Adaptive Retrieval of Semi-structured Data

Yosi Ben-Asher[1], Shlomo Berkovsky[1], Paolo Busetta[2],
Yaniv Eytani[3], Sadek Jbara[1], and Tsvi Kuflik[1]

[1] University of Haifa, Haifa, Israel
{yosi,slavax}@cs.haifa.ac.il, tsvikak@is.haifa.ac.il
[2] ITC-irst, Trento, Italy
busetta@itc.it
[3] University of Illinois at Urbana-Champaign, Illinois, USA
yeytani2@uiuc.edu

Abstract. The rapidly growing amount of heterogeneous semi-structured data available on the Web is creating a need for simple and universal access methods. For this purpose, we propose exploiting the notion of UNSpecified Ontology (UNSO), where the data objects are described using a list of attributes and their values. To facilitate efficient management of UNSO data objects, we use LoudVoice, a multi-agent channeled multicast communication platform, where each attribute is assigned a designated communication channel. This allows efficient searches to be performed by querying only the relevant channels, and aggregating the partial results. We implemented a prototype system and experimented with a corpus of real-life E-Commerce advertisements. The results demonstrate that the proposed approach yields a high level of accuracy and scalability.

1 Introduction

Nowadays, the amount of available semi-structured, heterogeneously represented, and highly dynamic data is growing, making it difficult for users to find and access the data relevant to their needs. Hence, mechanisms are required that will facilitate access to and matching of such data and thus free users from the need to know a priori how the data objects are structured (e.g., schemata or ontologies [9]).

This issue is being approached from different angles. Information retrieval techniques [10] cannot handle semantic and syntactic heterogeneity in the data. Semantic Web research [6] focuses on treating the Web as a knowledge base that defines semantic concepts and their relationships, whereas knowledge representation languages allow the meaning of concepts to be represented using ontologies. The key challenge of the semantic approach is the large number of such ontologies. Data integration [3] and schema matching [8] research studies aim at addressing this challenge by, respectively, reconciling the ontologies and matching between the concepts pertaining to different ontologies. A global ontology allows a uniform data access mechanism and defines rules encapsulating the differences between the ontologies. Despite their relative success, neither data integration nor ontology matching have fully succeeded in resolving the issues related to the dynamic and variable nature of the ontologies.

W. Nejdl et al. (Eds.): AH 2008, LNCS 5149, pp. 32–41, 2008.

This work uses an alternative approach to accessing the data: UNSpecified Ontologies (UNSO) [2]. UNSO assumes that the ontologies are not fully defined and can be dynamically specified by the data providers. Hence, instead of basing the data description on a set of a-priori defined ontologies, the data objects are described in the form of an unspecified list of attributes and their values, where both the attributes and the values in the data object descriptions are determined by the data providers. In this work we investigate the accuracy and efficiency of the UNSO approach implemented over the LoudVoice multi-agent platform [4]. In this setting, an agent can either provide a new data object (i.e., contribute data and define new concepts) or search for the existing data objects. To handle these functions, LoudVoice provides a set of channels that allow the agents to be tuned to existing channel or to create new ones.

The practical part of this work implements UNSO over LoudVoice, and experiments with a set of real-life E-Commerce ads from various domains. The experimental results demonstrate that the proposed approach yields both accurate and efficient data management and search capabilities. Hence, our contribution is two-fold. First, we propose and evaluate a scalable approach for storing data objects over a multi-agent platform, which allows search queries to be posed to a dynamic mechanism that captures the descriptions of the data objects. Second, the LoudVoice communication mode facilitates the extraction of domain meta-data reflecting the dynamic quality of the data objects, which is exploited for optimizing the search and improving the efficiency of the proposed approach.

2 Unspecified Data Management over Multi-agent Platform

Ontologies are referred to as standardized, well-defined, and formal models of a domain, agreed upon all the users [9]. Conversely, the main assumption behind UNSO is that the domain ontologies are not fully specified and their parts can be dynamically specified by the data providers [2]. As such, UNSO allows the data providers to describe the data objects in a relatively unconstrained form of a list of <*attribute:value*> pairs, where neither the attributes nor their values are defined a priori. Although such a description of the data objects may be inapplicable for complex entities, it is sufficient for simple data and real-life objects, e.g., files, products or computing resources.

