Towards Information Agent Interoperability

Stefan Haustein and Sascha Lüdecke

University of Dortmund, Computer Science VIII, D-44221 Dortmund, Germany
{haustein, luedecke}@ls8.cs.uni-dortmund.de

Abstract. Currently, many kinds of information agents for different purposes exist. However, agents from different systems are still unable to cooperate, even if they accurately follow a common standard like FIPA, KIF or KQML. Being able to plug agents together with little effort and exchange information easily, would be of a great use for several reasons. Among others, the agents could profit from each others' services. In addition, certain aspects of multi-agent systems could be evaluated without needing to build a complete system. Testing agent systems with standard components would allow simpler comparison. Furthermore, building different agent-based applications would be simplified by combining new software with "off the shelf"-components. In this paper, we explore the feasibility of practical software development and integration of existing systems, without developing "yet another abstract agent architecture".

1 Motivation

The demand for multi-agent systems has initiated an increasing amount of research in this area. Many different architectures, proposed interfaces, and surely good ideas are implemented in a broad range of systems. All existing multi-agent systems depend on information exchange. Building a population of information agents and building a corresponding population of consumers – let us call them application agents – are different tasks, performed by different people. It would be of a great use, if the agents could be plugged together with little effort and could exchange information easily, thus profiting from each other.

Information agent interoperability is important for research in information agent systems themselves, especially when building heterogeneous systems. For example, research on cooperative information agents often focuses on an isolated aspect of multi agent systems like mediation or brokering. Many information agent systems are build on top of very basic information agents, just wrapping a source into a format suitable for the more intelligent supervising entity. It should be possible to reuse at least these simple entities.

Furthermore, systems can be used as a testbed for each other to increase the overall quality and to prove or disprove certain theories and concepts. But this is not the situation one finds: looking at the current state, it reveals being rather babylonical. Except for the speech act level, each system speaks its own language, uses its own protocol and has its own ontology, despite some existing standards, which spread slowly.

In order to build intelligent components and to provide easy integrateable building blocks, the gap between the systems has to be closed. In this paper we will examine the areas, where the gap can be seen and propose a possible way out.

In the first section we give a short overview about the different places where the gap can be seen. The second section goes into detail an concentrates on the conversational level, which seems most problematic. In the third section we propose a FIPA-based interface, specialized for the needs of information agents. Related work will be presented in section four, whereas section five draws some conclusion and gives an outlook to future work.

2 Levels of Misunderstanding

To make agents interoperable, they obviously have to communicate with each other in order to exchange data, normally using messages transmitted between them. Since there are several standards involved in agent to agent communication at different levels, there is a great chance of incompatibility. This can happen both on the message exchange and on the conversational level. In this section, we give a rough overview of the different layers involved in the communication. We split our overview into two parts. Message exchange covers the range from basic transport layers up to speech acts. Conversations are based on the message exchange and follow mostly semantic and pragmatic conventions. The possible choices an agent designer has, are discussed more detailed in the next section. Let's start with the different levels of message exchange:

ISO/OSI Transport Layers: For agent communication, we can use the abstraction of these levels provided by application level protocols. Nevertheless, exceptions like agents at the physical level controlling a certain device are imaginable. But this normally doesn't touch agent to agent communication as we investigate here.

Application Level Protocol: The application level protocol, like HTTP, IIOP and SMTP, normally is the lowest relevant level for agent communication. It provides a mechanism for platform independent message exchange, but agents obviously cannot even "hear" each other if they use different protocols at this level.

Speech Acts: In many multi-agent systems a communication subsystem uses the application level protocol to provide a more abstract message exchanging interface to the software agents. The messages exchanged, like INFORM, QUERY, REQUEST etc., are based on the speech act theory. At this level, two relevant standards, namely KQML [7] and FIPA [8] exist. Even though two agents or systems are using the same application level protocol, they cannot know the others intention without using the same standard.

Besides the different options for simply composing and exchanging messages, the conversational dimension allows misunderstanding between several agents. Whereas the aforementioned distinction is of a rather syntactic nature, communication can further fail on a semantical and a pragmatic level, too. These are the content and the query language and the model of information access.

Content Language and Query Language: The content language is the language used by the information agent to encode the gathered information, the query language is used by the application agent to inform the information agent about what kind of information it is looking for.

The content and query languages are often related to each other, and in the case of expressive languages like KIF[9] or SL[18] they are normally identical.

