Reconciling Ontological Differences for Intelligent Agents

Kendall Lister & Leon Sterling

Intelligent Agent Lab
Department of Computer Science and Software Engineering
The University of Melbourne
Victoria, 3010, Australia
{krl,leon}@cs.mu.oz.au

Abstract

This discussion presents several alternative strategies for approaching the problem of semantic interoperability, based on recent projects to develop software agents and systems that attempt to reconcile ontological differences without explicit ontologies. The difficulties of reconciling explicit ontologies are discussed, and possibilities for meaning negotiation through implicit approaches are presented. To ground these ideas, two prototype systems are introduced that approach the issue of meaning negotiation without requiring the construction of formal ontologies.

Introduction

Developing intelligent software agents is a costly and time-consuming exercise. One significant contributing factor to the magnitude of the effort required is the need to conform to prescribed standards for knowledge representation, agent communication languages and ontologies. Such conformity is necessary because without agreement about what may be said and how to say it, communication becomes impossible and the numerous benefits of heterogeneity in multi-agent systems are lost. Two of the most common problems that arise when juxtaposing information from heterogeneous sources are synonymy and polysemy, in which data, labels and markers (in semiotic terms, signs in general) either have the same syntactic representation but have different meanings, or have different representations but equivalent meaning [2]. An example of synonymy is that in the context of advertising products for sale, the keywords "cost" and "price" will generally be semantically equivalent. As labels, any data to which they refer will tend to be directly comparable. This realisation is necessary for machines to process such data without human intervention, which is the primary goal of current efforts such as the development of a semantic web. The human and cultural factors that lead to this confusion of meaning are discussed in detail by [6]. The converse of this phenomenon, polysemy (e.g. "minute", a unit of time, and "minute", an adjective for very small) is likely to be less common but still present and solutions for reconciliation should deal with both sides of the problem, as do the approaches discussed later in this paper.

Formal ontologies are often introduced to represent an explicit statement of the meaning of the data communicated by an information agent or source. However, in a heterogenous environment even such explicit representations meet many problems when they are combined, as discussed in detail by [5, 7].

To date, global shared ontologies (that is, all participants in a system agree to use a common ontology or design) have become the standard solution to the problem of ontological differences, and they appear to work well in practice for small multi-agent systems [13, 14]. However, their success depends on several characteristics of the agent system for which they are constructed: the system must be relatively small and simple, the ontology must not change, the agents must not change, the life of the system must be short, the context of the system must be static. Obviously some of these restrictions are inter-related, and not all will apply to every system, but all place heavy limitations on the long-term, large-scale viability of software agents in open information systems.

A major consequence of the need for an *a priori* agreed global shared ontology is that the ontology for a system must be designed before the system is implemented. This is necessary because each agent must be equipped with the ontology before they can be initiated into the system. The process of distributing the ontology to all inhabitants of the system is complicated by the size of the system, including the number of agents in it, as well as the complexity of the communication infrastructure within the system. Inevitably, there is a point at which the difficulty of synchronising all agents across a large system to use the same ontology becomes too great. This is not just a technical problem; it is an organisational one as well. The development of the different software agents must be co-ordinated in line with the development of the ontology. As the number of groups or companies involved in the development increases, the administration required to align their efforts and enforce adherence to the standard ontology becomes overwhelming. A good analogy is the effort of the W3

Consortium to develop the HTML standard. As each new version of the standard was released browsers and servers had to be re-written to cope with the changes; additionally, there was no effective way to enforce compliance when the individual browser developers decided to deviate from the published standard. For a number of years, most web sites that attempted to present anything more sophisticated than plain text with headings were almost certain to render correctly with only one browser, with the result that users were left confused and frustrated.

Predetermined global ontologies also require the prediction of appropriate assumptions and generalisations that will be acceptable to all participants in the system. Likewise, a predetermined ontology tends to inhibit attempts to interact at variable levels of abstraction (this is discussed further in [6]). Any ontology in a large, long-term system will require maintenance over time, and in an open system backwards compatibility is another requirement to ensure that older agents can continue to operate. These issues all complicate the use of global shared ontologies, and the remaining chapters in this paper present the results of work to avoid these difficulties.

