University of Southampton

Electronics and Computer Science Department

Approaches to System Integration For Distributed Information Management

Samhaa El-Beltagy

Technical Report No MM98-7

October 1998

ISBN: 0854-326-812

1. Introduction

Information is vital resource to individuals and organisations. In 1995, a report by the Information and Interactive Services stated that more than 10,000 people sign up for on-line information services every day[EW95]. In a survey conducted by Georgia Tech Research Corporation, it was found that 86.03% of people browsing the Web do so for the purpose of gathering information, making this the number one activity of Web users[GTC97]. However, as information systems that are both distributed and heterogeneous in nature, continue to evolve rapidly, the problem of locating, integrating and organising relevant information continues to be a major one and still motivates much research. Surprisingly, such a problem is not only to be found within huge distributed systems such as the Web, but also within the information systems of large enterprises and organisations.

2. General Overview

A distinction has been made with respect to various sources of information dividing them into three different classes: structured, unstructured, and semistructured. Databases, whether relational or otherwise, are examples of structured information sources. Examples of unstructured information include text files and a large class of web pages. The Web in fact, is one of the largest repositories of unstructured information. Examples of semistructured information includes web pages with known fields of contents. The problem of attempting to find information, lends itself to significantly different approaches depending on whether the information source is structured or unstructured.

The most widely spread approach for handling unstructured information sources is that based on indexing and keyword searching. This approach is exemplified by the numerous search engines that can be found on the Web. Although, the different engines employ different techniques to build indices some of which are more intelligent than others, retrieving information from unstructured information still has many limitations. Users are often returned results which are irrelevant and often have to spend horrendous amounts of time sifting through a huge amount of pages returned[PAD98][GRGK97]. For the purpose of improving search results, adding metadata descriptors was suggested[C95]. However, current support for metadata does not solve the problem completely since it still fails to represent the semantic content of a given document.

Due to the limitations associated with indexing techniques, a number of novel approaches have emerged. Among these is the approach that uses the user's preference for documents to learn more about a query and attempt to refine and find more documents based on what it has learned[KB97][MZ97]. To enable complex queries and move away from simple keyword searching, another approach that entails adding semantic content to unstructured documents such as Web pages was proposed through the use of ontologies and languages such as KQML[LH97][LSRH97]. However, the format of unstructured documents is by nature uncontrolled and publishing documents in a system like the WWW is totally unrestrained. So, despite the usefulness of this approach and the improvement it yields in terms of results, it is unlikely to be adopted on a large scale in the near future.

Another consideration, which until recently was rather ignored, is that unstructured information sources may often contain multimedia information. Contents of multimedia components such as images, graphics, audio, video, etc. need specialised tools to classify and index them. Some approaches based on content based retrieval were suggested[SC97][MHH97]. However, such systems suffer from major limitations, because they depend on physical analysis of the media element such as the texture of an image rather than more meaningful information that the component really represents, and thus a lot of information conveyed by the element, is lost. To overcome some of these limitations another approach which attempts to understand the contents of a media element based on the contents of the document in which it is a part as well as its physical content, has been proposed[ARS98].

Approaches to structured information source integration and search, are significantly different than those of unstructured ones. Such sources do not yield themselves to the simple keyword search, and usually offer the users the ability to enter complex queries. Because of the diversity of information offered by the different sources and because each source must be queried in a certain way, the problem of entering a single query and obtaining results from multiple sources is complicated. Integration of such resources usually entails identification of relationships between data elements in each resource and using these relationships to formulate a unified schema or view for the integrated system. Allowing the dynamic addition and retraction of such resources, to a unified system, is a complex problem that continues to motivate research. In general, most approaches to integrating heterogeneous information resources relay on another external form of representation, usually based object models. Definitions of relationships between the various objects is usually defined on an abstract and global level rather than on the level of each resource. Each resource that wishes to make its services available, must implement a wrapper that is capable of mapping the resources' data to the global model.

A user interested in finding information, rarely cares whether the information is obtained from structured or unstructured. All he/she wants is all relevant information to a query he/she has entered. This means that another level of integration between the two sources of information is required in a seamless way. The objective of this document is to present some systems that have more or less, attempted to address this problem.