The problem with content languages is that even if two agents are using the same content language, they still may not be able to understand each other due to different vocabularies or different ontologies. This can be seen as semantic misunderstanding.

Access Model: In multi agent information systems, the access to the information agents is often directed through mediator, matchmaker, or broker agents. They group agents together and provide a unique simple access model to the requested information.

Furthermore, the information agents and can roughly be grouped into two general access models: information push and information pull. In the pull model, the client "pulls" information from the agent by sending an explicit query. The agent sends back information corresponding to the concrete request. In the push model, the information agents know what they need to do, and gathered information is "pushed" to the client whenever available. This level is addressed by the majority of information agent papers, though usually only one model is provided.

System Application	Communication Language	Content / Query Language	Ontology / Structure
SIMS-based network of agents for logistics planning, including information gathering [14].	KQML	Loom	Proprietary
WARREN: A multi-agent financial portfolio management system [4]	KQML[1]	unspecified	unspecified
Information gathering based on low level retrieval agents accessing HTML, supervised by planning, coordination and scheduling agents. [3].	KQML	unspecified	unspecified
Infomaster: information Integration [6].	unspecified	KIF[2]	unspecified
Agents for Hypermedia Information Discovery [16].	KQML	XML / Prolog	unspecified
Multi-agent Systems in Information-Rich Environments [13].	KQML	SQL	KIF-ontologies[3]
COMRIS [11, 17].	FIPA ACL (XML)	XML	Proprietary

Table 1. Communication properties of some information agent or related system implementations. Although most systems are using KQML for communication, they are not interoperable due to mismatches in lower or higher communication layers.

3 How can Information Agents "Plug and Play" be achieved?

After describing the several levels where misunderstanding can take place, we will now investigate some levels in more detail and point out, how interoperability can be achieved there. First of all, a possible solution for making information agents more interoperable is to provide $O(n^2)$ wrappers between all the incompatible standards available. Another possibility could be to agree on a suitable standard for each level.

[1] Using ASK, ASK-ALL, STREAM-ALL instead of RECRUIT, BROKER, RECOMMEND

[2] For describing information sources

[3] Created with the JAVA ontology editor JOE

But since standards have their specific benefits and disadvantages, it is difficult for the community to achieve an agreement on a certain one.

The following subsections provide a view on the conversational communication levels. The lower levels will not be visited again, since we think that they don't differ much in their properties and thus could be chosen easily. We try to include an imaginable way out of the dilemma of misunderstanding.

3.1 Speech Act and Communication Protocols

The two relevant protocols for the speech act protocol are KQML and FIPA. KQML was introduced 1992 by [7], and table 1 shows that KQML is currently used in the majority of existing information agent systems. FIPA ACL is a newer protocol introduced in 1997 by the Foundation of Physical Agents that is mainly based on [19]. In contrast to KQML, FIPA ACL does not attempt to cover categories like agent management, but provides separate entities for the corresponding purposes.

However, if we look at the question "which standard covers most levels", FIPA is the clear winner: in contrast to KQML it does not cover the speech act level solely, but includes all lower levels of communication. Additionally, the FIPA framework includes an agent management platform. FIPA provides simple services for administrating agent populations, without assuming a concrete agent model limiting possible implementations. Also, several more or less free FIPA platforms are available[4] and there is a noticeable trend of moving towards a FIPA compliant system. This weakens the argument from Huhns and Singh, that KQML has the broader support [12]. So misunderstanding at the speech act level and below will probably vanish in the near future.

3.2 Access Models

Even if we agree on FIPA as "lingua franca" for information agents, and the message exchange problem would be solved completely, problems concerning the protocols of information exchange or ways of collaboration still remain open.

The question, how information agents are able to collaborate is probably the question where most research effort in collaborative information system has gone. Several types of middle agents, namely mediators, matchmakers and brokers have been built. Each type of those middle agents abstracts from the concrete information agents or provides a search mechanism for concrete information agent meeting a certain purpose. In the following paragraphs, we summarize the approaches including their shortcomings for the purpose of general interoperability.

Mediators: Very often, a system needs access to information distributed over several, heterogeneous and even instable data sources like the world wide web, local and online (remote) databases. Mediators like Infomaster[6] typically aim to integrate these sources and to provide a single, consistent interface to all data. They act as intelligent proxies or filters. Mediators typically consist of a set of wrappers around several data-sources and an integrating component, which provides an interface to information requesting agents. Communication normally takes place bidirectionally. Both agents answer requests and update information in the data.