Implicit Ontologies

An alternative to global shared ontologies that minimises these issues is to give agents in a system the ability to learn each other's ontologies. If two agents, or a third middle agent, can interpret each other's communication and deduce the differences between their ontologies, the disadvantages of needing to design the agents with identical ontologies can be avoided. Obviously, this is not a simple matter. Complete understanding of all situations would require human-like comprehension abilities. Even when information agents are equipped with explicit ontologies, many problems arise when such agents attempt to communicate as the inevitable differences of scope, context and detail in their ontologies must be resolved. (Efforts to merge and align formal ontologies are directly related to the work presented in this paper, and several prominent projects are discussed in the final chapter of this paper.) However, in practice a number of strategies can be used to reduce the complexity of the task of ontological reconciliation.

All individuals, organisations and systems have their own ontology. For individuals, their ontology is based on their experiences and their beliefs; for organisations, on their established business practices; and for systems, on their design details. A software application can be considered as having an implicit ontology in the form of the classes and relationships from which it was constructed. In fact, some ontology development tools are designed to generate Java source code that represents the ontology; thus the ontology becomes an intrinsic part of the system. Similarly, an information source can been seen as having an implicit ontology that is defined by the labels it uses to present its data and the structure and representation it gives to that data.

The principle underlying the next section of this paper is that without formally specifying ontologies, the ontological beliefs encoded in information sources or the communications of software agents can be identified by analysing the way that similar objects are represented. If two communicating agents can be assumed to be expressing information of similar types, correspondences between data instances from each can be observed and semantic equivalences of classes recognised. This is significant in that it can allow interoperation without necessarily formalising the ontological opinions of the participating agents and constructing a translating map between their ontologies.

Automatic Reconciliation of XML Structures

A software prototype developed at the University of Melbourne to perform Automatic Reconciliation of XML Structures (AReXS) is able to analyse sample records from XML databases with different schema and deduce that fields with different names contain equivalent data [3, 6]. AReXS uses the previously developed Example-Based Frame Mapping (EBFM) algorithm [4] to identify similarities in the contents of data fields, and based on these similarities attempts to match fields across the databases. The EBFM algorithm relies on the fact that objects from different databases often contain almost identical contents, even when the field names and structures of the databases are quite different. This matching allows heterogeneous data sources to be reconciled and processed together. Using records from on-line book stores as sample data sources, AReXS was able to compare fields similar to `<TITLE>The Lord of the Rings</TITLE>` in one XML document and `<BOOK NAME>Lord of the Rings</BOOK NAME>` in another, and decide that the fields were semantically equivalent. Due to the EBFM algorithm used, the larger the sample size, the higher the confidence that can be placed in the resulting matches. AReXS considers each pairing of records from two heterogeneous sources and builds a table of hypotheses about potential matches between fields based on the contents of those fields across multiple records (refer to [3] for a detailed description of the actual formulae used by the AReXS algorithm).

The first phase of the reconciliation algorithm implemented in AReXS is to construct a set of frames from the input data. The data is entered in the form of first-order XML documents and converted to a frame representation in the form of slots and instances; the conversion consists of creating a frame for each data record, using each field tag as a slot name and the field data between tags as instances of the slots. This data is then used as input to a slightly customized implementation of the EFBM algorithm. Frame mapping in general involves identifying equivalent slot

pairs (ESPs) which can then be used to map between data source pairs. In the diagram below, the linking arrows indicate a mapping of slots from one source to another. It is worth noting that not all slots are mapped as some data sources are richer and cover wider ranges of information than others. Also, the slots that are mapped do not occur in the same order in both sources. Manual selection of slot mappings is a simple way to achieve semantic interoperability, but this is infeasible for the sorts of large-scale, open, dynamic and autonomous systems described in the earlier section of this paper. Since more and more systems are exhibiting these characteristics, automating the process of determining appropriate slot mappings is the underlying focus of the AReXS system.

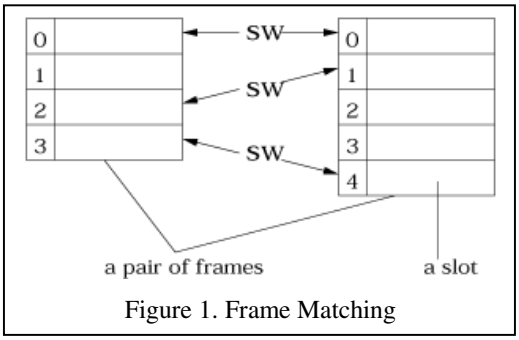

Figure 1. Frame Matching

However, the issues of synonymy and polysemy discussed earlier can manifest when differences occur in slot names. AReXS attempts to resolve these problems. In the diagram above, the labels SW represent a computed equivalence value of an ESP, or in other words the likelihood that a pair of slots are semantic equivalent. This differs from standard frame matching in that potential candidates for matching pairs of slots are computed without human intervention or external input.