3. TSIMMIS: The Stanford-IBM Manager of Multiple Information Sources

TSIMMIS[GHI95] is a joint project between Stanford University and the IBM Almaden research Center. The system was designed to address problems relating to integration of information among heterogeneous resources. Specifically, the goal of the TSIMMIS project was to develop tools that assist system builders to rapidly integrate heterogeneous information sources that may include both structured and semi-structured data.

The underlying premise in TSIMMIS, is that it is possible to extract information from unstructured sources as well as structured ones and store it in objects. The approach followed within the TSIMMIS project is one that is built around clustering related information sources into one integrated system. For example, a integrated bibliographic system is used to link a library retrieval system, a relational database holding bibliographic records and a file system with unstructured bibliographic entries. A simple common object model is employed to semantically describe the various components in a given application domain. The model has been called OEM(object exchange model) by its developers[PGW95]. The model derives its expressive power by enforcing objects to have self describing labels or tags. A query language, OEM-QL, was developed to enable requests of OEM objects.

3.1. System Components and Architecture

TSIMMIS employs classifiers/extractors, wrappers/translators, mediators, and constraint managers to achieve system integration. The classifiers/extractors attempt to identify simple patterns in unstructured sources and then export this information to the entire TSIMMIS system through a wrapper. A wrapper within TSIMMIS is responsible for translating queries expressed in the common object model into requests understandable by the source on which it is defined, and then converting the results back to the common object model. One of the goals of this project was to automate the development of wrappers. To this end, a wrapper implementation toolkit was developed[HBG97]. The toolkit allows for the semi-automatic creation of wrappers through the use of predefined templates. Mediators are defined on top of wrappers. Mediators in TSIMMIS contain some knowledge, expressed in terms of rules, of which resources to forward a query to, and of how to process returned answers, etc. Finally, there are the constraint managers which attempt to ensure semantic consistency across integrated resources.

3.2. Querying the System

End users can access information either by writing applications that request OEM objects or by using a generic browsing tool that has been developed within the project. One of the browsing tools developed provides WWW access through Mosiac. The tool enables the user to enter a query through a menu or by explicitly writing it. One of the limitations of the GUI, is that it assumes that the user of the system is capable of understanding and entering SQL like expressions to formulate a query.

4. Lore

Lore[MAG97][MW97] is a system that was also developed at the University of Stanford as an extension to TSIMMIS. While the focus of the TSIMMIS project was on the development of tools to automate resource wrapping, the main goal of Lore was the construction of a repository for managing semistructured information. Lore's data object model is based on the OEM introduced in TSIMMIS. To support and allow users to update and retrieve data with no known structure, a language called Lorel was developed. The novelty in Lore is that it could be thought of as a DBMS with an ability to represent data as objects with dynamic structure as opposed to traditional data base systems that represent data in a static and predetermined manner specified by a schema.

4.1. System Components and Architecture

The architecture of the Lore system is based on two layers: the query compilation layer and the data engine layer. Each of these layers is made up of several components. The query compilation layer, is composed of a query parser, a query preprocessor, a query plan generator and a query optimiser. Basically, this layer receives a query, converts it to Lorel, optimises it and sends it on to the data engine layer. The two most important components of the data engine layer are the OEM object manager, and the external data manger. The OEM object manager, is responsible for managing the Lore object database. The Lore database accepts addition in one of two ways: a user can explicitly issue update statements to add objects or issue a load file feature where an OEM database could be used. The external data manager keeps track of external data sources. Within Lore, an external data source is defined as any resource such as a Web page, a database, or a program, that is capable of packaging its contents in OEM format. During query processing, if the execution engines recognises the need to fetch information from an external object, the information is fetched and cached in the Lore database until it expires.

4.2. Querying the System

Lore offers the user a Java based GUI by which he/she can browse through objects supported by the system. Through browsing the different objects, the user is given the capability of entering a query as well as customising aspects of the different objects that he/she wants to appear in the result. The GUI is based on a novel component called a data guide. The interface is very friendly, but might prove confusing in huge systems. Another feature that is supported by the Lore's user interface, is the provision of keyword search based on selection of categories and specifiers. Of all the systems reviewed, Lore and another system(Dioroma) are the only ones that offer this capability.

