An Information Food Chain for Advanced Applications on the WWW

Stefan Decker, Jan Jannink, Prasenjit Mitra, Gio Wiederhold
Stanford University, CS Dept, CA 94306
{stefan, jan, prasen9, gio}@db.stanford.edu
Steffen Staab, Rudi Studer
University of Karlsruhe, Institut AIFB, 76128 Karlsruhe, Germany
{staab, studer}@aifb.uni-karlsruhe.de

Abstract
The Internet and especially the World Wide Web are growing at a tremendous rate. More and more information is becoming directly available for human consumption. But humans have a limited information processing capacity and are often not able to find and process the relevant information. Automation, in the form of processing agents that provide a powerful extension to human capabilities, is required. However, with the current technology it is difficult and expensive to build such automated agents and support human user with their information processing power, since agents are not able to understand the meaning of the natural language terms found on todays' webpages. The remedy this situation we look at different existing technologies and put them together to a new *information food chain* [Etzioni 1997] for agents, that enables advanced applications on the WWW.

1 Introduction

The Internet and especially the World Wide Web are growing at a tremendous rate. More and more information is becoming directly available for human consumption. But humans have a limited information processing capacity and are often not able to find and process the relevant information. Automation, in the form of processing *agents* [Genesereth 1994] that provide a powerful extension to human capabilities, is required. However, with the current technology it is difficult and expensive to build such automated agents that support human user with their information processing power, since agents are not able to understand the meaning of the natural language terms found on today's webpages.
Current software-agents extract information from webpages based on *wrappers* [Sahuguet 1999]. Wrapper are site-specific software modules that extract information based on the regular structure of webpages. But creating a wrapper requires human work. If the structure of the wrapped webpage is changed, the wrapper does not work anymore. So wrappers are expensive to built and difficult to maintain, and an automated agent can not access information before a wrapper is (manually) created. That means the agent can not just browse the web on its own. This explains the current very limited capabilities of automated agents searching the web, and why no general agents are available right now.
To really facilitate automated agents on the web, agent interpretable formal data is required. However, creating formal data about a particular domain is a high effort task, and it is not immediately clear what kind of data and tools should be created to support this task. This papers aims at clarifying the question of which data and tools are necessary for the creation and deployment of formal data on the web by presenting an *information food chain* [Etzioni 1997] for agents and formal data aiming at agents. Every part of the food chain provides information, which enables the existence of the next part. A part of the food chain is basically information on the web, which is used to create more advanced information.
That in contrast to other agent approaches we do not focus on inter-agent-communications approaches like KQML [Wiederhold 1990][Finin 1997]. Instead we investigated the infrastructure necessary to enable automated single agents on the Web, which we believed has to be created first, before inter-agent communications is necessary.
Our goal is not to provide a new technical solution to sub-problem, instead we aim at putting all pieces of current technology in the right place to enable automated agents.
In the next section we present an overview about the agent information food chain.

2 An Agents Information Food Chain: Overview

Representing information on the web requires a joint representation language for data on the web, such that the building of wrappers is no longer necessary. Given the current situation, this will be most likely XML based. However, XML itself is not suitable for this task (although the current hype around XML might suggest this) because the semantics common to all XML languages is just the parse tree of the documents, which is as useful as an HTML parse tree. If general XML is used, creation of wrappers becomes necessary again. So a specific XML-based language is necessary for formal agent communication.

For data exchange on the web it is also necessary to have a specification of the terminology of a particular domain, where the data is about. Raw data available on the web without an explanation about its *semantics* is useless (e.g. the tag <flighttime> from an aircargo carrier website might mean starting time of the flight or landing time).

Ontologies [Fridman Noy, 1997] are a means for Knowledge Sharing and Reuse and capture the semantics of a domain of interest. Ontologies are consensual and formal specifications of vocabularies used to describe a specific domain, and facilitate information interchange among information systems. Since there will be no ontology available describing all possible data, we need multiple ontologies for different application domains (DTDs as used e.g. at Biztalk (http://www.biztalk.com) can be regarded as a very limited kind of ontology. But DTDs describe the grammar (the structure) of a document, not the content).

However, ontologies itself are just formal data on the web, that needs to be exchanged. So it is desirable for the XML-based data exchange language to also be able to represent an ontology for a particular domain.