Once the input data is in the form of frames and slots, AReXS then iterates over the data a sequence of operations that isolate ESPs that are consistently similar across the range of frames. The first step is to locate, for each slot, values that are relatively unique in the data source. The aim is to find data elements that can serve to uniquely identify a particular record (frame) from among all the other records (frames) in a source. This data element can then act as a primary key for the record of which it is a part. If the same unique data value can be found in a record from the other data source, then the unique occurrence of this particular data element gives a strong indication that the records described the same object. Thus, the two slots that contain this data element in records from different sources can be considered equivalent, even if there is no correspondence between their slot names. At this point the potential ESP is added to an accumulating set of hypotheses.

The uniqueness function used for these calculations is that provided by [4] in their presentation of the original EBFM algorithm. A deviation from the standard EBFM is worth noting at this point. Whereas, in its original form, the algorithm would discard hypotheses that are based on a single slot pair, we deliberately retain these hypotheses. At first, AReXS implemented the EBFM algorithm purely, but it was found during evaluation that hypotheses based on a single slot pair actually contribute valuable information to the reconciliation process, and so the algorithm was modified to retain them for consideration.

So far along the reconciliation process, AReXS has assembled a set of candidate pairings of slots to map between the two data sources. These hypotheses are then sorted by strength of correspondence, also taking into consideration the similarity of the other slots in each frame. The less promising hypotheses are then removed so that each instance from an information source is only referred to once.

The final step is to apply to each hypothesis a short series of pruning operations that identify the most promising ESPs by testing the hypotheses against other data records from each source. This prevents anomalous correspondences between slot values from poisoning the final frame mapping. Hypotheses that prove to be unreliable are discarded, leaving only the most effective mappings. These mappings are then compiled to produce a set of mappings between ESPs, which is the final output of the AReXS system.

Sample Results for Implicit Ontological Reconciliation

AReXS was tested on information sources that were manually constructed XML documents containing data obtained directly from on-line book stores such as borders.com, amazon.com and angusandrobertson.com.au. These XML documents were created using the labels from each web site as tags for the data presented under these labels. The following set of data is representative of the form and substance of the tests run on the AReXS system. At this stage AReXS has constructed a frame representation of the XML documents – it is not productive to include the entire XML source as well as it can be easily inferred from this output trace:

```
Source [0] ==> Borders.com

   {Slot 0: ISBN}
   {Slot 1: Title}
   {Slot 2: By}
   {Slot 3: Format}
   {Slot 4: Availability}
   {Slot 5: Our Price}

   <Instance: 0>
      0679764410
      American Sphinx
      Ellis, Joseph J.
      Trade Paperback, 440 Pages, Vintage Books, April 1998
      In stock - ships in 24 hours
      $13.50 - Save $1.50 (10%)
   <Instance: 1>
      0802713521
```

```
      E=mc2: A Biography of the World's Most Famous Equation
      Bodanis, David
      Hardcover, 352 Pages, Walker & Company, September 2000
      In stock - ships in 24 hours
      $20.00 - Save $5.00 (20%)
   <Instance: 2>
      0684870185
      When Pride Still Mattered: A Life of Vince Lombardi
      Maraniss, David
      Trade Paperback, 54 Pages, Reprint, Simon & Schuster Trade
         Paperbacks, September 2000
      In stock - ships in 24 hours
      $14.40 - Save $1.60 (10%)
   <Instance: 3>
      0140143459
      Liar's Poker: Rising Through the Wreckage on Wall Street
      Lewis, Michael
      Trade Paperback, 249 Pages, Viking Penguin, October 1990
      In stock - ships in 24 hours
      $12.60 - Save $1.40 (10%)
   <Instance: 4>
      0671042815
      Dream Catcher: A Memoir
      Salinger, Margaret A.
      Hardcover, 464 Pages, Pocket Books, September 2000
      In stock - ships in 24 hours
      22.36 - Save $5.59

Source [1] ==> Amazon.com

   {Slot 0: }
   {Slot 1: by}
   {Slot 2: Our Price}
   {Slot 3: Availability}
   {Slot 4: Category}
   {Slot 5: }
   {Slot 6: ISBN}

   <Instance: 0>
      American Sphinx : The Character of Thomas Jefferson
      Jospeh J. Eillis
      $12.00
      Usually ships within 24 hours
      Biographies & Memoirs
      Paperback - 440 pages Reprint edition (April 1998) Vintage
         Books
      0679764410
   <Instance: 1>
      Liar's Poker: Rising Through the Wreckage on Wall Street
      Michael Lewis
      $11.20
      Usually ships within 24 hours
      Business & Innvesting
      Paperback - 249 pages (October 1990) Penguin USA (Paper)
      0140143459
   <Instance: 2>
      E=mc2: A Biography of the World's Most Famous Equation
      David Bodanis
      $20.00
      Usually ships within 24 hours
      Science
      Hardcover - 337 pages 0 edition (September 2000) Walker & Co
      0802713521
   <Instance: 3>
      Dream Catcher: A Memoir
      Margaret A. Salinger
      $22.36
      Usually ships within 24 hours
      Biographies & Memoirs
      Hardcover - 436 pages (September 6, 2000) Washington Square
      0671042815
   <Instance: 4>
      When Pride Still Mattered: A Life of Vince Lombardi
      David Maraniss
      $12.80
      Usually ships within 24 hours
      Sports
      Paperback - 541 pages Reprint edition (September 2000)
         Touchstone Books
      0684870185
```