5. InfoSleuth

InfoSleuth[BBB96][NU97] is a major project that was launched in 1995 by the Microelectronics and Computer Technology Corporation(MCC) with the goal of improving existing technology for locating and retrieving information across distributed information sources including the Internet. The project builds on another MCC project called Carnot[HJK92]. Carnot was developed for the purpose of integrating information in heterogeneous, distributed, enterprise databases. The focus of the InfoSleuth project revolves around information advertising, information discovery and information fusion. Information advertising entails augmenting information providers with the capability of advertising their availability and the information they are capable of providing. Information discovery involves the use of agents that explore knowledge bases or other information sources, watching for new additions or retractions of obsolete information. The task of information fusion is assigned to intelligent agents that are capable of combining information from multiple sites to form an integrated or fused response to a user's query. It has been stated that InfoSleuth will accommodate multimedia queries and responses and that it will provide a natural language interface to information sources and knowledge bases.

5.1. System Components and Architecture

Within the InfoSleuth framework, each resource is assigned a resource agent(a wrapper) that handles queries to and from that agent. A number of agents are implemented within the system to achieve intelligent system integration. InfoSleuth employs KQML[FWW93] in conjunction with KIF for the purpose of achieving agent interoperability. Other applications employed by InfoSleuth include LDL++ which is a deductive database system, and CLIPS which is a tool the enables the construction of rule based and object based expert systems. The agents that have to exist within InfoSleuth are described briefly as follows:

- User Agent: assists the user in formulating a query using the system's domain ontologies. A user agent is persistent and maintains the user's context between browser sessions.
- Ontology agent: answers queries about ontologies used within the system
- Broker Agent: matches the user's queries to the different resource capabilities. It offers an interface to different resources that have to register with this agent and to advertise their capabilities using KQML. A deductive database is employed to allow the broker to perform rule based matching of capabilities to user requests.
- Resource agent: provides the interface between the different agents and a given resource. It is the entity that actually registers with the broker agent.
- Task execution agent: the agent responsible for co-ordinating the activities pertaining to answering a given query. The agent may choose to formulate subtasks from the original task of answering a specific query and assign these subtasks to other agents based on their capabilities. This agent is designed to handle dynamic, incomplete and uncertain knowledge. After receiving the results from the various resource agents, the task execution agent integrates them and returns them to the User Agent via streaming protocol.
- Monitor agent: Monitors agent interactions and provides an interface for their visualisation

5.2. Querying the System

InfoSleuth provides a Java based GUI. A user agent handles user requests through Java applets, routing these requests to appropriate server agents and passing responses back to the user. A user agent is persistent and autonomous so it is able to maintain the user's context beyond a browser session. It stores data and queries for the user and can act as a resource for other agents. A user agent is implemented as a standalone Java application[JS96].

6. Garlic

Garlic[CHS95][RS97] is a system developed by IBM which provides an integrated view to a number of legacy data sources. Garlic was built with specific emphasis on large-scale multimedia information systems. Typically, Garlic resources include relational and non relational databases as well as document managers, image managers and video servers. Like most systems that address the issue of integrating various resources, Garlic relies on wrappers to provide an interface to the outside world. A wrapper within Garlic translates between Garlic's internal protocols and a resource's native protocols.

6.1. System Components and Architecture

To achieve system integration, Garlic employs a unified schema. Like TSIMMIS, Garlic adopts an object representation model. Garlic's model is based on the Object Database Management Group (ODMG) standard[CF94]. The system maintains a global metadata repository that serves as a description of the unified schema. Garlic's metadata repository does not support information on the query processing capabilities of the different resources. The main components of Garlic are:

- A metadata repository which is used to store the unified schema
- A complex object repository that contains definitions of relations between objects in different repositories.
- A query service and runtime system: provides query processing and data manipulation functionality. It also provides a unified view of objects within the Garlic Database.

A special language, the Garlic definition language (GDL), is used to describe the behaviour of the different objects within a resource. Each Garlic object has an interface that defines its behaviour on an abstract level and an implementation that provides the functionality given by the interface. One of the most important functions of wrappers within Garlic, is converting data contained in its underlying information source into Garlic objects. When a wrapper registers with the system it provides a description of its resource using GDL. This description is then merged into the global schema.