UNSO may suffer from inconsistent symbolism, since the data providers may insert different descriptions of the same objects. For example, consider the two descriptions of an object shown in Figure 1. There, the same object is described in different ways and using different attributes. Moreover, the same value of the attribute *engine* is described using different values. In [2], this problem is addressed by standardizing the <*attribute*:*value*> pairs mentioned in UNSO descriptions using WordNet [7]. In WordNet, nouns, verbs, adjectives, and adverbs are organized into synonym sets,

<**product**:car, **type**:Mazda>
<**volume**:engine1600, **year**:2000>
<**color**:red, **distance**:Km100000>

<**product**:car, **type**:Mazda>
<**volume**:1.6l, **year**:2000>
<**condition**:good, **owners**:2>

Fig. 1. Different descriptions of the same data object using UNSO

representing the underlying lexical concepts. To overcome the inconsistent symbolism, UNSO data object descriptions are standardized by substituting the original terms used in the descriptions with their most frequent synonyms.

To facilitate efficient data management of UNSO data objects, we extend the notion of implicit organizations [5]. An implicit organization is "a group of agents playing the same role and coordinating their actions." The term implicit stresses that there is no explicit group formation stage, and joining an organization is a matter of sharing functionality with other members of the organization. In the context of UNSO descriptions, implicit organizations reflect the attributes mentioned in the descriptions, such that each attribute is assigned to a single organization. The resulting set of organizations facilitates dynamic management and access to the underlying data objects.

The selected LoudVoice communication platform [4] has been designed to support implicit organizations inherently, as every LoudVoice channel represents an individual organization. To adjust UNSO to an agent-based environment, the data object descriptions are partitioned among agents, mimicking a real-life matching scenario, where real agents represent users offering or searching for a product. Each unique attribute mentioned in the descriptions is assigned a designated channel, such that the agents join the channels through 'tuning' to them. Hence, each agent joins multiple channels, according to the attributes mentioned in the descriptions it stores.

The above mapping of data objects to LoudVoice channels is shown in Figure 2. Data object descriptions in the form of an <*attribute*:*value*> list (left) are inserted by an agent (middle), which is tuned to a channel representing one of the mentioned attributes (right). Note that other agents, storing data object descriptions with the same attribute, are also tuned to that channel. For example, consider the following description of a file: [*name*:*myfile.txt*, *author*:*JohnDoe*, *size*:*1.23*K]. The agent storing this description is tuned to channels representing *name*, type *author*, and *size* attributes. For the sake of simplicity, we refer to the channels using the attribute names only.

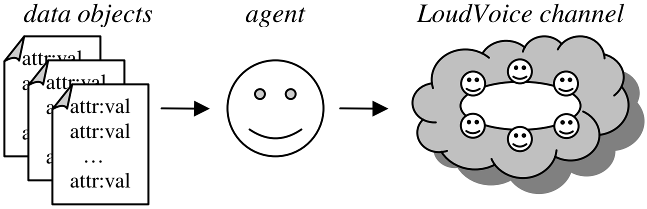

Fig. 2. Mapping of data object descriptions to implicit organizations

LoudVoice implements a channeled multicast communication mode, where messages sent over a channel are received by all the agents tuned to it. Channeled multicast reduces the amount of communication and allows overhearing, i.e., 'eavesdropping' on messages addressed to others. Overhearing, in turn, allows advanced data management functionalities, e.g., domain meta-data extraction (will be presented later), which are achieved through so-called *mediating* agents. These agents are tuned to both the channels pertaining to attributes and the inter-organization communication channels used for transferring information between the channels and coordinating between agents. For example, consider two channels *A* and *B* and their mediating agents tuned also to an inter-organization communication channel, as shown in Figure 3.

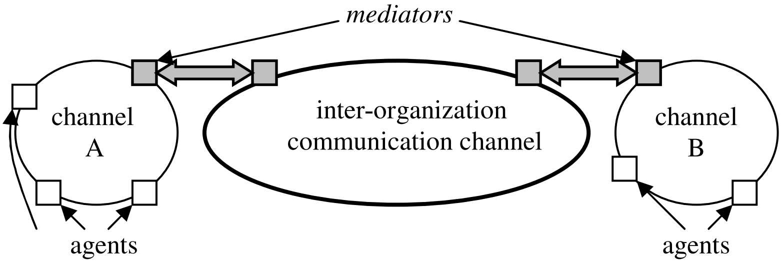

Fig. 3. Organization of LoudVoice channels

3 Semantic Search and Matching over LoudVoice

One of the main characteristics of UNSO is the efficient storage, search, and matching of data objects. This work focuses on the E-Commerce applications, and, in particular, on buy/sell interactions, where sellers offer products, buyers search for products offered by the sellers, and the products are represented by the data objects. We use this setting to demonstrate a protocol for search and matching of the data objects. Various enhancements, such as the highest bid selection and product ranking, can be added to this protocol. The protocol is schematically shown in Figure 4.

