# Enriching Information Agents' Knowledge by Ontology Comparison: a Case Study

Gustavo A. Giménez-Lugo[1], Analia Amandi[2], Jaime Simão Sichman[1] and Daniela Godoy[2]

[1] Laboratório de Técnicas Inteligentes
Escola Politécnica da Universidade de São Paulo
Av. Prof. Luciano Gualberto, 158 tv. 3
05508-900 São Paulo, SP, Brazil
{gustavo.lugo,jaime.sichman}@poli.usp.br
[2] Universidad Nacional del Centro de la Prov. de Bs. As.
Facultad de Ciencias Exactas - ISISTAN Research Institute
C.P. 7000 - Tandil, Buenos Aires, Argentina
e-mail: {amandi,dgodoy}@exa.unicen.edu.ar

**Abstract.** This work presents an approach in which user profiles, represented by ontologies that were learned by an interface agent, are compared to foster collaboration for information retrieval from the web. It is shown how the interface agent represents the knowledge about the user along with the profiles that were empirically developped. Departing from a specific matching model, briefly presented here, quantitative results were achieved by comparing such particular ontologies in a fully automatic way. The results are presented and their implications are discussed. We argue that an agent's knowledge can be enriched by other agents navigation experiences, possibly helping in connecting and reconciliating distinct knowledge views while preserving a degree of privacy.

## 1 Introduction

Information agents applied to Information Retrieval(IR) try to detect users preferences to help them in information search tasks. The registered preferences, considered as user profiles, are used as guides in the search processes. An active research field tries to improve the results of this type of agents, working generally isolated, through cooperation. Such an approach, in the form of Multi-Agent Systems (MAS), would foster knowledge sharing, allowing access to the results of other agents experiences that could potentially enrich the results achieved by individual agents. To pave the way for knowledge sharing it is necessary to determine the form by which recorded preferences can be compared.

The PersonalSearcher agent [5], was designed to assist users to filter information during Internet search sessions and to keep profiles that can be treated as ontologies. The agents were initially designed to run as stand-alone systems. An alternative to improve their search performance is to acquire knowledge about themes that are related to the ones that appear in a particular profile, possibly

available with other agents, acting as part of a MAS, although no documents should be automatically exchanged to better preserve the user privacy. The first step in this direction is to compare quantitatively concepts that belong to different user profiles, on their current form. This is the aim of the experiences that are related in this work, describing the experimental results obtained with the implementation of an algorithm for comparing different user profiles, considering the way this knowledge is stored by PersonalSearcher agents. The chosen similarity measure is the MD3 [18] model, that takes into account the words that describe a concept and also its semantic neighborhood. This model is briefly described along with the modifications that were necessary for its adoption.

The work is organized as follows: section 2 presents basic concepts related to information agents; section 3 outlines the model used to quantify the similarity between concepts of different user profiles (ontologies) and shows the implemented algorithm that screens the compared ontologies to map those concepts that may be suitable for an eventual sharing. Next, section 4 details the experimental conditions concerning the selection of the test collections used to generate the profiles and later compare them, as well as the modifications needed to adapt the similarity model to the characteristics of the PersonalSearcher profile representation. Section 5 discusses the results along with implications concerning related work. Finally, section 6 presents the conclusions about this experiment.

## 2 Information agents

Agents are usually processes that run continously, know what to do and when to intervene. Agents communicate with other agents, asking solicitations and executing the requested tasks. According to [8], an agent has a long list of properties, among which can be highlighted: *autonomy*, *social hability*, *reactivity* and *proactivity*. Due to the enormous ammount of information accessible through the Internet, and the short time a user generally has to find relevant information, a type of agent that has been widely researched is the so called *intelligent information agent* [9, 12, 10, 16, 3]. They are defined in [9] as computational software entities that can access one or multiple information sources that are distributed and heterogeneous and can acquire, mediate and mantain proactively relevant information on behalf of the user or other agents, preferably in a *just-in-time* fashion. They can be, in general, *cooperative* and *non-cooperative*. Additionally, both types can be *rational*, *adaptive* and *mobile*. Recommender agents are a special case of information agents. Two methods that are commonly used in recommender agents, based on Machine Learning(ML) and making part of a MAS applied to IR, are [16, 3]:

- *Content based approaches*: agents seek for items that are similar to those preferred by the user, comparing content. Rooted in the IR domain, they are popular for textual data and can be applied succesfully to isolated users;
- *Collaborative approaches*: or *social learning* [13]. They assume that there is a group of users of the system, computing the similarity between users (not items) and recommending items that similar users have found interesting.