6.2. Querying the System

Plans for allowing the user to enter a query, involve an elaborate GUI, by which users could interactively expand or refine their queries. The GUI also allows the users to browse different Garlic objects.

7. Infomaster

Infomaster[GGKS95] is a system that was developed at Stanford University. As a "virtual information system", it enables users to access a variety of heterogeneous and distributed information sources where each information source handles the queries over the data it stores through the use of wrappers.

7.1. System Components and Architecture

To access information stored in a database or knowledge base, Infomaster relies on an agent communication language(ACL) consisting of KQML, KIF, and a number of ontologies. It provides a WWW user interface that can be used to enter queries using menus, SQL or ACL. Regardless of how the query is entered, it is converted to ACL and passed to a facilitator. Infomaster facilitators have a similar functionality to brokers implemented in InfoSleuth. Each facilitator may specialise in a particular domain. The facilitator then calls on the resources of other agents and/or other facilitators it knows about. Virtual facilitators route requests to specialised facilitators based on their knowledge of what domain each facilitator covers. The system also contains a specialised HTTP daemon implemented in LISP, to speed up communications. The HTTP daemon implements the functionality of an ACL converter and as well as the functionality of a facilitator.

Within Infomaster, information sources notify one or more facilitators of their willingness to respond to requests. Since information resources may vary in the way in which they represent, query, describe and present their information, for the purpose of integration, queries as well as responses need to be translated appropriately. Information sources are allowed to integrate with Infomaster in a number of ways. First, an information source can be entirely represented in KIF. An agent that loads the KIF knowledge base then simply becomes an information source. The second method involves writing an agent to act as a wrapper between the facilitator and another information source. Building an Ontology for use within Infomaster is simplified by the fact that groups of agents are implemented for use in pre-defined domains with known relationships. For instance, the example presented as a demonstrator for the Infomaster architecture is that for rental housing. Information is collected for heterogeneous resources, but what is being searched for is known before hand.

7.2. Querying the System

A user is allowed to enter a query using a rather simple interface based on HTML links and forms where the user is more or less guided into filling fields for pre-defined queries.

8. Dioroma

Dioroma[LLP97][LP96] is a system being developed at the University of Alberta, Canada. The goal of the project is to develop a methodology and to implement tools for the intelligent integration and access of heterogeneous information sources in large-scale and rapidly growing enterprise-wide networking environments. Dioroma's scope of information sources covers structured data repositories, semistructured data and unstructured data.

8.1. System Components and Architecture

The Dioroma system consists of several components that extract properties from unstructured data or collect data through other information brokers/mediators, and dynamically convert and assemble gathered information into DIOM objects. DIOM stands for Distributed Interoperable Object Model, which is developed as an extension to ODMG object model[CF94]. The system is based on the concept of information producers and information consumers. Mediators with the system are application specific and are represented in terms of a DIOM interface definition language (DIOM IDL). The main function of the mediator is to use metadata provided by information producers and consumers to process a query. Queries are represented using an extension to SQL, called the DIOM Interface Query Language (IQL). The main components of DIOM are:

- DIOM Interface Manager: the component which interacts with the user and allows him/her to enter a query.
- Distributed Query Mediation Service Provider: responsible for processing a query, selecting relevant resources, decomposing queries into subqueries and assembling returned results.
- Runtime Supervisor: executes subqueries by communicating with wrappers.
- Information Source Catalog Manager: manages both the information source repository metadata and the interface repository metadata.
- Implementation Repository Manager: maintains the correspondence between the source information and their DIOM internal object representation.

8.2. Querying the System

To provide a Web interface, a combination of HTML and Perl scripts is used. A user can enter a query as a simple keyword, or he/she can enter it based on DIOM IQL. During the processing of long queries, users are presented with intermediate results. The user interface provides the user with facilities for adding or updating existing resources. In the case of adding a new resource, the type of the resource must be chosen from a set of predefined categories. The URL of the resource as well as the keywords that characterise that resource must also be entered.