After applying the Example-Based Frame Mapping algorithm, AReXS reports the final candidate slot pairs and its confidence in each match:

```
**********
FRAME MAP:
**********

Number of runs : 2

Slot [0] <---> Slot [6]. Initial Weight : 0.7868673239238453,
Final Weight : 1.0
Slot [1] <---> Slot [0]. Initial Weight : 0.7868673239238453,
Final Weight : 0.9254817790881276
Slot [2] <---> Slot [1]. Initial Weight : 0.7868673239238453,
Final Weight : 0.7868673239238453
Slot [5] <---> Slot [2]. Initial Weight : 0.7868673239238453,
Final Weight : 0.7868673239238453
```

The results of automatically reconciling these two information sources are that the two fields labelled “ISBN” have unsurprisingly been matched with the highest possible degree of confidence. This is due to the fact that although AreXS is currently biased towards string data (as discussed below), the numbers in the ISBN fields are very precise and there is no room for corruption of variation – if two books have the same ISBN, there is only one correct way to present that data. By contrast, book titles or author names can vary in presentation, leading to less than perfect confidence even though the confidence is still high. Next, the field labelled “Title” in the first information source has been matched with an unnamed field in the second source. This could not be done by only considering the field names, as would generally occur in an explicit ontology alignment algorithm. Also, it is worth noting that AReXS is able to cope with blank data tags, although if frames from both sources contain blank tags the matching algorithm will quickly become confused. Finally, two more fields were matched based on correspondences in their contents. Although the two “ISBN” fields could have easily been matched based on their tag names, the EBFM approach confirms that these tag names are accurate by cross-checking the data content. Other data sourced from more on-line book stores further demonstrated AReXS’s ability to match slots with completely different labels. For example, where borders.com uses the label “By” to mark author information, angusandroberston.com.au uses “Author”; AReXS recognises this and maps between the two fields.

AReXS is able to handle textual differences in both field names and field contents, as the implementation of the EBFM algorithm is reversible. AReXS is inherently flexible, as if a matching record from one source is not immediately found, subsequent records can be assessed until a suitable candidate is discovered.

The flexibility of the EBFM algorithms also gives AReXS robustness to cope with incomplete records and missing fields. Because thresholds are employed when determining likely semantic matches, if one information source contains a field that appears not to correspond to any fields in the other source, AReXS will decide that this is a singleton field and not force a match.

