p synonyms Tlemma$_i$ and a set of q relations Trel$_j$) and calculate the combination function:*

$$f_{SYM}(Q_{syn}, T_{syn}) = \gamma f_{SEM}(Q_{syn}, T_{syn}) * f_{SYN}(Q_{syn}, T_{syn}) \qquad (2)$$

where $\gamma \in]0, 1]$, $f_{SEM}(Q_{syn}, T_{syn})$ is the semantic contribute from the similarity between the two meaning definitions while $f_{SYN}(Q_{syn}, T_{syn})$ is the syntactic contribute that considers the similarity between the sets of synonyms.

Going into details about syntactic similarity function, the approximate text match can be performed on the basis of the edit distance or the name match[13]. However, the semantic similarity function is the most important index we can adopt in order to determine whether the definitions of two concepts express the *same* concept. The function exploits the *definition match* approach. Two different well-known IR techniques are implemented: *vector space model match*[1] and *Latent Semantic Indexing match*[8].

It is necessary to point out that the combination formula should increase with increasing values of the semantic and syntactic similarity functions. Moreover, it should mitigate the syntactic match's tendency to grow large quickly (e.g applying a square root computation)[7]. Given a synset Q_{syn}, the tool displays a list of proposed

[6] Henceforth, the synset whose existence has to be tested will be referred as the *query synset*.

[7] In general the γ value is a-priori defined or obtained though an automated training.

target synsets only; then, the human supervisor is requested to confirm the proposed match. Every time a source synset is confirmed equal to the reference one, the two objects are superimposed in the ontology network. Future work will be addressed to design a *conflict resolution* algorithm, to perform a mapping between the relations' graph associated to the query network (i.e. the network that relates the query synset each others) and the underlying relations graph in the reference ontology.

5 RELATED WORK

According to the Gruber's definition, an ontology is an *explicit specification of a conceptualization* [9]. The idea beyond this definition is that the declarative formalization of the domain knowledge starts from the conceptualization of the domain, that is the identification of the objects that are supposed to exist in the world and the relationships between them. As discussed in [10], every information system has its own ontology, according to a particular vision of the world. The alignment of two different ontologies implies mismatch resolution[15]. Mismatches may reside at a *language or meta-model* level, because different mechanisms (*syntax, logical representation, semantics of primitives, language expressivity*) are used to define ontology items (classes) and their relations. Several approaches to meta-model level mismatches are the *Superimposed MetaModel*[5], *OKBC*[7] and *OntoMorph*[6].

In our case, we can assume that no mismatch of the meta-model level can exist since we are addressing to the issue of aligning two or more extension of an original lexicon ontology. Despite this, two type of mismatches are the most possible: *conceptualization mismatches* and *terminological mismatches*[8]. The first type regards mismatches in *scope*, i.e. when two ontology classes seem to represent the same concept but do not have the same instances, and in *model coverage and granularity*, i.e. mismatch in the part of the domain covered by the ontology or the level of detail to which that domain is modelled. The second type regards mismatches that arise when the same concept is represented by different names in the ontologies (synonym terms, i.e term mismatch) and when the meaning of a term is different in different context (homonym terms, i.e. concept mismatch). In particular, *Chimaera*[18] is an interesting ontology merging and diagnosis tool. The major supported tasks are the coalesce two semantically identical terms from different ontologies, so the resulting ontology refers to them with the same name. Also *PROMPT*[20] is an interactive ontology merging tool that guides the user by making suggestions, determining conflict and proposing possible solutions.

6 CONCLUSION AND FUTURE WORK

The need to extend the reference ontology in an integration information system is due to the current way in which sources are annotated. In this document, we presented an approach to make possible the extension of a reference lexicon ontology. The complexity of the adopted lexicon ontology, i.e WordNet, is handled with a intuitive an user-friendly graphical tool. Moreover, in order to exploit other people's annotation we propose a solution for reconciling two WordNet extensions developed by different designers. In future work we will define a test benchmark to evaluate the quality of the ontology alignment achieved through the implemented similarity functions.

[8] It is necessary to point out that lexicons supply a partial solution about semantic interoperability, since in the Semantic Web area formal languages such as description logics are used to make explicit the intended meaning of concepts.