Example for a XML-based languages that allows explicitly ontology definition are e.g. RDF (Resource Description Framework) [Lassila 1999], RDF Schema [Brickley 1999], XOL (XML-based Ontology Exchange Language) [Karp 1999], or OML/CKML (Ontology Markup Language/ Conceptual Markup Language) [Kent 1999] or the recent DAML program (DARPA Agent Markup Language: http://www.darpa.mil/iso/ABC/BAA0007PIP.htm)

An XML based representation language and formal ontologies are one foundation for automated agents on the web. However, to deploy the formal data and to make the creation of the formal data manageable infrastructure (e.g. support- and deployment tools) - the *information food chain,* is necessary. A pictorial description of the information food chain is depicted in Figure 1.

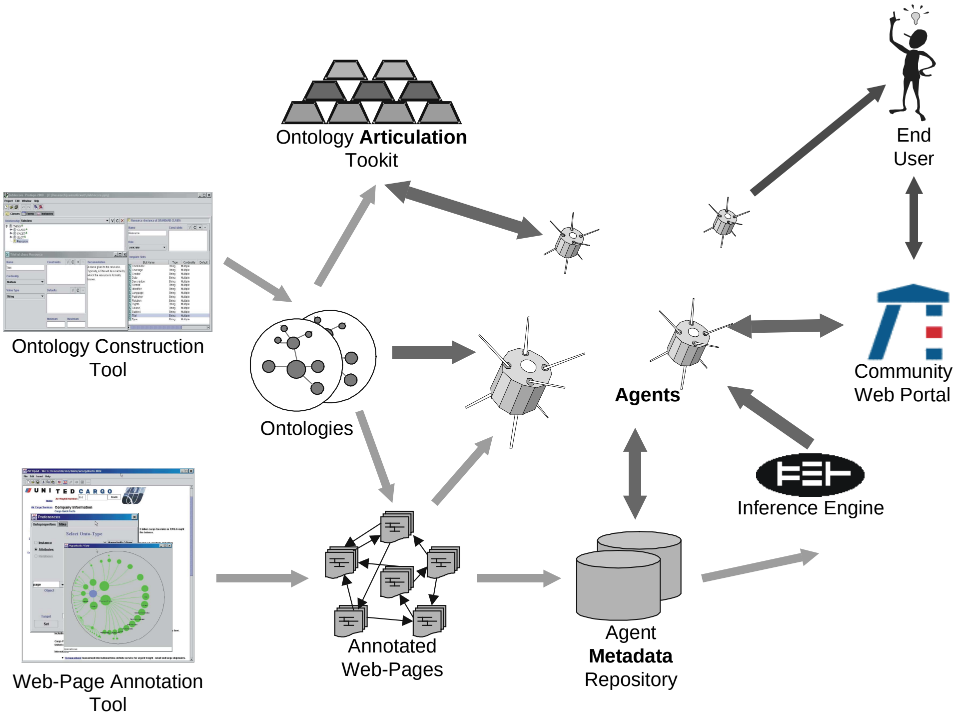

Figure 1: Agents Information Food Chain

Since the ontology is foundational for a set of data items, the food chain starts with the construction of an ontology, preferably with an support tool, an Ontology Construction Tool. An ontology is the "explicit specification of a conceptualization", which provides all the required terminology, and a basis for a community of interest for information exchange. The ontology defines the terms that are possible to use for annotation information in webpages, using the former mentioned XML-based representation language.

However, somehow the formal data formulated in terms of the ontology has to be created. We expect that much of the formal data will be generated from structured information sources, eg. Databases. But most of the available data on the web consists now of HTML pages [Lawrence 1999]. So the major task is to make this information available. The next part of the information food chain describes a tool to support this task:

A *Webpage Annotation Tool* has means to browse the ontology and to select appropriate terms of the ontology and map them to sections of a webpage. The webpage annotation process creates a set of annotated webpages, which are available to an automated agent to achieve its tasks. Of course the annotation process itself has a social component: although the effort for generating the annotation of a webpage is an order of a magnitude lower than the creation of the webpage itself, there has to be some incentive to spend the extra effort. The incentive for the creation of the annotation (which is metadata for the webpage) is *visibility on the web*, e.g. for a *Community Web Portal*, which presents a community of interest (distributed on the web) to the outside word in a concise manner. The data collected from the annotated webpages simplifies the task maintaining a Community Web Portal drastically, because changes can be incorporated automatically, without any manual work required.

An automated agent itself needs several sub-components: an important task of the agent is the integration from data from several distributed information sources. Because of the need to describe the relationships between the data in a declarative way (otherwise a new agent has to be programmed for every new task), an agent needs *Inference Engine* for the evaluation of rules and queries (see Figure 2 below for an example). The inference engine has to be coupled with a *metadata repository*, the memory of an agent where retrieved information is cached.