The performance of such systems crucially depends on the type of modeling used to represent their users.

### 2.1 User Profiles

Each agent in a MAS can be considered to have a particular vision of its environment. This vision can be made explicit through an ontology. An ontology is defined in [7] as a logical theory which gives an explicit, partial account of a conceptualization. Furthermore, a conceptualization is an intensional semantic structure which encodes the implicit rules constraining the structure of a piece of reality [7]. Thus, an ontology is a compromise with a specific conceptualization of the world. Guarino's definition refines another definition given by Gruber [6], stating that an ontology is a explicit specification of a conceptualization.

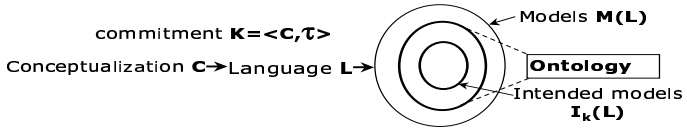


**Fig. 1.** Relating ontologies, language and conceptualization [7].

The use of ontologies to explain implicit knowledge is a viable approach to overcome the problem of semantic heterogeneity. Interoperability between agents can be achieved reconciliating the different world views through the compromise to a common ontology [19]. Also, information agents can use ontologies to represent explicitly isolated user profiles [17, 1], which is the view adopted in this work concerning the profiles generated by the PersonalSearcher agent.

### 2.2 PersonalSearcher

The PersonalSearcher [5] agent is dedicated to the task of retrieving information from the web. It uses Case Base Reasoning to build a profile of the thematic preferences of the user, using an approach detailed in [5]. This profile is automatically generated and it is incremental, being used to expand the user requests with relevants terms, as well as to filter relevant pages from the set of results obtained from general purpose search engines like Google, Yahoo or Altavista.

The profiles are generated by the agent watching the user's search behaviour. Parameters as the time dedicated to read a document, its source address, content, etc. are taken into account to describe the document as a *case* which is compared to previously recorded cases. When a certain degree of similarity is reached between cases, they are grouped into a cluster. Clusters are monitored, and when the chosen parameters remain invariant, a group of words is taken to generate a *theme*, that will be used from then on to reduce the space of options

when verifying if a document retrieved from the web is relevant for the user. A theme has an associated *preference* value, ranging from zero to one. The preference value is given by the time spent by an user reading the cases that are classified under a theme, when compared to the overall reading time of the user. The themes successively generated by the PersonalSearcher agent are hierarchically organized. The thematic hierarchies, or *theme trees*, can be considered as representing particular ontologies. This way, themes can be treated as *concepts* having *features*, actually words that describe them.

## 3 Ontological matching as a quantitative measure of profile similarity

In the literature there are some works that deal with the comparison of ontologies [15, 4, 14] referencing specific tools for the construction of complex ontologies with the intervention of human experts. On the other hand, in [18] it is presented a relatively simple model, called MD3, to compare ontologies, that allows a completely automated comparison, combining three forms of similarity measurement. Furthermore, it can be applied with some modifications to the hierarchies that appear in the PersonalSearcher user profiles or even to more detailed ontologies. Even though the use of the MD3 model can cause some precision loss in the results, when a society of information agents is considered it is essential that the comparison process could be made effectively with the highest degree of automation, so to avoid a situation in which the user is continously asked to decide upon conflicts that would distract her/him from her/his real goals. The MD3 model evaluates the similarity between classes belonging to different concept hierarchies. It was used as a basis to allow the comparison of profiles of different users of PersonalSearcher. It considers that two independent concept hierarchies (ontologies) are connected through a more general (imaginary) class as it appears in figure 2.a).