To compare the contents of fields, the Character-Based Best Match algorithm [10] is used. This is more flexible than direct lexicographical comparison of data and so enables AReXS to cope with slight syntactic corruption of data. For example, given the input data `<TITLE>The Lord`

of the Rings</TITLE> and <BOOK NAME>Lord of the Rings</BOOK NAME>, it would be expected that a useful system be able to match the labels "title" and "book name". Even though the field contents are not identical, they are so close as to permit humans to make the assumption that they are equivalent. Part of the research required to advance this work is understanding what extra knowledge we have at our disposal that enables us to easily perceive such correspondences, as this knowledge is clearly not present in the raw data. In this case, AReXS is unconsciously deciding (according to its internal threshold levels) that the word "The" at the beginning of a data unit is not significant (although internally AReXS does not have any conceptualisation of a word, nor has it been taught anything about the English language). In its current implementation, AReXS is not so clever when presented with numerical data or field contents in other forms. Current work will overcome this by adding heuristics for recognising and handling other common types of data, although it seems quite sensible to keep such heuristics separate from the algorithms themselves to permit adaptability and reuse of the knowledge.

The primary significance of the AReXS implementation of EBFM is that semantic equivalences can be deduced from data that has no inherent semantic encoding. From flat XML databases AReXS is able to build a set of potential semantic links, effectively inferring the ontological knowledge of the designers of the databases even though such knowledge is not actually specified. Further work will aim to enhance these abilities by considering extensions such as resolving nested XML documents and composite fields. For example, it would be helpful to be able to extract multiple data units from within a single field, so that <SUBJECT>433-380 Graphics</SUBJECT> could be matched with <SUBJECT NAME>Graphics</SUBJECT NAME> and <SUBJECT CODE>433-380</SUBJECT CODE>. In both cases, the algorithms implemented in AReXS appear to be useful.

Knowledge Categories for Reuse and Interoperability

Another strategy for reducing the problem of ontological differences has been implemented in the Classified Advertisement Search Agent system [1, 12]. Also a prototype system, CASA was designed to evaluate the potential advantages of compartmentalising knowledge into different contexts. Similarly, CASA also separates the knowledge of an information agent from its architecture, easing the re-use and transmissions of knowledge between agents.

CASA considers knowledge in three different categories: general knowledge, domain knowledge and site or source specific knowledge. General knowledge is that which is true for all information sources, and gives an agent the ability to operate in its environment. For a web-based information agent such as CASA, general knowledge might include the components that make up a web document and how to retrieve web pages from the Internet. Domain specific knowledge prepares an agent for working with information from a particular area, such as car classified or real estate advertisements. Domain specific knowledge is true across all information sites that cover the same subject. Site specific knowledge is true only for a particular information source or site. For a web-based information agent, site specific knowledge might include the layout of a certain web site, the format of tables or data records it contains and markers that identify the locations of certain data. Once knowledge is separated into these different classifications, constructing and modifying information agents becomes simpler. For example, an agent that possesses domain specific knowledge can use that knowledge to learn how to use a new information source; agents can teach each other about new information sites by transmitting site specific knowledge that will make sense if they already share domain specific knowledge. From the other end, an information agent placed in a new environment requires only that its general knowledge be updated; for example, an agent designed for a corporate intranet could be released into the Internet with very little modification.

Related and future work

CASA and its underlying principle of categorisation of knowledge are significant to the problem of meaning negotiation because it provides a mechanism for distinguishing between content and representation. Much of the difficulty of semantic interoperability comes from the inconsistencies often employed in representing information. As discussed earlier, the polysemy and synonymy addressed by AReXS are contributors to the problem. Similarly, formatting and layout are also potential sources both of confusion and of meta-level knowledge. CASA goes some way toward addressing this. Current work at the Intelligent Agent Lab at the University of Melbourne aims to combine the technologies of both systems, creating an agent that can extract content from heterogenous information sources and then reconcile that content without requiring manual construction of ontology maps and translations. With the ability to automatically deduce information structure from an HTML source and extract knowledge units, the initial preparation of data that AReXS currently requires be done manually could be streamlined. Once the data is in a form suitable for conversion to the frame representation required by AReXS, automatic reconciliation can occur and agreed meanings of data can be negotiated.

The work done in developing and evaluating the AReXS and CASA systems also provides insight to answering questions of what types of knowledge should be pre-defined when developing an information agent, and what is best left to be learnt dynamically. AReXS relies on

information sources containing some structure that it uses to detect frame and slot boundaries. CASA, on the other hand, is much less dependent on the syntactic structure of the information sources as it attempts to identify their internal semantic structure. Considering the abilities of each system gives practical indications of what can be achieved with different classes of knowledge, and this can then inform future development of information agents. It is still unclear exactly what is the most beneficial and efficient combination of predefined knowledge and dynamically learnt knowledge, but current work will experiment to improve understanding of this issue.