Furthermore if an automated agent browses the web it will usually encounter data formulated in unknown ontologies. Therefore it needs a facility to relate unknown ontologies to ontologies it already knows. This facility is an *Ontology Articulation Toolkit* for information mediation.

Given an overview about the agents information food chain, we will describe the most essential parts in greater detail in the next section.

3 Information Food Chain Components

3.1 OnTo-Agents Ontology Construction Tool

Ontologies are engineering artifacts, and constructing an ontology involves human interaction. Thus means are necessary to keep the costs for ontology creation as low as possible. Therefore an Ontology Construction tool is necessary to provide means to construct ontologies in a cost-effective manner.

Because ontologies evolve and change over time (as our knowledge, needs, and capabilities change), reducing acquisition and maintenance cost of ontologies is an important task. An examples of ontology editors are e.g. the Protégé framework for customized knowledge-base system construction [Grosso et al., 1999] or the WebOnto-Framework [Domingue 1998]. WebOnto is aiming at the distributed development of ontologies. WebOnto supports collaborative browsing, creation and editing of ontologies by providing a direct manipulation interface that displays ontological expressions. However, none of those tool is yet used for creating ontologies for web based agent applications.

3.2 Webpage Annotation Tool

Creating formal data on the web is a costly process. Hence, an important goal of an information food chain is to reduce cost of ontology-based data creation as much as possible. The annotation process requires tools that support user in the creation of this kind of data. The biggest source of information on the web today are HTML pages. The creation of explicit ontology-based data and metadata for HTML-pages requires much effort if done naively. Most HTML pages we find on the web are created using a visual HTML editor, so that annotators often already know how to use these tools. However, HTML-Editors focus on the *presentation* of information, whereas we need support for the creation of ontology-based metadata to describe the content of information. Thus the information food chain needs a practical, easy-to-use, ontology based, semantic annotation tool for HTML pages.

A basic tool, Ontopad, was already created in the project Ontobroker [Decker, 1999][Fensel 1999][Benjamins 1999]. Ontopad is an extension of a Java-based HTML editor, which allows normal browsing and editing of the HTML page, and supports the annotation of the HTML-page with ontology-based metadata. The annotator can select a portion of the text from a webpage and choose to add a semantic annotation, which is inserted into the HTML source. However, for significant annotation tasks a basic annotation tool is not sufficient. It still takes a long time to annotate large pages, although a significant improvement was reported when compared to the manual task.

So a practical tool should also exploit information extraction techniques for semi-automatic metadata creation. The precision of linguistic processing technology is far from perfect and reasonably exact automatic annotation is not possible. However, there exists currently much linguistic processing technology that is highly appropriate to help users with their annotation tasks [Alembic 1999][Day 1997][Neumann 1997]. Often it is sufficient to give the user a choice to annotate a phrase with one of two or three ontological expressions. Resources like WordNet [Fellbaum 1998] and results obtained from the Scalable Knowledge Composition (SKC) project [Mitra 2000] to provide high-level background knowledge to guide the annotation support.

Often a collection of similar pages has to be semantically annotated. Here document templates with built-in annotations simplify the creation of collections of annotated documents considerably. Document templates enable the reuse of annotations: a document template has to be only created once.

We can also benefit from the intensive, ongoing efforts in the W3C Extended Mark-up Language (XML) developments. XML promotes the creation of semantic markup. With the definition of a mapping between XML-elements and the concepts of an ontology [Erdmann 1999], XML-documents can be treated as formal information sources

3.3 Ontology-Articulation Toolkit

In order to solve a task that involves consulting multiple information sources that have separate ontologies, an automated Agent needs to bridge the semantic gap between the several ontologies we finds on the web.