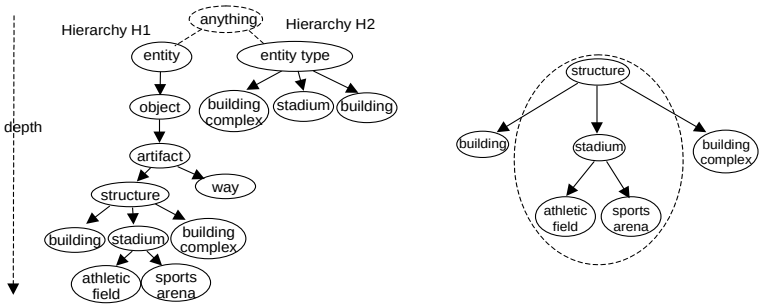


**Fig. 2.** a)Connecting two ontologies using a common imaginary root (left); b)semantic neighborhood with $r$=1 for $stadium^{H1}$, containing 4 classes (rigth) [18].

The MD3 model combines *feature-matching* and *semantic distance*. The global similarity value $S(a,b)$ between two concepts $a$ and $b$, belonging to different ontologies, say $p$ and $q$, is given by the following equation:

$$S(a^p, b^q) = w_l * S_l(a^p, b^q) + w_u * S_u(a^p, b^p) + w_n * S_n(a^p, b^q) \quad (1)$$

$S_l$, $S_u$ and $S_n$ denote the lexical, feature and semantic neighborhood similarities between the concepts $a$ and $b$, and $w_l$, $w_u$ and $w_n$ are the weights of each component when calculating the global similarity value. As the lexical similarity can be influenced by polysemy, the feature and semantic neighborhood similarities should be used. The values of $w_l$, $w_u$ and $w_n$ depend on the characteristics of the ontologies to be compared and they must add up to one. However, only common specification aspects can be used to verify the degree of similarity. E.g., if an ontology labels a concept with symbols that carry no information about it (say "ABC0027"), the lexical similarity is useless. In the similarity functions $S_i(a^p, b^q)$, $a$ and $b$ are classes (in the case of PersonalSearcher, *themes*) and $i$ denotes the type of similarity (lexical, feature and semantic neighborhood). Let $A$ and $B$ be the sets of features of $a^p$ and $b^q$. The *matching* process determines the cardinality of the intersection and the difference between $A$ and $B$.

$$S_i(a^p, b^q) = \frac{|A \cap B|}{|A \cap B| + \alpha(a^p, b^q) * |A - B| + (1 - \alpha(a^p, b^q)) * |B - A|} \quad (2)$$

The value of $\alpha$ is calculated using the *depth* of the classes in the connected hierarchies: $\alpha(a^p, b^q) = \begin{cases} \frac{depth(a^p)}{depth(a^p)+depth(b^q)} & \text{If } depth(a^p) \leq depth(b^q) \\ 1 - \frac{depth(a^p)}{depth(a^p)+depth(b^q)} & \text{If } depth(a^p) > depth(b^q) \end{cases}$

An important aspect is that the similarity $S(a^p, b^q)$ between concepts is not a necessarily symmetric relation [18], e.g. "a *hospital* is similar to a *building*" is more generally accepted than the expression "a *building* is similar to a *hospital*".

To calculate the semantic neighborhood similarity is considered the notion of *semantic neighborhood*, that is the set of classes which distance to a given class is within a specified *radius*. The semantic neighborhood of a class contains the given class, as can be seen in fig. 2.b). Formally: $N(a^o, r) = \{c_i^o\}$, where $a^o$ and $c_i^o$ are classes of an ontology $o$, $r$ is the radius and $N(a^o, r)$ is the semantic neighborhood of $a^o$. Furthermore, $\forall c_i^o \in N(a^o, r)$, $d(a^o, c_i^o) \leq r$. This notion of similarity is based on shallow matching, associated to the evaluation of the immediate neighborhood of a class, i.e. the *radius* is of value 1.

