A Framework for Inter Society Communication in Agents

R. Abbasi, F. Mitchell & S. Greenwood

School of Computing and Mathematical sciences
Oxford Brookes University
Oxford, OX3 0BP
UK.
<rabbasi, frmitchell, sgreenwood>@brookes.ac.uk

Abstract. Normally a society of agents is developed to cope with and solve a particular set of problems or to achieve some particular goals and objectives. Different people, organizations and groups have developed and deployed agent societies for their own particular needs. These societies have to communicate, coordinate and compete with each other. In this paper an explanation is provided as to what is meant by inter society communication in agents, why it is needed and some basic problems in the area of inter-society communication are identified. A framework for inter-society communication in agent systems is also proposed.

1 Introduction

A typical agent society is composed of more than one agent, who work together to achieve a shared purpose or goal, through a division of work, and are integrated by decision processes. In order to achieve its goal, a society has to provide some mechanism for its component agents to communicate with each other.

Communication between agents can be direct or indirect. Direct communication is based on theory of communication, which emerged from the telecommunications research of Shannon and Weaver in the 1940's. In this model [25,34] an act of communication consists of the sending of some information from one party (sender) to another party (receiver), while the information being passed is normally encoded in some form, sent on a medium and decoded on reception. This is the most commonly used method of communication in agent societies on which many models are based. [10,13,20,22,26]. A Model of communication in agent societies is normally known as Agent Communication Language (ACL)[30]. Most of these ACL's are semantically based on speech act theory [23,29]. KQML [13] and FIPA ACL [10] are the most commonly used ACL.

2 Inter-Society Communication

It is possible to have different agent societies, each using different ACL's for their internal communication purposes. Normally agents communicate only or mainly with other agents within their society. However, there is gradual recognition of the need for these societies to interact with each other. Agent societies can learn from each other, can use services provided by other societies and can offer services to other agent societies. As human societies collaborate, coordinate and also compete with each other, similarly agent societies will also need to collaborate, coordinate and compete, either for their own purpose or for achieving the goal set for them by users, in order to achieve this, agent societies have to communicate with each other.

From software engineering viewpoint just in the same way that there is a need for different distributed object systems, written in different programming languages to communicate, it is becoming increasingly apparent that there is a need for different agent societies, with different communication methods, to be able to communicate. CORBA [3] and SOAP [27] are possible solutions for distributed objects, but due to the differences between objects and agents, they are currently unsuited for the challenges posed by Inter Society Communication (ISC). What is needed is a CORBA like solution for agent societies.

3 The Framework.

There are many issues and problem associated with providing a mechanism for Inter-Society Communication in Agents (ISCA). Five major issues and problems have been identified and a framework has been proposed and partially implemented and tested. This framework consists of five layers and each layer addresses one issue or problem. A detailed description of each layer follows.

3.1 Naming

Every agent society can be considered as an organizational unit. These societies cannot communicate with each other on a society level unless and until they recognize each other's existence. Another, related problem is discovering the existence of other societies which is not possible without a unique identification for each society.

It is worth mentioning that at an intra-society level there are few communication methods which do not require unique naming scheme for agents within society. These methods include communication via environment, blackboard or broadcasting. We do not consider these as suitable for inter-society communication for the following reasons. Communication through the environment is infeasible simply because agent societies do not have to share a common environment. A blackboard is potentially more useful, however blackboards can become a bottleneck and a central point of failure. One solution to this is to provide a mechanism for providing replication of the blackboard, however, reliable replication is a known hard problem in distributed systems (and most solutions require a naming scheme for the replicants, !) therefore

we believe blackboards are not suitable as a mechanism for inter-society communication. Similarly broadcasting may serve at an intra-society level where agents are on same network or subnet and also the number of agents within a society is normally not in thousands or millions, but this is not suitable for inter-society communication[1]. Finally although we have not considered system security in this framework, it is an important concept, and security is severely weakened (if not removed), if there is no form of identification available. Therefore at an inter-society level we have to rely on a direct communication method, which will require the unique identification of each individual society.

