Agent Communication Semantics for Open Environments: Issues and Challenges

Steven Willmott
Ecole Polytechnique Fédérale de Lausanne, Switzerland
CH-1015
steven.willmott@epfl.ch

Jonathan Dale
Fujitsu Laboratories of America
Sunnyvale, California
94086, USA
jonathan.dale@fla.fujitsu.com

Patricia Charlton
Motorola
Espace Technologique – St Aubin
Gif-sur-Yvette 91193, France
patricia.charlton@motorola.com

ABSTRACT

Explicit communication semantics are often considered essential for rich interaction between heterogeneous automated systems. While extensive work on semantic models has been carried out, much of this work remains to be tested in real open environments.

This paper describes some of the issues and challenges to be considered when combining existing semantic frameworks for effective agent communication. In particular, we describe issues related to the development of the Agentcities Network which is a large-scale open test bed for agent systems that aims to enable on-line experimentation with semantic frameworks for agent communication.

The presentation is kept discursive in nature: characterizing different aspects of communication, outlining research challenges, commenting on possible strategies and describing the current status of activities in the Agentcities Network.

Keywords

Semantics of Agent Communication, Interaction Protocols, Agent Communication Languages, Content Expressions, Ontologies, FIPA Semantics, Open Environments, Deployment and Application

1. INTRODUCTION

Open environments such as the Internet create conditions where on-line systems can freely interact with one another. Whilst current Web sites are essentially static and interaction is driven by human users, developments such as Web Services [Web Services], GRID computing [GRID], P2P networks and e-business standards are making it possible to establish complex system-system interactions.

However, how will these systems communicate? In particular, how will developers deal with the increasing *complexity* (accomplishing more complex tasks), *flexibility* (dealing with heterogeneity in the environment) and *adaptivity* (providing component re-use and dealing with change in the environment) that their systems are likely to require? Furthermore, how will they ensure that their systems communicate in ways which are meaningful enough to support real, automated business interactions? Although there is no single answer to these questions it seems clear that many of the issues faced will be exactly those considered by researchers working on agent communication over the past decade or more.

The extensive work on semantics for agent communication addresses many aspects of the communication problem, from conversation protocols [AUML] and agent communication languages [FIPA-ACL][KQML] to content expressions [KIF][FIPA-SL] and shared ontologies [ONTOL]. While much of this work provides a promising source of solutions for communication in open and dynamic environments, many of the existing formalisms have not yet been tested in large-scale applications and it remains to be seen how effectively they can be used to support interoperability in truly open environments.

This paper is motivated by some of the communication issues which have arisen in the Agentcities Network[1] [Willmott 00] that is working to deploy a large-scale open network of agents and services which are able to interact with each other flexibly and dynamically. In particular, it aims to enable agents deployed in publicly accessible servers to begin communicating with each other on a rich, semantic basis.

The work in the Network is intended to enable:

1. The application of existing semantic frameworks to evaluate how effective they are at enabling semantic interoperation between agents in open environments.
2. The development of new or refined frameworks which might be applied to future generations of information systems.

The objectives of this paper are to describe a roadmap for work on semantic frameworks in large-scale environments such as the Agentcities Network (Section 2), present some of the issues that must be dealt with (Sections 3 and 4) and to briefly outline a first draft at a top-to-bottom semantic model that is being used in the Agentcities Network (Section 5). Section 6 concludes the paper.

2. CONTEXT AND OBJECTIVES

Agentcities is an on-going initiative to develop and exploit a large-scale test bed in which any researcher or developer can deploy their own agent-based systems and services. The objective of the initiative is to create a global open system to provide the conditions in which to test agents, services and other technologies, such as delegation, coordination, modeling of dynamics and, in particular, communication based on formal semantics.

At the time of writing there are over 100 organizations involved in various Agentcities-related projects, running in 20 countries. An initial network of agent platforms based on the FIPA agent standard[2] has been deployed, called the Agentcities Network.[3]

[1] http://www.agentcities.org/

[2] http://www.fipa.org/

2.1 Need for Semantic Frameworks

Agentcities aims to enable a large number of agent systems developed by different organizations to interoperate. There are a number of reasons why explicit semantic frameworks are essential in achieving this goal:

- With agents and services being deployed by many individuals, it becomes impossible to hand engineer interactions and it becomes essential for interactions to be defined clearly and unambiguously.
- Agents in the environment will be able to coordinate their actions, make commitments and, in the long term, engage in business transactions. Such activities are very difficult to achieve without being able to ascribe precise meaning to interactions and derive resulting obligations.
- Explicitly shared semantics allow agents to reason flexibly about their communication which enables them to react to exploit the environment around them more effectively.

While individual solutions may be heterogeneous or application-dependent, there is a need to provide broad frameworks that allow descriptions of the communication framework used in the Agentcities Network.