Given two concepts $a^p$ and $b^q$ from ontologies $p$ and $q$, where $N(a^p, r)$ has $n$ concepts and $N(b^q, r)$ has $m$ concepts, and the intersection of the two neighborhoods is denoted by $a^p \cap_n b^q$, the value of $S_n(a^p, b^q, r)$ is calculated by:

$$\frac{|a^p \cap_n b^q|}{|a^p \cap_n b^q| + \alpha(a^p, b^q) * \delta(a^p, a^p \cap_n b^q, r) + (1 - \alpha(a^p, b^q)) * \delta(b^q, a^p \cap_n b^q, r)} \quad (3)$$

where $\delta(a^p, a^p \cap_n b^q, r) = \begin{cases} |N(a^p, r)| - |a^p \cap_n b^q| & \text{If } |N(a^p, r)| > |a^p \cap_n b^q| \\ 0 & \text{In other case} \end{cases}$

The intersection between the neighborhoods is approximated by the similarity of classes within them: $a^p \cap_n b^q = \left[\sum_{i \leq n}\left(max_{j \leq m} S(a_i^p, b_j^q)\right)\right] - \varphi * S(a^p, b^q)$

with $\varphi = \begin{cases} 1 & \text{If } S(a^p, b^q) = max_{j \leq m}\ S(a^p, b_j^q) \\ 0 & \text{In other case} \end{cases}$

As $S(a^p, b^q)$ is asymmetric, $a^p \cap_n b^q$ also is. The similarity between two classes $a_i^p$ and $b_j^q$, from the neighborhoods of $a^p$ and $b^q$ is calculated using lexical and attribute similarity, with equal weights (0.50): $S(a_i^p, b_j^q) = w_l * S_l(a_i^p, b_j^q) + w_u * S_u(a_i^p, b_j^q)$. From here on, $S_n(a^p, b^q)$ should be read as $S_n(a^p, b^q, 1)$.

### 3.1 Comparison algorithm

Using the MD3 model, a comparison algorithm was implemented. It has two procedures, $Trav\_1$ and $Trav\_2$, shown in fig 3, being both essentially breadth-first traversals of the PersonalSearcher user profiles theme trees. $Trav\_1$ traverses the first profile $profile\_1$, considered as a $reference$. $Trav\_2$ receives a node of $profile\_1$, with its neighborhood, and compares it with all the concepts in $profile\_2$. The MD3 model was adjusted as follows:

- As a profile concept label carries no information (e.g. "$SBJ48$"), $w_l$ is set to zero: $S(a^p, b^q) = 0.0 * S_l(a^p, b^q) + 0.5 * S_u(a^p, b^p) + 0.5 * S_n(a^p, b^q)$;
- Theme features have associated weights that affect the values of $|A \cap B|$, $|A - B|$ and $|B - A|$ in eq. 2 when calculating feature similarity. In eq. 2 the cardinality is an integer, now is a real number. A possible interpretation is that the values have attached a confidence degree. A feature is now denoted by an ordered pair indicating the weight of the feature for a concept;
- When two *isolated themes* (i.e. themes that are linked to the root and have no children) are compared, only the feature similarity is calculated.

**Procedure 1** Trav_1(profile_1, profile_2)

```
queue_1.add(profile_1.root)
while queue_1 not empty do
  node_1← queue_1.head
  for k=1 to num_children(node_1) do
   child_k ← next_child(node_1)
   queue_1.add(child_k)
   queue_neighb.initialize()
   queue_neighb.add(node_1)
   queue_neighb.add(child_k)
   for j=1 to num_children(child_k) do
     queue_neighb.add(next_child(child_k))
   end for
   Trav_2(child_k, queue_neighb, profile_2)
  end for
  queue_1.retrieve(node_1)
end while
```