Mediators are mostly passive components. That means that no information gathering is initiated automatically and all requests are answered "online". Even the accessed information systems are forced to stay passive and cannot provide information in advance. However, some might gather information depending

[4] see `http://www.fipa.org/fipa9706.pdf`

on their own internal state and therefore cannot be made accessible through a mediator. Because information gathering and information requesting takes place synchronously, both parts, providers and consumers, are tightly coupled. This often implies a shared interface, which hinders the interoperability and exchangeability of agents. The information is provided by a pull mechanism.

General Matchmaking Agents: Matchmaking agents are agents that are able to bring service providing and service requesting agents together. To get this done, specifying each service agent and each request in a meta-language is necessary. The matchmaking agent then evaluates them against each other by certain heuristics to produce a list of service agents that fulfill a request. The requesting agent must now choose from this list and contact the service agent directly. This approach is used by [22]: services and requests must be specified by context, input/output, and, optionally, further in-/out-constraints and a description of used concepts. Five different methods for match-making are provided, which can be combined by the user. The representation language and semantic matchmaking process is roughly based on KL-ONE [21].
The general matchmaking approach does a good job on mediating between information services and requesters. Providing even plug-and-play functionality, it doesn't decouple requesting agents from information agents, since they have to communicate directly. Thus all communication modalities like the interfaces, the communication language, together with a common ontology, still need to be defined.

Broker Systems: The difference between a matchmaker and a broker is that the matchmaker only introduces matching agents to each other, whereas a broker also remains active when the matching agents are found: all communication between the matched agents goes through the broker, the agents perform indirect communication only. Thus, a broker service necessarily needs to care about possible language and protocol problems.
In contrast to mediators, existing broker systems like Ontobroker [5] are able to collect information from different providers similar to data warehouses. Unfortunately, they are not able to forward queries to external databases like mediators do. Thus, information agents that gather information on demand only, cannot be integrated into systems like Ontobroker and no information pull will be possible.

A general problem with the existing systems is that they do not overcome the gap between push and pull access to information. Each of the systems is fixed to one access model or has either gaps in the interfaces for accessing information or in the conversation definition such as a missing ontology. Because of the typically different natures of information needs in practical multi-agent system, both types of access to information are applicable, especially in heterogeneous populations. For example, an agent controlling a satellite on a fixed orbit sending weather forecast images would probably be an information pushing agent, whereas the corresponding proxy on the ground station could forward the pictures to different clients on demand, thus provide a pull mechanism.

By caching data from a "push"-source, a combination of a mediator and a broker could solve the access mismatch problem, providing both access modes. Systems like Ontobroker [5] or the COMRIS information layer [11] could be extended to such an entity.

3.3 Content and Query Language

For content languages, there is even a broader range of choices which can lead to misunderstandings. The following list gives a rough overview only. Several other content and query languages are used in agent systems, as shown in table 1.

Knowledge Interchange Format (KIF): KIF was introduced 1992 by Genesereth, Fikes and others[9]. It provides a prefix notation for predicate calculus with functional terms and equality, developed at the Stanford Knowledge System Laboratory in the ARPA Knowledge Sharing Effort.

Semantic Language (SL): SL is a content language used by the directory faciliatior, agent management system, and agent communication channel of the FIPA agent platform. Similar to KIF, SL provides full first order logic (see [18] for details) and three different levels of less expressive subsets. SL is one of the content languages proposed for FIPA ACL messages.

Extensible Markup Language (XML): XML[1] is becoming popular as a simplified successor of SGML with the potential to replace HTML in many application. XML is a simple generic markup language, XML-Document Type Definitions are used to create concrete languages for different purposes. A disadvantage of (plain) XML is that the data model is limited to simple trees. The great advantage of XML is that several XML-based schemas for information interchange are under development or are already available.

Resource Description Format (RDF): RDF[15] was introduced by the World Wide Web consortium as a generic meta language for inclusion in HTML and XML. It provides a more powerful data model (conceptual graphs) than XML. Similar to the DTD in XML, RDF provides a mechanism for schema definition, RDF-Schema (RDFS). RDF-Schema itself is specified in RDF, and a RDF Schema for RDF schema exists. Like SL, RDF is one of the content languages proposed for FIPA messages [2]. A disadvantage of RDF is that it uses a more complicated syntax that allows RDF to be embedded in XML. However, there are approaches to use a simplified RDF syntax[5].

A "General" language, fitting all possible needs, currently seems as far away as a general problem solver. SL and KIF have the advantage of using the same language for content and queries. But their expressiveness has the corresponding tradeoff, too.