ACKNOWLEDGEMENTS

This work is supported in part by the EC's 5th Framework IST program through the SEWASIE project (coordinated by Sonia Bergamaschi) in the Semantic Web Action Line. Further information at *http://www.sewasie.org*. We would like to thank Veronica Guidetti[11] for her contribution in early design of WNEditor and the anonymous reviewers for their helpful advice.

REFERENCES

[1] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison-Wesley, 1999.
[2] C. Batini, M. Lenzerini, and S. B. Navathe, 'A comprehensive analysis of methodologies for database schema integration', *ACM Symp. of Principles of Database Systems*, 51–61, (1997).
[3] D. Beneventano, S. Bergamachi, F. Guerra, and M. Vincini, 'Synthesizing an integrated ontology', *IEEE Internet Computing, The Zen of the Web*, 42–51, September–October, (2003).
[4] S. Bergamaschi, S. Castano, and M. Vincini, 'Semantic integration of semistructured and structured data sources', *SIGMOD Record, Special Issue on Semantic Interoperability in Global Information*, **28(1)**, 54–59, (1999).
[5] M. Bowers and L. Delcambre, 'Representing and trasforming model-based information', *First Workshop of the Semantic Web at the Fourth European Conference on Digital Libraries, Lisbon, Portugal*, (2000).
[6] H. Chalupsky, 'Ontomorph: A translation system for symbolic knowledge', *Principles of Knowledge Representation and Reasoning, Proceedings of the seventh international conference (KR2000)*, (2000).
[7] V.K. Chaundri, A. Farquhar, R. Fikes, P.D. Karp, and J.P. Price, 'Okbc: a programmatic foundation for knowledge base interoperability', *Proceedings of the AAAI-98, AAAI Press/The MIT Press, Madison, Wisconsin*, (1998).
[8] S. Deerwester, S.T. Dumais, T.K. Landauer, G. W Furnas, and R. H Harshman, 'Indexing by latent semantic analysis', *Journal of the American Society of Information Science*, **41**(6), 391–407, (1990).
[9] T.R. Gruber, 'A translation approach to portable ontology specifications', *Knowledge Acquisition*, (1993).
[10] N. Guarino, 'Formal ontologies and information systems', *Proceedings of the International Conference on Formal Ontology and Information Systems (FOIS'98)*, Trento, Italy, (1998).
[11] V. Guidetti, 'Intelligent information integration systems: Extending a lexicon ontology', *Master thesis in Computer Science, University of Modena and Reggio Emilia*, (2002).
[12] J. Hendler, O. Lassila, and T. Berners-Lee, 'The semantic web, a new form of web content that is meaningful to computers will unleash a revolution of new possibilities', *Scientific American*, (2001).
[13] E. H. Hovy, 'Combining and standardizing large-scale, pratical ontologies for machine translation and other uses.', *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, Granada, Spain, May 28–30, AAAI Press, (1998).
[14] R. Hull, 'Managing semantic heterogeneity in databases: a theoretical perspective', *ACM Symp. of Principles of Database Systems*, 51–61, (1997).
[15] M. Klein, 'Combining and relating ontologies: an analysis of the problems and solutions', *Workshop on Ontologies and Information Sharing, IJCAI'01*, (2001).
[16] V.I. Levenshtein, 'Binary codes capable of correcting spurious insertions and deletions of ones', *(Russian) Problemy Peredachi Informatsii*, (1965).
[17] S.E. Madnick, 'From vldb to vmldb (very many large data bases): dealing with large-scale semantic heterogeneity', *VLDB International Conference*, 11–16, (1995).
[18] D.L. McGuiness, R.E. Fikes, and S. Wilder, 'An environment for merging and testing large ontologies', *in Selman, Cohn, Giunchiglia, Principles of Knowledge Representation and Reasoning. Proceedings of the 7th international conference (KR2000)*, 483–493, (2000).
[19] A.G. Miller, 'Wordnet: A lexical database for english', *Communications of the ACM*, **38(11)**, 39–41, (1995).
[20] F. N. Noy and M.A. Musen, 'Prompt: Algorithm and tool for automated ontology merging and alignment', *Proceedings of the 7th AAAI-2000*, (2000).