**Procedure 2** Trav_2(node, queue_neighb,profile_2)

```
queue_2.add(profile_2.root)
while queue_2 not empty do
   node_2← queue_2.head
   for m=1 to num_children(node_2) do
    child_m ← next_child(node_2)
    queue_2.add(child_m)
    queue_neighb_2.initialize()
    queue_neighb_2.add(node_2)
    queue_neighb_2.add(child_m)
    for n=1 to num_children(child_m) do
      queue_neighb_2.add(next_child(child_m))
    end for
    element_queue.sim=S(node,queue_neighb,child_m,queue_neighb_2)
    node.queue_sim.add(element_queue)
   end for
   queue_2.retrieve(node_2)
end while
```

**Fig. 3.** Traversal procedures used in the comparison process.

## 4 Experimental results

The described model, implemented as part of the algorithm, was applied to sample user profiles, acquired feeding PersonalSearcher with subsets of test collections of web pages. The profiles correspond to fictitious users. This approach was taken to ensure repeatability, as several parameters can be used to tune the PersonalSearcher theme generation: threshold to add a case to a cluster; number of cases included in a cluster without introducing changes in the set of candidate attribute words (used to transform a cluster into a theme); threshold to classify a case under a theme; threshold to choose a word to be a candidate attribute.

The used collections were made publicly available by the CMU Text Classification Group: the first, more structured, called $WebKB$, has 8282 pages collected in 1997 from Computer Science Departments of four USA Universities, organized into 7 main directories with 5 subdirectories each; the second, rather flat, called *Sector*, has 9548 pages distributed homogeneously into 105 directories corresponding each to an industrial activity sector. The subsets used to fed PersonalSearcher were: 973 pages from one of the 7 main directories of the $WebKB$ collection; 521 pages from 5 directories of the *Sector* colection. From here on, citing a collection denotes the used subset. Three profiles, shown in fig. 4, were generated: *UserONE*, using *Sector* as input. Its structure is rather flat, containing just 6 *isolated themes*(i.e., a theme linked to the root, having no children), neither of which is subclassified under another; $UserTWO$, using $WebKB$ as input. It is more structured than $UserONE$, with 5 themes, 2 of which are subcategories of more general themes; and *UserTHREE*, obtained feeding initially *Sector* followed by $WebKB$. The order is relevant as it affects the theme generation. A sample concept is shown with its features in table 4.

**Table 1.** A sample concept from the PersonalSearcher *UserONE* profile

| **Theme label** | Theme preference | *Feature (feature weight)* |
|---|---|---|
| *SBJ24* | (0.08) | profil(1.0), innov(1.0)<br>corpor(1.0), entertain(1.0)<br>fall(1.0), lodgenet(1.0)<br>sioux(1.0), employ(1.0) |

The comparison of the generated profiles was performed in two steps. Initially, each one of the three profiles was compared with itself, in order to have reference cases. Next, they were compared with each other. Each concept of the $reference$ ontology keeps a list of the concepts of the other ontology upon with it has a similarity value above a threshold than can be set by the user. Additionally, the preference degree of a theme can be taken into account, causing the similarity to be multiplied by it, decreasing the similarity value. The observations confirmed the predictions stating that the similarity is non-symmetric. Still, identical themes received relatively high similarity values. The lower similarity