In our Scalable Knowledge Composition Project (SKC [Mitra 2000]) we have developed tools that support an expert with the process of defining rules that link different ontologies, i.e., ontologies from different sources that differ in their terminology and semantic relationships. These rules define a new *articulation* ontology, which serve the application, and translates terms that are involved in the intersection to those in the source domain. Typical disjoint source domains with an important articulation in a logistics example would be trucking and air-cargo. Information produced within the source domain in response to a task can then be reported up to the application using the inverse process. Now the results are translated to a consistent terminology, so that the application can produce an integrated result. The major advantage of the SKC approach is that not all of the terms in the source ontologies have to be aligned, i.e., be made globally consistent. Aligning just two ontologies completely requires a major effort for a practical application, as well as ongoing maintenance. For instance, it is unlikely that United Airlines and British Airways will want to agree on a fully common ontology, but a logistics expert can easily link the terms that have matching semantics, or define rules that resolve partial match problems. Partial matches occur when terms differ in scope, say that one airline includes unaccompanied luggage in its definition of cargo, and another does not. Partial match is dealt with by having rules in the articulation that indicate which portions of two partially matching concepts belonging to different ontologies are semantically similar and can thus be combined. Articulation also provides scalability. Since there are hundreds of domains just in the logistics area, and different applications need to use them in various combinations, the global-consistency approach would require that all domains that can interact at some time must be made consistent. No single application can take that responsibility, and it is unlikely that even any government can mandate national semantic consistency (the French might try).

However, an automated agent can also come across a new domain, with its own ontology about which it has no prior knowledge. The customer still needs information to be extracted from the new domain. This is a hard problem because the agent does not possess any prior information about the semantics of the terms and the relationships in the new information source.

To address this problem methods for the generation of dynamic articulations are necessary, on an as-needed basis. This will allow the creation of working articulation ontologies or linkages among the source ontologies. If the end-user has the expertise to perform the articulation, then no specialist expert who understands the linkages among the domains is needed. For instance, any frequent traveler today can deal with most domain differences among airlines.

3.4 Agents Inference System

Combining information from different source enables new applications and exploits available information as much as possible. This requires inference techniques to enable the ontology language to relate different pieces of information to each other. An agents inference system provides the means for an effective and efficient reasoning for agents. We expect that this technology will enable agents to reason with distributed available knowledge.

Metadata on the Web is usually distributed and added value can be generated by combining metadata from several metadata offerings: for example, in the transport domain**Fehler! Verweisquelle konnte nicht gefunden werden.** one metadata offerer might state that a flight is available from Washingon D.C. to New York.

Another statement from another server states the availability of a flight from New York to Frankfurt. Combining these two pieces of information an agents inference system can deduce that a connection is available from Washington DC to Frankfurt, although this it not explicitly stated anywhere. Please note that even such simple rules as used in Figure 2 are neither evaluable by commercial databases (because of their inability to perform recursion)

nor by simple PROLOG systems (because of potential infinite looping). Thus either fundamental type of system is not usable for this task.
Inference engines also reduce metadata creation and maintenance cost. Having to state every single assertion of metadata explicitly leads to a large metadata creation and maintenance overhead. So tools and techniques are necessary that help to reduce the amount of *explictly* stated metadata by infering further *implicit* metadata [Decker 1998]. Implicit metadata can be derived from already known metadata by using general background knowledge, as stated in Ontologies.
To exploit axioms defined in ontologies for available metadata, an automated agent needs to have reasoning capabilities. The properties of these reasoning capabilities have to be carefully chosen. A reasoning mechanism, which is too powerful, has intractable computational properties, whereas a too limited approach does not enable the full potential of inference. Deductive database techniques have proven to be a good compromise between these tradeoffs.

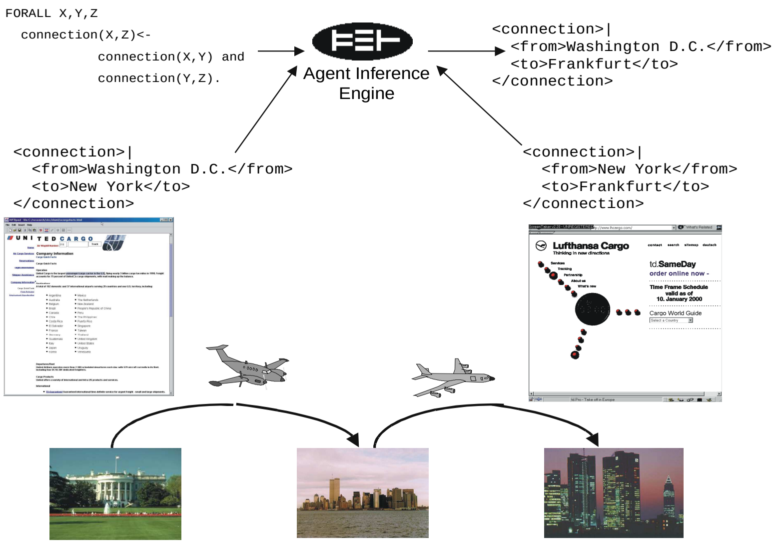

Figure 2 Inference Enabling Formal Markup Deployment

3.5 Semantic Community Portal Site

From the very beginning communities of interest have formed on the web that covered topics that they deemed to be of interest to their group. These users create what is now commonly known as *community web portals* [Staab 2000]. Community web portals are hierarchical structured similar to Yahoo! and require a collaborative process with limited resources (manpower, money) for maintaining and editing the portal. Since a number of people are involved, update and maintenance is a major effort. Strangely enough, technology for supporting communities of interest has not quite kept up with the complexity of the tasks of managing community web portals.