Consider three objects offered by two sellers: (O1) Mazda produced in *2000* and having *2* past owners, (O2) Nissan produced in *2002* that costs *$12,000*, and (O3) red Van that costs *$9,000*. Objects O1 and O2 are stored by *seller1* agent, while O3 is stored by *seller2*. In addition, there are two buyers: *buyer1* and *buyer2*, such that *buyer1* is searching for a red car below *$10,000* and *buyer2* is searching for a *Mazda* car having *2* or less past owners. The proposed protocol consists of two steps executed by the buyers and two steps executed by the sellers.

- Step1 (sellers listen): The sellers are tuned to the channels corresponding to the attributes of their objects and wait for queries indicating that the buyers are looking for products represented by <*attribute:value*> pairs. For example, *seller1* is tuned to the channels *product*, *type*, *year*, *price* and *owners*.
- Step 2 (buyers broadcast): The buyers send the desired values of the attributes they are searching. For example, *buyer2* broadcasts *car* on the channel *product*, *Mazda* on the channel *type*, and *less than 2* on the channel *owners*.
- Step 3 (sellers respond): The sellers inform the buyers that they have a product with the mentioned <*attribute:value*> pair. For each buyer, the seller sends one message $seller_{id} \rightarrow buyer_{id}$ specifying all the objects matching the desired value. For example *seller2* sends two messages on the *product* channel indicating that the Van matches the *car* value of *buyer1* and *buyer2*.
- Step 4 (buyers aggregate responses): The buyers collect all the messages sent at the previous step and identify matchings. Thus, *buyer1* obtains a complete matching for the O3 and *buyer2* obtains a complete matching for O1.

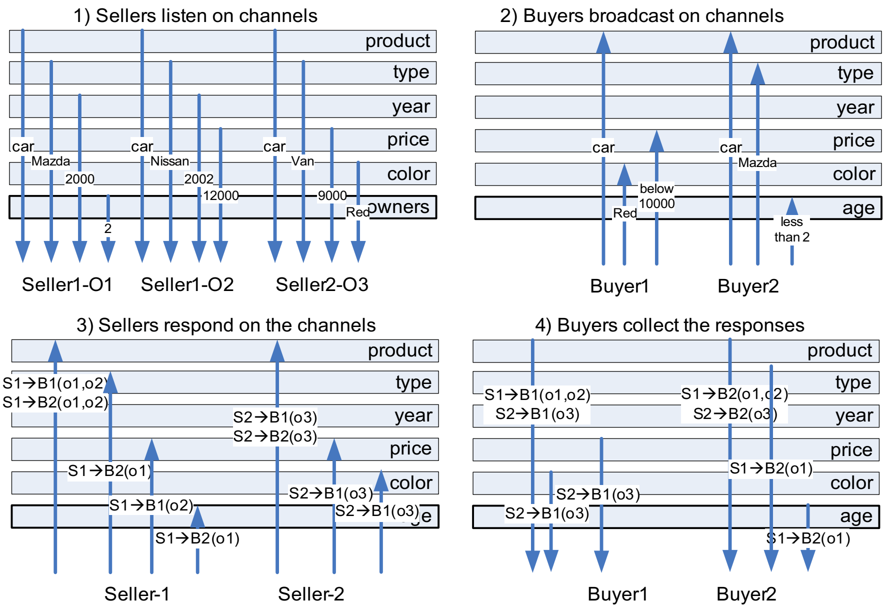

Fig. 4. Stages of the matching protocol

An interesting observation appears when analyzing the overhearing in LoudVoice channeled multicast communication. Due to the overhearing, the mediating agents can learn the attributes used to describe domain objects. Hence, in addition to the matching capability, the proposed structure facilitates autonomous generation of domain meta-data. This is achieved as follows. The mediating agents receives all the messages sent over the channel. Hence, it can collect the data referring to the statistical properties (e.g., distribution, values, frequencies and so on) of the attributes. Using the inter-organization channel, the mediating agents can communicate and collect domain meta-data. For example, the mediating agent of a channel can collect meta-data regarding the possible values of the attribute, or several agents can manage a list of the most frequently mentioned domain attributes.