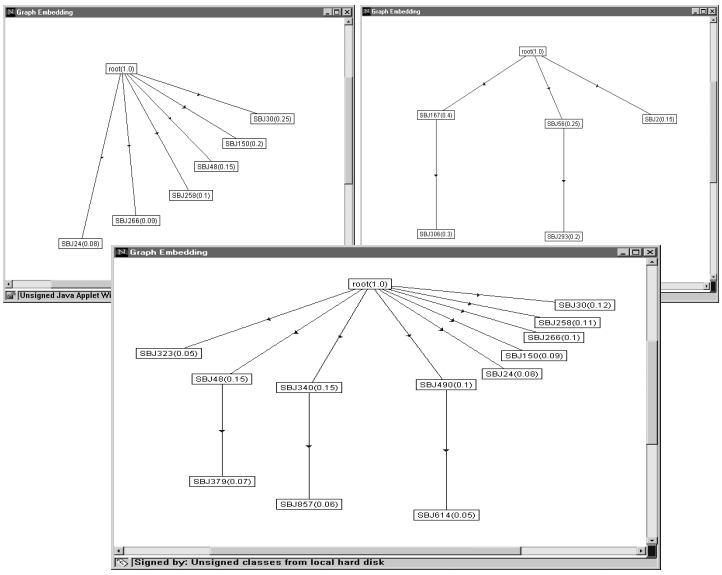


**Fig. 4.** *UserONE* (upper left), *UserTWO* (upper rigth) and *UserTHREE* (down).

value for identical themes was 0.50, resulting from feature similarity, with gradually higher values for correspondingly similar semantic neighborhoods. *Isolated themes* (see section 3.1) reached a similarity of 1.0. Sample comparison results are shown in fig. 5 and in table 4.

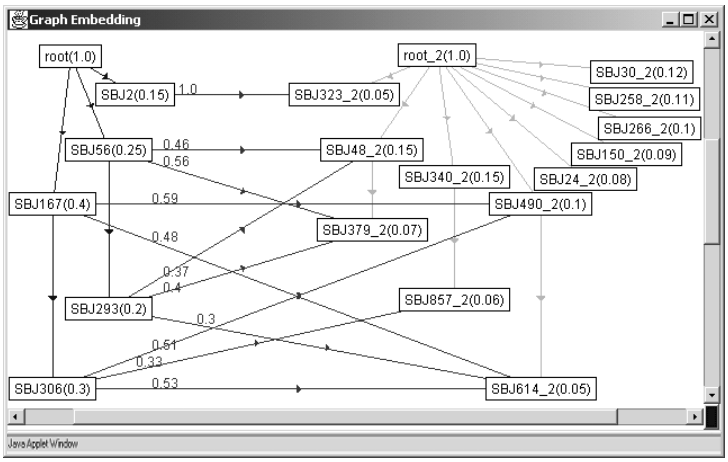


**Fig. 5.** Similarity links values when comparing *UserTWO* with *UserTHREE*.

**Table 2.** Relations and similarity link values between sample concepts, from fig. 5

| UserTWO theme | UserTHREE theme | Similarity value | Direct relation |
|---|---|---|---|
| *SBJ2* | *SBJ323_2* | 1.00 | Identical features |
| *SBJ56* | *SBJ379_2* | 0.56 | Identical features |
| *SBJ167* | *SBJ490_2* | 0.59 | Identical features |
| *SBJ293* | *SBJ614_2* | 0.30 | The 4 *SBJ614_2* features are a subset of the 9 *SBJ293* features |

## 5 Discussion and related work

At the present stage, PersonalSearcher user profiles still need improvements in order to have a more accurate representation of the user knowledge, a problem that is common to fully automated knowledge acquisition approaches.

In this sense, a confidence degree can be worked out for the acquired concepts as part of further work on the acquisition of the profiles. Anyway, the comparison process shown in this work doesn't need modifications to manipulate more detailed and accurate ontologies and will only benefit from better acquisition results, as far as the PersonalSearcher representation model remains the same.

There are few related works reporting the representation and comparison of individual user ontologies mantained by agents as user profiles. The CAIMAN [11] system represents ontologies only as vectors of words, on the other hand PersonalSearcher favors the representation of ontologies in a way that is suitable to cope with a frame based knowledge representation model like OKBC [2], enabling a much richer representation that allows to take advantage of the relations that can be made explicit in a hierarchical structure. In DOGGIE [20], agents exchange full documents to learn new concepts, a highly desirable but rather restricting approach, e.g. for privacy reasons; our approach overcomes much of the associated problem because it allows the exchange of (portions of) structured knowledge, i.e. ontologies, without exposing the actual user documents.