A major problem for information agent plug-and-play is ontology mismatch, which is still an unsolved problem. Thus, the availability of commonly used and accepted data structures is of major importance for choosing the "right" content language.

An advantage of XML and RDF is that a growing range of concrete schemas is developed for these languages. Also, XML and RDF are already designed for use on the WEB, a major "hunting ground" for information agents. Since XML and RDF are used not only in research but have a broad acceptance in industry, it is likely that more and more information will be available in these formats.

4 An Interface for Information Agents

If we assume that FIPA standards are used for agent communication, and XML or RDF with the corresponding schema definitions and query languages are used for content, the only gap that needs to be filled exists at an intermediate level: What does the concrete registration at the FIPA Directory Faciliator (DF) look like, and which performatives is an information agent required to supported. The FIPA Directory Facilator is a component of the FIPA infrastructure providing a kind of "yellow pages" for software agents.

If we choose simplicity as main goal, the interface between a client and a basic low level information agent should be similar to the abstraction of a complete multi

[5] see `http://www.w3.org/DesignIssues/Syntax` and `http://WWW-DB.Stanford.EDU/~melnik/rdf/syntax.html`

information agent system. On the other hand, strategies and features of intelligent cooperative information agent systems should not be limited in any way by imposing a complete agent system structure. If the system internally makes use of simple basic information gathering agents, the interface may also be useful for advanced middle agents to control the information agents. For other information agent systems, using proprietary communication internally may be a more suitable approach.

Figure 1 shows an example, how agents could be plugged together using unique interfaces. Basic information agents "push" discovered knowledge to an intelligent broker. The broker provides an information service that abstracts from the information sources. Since the application requires pull access to the gathered information, a pull/push converter stores the gathering results and forwards them on demand only. Like the pull/push converter, other intermediate agents like DTD translators are imaginable. Filtering agents can just republish the properties of the covered information providers with their additional or changed features at the DF.

The minimum requirements for an information agent interface are

- a unique definition of how information providers register themselves at the FIPA Directory Faciliator, and
- a concrete description of the performatives the information providers and clients need to implement.

4.1 Registration at the Directory Faciliator

While it is more or less clear, which properties the information agents needs to publish using the Directory Faciliator, there are several ways to map them into the structure the DF provides. For example, basic information agents could put all information into the FIPA `agent description`, while large information agent systems may require the `df-service-description` structure in order to provide all meta-information a client may be interested in. Tables 4 and 3 show an allocation of the corresponding FIPA DF structures suitable for both kinds of agents.

4.2 Client and Provider interfaces

Table 2 shows the minimum set of FIPA performatives that need to be supported by the corresponding types of agents.

A client requesting information or subscribing to information providers just needs to be able to understand the corresponding ACL `inform` messages. In order to receive information, it should also be able to generate `query-ref` or `subscribe` messages.

The minimum requirement for an information agent or information agent system is that it is able to send inform messages to the client. If the information agent is able to perform queries, should also be able to respond to corresponding query messages with an inform message.

The following paragraphs show a short description of the corresponding FIPA performatives.