There are different schemes and standards available for naming of agents within a society. The research in this area does not highlight any suitable naming standard for agent societies, which can guarantee a unique name for each society. The required characteristics for a naming scheme for agent societies are listed below.

- It should generate a unique name for each society.
- It should be to able handle an arbitrary number of names.
- It should be able to deal with an arbitrary administrative organization for names.
- It should be dynamic i.e. able to handle changes made to names, attributes associated with the names and the distribution of the naming database.
- It should be efficient, providing speedy retrieval of names and their attributes.
- It should provide transparent distribution.
- It should be reliable i.e. providing fault-tolerance.
- It should be secure.
- It should be distributed
- It should be hierarchal

Using the above requirements a naming system for agent societies has been designed and implemented, which forms the first layer of the framework. In this system any society, which wants to be a part of the global environment, has to acquire a unique name. There are three main components of this naming system.

1. **Name Resolution Language**: - A language used for name resolution
2. **Societies Naming Agent (SNA)**: - An agent responsible for the generating a unique name for the societies
3. **Local Naming Agent (LNA): -** an agent responsible for acquiring a name for the society

The process of acquiring a unique name for the society requires a dialogue or communication between the SNA and the LNA. Different LNA's belonging to different societies will be speaking different ACL's. It is currently infeasible to design an SNA that could understand all the ACL's. However the dialogue between the SNA, and the LNA will always be very specific hence there are only a few different ACL messages used during this dialogue. Keeping these two things in mind a Name Resolution Language (NRL) has been proposed. This NRL is designed as a native language for the SNA. There is no set of performatives in the NRL. A performative is a speech act, and if it was used we would have to use some content language as well

[1] What would had happened if broadcasting was being used for email? It might work in the beginning when emails addresses are few but later with billions of email addresses and millions of email servers it would not be workable.

to code the message. As the case under consideration is quite simple we can omit both the performative and the content language. In NRL a field called 'status' is used, that is not a speech act. Status just depicts what is the status of the dialogue between the SNL and the LNA. Other fields of the message are 'sender', 'receiver' and 'name'. The name, which is being negotiated, is coded in the field 'name'.

The main function of the SNA is to keep the information about the names of all the existing societies in its domain[2] and make sure that all the new societies in that domain are allocated a unique name. In order to avoid a bottleneck and also for the purposes of reliability, efficiency and robustness it is proposed to have more than one SNA for a specific domain. These SNA's work together as a team to ensure that all the societies in one domain have a unique name. Whenever a SNA receives a 'check' message for a name, it checks in its local knowledgebase to see whether the name has already been allocated or not. If the name has been already allocated, it proposes another name to the requesting party, which might be of interest to the requesting party, but is not being used at that time. Currently a simple algorithm for proposing a new name is being used, but this could be modified at a later stage if required. If the name is not being used according to local information of the SNA then it tells the requesting party that the name is available by sending them a message with status 'available'. If the SNA receives a request 'request-allocate', it means that the requesting party wants a name to be allocated to them. Now the SNA has to make sure that this name is not being proposed to some other party by another SNA. In order to do that the SNA talks with it's fellow SNA's and checks the availability of name by sending a 'can-i-offer-it' message. Then it waits for the responses from all other SNA's. If the responses from all the other SNA are positive (can-offer) then the name is allocated to requesting party and all other SNA's are informed that the name has been allocated. If the response from even one SNA is negative (do-not-offer) then a new name is proposed to the requesting party. If, after a certain time period, no message is received from one or more fellow SNAs, a time out will occur and the message will be sent out again.

The Role of the LNA in this system is to act on the behalf of the agent society, and to obtain a unique name for the society. This is a simple process and may only need to be carried out once. As work progresses it might be desirable to allocate other functionality to the LNA. In the test implementation this job is assigned to the Name Server/Facilitator agent.