## 6 Conclusions

We have shown an approach in which a quantitative comparison of user profiles, considered as ontologies maintained by information agents, was implemented. We plan to apply the results in a MAS system for IR in which existing interface agents' reasoning capabilities will be extended to support a model in the context of a cooperative MAS in which different degrees of knowledge sharing (say single features, concepts or whole ontologies) will be possible.

## 7 Acknowledgements

Part of this work was done while the first author was studying at the ISISTAN Research Institute/UNCPBA, Tandil, Bs.As., Argentina, funded by CAPES,

Brazil, CAPES/SCyT cooperation program, grant BEX 0510/01-7. The third author is partially financed by CNPq, Brazil, grant 301041/95-4; and by CNPq/NSF PROTEM-CC MAPPEL project, grant number 680033/99-8.

## References

1. J. Chaffee and S. Gauch. Personal ontologies for web navigation. In *CIKM*, 2000.
2. V. Chaudhri, A. Farquhar, R. Fikes, P. Karp, and J. Rice. OKBC: A programmatic foundation for knowledge base interoperability. In *AAAI-98*, 1998.
3. T. Finin, C. Nicholas, and J. Mayfield. Agent-based information retrieval. In *IEEE ADL'98, Advances in Digital Libraries Conference '98*, 1998.
4. N. Fridman and M. Musen. An algorithm for merging and aligning ontologies: Automation and tool support. In *AAAI Workshop on Ontology Mangmt.*, 1999.
5. D. Godoy and A. Amandi. PersonalSearcher: An intelligent agent for searching web pages. In *7th IBERAMIA and 15th SBAI. LNAI 1952.* M. C. Monard and J. S. Sichman eds., 2000.
6. T. Gruber. Towards principles for the design of ontologies used for knowledge sharing. *Intl. Journal of Human and Computer Studies*, 43(5/6):907–928, 1995.
7. N. Guarino. Formal ontological distinctions for information organization, extraction, and integration. In *Information Extraction: A Multidisciplinary Approach to an Emerging Inf. Technology. LNAI 1299.* M. T. Pazienza(ed). Springer, 1997.
8. N. Jennings and M. Wooldridge. Applications of intelligent agents. In *Agent Technology: Foundations, Applications and Markets.* Jennings, N. and Wooldrigde, M.(eds.). Springer Verlag., 1998.
9. M. Klusch. Intelligent information agents. In *Third European Agent Systems Summer School. Advanced Course on Artificial Intelligence ACAI-01.*, 2001.
10. M. Kobayashi and K. Takeda. Information retrieval on the web. *ACM Computing Surveys*, 32(2):144–173, 2000.
11. M. S. Lacher and G. Groh. Facilitating the exchange of explicit knowledge through ontology mappings. In *14th FLAIRS Conf.* AAAI Press, May 2001.
12. A. Y. Levy and D. S. Weld. Intelligent internet systems. *Artificial Intelligence*, 118(1-2):1–14, 2000.
13. P. Maes. Agents that reduce work and information overload. *ACM Comm.*, 1994.
14. D. McGuinness, R. Fikes, J. Rice, and S. Wilder. An environment for merging and testing large ontologies. In *Proc. of the KR2000 Intl. Conf.*, 2000.
15. P. Mitra, G. Wiederhold, and M. Kersten. A graph oriented model for articulation of ontology interdependendies. In *VII EDBT Conference*, 2000.
16. D. Mladenic. Text-learning and related intelligent agents: a survey, 1999.
17. A. Pretschner. *Ontology Based Personalized Search. MSc. Thesis.* Kansas Univ., USA, 1999.
18. M. A. Rodríguez. *Assessing semantic similarity among spatial entity classes. PhD. Thesis.* University of Maine, USA, May, 2000.
19. H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner. Ontology-based integration of information - a survey of existing approaches. In *IJCAI Workshop on Ontologies and Information Sharing*, 2001.
20. A. B. Williams and Z. Ren. Agents teaching agents to share meaning. In *Proc. of the 5th Intl. Conf. on Autonomous Agents.* ACM, May 2001.