3.1 Search Optimizations

Now we consider two optimizations of the above protocol, aimed at reducing the number of messages in the system and the overhead related to their processing. The first optimization, referred to as a *dedicated channel*, allows each buyer to use a dedicated channel, where the sellers respond in step 3 of the protocol. In this way, the number of processed messages is reduced, as the sellers send the messages destined for a buyer on a separate channel, to which only the seller and the buyer (and, possibly, a small number of other agents concurrently sending their responses) are tuned. As a result, the number of messages that are received and processed is reduced in comparison to the original protocol, where all the responses are received by all the sellers and buyers that are tuned to the channels.

The second optimization, referred to as *meta-data*, uses the collected domain meta-data to reduce the number of response messages. This is done by substituting the concurrent execution of steps 2 and 3 over multiple channels with sequential execution, such that the order of sequential operations is determined by the frequencies of the attributes. We will illustrate this optimization with an example. Assume that a buyer is searching for an object $<a_1:v_1,..., a_k:v_k>$, whose attributes are ordered according to the attribute frequency (a_1 is the most frequent attribute). The cardinality of the set of objects having a complete match with the query is bounded by the frequency of the least frequent attribute a_k. Hence, sequential querying of channels is ordered such that they are launched first on the channels of the least frequent attributes (starting from the channel of a_k) and then on the channels of more frequent attributes.

To implement this optimization, we use an additional set of candidate object identities, a *CO*. Initially, a *CO* contains the identities of all the available objects. The following operations are repeated for the $<a_i:v_i>$ pairs, according to the attribute frequencies, from the least frequent to the most frequent attribute:

- Broadcast the query for the desired v_i and the current *CO* on the channel a_i
- Sellers, storing the objects containing the desired v_i, respond to the query only if the identity of the data object having the desired v_i appears in the *CO*
- Buyers receive the responses and remove from the *CO* the identities of the objects that were not included in the sellers' responses.

Hence, in each iteration the buyers filter out from the *CO* the identities of the objects not satisfying the desired $<a_i:v_i>$ pair, but only previous $<a_{i+1}:v_{i+1}, …., a_k:v_k>$ pairs. This reduces the number of messages sent, received and processed, as the sellers do not respond to all the identified matchings.

4 Experimental Evaluation

To evaluate the proposed approaches, we collected *5* corpora of real-life E-Commerce *supply* ads from the following application domains: refrigerators, cameras, televisions, printers, and mobile phones. The ads were downloaded from *www.recycler.com* and converted by annotators to UNSO format. For example, "Nokia 5190 phone, charger and leather case, good condition, $125" ad was converted to <*manufacturer*:*Nokia*, *model*:*5190*, *charger*:*yes*, *case*:*leather*, *condition*:*good*, *price*:*$125*>. Conversions were kept as close as possible to the original contents of the ads. A set of *demand* ads was built by modifying the attributes and values of the supply ads. Due to space limitations, we present in this section only the results obtained for a corpus of mobile phones ads. The other corpora demonstrate a similar behavior.

In the first experiment, we evaluated the matching accuracy of the system through the traditional Information Retrieval metric of recall[1] [10]. For this, we used a corpus of *130* supply ads imitating the available data objects and a corpus of *64* demand ads imitating the search queries. For each one of the *64* queries, the recall was computed as the number of retrieved relevant ads divided by the total number of relevant ads in the system. The average recall for all *64* queries was computed in two conditions: (1)

[1] Precision was not measured, as all the ads pertaining to one domain are considered relevant.

for the original terms mentioned in the ads, and (2) after standardizing the <*attribute:value*> pairs using WordNet.

The original average recall was *0.29*. The low result is explained by the observation that when the data objects were defined using UNSO format, the users mentioned different terms in their UNSO descriptions, and only the exact string matching ads were retrieved. Using WordNet standardization, the average recall was *0.8*. This is explained by the nature of the standardization, which substitutes semantically close terms with their most frequent synonym. Note that even after the standardization the recall did not reach the optimal value of *1* due to the fact that WordNet standardization with the most frequent synonym failed to identify syntactic errors, hyponyms and hypernyms, polysemy, and other discrepancies.

We used the same corpora of *130* supply and 64 demand ads also in the second experiment, which was aimed at measuring the communication overhead of the proposed mechanism. We gradually increased the number of inserted ads N_c from *1* to *125* and for each value of N_c launched the same set of *64* queries. In the experiments we measured four metrics: (1) the number of established channels, i.e., channels to which at least one agent was tuned, (2) the overall number of messages sent for every query, (3) the overall number of ads received and processed for every query, and (4) the average size of the messages. For each value of N_c, the *64* searches were repeated *1,000* times, for randomly selected sets of supply ads. Figures 5, 6, and 7 show the results of all four metrics for the original protocol (Figure 5), *dedicated channel* optimization (Figure 6), and *meta-data* optimization (Figure 7). Note that the average message size values were scaled down to be shown with the other metrics.