The Naming system described above has been implemented for a single domain of societies and tested successfully. For the purpose of inter-society communication all the societies at the global level have to be uniquely named so that communication between agent societies at the global level is possible. A hierarchical naming system is proposed which can be based on geography and functionality.

---

[2] We assume that the societies of the agent are well organized into domains. Each society belongs to at least one domain and there could be many domains of the societies. One domain can have more than one sub-domain.

3.2 ACL Resolution.

Within a society of agents the same ACL is normally used for communication. MostACL's are based on speech act theory. In a typical ACL, agents communicate with each other using messages. Each message starts with a speech act (sometimes called a performative in some ACL's) followed by some other predefined fields. These messages normally travel from one agent to another in the format of a string. When an agent receives a message from an agent belonging to another society without prior knowledge of the ACL used by the other society, the outcome is unpredictable and can cause the following results:

1. Agent is not able to parse the message because it does not contain some mandatory fields.
2. Agent is able to parse the message but the speech act or performative is not understood.
3. Agent is able to parse the message and is even able to identify the speech act but there is a chance that the meaning of speech act is different to both ACL's so in this case the agent would misunderstand the message.
4. Agent does not understand the ontology used in the message
5. Agent does not understand the language used for the content field of the message.

In all these cases the agent is not able to understand the message properly resulting in failure of possible communication between the agents. Also the agent cannot tell other agents that it is not able to understand the message, as any message send may cause similar kinds of problems for the agent belonging to the other society.

It is known that translation between different ACL's is a non-trivial problem. But without tackling this problem, the dream of inter-society communication cannot be achieved. Agent societies normally do (or will need to) communicate with each other mainly for information sharing, service provision and service request

This means that all the communication between agent societies is generally limited to the types of communication outlined above, hence it is possible to propose some basic performative set which can be used to design messages. This limited scope allows the proposal of a reduced set of fields in message. This reduction in fields in a message and the limited number of primitive performatives will simplify the problem and enable the design of a mechanism for translation between different ACL's. Three performatives have been proposed as primitive performatives, which are Inform, Query and Request.

The next step is to propose a generic set of fields. In any ACL a message is sent from one agent to another so sender and receiver fields are needed. The content field is needed for the content of the message, language is needed to tell the recipients what language is used for the content and ontology is needed to tell what ontology the sender is using. These fields have to be supported by every ACL to be able to use the framework. It is possible to envisage a situation where fields like reply-with and in-reply-to may be advantageous, especially when multiple conversations are going on at the same time between two societies. Some people prefer to use conversation-id instead of reply-with and in-reply-to therefore this field is also included in the list.

Here generic names for fields have been used and it is possible that different ACL's will use other names for those fields. As long as any message is being coded

using only the above-mentioned fields or their equivalent fields the translation mechanism will be able to translate the message in other ACL's.

All the messages for inter society communication will use any of the above listed performatives and should be coded using the above-mentioned fields. In different ACL's there could be different equivalents for these performatives and fields. It is not possible, or even desirable to force every one to use exactly the same performatives and fields because this would be the equivalent to introducing a new ACL. The idea is to let every society use its own ACL with its own fields and performatives for communication within that society, and also let it communicate with external societies using the same language. It is the responsibility of a translator to ensure that each society receives a message in a format, which it understands. When a society wants to become part of a global environment in order to provide services to other societies or wants to make use of services provided by other societies, it will have to inform the translator what ACL it is using and what are the equivalent of the primitive performatives and generic fields. For the purpose of encoding this an ACL Description Format (ADF) is proposed. The fields of the ADF and their description are given below.

- **SocietyName**:- States the name of the society using the ACL being described.
- **BaseLanguage**:- States the base language for the ACL being described.
- **NoOfFields**:- States how many fields are used in the message of the ACL.
- **NoOfPerformatives**:- States how many performatives are supported by the ACL.
- **FieldsList**:- Includes all the fields being used in ACL with the equivalence information. If the ACL is using any field for which there is no equivalent generic field name then slot for generic name is filled with the reserved word "NULL". The syntax is (field name1 : primitive field name1, field name2 : primitive field name2,….)
- **PerformativeList**. Includes all the perfomatives supported by the ACL being described, along with the equivalence information. The reserved word "NULL" is used instead of the primitive performative where there is no primitive performative available. The syntax of this field is similar to the one of FieldList.