Recent research has demonstrated that authoring, as well as reading and understanding of web-sites, profits from conceptual models underlying document structures [Fröhlich 1998][Kesseler 1995]. The ontology underlying a community of interest provides such a conceptual model, since the ontology formally represents common knowledge and interests that people share within their community. It is used to support the major tasks of a portal, namely. *accessing* the portal through manifold, dynamic, and conceptually plausible views onto the information of interest in a particular community, and *providing* information in a number of ways that reflect different types application requirements posed by the individuals.

4 Related Work and Conclusion

In the Ontobroker [Decker 1999] and SHOE project [Henflin 1998] means were investigated to annotate webpages with ontology-based metadata. - thus realizing part of the food chain. Agent based architectures usually focus on

inter- agent communication instead to ontology creation and deployment (see [Nwana 1999] for a complaint about neglecting of the ontology problem).
We presented an information food chain, that delivers an infrastructure that empowers intelligent agents on the web and deploys applications will facilitate automation of information processing on the web. Fundamental to that approach is the use of a formal markup language for annotation of web resources.
We expect this information infrastructure to be the basis for the "Semantic Web" idea - that the World-Wide-Web (WWW) will become more than a collection of linked HTML-pages for human perusal, but will be the source of formal knowledge that can be exploited by automated agents. Without automation, and precision of operations, business and governmental uses of the information will remain limited.

Bibliography

[**Alembic 1999**] http://www.mitre.org/technology/alembic-workbench/

[**Benjamins 1999**] V R. Benjamins, D. Fensel, S. Decker, and A. Gomez-Perez: (KA)2: Building Ontologies for the Internet: a Mid Term Report. *International Journal of Human-Computer Studies (IJHCS)*,51, 687--612, 1999.

[**Brickley 1999**] D. Brickley, R. Guha: Resource Description Framework (RDF) Schema Specification
W3C Proposed Recommendation 03 March 1999, http://www.w3.org/TR/PR-rdf-schema/

[**Day 1997**] D. Day, J. Aberdeen, L. Hirschman, R. Kozierok, P. Robinson, and M. Vilain: Mixed-Initiative Development of Language Processing Systems. Appeared in: *Fifth Conference on Applied Natural Language Processing*, 1997, Association for Computational Linguistics, 31 March - 3 April, Washington D.C., U. S. A.

[**Decker 1998**] S Decker, D. Brickley, J. Saarela, and J. Angele: A Query and Inference Service for RDF. In: *Proceedings of the W3C Query Languages Workshop (QL'98)*, http://www.w3.org/TandS/QL/QL98/pp.html, 1998.

[**Decker 1999**] S. Decker, M. Erdmann, D. Fensel, and Rudi Studer: Ontobroker: Ontology Based Access to Distributed and Semi-Structured Unformation. In*: Database Semantics - Semantic Issues in Multimedia Systems, IFIP TC2/WG2.6 Eighth Working Conference on Database Semantics (DS-8)*. Eds: R. Meersman et. all Kluwer, 351-369, 1999.

[**Domingue 1998**] J. Domingue: Tadzebao and webonto: Discussing, browsing, and editing ontologies on the web. In*: Proceedings of the Eleventh Workshop on Knowledge Acquisition, Modeling and Management, KAW'98*, Banff, Canada, 1998, http://ksi.cpsc.ucalgary.ca/KAW/KAW98/domingue/

[**Erdmann 1999**] Michael Erdmann, Rudi Studer: Ontologies as Conceptual Models for XML Documents. In: *Proceedings of the 12th Workshop on Knowledge Acquisition, Modeling and Management (KAW'99)*, Banff, Canada, October 1999.

[**Etzioni 1997**] O. Etzioni: Moving Up the Information Food Chain: Deploying Softbots on the World Wide Web, *AI Magazine*, 18(2): Spring 1997, 11-18.