2.2 Challenges

The realization explicitly semantic frameworks raise a number of challenges:

- The benefits of many proposed semantic frameworks have rarely been demonstrated in real applications/systems[4].
- It is not clear to what extent these solutions can be applied together to address the need for semantics at multiple levels of communication.
- There has yet to be widespread adoption (in research or industry) of even the best-known frameworks, such as FIPA-ACL.

While good interoperability has been shown in particular agent architectures such as AAA/OAA (see [Cohen 97] [Kimur 00]), it has not been demonstrated in truly open environments.

2.3 Milestones

Here we describe a number of milestones that represent how semantic frameworks can be used within software systems:

M1 **Complete off-line descriptions**: Formalisms can be used off-line, for example, in the design or analysis phases of software development, to precisely define the meaning of any message instance in the context that it was sent.

M2 **On-line message generation**: Agents are able to use encoded knowledge of the semantic formalisms to generate messages in predetermined ranges depending upon need.

M3 **On-line message interpretation in limited domains**: Agents can reason about the meaning of a particular message within a limited domain in context as it arrives and thus determine an appropriate response.[5]

M4 **Universal flexible artificial languages**: Restrictions on generality are reduced as agents are able to perform complex reasoning to interpret/generate messages in multiple domains based on their core knowledge and external resources.

As a first step, agents may have very rigid interaction possibilities (for example, canned messages) but the meaning can be precisely defined[6]. Subsequent steps increase the flexibility of both the formalisms and their usage.

The intention is not to suggest that all of these milestones will be achieved in the context of the Agentcities initiative but to outline long term research challenges which are relevant to Agentcities now and potentially to other networks / systems in the future.

2.4 Current Status

Given these four steps, it is interesting to reflect on what the current status of research is:[7]

- Overall, it appears research has not yet fully achieved M1 except in isolated demonstration cases where domains, interactions and world models were strongly limited.
- Message generation is used in many agent systems but often the range of messages generated is not based on explicit semantics but on tailored program code produced by the developer, that is, the semantics remain implicit.

Whilst reasoning results have been shown for some formalisms, research is still needed in the area of combining many of the semantic frameworks at each level in the semantic communication stack to the extent that effective reasoners could be developed for them. For example, reasoners have been constructed for KIF, but these have not been combined with reasoners for conversation protocol formalisms, ACLs or agent world models.

3. SEMANTICS OVERVIEW

There are a large number of approaches to addressing semantics from the area of linguistics and philosophy. Computer science has often drawn from these areas to formalize a semantic framework for communicating systems such as agents, e.g. speech acts, which are concerned with the actual use of a language by its speakers and receivers. However, the utterance (usually for software agents is packaged in the content part of a message) that is conveyed between two agent systems can be represented by a combination of *explicit* and *implicit* semantics. Also, communication between a number of agents may require distinguishing between *extension* and *intension* within any particular communication. Finally it often proves useful to

[3] There are currently 40 agent platforms registered in the network of which 30 are regularly running – a continuously updated list of which platforms are up and running can be found at http://www.agentcities.net/

[4] Whilst this has been done for individual aspects, such as the utility of domain ontologies [Fensel 01], to our knowledge, there are few examples of semantic-rich agent communication in general. In some cases, the semantic frameworks have not moved beyond small-scale demonstrators.

[5] Determining meaning here is taken to mean identifying a particular world state which accounts for information imparted by the message.

[6] In this case, messages can be seen as method calls based on a particular API. The advantage of explicit semantic frameworks is that the meaning of each method call is precisely defined and determined in the context of the world/agent state when it is sent or received.

[7] This refers to systems that have been tested in large-scale open environments.

characterize the meaning of communication in terms of semantic stack.

In this section we discuss further these aspects of semantic modeling and their importance.

3.1 Implicit versus Explicit Semantics

As pointed out by Uschold [Uschold 02], an important distinction to be made in the discussion of semantic frameworks is not on whether a "system has semantics" but on whether the semantics in the systems are "*explicitly* or *implicitly* encoded".

If two systems, A and B, are able to communicate with each other and act appropriately for each message they can be said to have *shared semantics*. If these semantics are *explicit* a formal specification of the meaning of each message should exist. However, if such an explicit definition does not exist (and the systems communicate by sending short codes known only to the system designers), the semantics are in practice *implicitly* encoded in the software implementation of each system, that is, they are hard coded.

A second question is whether a set of explicit semantics is written in its own formal and machine-readable language, which itself then may have explicit (a second level of formalism) or implicit semantics (through an interpreter, for example).

3.2 Meaning through Extension and Intension

Although explicit representation may lead to some degree of semantic interoperability, there is a second separation of representation required. It is often hoped that explicit semantics means that the meaning of any set of communication can be completely determined by the meaning of its constituents. This idea is commonly called the *principle of compositionality* [Jackendorf 99].