The job of the translator is to translate a message from an ACL to a generic format and vice-versa. Each society can have its own translator. The translator always receives a message in the generic format from the outside world and then translates it into its own native language. Whenever anyone from within the society has to communicate with an agent in another society, it has to do it via the translator. The translator will translate the message into the generic format before sending it out. In this scenario the translator is not only responsible for translating the messages but also acting as a Foreign Minster[3].

Another approach can be to divide the above-mentioned role into two parts, where one agent is the translator while another is responsible for communicating with the external world, the Foreign Minster (FM).

Yet another option is to move the translator outside the society and making them available globally. A FM is still needed who will be responsible for the

[3] In our Framework a Foreign Minister is an agent who is a link with other societies. It is able to discover information about other societies such as their location, services provided by them and the ACL, content language and the Ontology being used.

communication with the outside world. All the communication with the outside world is achieved by this FM.

In all cases the address of the FM has to be made available to other societies, so that any agent from another society can discover how that society can be accessed. When an agent from within a society wants to communicate with the outside world it sends its message to the FM. The FM finds some suitable translator who understands its ACL. Then the FM sends this message to that the translator. If the translator understands the ACL spoken by the receiver society it translates the message for the agent and sends it to the respective FM of the receiver society, which then receives the message and passes it to the appropriate agent. If the translator does not understand the ACL used by the receiver's society it can make use of the services of another translator, who does understand the ACL used by the receiver's society before forwarding it to respective FM.

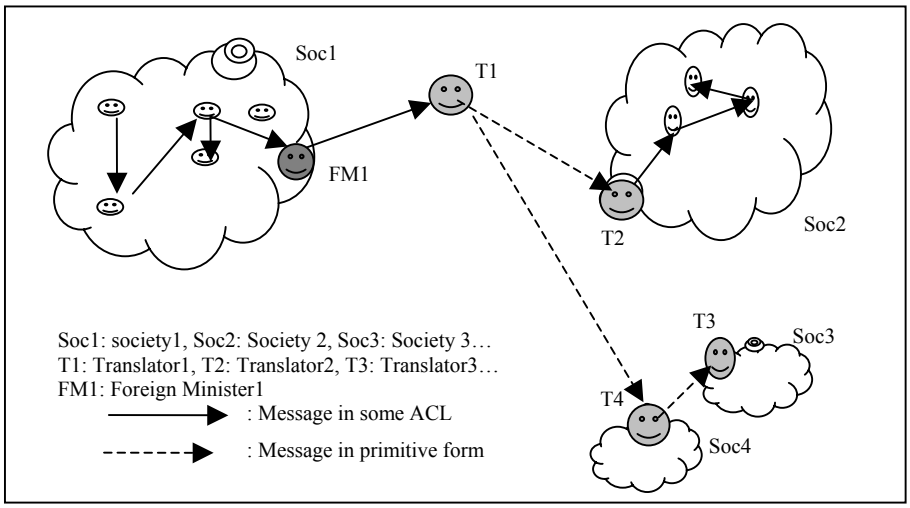


Fig. 1. Translators and Foreign Ministers

The above mentioned translation mechanism has been implemented and tested successfully for translation between FIPA-ACL and KQML and the societies using these languages were able to communicate with each other with the facilitation of the translator. In future this mechanism will be tested on other ACL's, which are substantially different from KQML and FIPA ACL in order to demonstrate its generality for ACL's based on speech act theory.

3.3 Service Advertisement and request Mechanism

When a society needs a service, the first hurdle is to find a suitable society that can provide the required service. There is a requirement that there should be some available mechanism, which can be used by the societies to advertise their services.