[**Fellbaum 1998**] Christiane Fellbaum (ed): Wordnet: An Electronic Lexical Database; MIT Press, 1998, ISBN 0-262-06197-X, available on the web as http://www.cogsci.princeton.edu/~wn, 1998.

[**Fensel 1999**] D. Fensel, J. Angele, S. Decker, M. Erdmann, H.-P. Schnurr, S. Staab, R. Studer, and Andreas Witt: On2broker: Semantic-based access to information sources at the WWW. In*: Proceedings of Webnet'99*, Eds.: P. de Bra and J. Leggett, AACE, 1065-1070, 1999.

[**Finin 1997**] T. Finin, Y.D. Labrou, KQML as an Agent Communication Language, In *Software Agents*. Bradshaw, J.M. (ed.), MIT Press, Cambridge, MA, pp. 297-316, 1997.

[Fridman Noy, 1997] N. Fridman Noy and C. D. Hafner: The State of the Art in Ontology Design, *AI Magazine*, 18(3):53—74, 1997.

[**Fröhlich 1998]** P. Fröhlich, W. Neijdl, and M. Wolpers: KBS-Hyperbook - An Open Hyperbook System for Education. in: *10th World Conf. on Educational Media and Hypermedia (EDMEDIA'98)*, Freiburg, Germany, 1998.

[**Genesereth 1994**] M. R. Genesereth, S.P. Ketchpel: Software Agents. In: Communications of the ACM 37(7, July), 48-53, 1994.

[**Heflin 1998**] J. Heflin, J. Hendler, and S. Luke: Reading Between the Lines: Using SHOE to Discover Implicit Knowledge from the Web. In: *AAAI-98 Workshop on AI and Information Integration*, 1998.

[**Karp 1999**] Peter D. Karp, Vinay K. Chaudhri, and Jerome Thomere: XOL: An XML-Based Ontology Exchange Language. ftp://smi.stanford.edu/pub/bio-ontology/xol.doc

[**Kent 1999**] Robert E. Kent: Conceptual Knowledge Markup Language: The Central Core . In Proceedings of the 12 Workshop on Knowledge Acquisition, Modeling and Management, Banff, 1999. http://sern.ucalgary.ca/KSI/KAW/KAW99/papers.html

[**Kesseler 1995]** M. Kesseler: A Schema Based Approach to HTML Authoring. in: *Proceedings of the 4th Int. World Wide Web Conf. (WWW'4).* Boston, December 1995.

[**Lassila 1999**] O. Lassila, Ralph Swick: Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation 22 February 1999, http://www.w3.org/TR/REC-rdf-syntax/

[**Lawrence 1999**] S Lawrence, and C. L. Giles: Accessibility of information on the web. In: *Nature*. Vol. 400, July 1999, 107-109.

[**Mitra 2000**] P. Mitra, G. Wiederhold , and M. Kersten. A Graph-Oriented Model for Articulation of Ontology Interdependencies. In: *Proceedings Conference on Extending Database Technology 2000 (EDBT'2000)*, Konstanz, Germany, 2000.

[**Neumann 1997**] G. Neumann, R. Backofen, J. Baur, M. Becker, and C. Braun: Information Extraction Core System for Real World German Text Processing. Appeared in*: Fifth Conference on Applied Natural Language Processing*, 1997, Association for Computational Linguistics, 31 March -- 3 April, Washington, D.C., U. S. A.

[**Nwana 1999**] H. S. Nwana & D. T. Ndumu, A Perspective on Software Agents Research, In: *The Knowledge Engineering Review*, Vol 14, No 2, pp 1-18, 1999.

[**Sahuguet 1999**] A. Sahuguet and Fabien Azavant: Building light-weight wrappers for legacy Web data-sources using W4F. In: International Conference on Very Large Databases (VLDB99) 1999.

[**Staab 2000**] S. Staab, J. Angele, S. Decker, M. Erdmann, A. Hotho, A. Mädche, H-P. Schnurr, R. Studer, Y. Sure: Semantic Community Web Portals. To appear in*: WWW9 - Proceedings of the 9th International World Wide Web Conference*, Amsterdam, The Netherlands, May, 15-19, 2000, Elsevier. URL: http://www.aifb.uni-karlsruhe.de/WBS/ama/publications/WWW2000/

[**Wiederhold 1990**] G. Wiederhold and T. Finin: KQML, Partial report on a proposed knowledge acquistion language for intelligent applications; prepared subsequent to a DARPA, NSF, AFORS sponsored Workshop in Santa Barbara, Stanford CSD technical note, March 1990; revised January 1991.