	Client	Provider
Pull	`query-ref`	`inform`
Push	`subscribe` and `cancel`	`inform`

Table 2. Performatives used in pull and push based communication

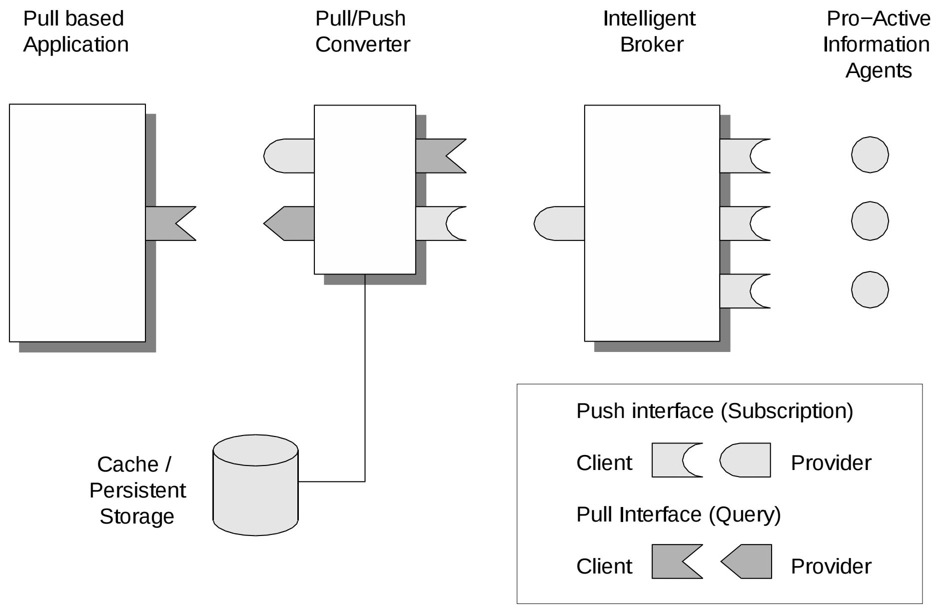

Fig. 1. Architecture Example

Attribute	Content
`:agent-name`	name of the agent (see FIPA specification)
`:services`	see table 4
`:type`	type of the agent: `information-agent`
`:interaction-protocols`	supported access models: `pull`, `push`
`:ontology`	not filled, because defined in service description
`:address`	address of the agent (see FIPA specification)
`:ownership`	ownership of the agent (see FIPA specification)
`:df-state`	state of the agent (see FIPA specification)

Table 3. Content of the `FIPA-DF-Description` structure

Attribute	Content
`:service-name`	name of the service, chosen freely
`:service-type`	type of the service: `information-provider`
`:service-ontology`	reference to a certain XML-DTD or RDF-schema
`:fixed-properties`	`(content-language rdf)`, `(content-language xml)`
	`(query-language XPATH)`, ...

Table 4. Content of the `FIPA-Service-Desc` structure

Subscription

Fipa-Performative	`subscribe`
Content	Query expression

Subscribes an agent to a kind of information defined by a query expression. The provider returns a confirmation including a handle the agent can use to unsubscribe.

Cancel

Fipa-Performative	`cancel`
Content	Subscription handle obtained from the broker when subscribing

Unsubscribes an agent from a subscription previously sent to the broker.

Query-Ref

Fipa-Performative	`query-ref`
Content	Query expression

The information broker performs a query. The query is possibly forwarded to other agents. The querying agent may get multiple responses. The broker is allowed to instruct the query processing agents to send the results directly to the originator of the query.

Inform

Fipa-Performative	`inform`
Content	Query or gathering results from information agents

An information agent informs the broker about newly discovered knowledge or the result of a query. The broker determines from the `in-reply-to` parameter if the content is the answer to a previously sent query or newly discovered information from an autonomous information agent. It distributes the information to the subscribers of the corresponding category or to the query originator.

5 Related Work

This work was inspired by other interoperability and agent communication papers like [20, 10], but we address a more concrete level: While the previous work concentrates on higher levels like semantic brokering and general communication structures, we are addressing more basic issues. These basic issues need to be solved first in our opinion. However, we are keeping the higher levels in mind in order to avoid misdesigning the lower communication levels we address, possibly blocking some higher level features.

The REusable Task Structure-based Intelligent Network Agents System model (RETSINA) [4] tries to provide an abstract framework for the implementation of information agents. This framework related to the interfaces provided here. However, the RETSINA framework does not limit itself to a simple set of interfaces. In this respect, RETSINA is more powerful, but the tradeoff is that it limits the implementation to a certain agent model (i.e. the RETSINA agent model).

6 Conclusions and Outlook

In this paper we have demonstrated the different areas where misunderstanding can take place when exchanging information components between multi-agent systems. We further presented a novel approach to enable interoperability of information agents by specifying a corresponding, simple FIPA compliant interface. In contrast to complete abstract multi agent systems like RETSINA, the described approach does not limit information agent implementation to a certain agent model.

The interface definition will hopefully help integrating information agents of different vendors. Development of large agent systems and new approaches to mediation, can profit from the synergy effects obtained when basic information agents or agent systems for different purposes are available "off-the-shelf". Our next step is to make our own agent system completely compliant to the proposed interfaces, making a set of open source information agents for bibliography search available for the community.

Furthermore, the proposed interfaces provide a base for new intermediate agent concepts like a combined mediator and broker with a local cache, translating between pull and push access. Generally, fixed interfaces are an important precondition for developing intermediate or translating agents that are independent from a concrete agent system or application.