The societies should also be able to find suitable services for them using the same mechanism.

The main goal is to enable requester societies to obtain a response according to their needs and request. For this purpose we have proposed a service broker mechanism based on classification of problems and information.

Classification System

A classification system, which will enable societies to search for required services in some respective class is needed. With the provision of such a classification system, it will become easier for societies to advertise and request the services. Also the process of request resolution will be able to use class information and will become simpler. For the purpose of ISCA a classification system is needed that:

- can deal with classification of both information and services.
- is dynamic i.e. services could be added, deleted and modified dynamically.
- can deal with the fact that one particular service can belong to more than one class at the same time.
- can give a unique identification to each service with respect to classes it belongs to.
- can deal with the fact that one service can be provided by more than one society.
- provides some provision to relate services with the provider societies.
- can be used by agents without involvement of human users.

The design of a complete classification system is not the subject of this work but in order to demonstrate the principles, a prototype for classification system is proposed and used in the ISCA framework for a limited set of societies. However any classification system which had the desired properties listed above could be used.

The proposed classification system classifies the problems and information into a variable-level tree structure and each level in this tree is identified by a series of numerical values. Each main class represent a specific domain for example one for science, the second for industry, the third for business and so on. Each main class is divided into more than one sub-classes and each sub-class is divided into further sub-sub-classes. The depth of the classification tree can be extended to any reasonable level needed.

Every service or type of information provided by a society can be associated with a description based on its position in the tree. Because services can have multiple descriptions, the system allows a service to be listed in more than one part of the tree

This information has to be supplied by the service provider society to the broker. For the purpose of service advertisement and request a language has been proposed called "Services Advertisement and Query Language for Societies" (SAQLS).

Services Advertisement and Query Language for Societies (SAQLS)

The provider of a service not only has to provide class information to the broker but also has to specify the input and output signatures of its service. Other information that may be provided includes the ACL, content language and ontology used. Based on this information the broker generates a token for each service. Some work, like LARKS [32], has been carried out in this regard but it is not primarily designed for inter-society communication, and also it does not address the concept of

classification. Therefore a language has been proposed called Services Advertisement and Query Language for Societies (SAQLS). Societies can also use this language to find a service and can specify the required upper limit on the expected number of results. The brief description of fields in a SAQLS message is providedd in table 1.

Table 1: - Description of different fields used in a typical SAQLS message

Name of Field	Descriptions	Message Directions
Performative	It is used to describe type of message it can have one of these value:-Advertise, Query, Advertised-ok, Query-response, Unadvertise, Unadvertised-ok	Society→Broker Broker → Society
Sender	Tells who is the sender of message. It will have either name of broker or name of FM/Translator of society	Society→Broker Broker → Society
Receiver	Tells who is the receiver of message. It will have either name of broker or name of FM/Translator of society	Society→Broker Broker → Society
Classlist	Tell to which classes a service belongs to	Society→Broker Broker → Society
Input	Used for input specification of services	Society→Broker Broker → Society
Output	Used for output specification of services	Society→Broker Broker → Society
ACL	Tells ACL being used by the society	Society→Broker Broker → Society
CL	Tells CL used being by the society	Society→Broker Broker → Society
Ontology	Tells ontology being used by the society	Society→Broker Broker → Society
Descriptor	Is used for encoding descriptor of services	Society→Broker Broker → Society
Provider	Is used to encode name of provider society	Broker → Society
Threshold	Tells what threshold is required	Society→Broker
Message-No	Tells the serial number of message if multiple messages are being send and is of the form 2/5.	Broker → Society
Weight	Describes the weight of proposed service as measured by the broker	Broker → Society

Service Broker

The service broker is an agent, which stores the information provided by the service provider, and also helps the requester societies to find a service suitable for their requirements. The service broker maintains a knowledgebase about the services provided by the agent societies and it stores all the information provided about the services by the societies. In order to accomplish this it has to transform the information provided by the societies via a SAQLS message into some internal form. In the test implementation tree and hash-table data structures have been used to maintain the information so that searching through the knowledgebase is made easier.