The flexibility and complementary nature of both systems described in this paper opens up the possibility of other approaches to meaning negotiation. Currently, there exist a number of ontology reconciliation tools and methodologies that assist developers to align, merge and combine ontologies (for example, SMART [9], Chimaera [8], and Klein's methodology [5]). However, all these tools are at best semi-automatic, either by design or by necessity. Simple syntactic or linguistic matching of concept and class names can identify some common elements in the explicit ontologies being reconciled, and this is the most common technique used to reconcile ontologies. Similarly, analysis of the local structure of each ontology can sometimes reveal slightly more semantic equivalences, although this does not appear to be as well developed. Both AReXS and CASA were implemented without consideration of explicit ontologies, but the algorithms used in each also offer contributions to the problem of aligning explicit ontologies. If the ontologies that are to be reconciled include instances as well as basic concepts, the example-based approach to determining similarity should be able to reveal many equivalences that cannot be identified using only syntactic or linguistic comparisons. This approach will be explored in another ongoing project at the Intelligent Agent Lab, an extension to the functionality of tools such as Chimaera that will use the instances provided with each ontology to guide concept matching.

References

[1] Gao, X., Sterling, L. *Classified Advertisement Search Agent (CASA): A Knowledge-Based Information Agent for Searching Semi-Structured Text*, Department of Computer Science and Software Engineering, The University of Melbourne, Technical Report 98/1, 1998

[2] Hendler, J., Heflin, J. *Semantic Interoperability on the Web,* Proceedings of Extreme Markup Languages 2000, 2000

[3] Hou, D. *Automatic Reconciliation of XML Structures*, Honours thesis, Department of Computer Science and Software Engineering, The University of Melbourne, 2001

[4] Ikeda, Y., Itoh, F., Ueda, T. *Example-based frame mapping for heterogeneous information agents*, Proceedings of the International Conference on Multi-Agent Systems, IEEE Press, 1998

[5] Klein, M. *Combining and relating ontologies: an analysis of problems and solutions.* Asuncion Gomez-Perez, Michael Gruninger, Heiner Stuckenschmidt, and Michael Uschold (eds), *Workshop on Ontologies and Information Sharing, IJCAI'01*, Seattle, 2001

[6] Lister, K., Sterling, L. *Agents in a Multi-Cultural World: Towards Ontological Reconciliation*, Markus Stumptner, Dan Corbett and Mike Brooks (eds), AI 2001: Advances in Artificial Intelligence, Proceedings of the 14th Australian Joint Conference on Artificial Intelligence, Adelaide, 2001, pp 321-332

[7] McGuinness, D., Fikes, R., Rice, J., Wilder, S. *An Environment for Merging and Testing Large Ontologies,* Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000), Breckenridge, 2000

[8] McGuinness, D., Fikes, R., Rice, J., Wilder, S. *The Chimaera Ontology Environment,* Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI 2000), Austin, 2000

[9] Noy, N., Musen, M. *An Algorithm for Merging and Aligning Ontologies: Automation and Tool Support,* Proceedings of the Workshop on Ontology Management at the Sixteenth National Conference on Artificial Intelligence (AAAI'99). Orlando, 1999

[10] Sato, S. *CTM: An example-based translation aid*, Proceedings of the Fifteenth International Conference on Computational Linguistics, 1992

[11] Sterling, L. *A Knowledge-Biased Approach to Information Agents*, Proceedings of the International Workshop on Information Integration and Web-based Applications and Services (IIWAS'99), Yogyakarta, Indonesia, 1999

[12] Sterling, L. *On Finding Needles in WWW Haystacks*, Advanced Topics in AI, Proceedings of the 10th Australian Joint Conference on Artificial Intelligence, Abdul Sattar (ed), Springer-Verlag LNAI, Vol 1342, 1997

[13] Sycara, K. *Multi-agent Infrastructure, Agent Discovery, Middle Agents for Web Services and Interoperation*, Multi-Agent Systems and Applications, Proceedings of 9th ECCAI Advanced Course, ACAI 2001 and EASSS 2001

[14] Theodosiev, T. Multi-Agent System with Sociality, Proceedings of Adaptability and Embodiment Using Multi-Agent Systems, Prague, 2001