9. HERMES

In an attempt to develop a principled methodology for integrating various data sources, HERMES (a heterogeneous Reasoning and Mediator System)[SAA95] was initiated. HERMES is a project that originated at the University of Maryland. Developers of HERMES distinguished between two levels of integration: Domain and Semantic. While domain integration refers to the physical linking of the various data sources and activities related to adding new sources to a mediated systems, semantic integration refers to the extraction and combination of information obtained from the different data sources in a meaningful way. Domain integration requires knowledge of the nature of the sources to be included and their dependencies so as to be able to integrate them into the mediated system. Unlike other mediated systems, within HERMES, there is a clear distinction between the two levels of integration and their implementations into a mediator. HERMES draws from the theory of Hybrid Knowledge Bases. To achieve semantic integration within HERMES, rules represented by a logic-based declarative language, were employed. Currently, the system has two versions, one that runs under Dos/Windows and another that runs under UNIX. The PC version has been used to integrate systems with data stored in text files, pictures in GIF format, databases developed under Borland's Paradox, and DBase V DBMS as well as spatial data. The UNIX version integrates that same set of sources except for the databases. Databases supported under UNIX are INGRES andObjectStore.

9.1. System Components and Architecture

HERMES employs a number of mediators each of which is capable of handling a cluster of related information. HERMES also employs a facilitator which implements a Yellow page service. Unlike other agent systems, this service is not provided for use by other agents, but rather to assist mediator developers in finding and integrating resources.

9.2. *Querying the System*

HERMES provides a number of interfaces through which the user can query the system. Some of the interfaces are platform dependant, but one is also provided for Web users. The Web interface is based on CGI scripts. Initially, the user is presented with a list of all mediators and domains to choose from. The user is then presented with other HTML pages that guide him/her through the query.

10. SIMS

SIMS[ACHK93][KA97]is a system that was initiated and developed at the University of Southern California. Over the years the SIMS system has significantly evolved and currently employs agents for the purpose of retrieving information from multiple resources. SIMS relies on the LOOM knowledge representation language for integrating the different resources. KQML[FWW93] is used to achieve agent interoperability.

10.1. *System Components and Architecture*

Within SIMS, each resource is considered an information agent once a wrapper has been built around it. Some of agents within the architecture are task specific. These are viewed in terms of the tasks they perform. In order to accomplish their tasks, these agent might only integrate with portions of the ontologies that are important to the accomplishment of their tasks. A SIMS agent contains a detailed domain model representing its “expertise” and information resources available to it. Upon receiving a query, an agent attempts to identify sources that are capable of answering that query, generates a query plan accordingly and then executes that plan by sending it to the various resources. Each agent is also augmented with learning capabilities which enables it to formulate rules about other agents and how best to address them. The architecture also employs a cache to store information that is either expensive to retrieve or which is requested frequently.

10.2. *Querying the System*

Queries in SIMS are entered in the form of a class or object description for which related information is needed expressed in LOOM. The user is assumed to be familiar with the terminology used within the domain for which a query is directed. As an extra aid, the user is provided a utility for browsing a domain model.

11. Other Systems

This section introduces some systems that although relevant to the problem of system integration are more or less specialised in either a specific domain or focused on a specific aspect of system integration.

11.1. *SIGAL*

SIGAL[MM97] was developed at Laval University to provide an interoperable environment for geo-referenced data stored in a number of distributed and heterogeneous georeferenced digital libraries. The system employs software agents as front-ends to existing systems in order to enable interoperability. In order to provide for a common terminological basis and hence resolve knowledge disparities, the concept of an ontology is employed. An ontology is constructed using meta-data which allows the specification of data structures, domain values and functional and semantic interpretations for each goe-referenced digital library. To enable co-operation between the various software agents, agents are organised into teams. Within this work, a software agent oriented framework is defined as an entity that offers a set of services for use by users or other frameworks where a framework environment is composed of a framework supervisor and one or more software agents teams. The teams are composed of agents selected from a bank of software agents. Teams of agents are structured according to their responsibilities. When a service is invoked, four steps known as a realisation scenario are followed to carry out the request . The first step determines the user’s needs by matching his/her query with ontology concepts. The second step identifies the GDLs to be queried accordingly. The third step involves the processing of these request. The last step provides the user with the results.