References

1. Bray, T., and Sperberg-McQueen, C. M. Extensible Markup Language (XML): Part I. syntax. Tech. rep., World Wide Web Consortium, 1997. http://www.w3.org/pub/WWW/TR/WD-xml-lang-970331.html.
2. Brickley, D., and Guha, R., Eds. *Resource Description Framework (RDF) Schema Specification.* World Wide Web Consortium, 1999. http://www.w3.org/TR/1999/PR-rdf-schema-19990303.
3. Decker, K., Lesser, V., Nagendra Prasad, M. V., and Wagner, T. An architecture for multi-agent cooperative information gathering. In *CIKM Conference, Workshop on Intelligent Information Agents* (1995).
4. Decker, K. S., and Sycara, K. Intelligent adaptive information agents. *Journal of Intelligent Information Systems*, 9 (1997), 239–260.
5. Decker, S., Erdmann, M., Fensel, D., and Studer, R. Ontobroker: Ontology based access to distributed and semi-structured information. In *Semantic Issues in Multimedia Systems, Kluwer Academic Publisher, Boston, 1999*, R. Meersman and other, Eds. Kluwer Academic Publisher, Boston, 1999.
6. Duschka, O. M., and Genesereth, M. R. Infomaster - An Information Integration Tool. In *Proceedings of the International Workshop "Intelligent InformationIntegration"* (1997), vol. 1303 of *LNAI*, Springer.
7. Finin, T., Weber, J., Wiederhold, G., Genesereth, M., Fritzson, R., McKay, D., McGuire, J., Pelavin, P., Shapiro, S., and Beck, C. Specification of the KQMLagent-communication language. Tech. rep., Enterprise Integration Technologies, Palo Alto, CA, 1992.
8. FIPA. Agent communication language. Tech. rep., FIPA, 1999. FIPA Specification 2 1999.
9. Genesereth, M. R., and Fikes, R. E. Knowledge interchange format, version 3.0 – reference manual. Tech. rep., Computer Science Department, Stanford University, Stanford, California 94305, 1992.
10. Gomaa, H. Inter-agent communication in cooperative information agent-based systes. In *Cooperative Information Agents III* (1999), vol. 1652 of *LNAI*, Springer.
11. Haustein, S. Information environments for software agents. In *KI-99: Advances in Artificial Intelligence* (Bonn, Germany, September 1999), W. Burgard, T. Christaller, and A. B. Cremers, Eds., vol. 1701 of *LNAI*, Springer Verlag, pp. 295 – 298.
12. Huhns, M. N., and Singh, M. P. Conversational agents. *IEEE Internet Computing 2* (1997), 73 – 75.

13. Huhns, M. N., and Singh, M. P. *Readings in Agents.* Morgan Kaufmann, San Francisco, Calif., 1998.
14. Knoblock, C. A., and José. Agents for information gathering. In *Software Agents*, J. M. Bradshaw, Ed. AAAI Press, Menlo Park, Calif., 1997, pp. 347 – 374.
15. Lassila, O., and Swick, R. R., Eds. *Resource Description Framework (RDF) Model and Syntax Specification.* World Wide Web Consortium, 1999. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222.
16. Lazarou, V. S., and Clark, K. L. Agents for hypermedia information discovery. In *Cooperative Information Agents II* (1998), no. 1435 in Lecture Notes in Artificial Intellignce.
17. Plaza, E., Arcos, J. L., Noriega, P., and Sierra, C. Competing agents in agent-mediated institutions. *Personal Technologies Journal 2*, 3 (1998), 1–9.
18. Sadek, M. *Attitudes mentales et interaction rationelle: ver une theéorie formelle de la communication.* PhD thesis, Université de Rennes I, France, 1991.
19. Sadek, M. D., Bretier, P., Carodet, V., Cozannet, A., Dupont, P., Ferrieux, A., and Panaget, F. A cooperative spoken dialogue system based on a rational agent model: A first implementation an the ags application. In *Proceedings of the ESCA/ETR Workshop on Spoken Dialogue Systems: Theories and Applications* (Vigso, Denmark, 1995).
20. Sheth, A., Kashyap, V., and Lima, T. Semantic information brokering – how can a multi-agent approach help? In *Cooperative Information Agents III* (1999), vol. 1652 of *LNAI*, Springer.
21. Sycara, K., Lu, J., and Klusch, M. Interoperability among Heterogeneous Software Agents on the Internet. Tech. Rep. CMU-RI-TR-98-22, CMU Pittsburgh, USA, 1998.
22. Sycara, K., Lu, J., Klusch, M., and Widoff, S. Dynamic Service Matchmaking among Agents in Open Information Environments. *ACM SIGMOD Record* (1999). Special Issue on Semantic Interoperability in Global Information Systems.