Matchmaking

The resolution of requests is an important problem. In the proposed solution the class information is the main parameter. A simple algorithm has been designed for distance calculation between classes. The information about input and output specification, ACL, content language and ontology is also used in the request resolution mechanism. The main goal is to generate a minimum number of results, which are as near to the request as possible. The broker agent also returns sufficient information to the requester along with the results so that the requester can utilizes its own matchmaking criteria if desired.

The above service advertisement and request mechanism has been implemented for two simple agent societies, each performing simple mathematical functions. The next step is to test it for a larger number of societies offering different types of services.

3.4 Content Language Resolution

One of the most important fields of any ACL message is the content field. This field, in combination with the speech act, determines the meaning of the message. Within a society agents normally use the same language for the content field and even if, as in some cases, multiple languages are used, they are understood by the agents in the society. However, in the area of inter-society communication there is a very strong possibility that content language used by one society is not understood by the other society. Another service, which can resolve this problem associated with use of different CL is required.

In the framework being proposed, the fourth layer will consist of some solutions for the problems associated with the use of multiple content languages. For this layer, a mechanism will be designed to facilitate the use of the formal description of content languages for translation of the messages from one content language into anther without any loss or alteration in the semantics of the messages.

3.5 Ontology Resolution

Agents use different symbols and words, which are understood by all the agents within a society as all the agents within a society share the same (implicit or explicit) ontology. However, when an agent tries to send a message to another agent belonging to another society that is using a totally different ontology then even use of the same ACL and content language does not guarantee that message will be properly understood by the receiving agent. There is a very good chance that the receiving agent will not be able to understand the message at all. In order to address this problem we need another service, which allows the different societies to share the ontologies being used.

When requesting a service it needs to be guaranteed that the ontology used by both the service provider and consumer is the same. We do not want to end up in a situation where a society looking for type of computer mouse ends up talking with a society that is dealing with different classes of species of mouse.

One idea is to classify ontologies on the basis of functional domain. A good starting point is to look at the work carried out by the semantic web initiative (W3C) [35,37] in this field.. If every ontology is described formally, it is often theoretically possible to transform a message between different ontologies. The fifth layer of the framework will consist of a solution to the problems associated with the use of multiple onotologies. The aim for this layer is to use available formal descriptions of ontologies in order to resolve differences between them.

4 Discussions and Further Work

A framework for inter-society communication in agents has been proposed. This Framework is still under development, however the design and implementation of the first three layers has been completed. The initial testing of these layers has been carried out and the usability of the framework has been demonstrated for some simple domains. The next step will be the design and implementation of the remaining two layers. Once all five layers are completed the whole system will be tested together as an entity. The Framework is being designed and implemented in such a way that each of its layers will form a complete system and will work independent of the other layers. This will provide freedom for agent societies to make use of the layers as and when they wish.

5 References

1. Agent Interoperability : http://www-2.cs.cmu.edu/~softagents/muri/interagent.html (May 2002)
2. Agentcities : http://www.agentcities.org/ (May 2002)
3. Common Object Request Broker Architecture (CORBA), Homepage: http://www.corba.org/ (2002)
4. Communication Working Group of Agentcities : http://www.agentcities.org/Activities/WG/Comms/ (May 2002)
5. Dale, J and Mamdani, E. : Open Standards for Interoperating Agent-Based Systems. In: Software Focus, 1(2), Wiley (2001)
6. DARPA Agent Markup Language (DAML) homepage: http://www.daml.org/ (2002)
7. Davies, W.; Edwards, P. : Agent-K: An Integration of AOP and KQML, King's College Technical Report AUCS/TR9406 (1994)
8. Ferber, J. : Multi-Agent System. An Introduction to Distributed Artificial Intelligence. Addison Wesley (1999)
9. FIPA Messaging Interoperability Service Specification: http://www.fipa.org/specs/fipa00093/ (May 2002)
10. FIPA, Agent Communication Language : http://www.fipa.org/repository/aclspecs.html (2001)
11. Gensereth, M.; Fikes, R. : Knowledge Interchange Format, Version 3.0 Reference Manual (1992)
12. Jennings, N.; Wooldridge, M. (eds.) : Agent Technology, Springer (1997)