11.2. The Information Manifold

The Information Manifold(IM)[LRO96a][LRO96b] is a knowledge based information retrieval system developed at AT&T laboratories. The primary aim of the project was to address the problem of query optimisation through the use of high level knowledge representation techniques to model data in structured repositories. Description logic was used to model relations within information source. This representation of the resources' content is used determine relevant sources to consult and formulates a query plan accordingly.

11.3. KRAFT

Kraft[GCE98] is a project being carried out at the Universities of Aberdeen, Cardiff, and Liverpool and funded by EPSRC and BT. The project name KRAFT stands for Knowledge Re-use And Fusion/Transformation. The primary aim of the project is to collect and fuse knowledge from various resources. KRAFT employs an agent based architecture and uses KQML to achieve interoperability. The primary focus of KRAFT is on "knowledge level mediation" which is achieved by the development of mediators capable of handling knowledge in the form of constraints. Other than that, all components within the KRAFT architecture are quite similar to those of InfoSleuth. Agents in KRAFT are implemented in either Java or Prolog.

11.4. TAMBIS

TAMBIS[BBB97] is a project that is currently being carried out at the University of Manchester. The goal of TAMBIS is to integrate distributed Bioinformatics resources. To achieve this goal, TAMBIS implements a knowledge base(KB) of biological terminology, that captures relationships between various biological concepts. The KB, which acts as a common schema between the various resources, is represented by a description logic language called GRAIL. Wrappers are employed to map between concepts in the KB, and data in a wrapper's underlying resource and to convert TAMBIS queries into a form that a resource can understand.

12. Conclusion

Most of the systems presented attempt to identify and represent relationships between the various entities contained in the different information sources to achieve integration across these resource. To achieve this goal, some have used object models while other employed powerful domain models through the use of ontologies. The semantic level of integration varied across different systems depending on the model each adopted. To address the problem of integrating unstructured information sources with structured information ones, the search for structure where it does not exist seems to be the trend. However, this approach was found to apply only to semi-structured information sources rather than to totally unstructured ones. Nevertheless, this could be remedied by the fact that search results retrieved by search engines spanning unstructured information source, are themselves semi-structured in nature. Thus, integration between the two different resources could be achieved, using a two step procedure.

Most of the reviewed systems' user interfaces are not as intelligent as the systems to which they are connected. These interfaces offer little room for customisation of search presentation, and operation. Also, only a fraction of the systems presented offer support for off-line operation.

Systems that apply an agent based approach reveal that such an approach is rather powerful and suitable for application in the area of system integration. Firstly, agents offer a simple and powerful way to advertise capabilities which is rather important for cutting down search times intelligently. Secondly , one of the design goals of agent based systems is to ensure that all agents are autonomous and none interdependent. This facilitates dynamic addition and removal of resources.

Providing tools to assist in the process of system integration appears to be an area which is likely to remain active for a while yet. To this end, some of the systems have provided templates for allowing semiautomatic construction of wrappers. Efforts to provide higher level tools, are still underway.

13. References

[ACHK93] Yigal Arens, Chin Y. Chee, Chun-Nan Hsu, and Craig A. Knoblock. Retrieving and Integrating Data from Multiple Information Sources. International Journal of Intelligent and Cooperative Information Systems. Vol. 2, No. 2, pp 127-158, 1993.

[ARS98] Guiseppe Amato, Fausto Rabitti and Pasquale Savino. “Multimedia Document Search on the Web”. In Proceedings of the Seventh International World Wide Web Conference, Brisbane, Australia, April, 1998.

[BBB96] R. Bayardo, W. Bohrer, R. Brice, A. Cichocki, G. Fowler, A. Helal, V. Kashyap, T. Ksiezyk, G. Martin, M. Nodine, M. Rashid, M. Rusinkiewicz, R. Shea, C. Unnikrishnan, A. Unruh, D. Woelk: "InfoSleuth: Agent-Based Semantic Integration of Information in Open and Dynamic Environments". MCC Technical Report MCC-INSL-088-96, October, 1996.

[BBB97] Patricia G. Baker, Andy Brass, Sean Bechhofer, Carole Goble, Norman Paton, Mark Quinna.“Transparent Access to Multiple Biological Information Sources, An Overview”. A Technical Report, 1995. http://www.cs.man.ac.uk/mig/tambis/frames/papers/tambis_overview/

[C95] Caplan, P., You call it Corn, we call it syntax-independent metadata for document-like objects, The Public-Access Computer Systems Review 6(4), 1995.