Figure 5 shows that the number of channels established converges fast. This is explained by the observation that even a small number of ads provides most of the domain attributes, whereas further ads contribute few new attributes. Also the number of messages sent in a single search converges. This is explained by the fact that the sellers respond to the queries regardless of the number of matching ads they store. Hence, even for a small number of ads they respond if a matching is identified, whereas further insertions contribute few new responses. It can be seen that both the number of channels and the number of messages sent reach over *80%* of their maximal values when inserting approximately *20%* of the ads.

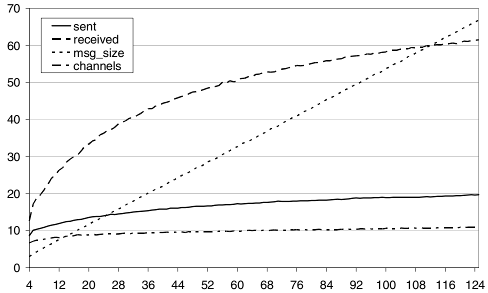

Fig. 5. Original protocol

The number of received messages converges more slowly than the number of sent messages, as every message sent on a channel is received by multiple agents tuned to it. Thus, a minor increase in the number of sent messages is reflected by a stronger increase in the number of received messages. Since the response messages include the identity of the data object having the desired attribute value, the average message size increases linearly with the number of ads. The results obtained for the original protocol serve as a baseline for following two protocol optimization experiments.

Figure 6 shows that *dedicated channel* optimization leads to a major improvement in terms of the number of received messages. Since in this case the messages are sent over a dedicated channel, they are received by a smaller number of agents. As a result, the number of received messages is significantly lower than in the original protocol. However, this is reflected by the number of channels established, which is higher by *1*, i.e., the channel where the responses are sent, than in the original protocol.

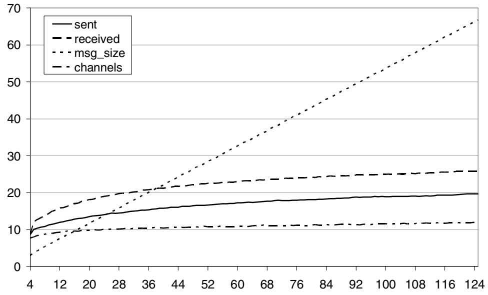

Fig. 6. *Dedicated channel* optimization

Figure 7 shows that *meta-data* optimization leads to an improvement in three metrics: the number of messages sent and received, and the average message size. The first two are explained by the fact that the set of seller responses in this optimization is smaller than in the original protocol, as the set of candidate objects *CO* inherently limits the data objects for which the sellers can potentially respond. Hence, the number of sent messages decreases and, as a result, the number of received messages. Surprisingly, also the average message size decreases. Unlike in the original protocol, where the sellers respond to all the matchings, in this optimization they respond to

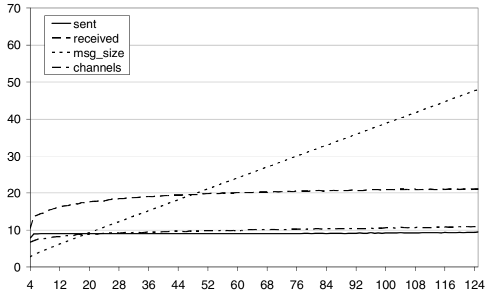

Fig. 7. *Meta-data* optimization

only the identity of the data object appearing in the *CO*. Thus, the average size of the response messages decreases. As for the number of channels established, the performance of this optimization is identical to the performance of the original protocol.

Finally, to compare the advantages and disadvantages of the above optimizations, we computed the overall communication overhead by multiplying the average message size by the sum of the number of sent messages and the number of received messages, i.e.,

$$overall\ overhead = (sent\ messages + received\ messages) * message\ size\ .$$

Intuitively, this computation reflects the observation that every message sent over a channel requires communication and every message received requires processing overhead, where the overheads are proportional to the size of the message. The overall communication overheads of the original protocol and two optimizations are shown in Figure 8. As can be seen, both optimizations are superior to the original protocol, while *meta-data* optimization outperforms *dedicated channel* optimization. These results allow us to conclude that *meta-data* optimization leads to the most significant improvement in terms of the communication overhead.