13. Labrou, Y.; Finin, T.: Semantics for an agent communication language. The fourth International Workshop on Agent Theories, Architectures, and Languages. Rhode Island, USA (1997)
14. Labrou, Y.; Finin, T.: Agent Communication Languages: The Current Landscape, Intelligent Systems, Vol. 14, No. 2, IEEE Computer Society (1999)
15. Lander, S. : Issues in Multiagent design systems. IEEE Expert12 (2): (1997) 18-26
16. Mai, C. Comaromi, J. Satija, M.: Dewey Decimal Classification: a practical guide. Albany, N.Y.: Forest Press (1994)
17. Mockapetris, P. : Domain Names- Concepts and Facilities. Tech. Report RFC1034 (1987)
18. Muller, P. : The Design of Intelligent Agents - A Layered Approach, volume 1177 of LNAI (1996)
19. Ontology Working Group of Agentcities http://www.agentcities.org/Activities/WG/Ontology/ (May 2002)
20. Open Agent Architecture. : http://www.ai.sri.com/~oaa/whitepaper.html (2002)
21. Reilly, D. : Simple Agent Communication Protocol, http://www.davidreilly.com/sacp (2002)
22. Sadek, D.; Bretier, P.; Panaget, F. : ARTIMIS technology, http://www.cselt.it/fipa/torino/cfp1/propos97_015.htm (1997)
23. Searle, J. : Speech Acts. An Essay in the Philosophy of Language. Cambridge (1969)
24. Service Description and Compositions Working Group of Agentcities: http://www.agentcities.org/Activities/WG/SDC/ (May 2002)
25. Shannon, C. : A mathematical theory of communication, Bell System Technical Journal, vol. 27, (1948) 379-423 and 623-656.
26. Shoham, Y. : Agent Oriented Programming: an overview of the framework and a summary of recent research, Knowledge Representation and Reasoning Under Uncertainty (1994) 123-129.
27. Simple Object Access Protocol (SOAP) homepage: http://www.w3.org/TR/SOAP/ (2002)
28. Smith, B. : Materials Towards a History of Speech Act Theory, in Eschbach, A. (ed.), Karl Bühler's Theory of Language. Amsterdam, (1988) 125-52.
29. Singh, M. : Multiagent systems, A Theoretical Framework for Intentions, Know-How, and Communications, Lecture Notes in Computer Science, Volume 799, Springer-Verlag (1994)
30. Singh, M. : Agent Communication Languages: Rethinking the Principles. IEEE Computer, volume 31, number 12, (1998) 40- 47.
31. Sycara, K,; Lu, J.; Klusch, M.; Widoff, S.: Matchmaking Among Heterogeneous Agents in the Internet . Proceedings AAAI Spring Symposium on Intelligent Agents in Cyberspace , Stanford, USA (1999)
32. Sycara, K. Widoff, S. Klusch, M and Lu, J. : LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace. Autonomous Agents and Multi-Agent Systems, 5, (2002) 173–203.
33. Vizine-Goetz, D. : Using Library Classification Schemes for Internet Resources. Proceedings of the OCLC Internet Cataloging Colloquium, San Antonio, Texas (1996).
34. Weaver, W. : Shannon, C: The Mathematical Theory of Communication, Urbana, Illinois: University of Illinois Press (1963).
35. Web Ontology Working Group of W3C : http://www.w3.org/2001/sw/WebOnt/ (May 2002)
36. Wooldridge M. : Verifiable Semantics for Agent Communication Languages, ICMAS'1998 Paris (1998).
37. World Wide Web Consortium (W3C) : http://www.w3c.org/ (May 2002)