[CF94] R. G. G. Cattell, Guy Ferran: ODMG-93: A Standard for Object-Oriented DBMSs. GI Datenbank Rundbrief 14: 6-7 (1994)

[CHS95] Michael J. Carey, Laura M. Haas, Peter M. Schwarz, Manish Arya, William F. Cody, Ronald Fagin, Myron Flickner, Allen W. Luniewski, Wayne Niblack, Dragutin Petkovic, John Thomas, John H. Williams and Edward L. Wimmers. “Towards Heterogeneous Multimedia Information Systems: The Garlic Approach”. In proceedings of the Fifth International Workshop on Research Issues in Data Engineering(RIDE): Distributed Object Management, 1995.

[EW95] Oren Etzioni and Daniel S. Weld. “Intelligent Agents on the Internet: Fact, Fiction, and Forecast”. IEEE Expert/Intelligent Systems & Their Applications Vol. 10, No. 4, August 1995.

[FWW93] T. Finin,, J. Weber , G. Wiederhold , M. Genesereth , R. Fritzson , M. McKay, J. McGuire , R. Pelavin, S. Shapiro , C. Beck. “Specification of the KQML Agent-Communication Language”. Technical report by The DARPA Knowledge Sharing Initiative External Interfaces Working Group 1993. *URL:http://www.cs.umbc.edu/kqml/papers/kqmlspec.ps*

[GCE98] P M D Gray, Z Cui, S M Embury, W A Gray, K Hui, A Preece. Accepted for Workshop on Agent-Based Manufacturing at Agents'98 International Conference, Minneapolis, USA . http://www.csd.abdn.ac.uk/~apreece/Research/KRAFT/kraft_agents98.html

[GGKS95] Donald F. Geddis, Michael R. Geneserth, Arther M. Keller, and Narinder P. Singh. “Infomaster: A Virtual Information System”. *Proceedings* of the *Intelligent Information Agents Workshop at SIKM’95*, Dec 1995. http://infomaster.stanford.edu/

[GHI95] H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and Jennifer Widom. "Integrating and Accessing Heterogeneous Information Sources in TSIMMIS". In Proceedings of the AAAI Symposium on Information Gathering, pp. 61-64, Stanford, California, March 1995.

[GRGK97] Venkat N. Gudivada, Vijay V. Raghavan, William I. Grosky, and Rajesh Kasanagottu. "Information Retrieval on the World Wide Web". IEEE Internet Computing, Vol. 1, No. 5, pp 58-68, September/October 1997.

[GTC97] Georgia Tech Research Corporation. GVU's 7th WWW user survey home page. http://www.cc.gatech.edu/gvu/user_surveys/survey-1997-04/

[HBG97] J. Hammer, M. Breunig, H. Garcia-Molina, S. Nestorov, V. Vassalos, R. Yerneni. "Template-Based Wrappers in the TSIMMIS System". In Proceedings of the Twenty-Sixth SIGMOD International Conference on Management of Data, Tucson, Arizona, May 12-15, 1997.

[HJK92] M. Huhns, N. Jacobs, T. Ksiezyk, W. Shen, M. Singh and P. Cannata. "Enterprise Information Modelling and Model Integration in Carnot", in Charles J. Petrie Jr., ed.,. Enterprise Integration Modeling: Proceedings of the First International Conference, MIT Press, Cambridge, MA, 1992.

[JS96] Jacobs, N. and R. Shea. "The Role of Java in InfoSleuth: Agent-based Exploitation of Heterogeneous Information Resources", MCC Technical Report MCC-INSL-018-96, March, 1996. Presented at the IntraNet96 Java Developers Conference.

[KA97] Craig A. Knoblock and José Luis Ambite. "Agents for Information Gathering". Software Agents, J. Bradshaw ed., AAAI/MIT Press, Menlo Park, CA, 1997.

[KB97] Bruce Krulwosh, Chad Burkey. "The InfoFinder Agent: Learning User Interests through Heuristic Phrase Extraction". IEEE Expert/Intelligent Systems & Their Applications Vol. 12, No. 5, pp 22- 27, September/October 1997.