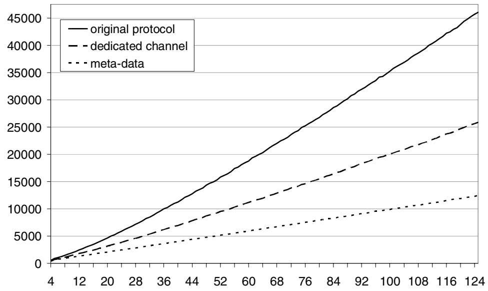

Fig. 8. Overall communication overhead

5 Conclusions and Future Research

This work was motivated by the abundance and dynamic nature of semi-structured and heterogeneous data on the Web. While many previous studies were focused primarily on overcoming the heterogeneity challenge, this study focused on overcoming the dynamicity of the data. For this, we used the flexible UNSO descriptions of the data objects and implemented a multi-agent system for the management and matching of the data objects over the LoudVoice channeled multicast communication platform.

Experimental evaluation comprised two parts. In the first part, we evaluated the contribution of semantic standardization to the retrieval capabilities. The results showed their dramatic improvement with respect to the recall metric. In the second part, we evaluated the scalability of the proposed approach. For this, we measured four factors: the number of channels established, the number of messages sent and received, and the average size of the messages. We evaluated the original search protocol and two optimizations based on (1) slightly modified communication policy, and (2) domain meta-data learned from the existing data objects. Experimental results showed that both optimizations lead to a decrease in the communication overheads. Although the reported results pertain to one corpus of ads from one application domain, in other corpora and domains the results were similar, allowing us to hypothesize that the results will be valid also for large-scale Web-based data.

In the future, we plan to evaluate the performance of another optimization, combining the advantages of the optimizations presented and evaluated in this work. That is, according to the envisaged optimization, the sellers will use a dedicated channel for responding to queries, whereas querying the channel will be sequential and ordered according to the frequencies of the attributes. We hypothesize that this optimization will decrease further the communication overheads of the proposed approach.

The manual generation of UNSO data object descriptions by human annotators constitutes a serious drawback (and, in fact, scalability limit) of the current work. However, to provide their descriptions in UNSO format may emerge as a controversial and unreliable task for the users. Hence, we plan to investigate text processing and language technologies for the purpose of automatically extracting UNSO descriptions from the free-text natural language descriptions inserted by the users.

We also plan to exploit the collected domain meta-data for advanced functionalities aimed at improving users' interaction with the system. For example, we plan to consider the deployment of user modeling agents, which will collect information about users and their needs by analyzing the queries launched by them. In turn, availability of this information will facilitate the use of personalization agents, which will suggest query modifications (e.g., a widely-used domain attribute, or another value) and notify the users about recently inserted data objects that would match their needs.

References

[1] Adali, S., Candan, K., Papakonstantinou, Y., Subrahmanian, V.: Query Caching and Optimization in Distributed Mediator Systems. In: Proceedings of the SIGMOD Conference, Montreal (1996)

[2] Ben-Asher, Y., Berkovsky, S.: Management of Unspecified Semi-Structured Data in a Multi-Agent Environment. In: Proceedings of the SAC Conference, Dijon (2006)

[3] Bernstein, P.A., Melnik, S.: Meta Data Management. In: Proceedings of the ICDE Conference, Boston (2004)

[4] Busetta, P., Dona, A., Nori, M.: Channeled Multicast for Group Communications. In: Proceedings of the AAMAS Conference, Bologna (2002)

[5] Busetta, P., Merzi, M., Rossi, S., Legras, F.: Intra-Role Coordination Using Group Communication: A Preliminary Report. In: Proceedings of the Workshop on Agent Communication Languages and Conversation Policies, Melbourne (2003)

[6] Dean, M., Connolly, D., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D., Patel-Schneider, P., Stein, L.: OWL Web Ontology Language. W3 Consortium (2002)

[7] Fellbaum, C.: WordNet - An Electronic Lexical Database. MIT Press, Cambridge (1998)

[8] Gal, A., Modica, G., Jamil, H.M., Eyal, A.: Automatic Ontology Matching Using Application Semantics. AI Magazine 26(1), 21–31 (2005)

[9] Gruber, T.R.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition Journal 6(2), 199–220 (1993)

[10] Witten, I.H., Moffat, A., Bell, T.C.: Managing Gigabytes: Compressing and Indexing Documents and Images. Morgan Kaufmann Publishers, San Francisco (1999)