[LH97] Sean Luke and James Hendler. "Web Agents that Work". IEEE Multimedia Vol. 4, No. 3, July-September 1997.

[LLP97] Yoo-Shin Lee, Ling Liu, Calton Pu. " Towards Interoperable Heterogeneous Information Systems: An Experiment Using the DIOM Approach", In the Proceesings of the 12th Annual Symposium on Applied Computing(SAC'97) Special track on Database Technology, February 28-March 2, 1997, San Jose, California, USA.

[LP96] Ling Liu and Calton Pu. " An Object-oriented Approach to Interoperable Heterogeneous Information Sources", In: Proceedings of the Seventh International. Hong Kong Computer Society Database Workshop, Hong Kong (May 1996) (Springer Verlag) .

[LRO96a] Alon Y. Levy, Anand Rajaraman and Joann J. Ordille. "Querying Heterogeneous Information Sources Using Source Descriptions". Proceedings of the 22nd International Conference on Very Large Databases, VLDB-96, Bombay, India, September, 1996

[LRO96b] Alon Y. Levy, Anand Rajaraman and Joann J. Ordille. Query Answering Algorithms for Information Agents. To appear in the Proceedings of the 13th National Conference on Artificial Intelligence, AAAI-96, Portland, Oregon, August, 1996.

[LSRH97] S. Luke, L. Spector, D. Roger, J. Handler. “Ontology Based Web Agents”. Proceedings of the First International Conference on Autonomous Agent, 1997.

[MAG97] J. McHugh, S. Abiteboul, R. Goldman, D. Quass, and J. Widom. “Lore: A Database Management System for Semistructured Data”. SIGMOD Record, 26(3):54-66, September 1997.

[MHH97] S. Mukherjea, K. Hirata and Y. Hara, Towards a multimedia World Wide Web information retrieval engine, in: Proc. of the 6th WWW International Conference, S. Clara, CA, 6–11 May 1997.

[MM97] Zakaria Maamar, and Bernard Moulin. "Softeware Agent-Oriented Frameworks for Heterogeneous Information Access". *Proceedings* of the 4th Knowledge Representation meets Databases (KRDB), Athens, Greece, August 1997.

[MW97] J. McHugh and J. Widom. “Integrating Dynamically-Fetched External Information into a DBMS for Semistructured Data”. Proceedings of the Workshop on Management of Semistructured Data, pages 75-82, Tucson, Arizona, May 1997.

[MZ97] A. Moukas , G. Zacharia. “Evolving a Multi-agent Information Filtering Solution in Amalthea” Proceedings of Agent’ 97, Marina Del Rey, 1997.

[NU97] M. Nodine and A. Unruh. "Facilitating Open Communication in Agent Systems: the InfoSleuth Infrastructure". Submitted to The Fourth International Workshop on Agent Theories, Architectures, and Languages (ATAL). MCC Technical Report MCC-INSL-056-97, April 1997.

[PAD98] Glen Pringle, Lloyd Allison and David L. Dowe. “What is a tall poppy among Web pages?”. In Proceedings of the Seventh International World Wide Web Conference, Brisbane, Australia, April, 1998.

[PGW95] Y. Papakonstantinou, H. Garcia-Molina and J. Widom. "Object Exchange Across Heterogeneous Information Sources". IEEE International Conference on Data Engineering, pp. 251-260, Taipei, Taiwan, March 1995.

[RS97] Mary Tork Roth and Peter Schwarz. "Don’t Scrap it, Wrap it! A Wrapper Architecture for Legacy Data Sources". In proceedings of VLDB’ 97, Athens, Greece, August 1997.

[SAA95] V.S. Subrahmanian, Sibel Adali, Anne Brink, Ross Emery, J.ames J. Lu, Adil Rajput, Timothy J. Rogers, Robert Ross, Charles Ward. “HERMES: A Heterogeneous Reasoning and Mediator System”. A Technical Report, 1995. http://www.cs.umd.edu//projects/hermes/overview/paper/

[SC97] John R. Smith and Shih-Fu Chang. “Visually Searching the Web for Content”. IEEE Multimedia Vol. 4, No. 3 pp 12-20, July-September 1997.