This principal is based solely on the extension meaning but this is often not sufficient in understanding a communication, it may also require the expression of *intension*. The extension of an expression means the object or set of objects in the real world to which this refers. The intension of an expression means its sense (meaning) that a person or software agent normally understands by the expression. Although, a number of systems may provide an explicit semantics of extension the intension is often made implicit being captured by the feature of context, which is usually domain specific aspects that are well engineered into the application. To represent intensional aspects, in theory, of the formulae, model operators are used. Such operators, and their logics, have been described for the modalities knowledge, belief, necessity, possibility, desire, obligation, the passing of time etc.

3.3 Semantic Communication Stack

Communication between software systems is often characterized as a stack of levels to separate different functional aspects of communication. Table 1 provides a level of decomposition drawn from those often used for the Foundation of Intelligent Software Agents[8] agent standard [FIPA-ACL] and KQML/KIF [KQML] [KIF].

Level	**Description**	**Semantic Description**
Context	State of the world in which the *conversation* takes place	Formalism for describing the meaning of states of the world, an institution, a market, etc
Conversation	Sequence of *messages*	Formal account of the meaning of statements in the protocol description formalism – which can ideally be interpreted to give the meaning of any particular state in the conversation sequence. (AUML, Finite State Machines etc.)
Message	A single communication from one or more originators to one or more listeners which expresses the speaker's opinion about the *content*[9]	Formal account of the meaning of messages represented in a particular language, for example: • FIPA-ACL semantics in Modal logic • KQML semantics in Definite Clause Grammar formalism • ebXML message semantics in natural language [ebXML]
Content	The description of a partial world state (or a world) which may contain references to *objects*, *actions*, *functions*, … in one or more *domains*	Formal grammar, semantics represented in particular language and a definition of those semantics, for example: • FIPA-SL: logic base • KIF: logic base • Prolog: logic base + interpreter • Java: language + Java virtual Machine specification
Domain Description	References to and descriptions of *objects, action, function* and other instances	Formalism for defining possible classes and/or instances of things in the world, for example, DAML+OIL, Ontolingua KIF

Table 1: Semantic communication stack

The domain description may be arbitrarily types according to how its description is formalized, for example, in DAML+OIL [DAML+OIL] everything is a subtype of the Thing class but every class defined has its own identifier. Content would normally be expressed in a content language (KIF, FIPA-SL), the

[8] FIPA, see http://www.fipa.org/

[9] The simplest and most usual case would be one receiver and one sender, but with more powerful semantic formalisms it could be more.

message in an agent communication language (FIPA-ACL, KQML) and the conversation/context perhaps in a logical formalism such as situation calculus with the protocol sequence specified in AUML, for example.

Finally the notion of context in this case can vary considerably from application to application. Ideally a semantic system supporting a set of communicating agents would have an organizational model to create context grounding. Currently in most engineered systems the main context is derived from the domain description, the conversation/dialogues determining an exchange of content and actions/response, and some reference to an ontology and potentially grounded references to entities existing in the environment (such as object references or access to physical entities). Even when there is the use of modal operators in the representation of certain aspects of this semantic stack e.g. BDI semantics for speech acts, the intensional thread through the stack is not modeled with any explicit semantics, hence the integration is usually engineered per solution.

4. REASONING ABOUT SEMANTICS

The problem that we are trying to address is:

> *"How to reason over the semantic definitions given for a particular message instance?"*

In particular, the definitions relevant to a message may span different communication levels and different semantic frameworks at each level. In the off-line case (M1), this would be done at compilation or analysis time and probably by human developers. The on-line cases (M2-M4) require computational mechanisms to be embedded into the software systems to be able to manipulate semantic definitions relevant to the message input, often in very time-constrained circumstances.

4.1 Dependencies between Levels

Ideally, each level in the communication stack should be opaque in terms of reasoning with regard to the layer above or below, but in practice there are constraints across the levels that hinder this:

- **Context**: Some conversation protocols may not be allowed because they are not understood by both parties or are inappropriate at this stage in the conversation. The context also generally *grounds* the entire communication, that is, it must contain instances of and descriptions of elements referenced or assumed in any given message.
- **Conversation**: The types of messages that can be sent or received may be restricted because of limitations imposed by the current conversation protocol.
- **Message**: The performative and parameters of the message may impose restrictions on the content expression of the message, such as using only entities from specific ontologies or using only the content type specified by the performative. For example, in FIPA-ACL, the *inform* performative can only have a proposition in the content expression.
- **Message content**: The content expression may have constraints imposed on it from the type and expressivity of its domain descriptors, for example, functions may have typed input/output variables, etc. The same can be true for other entities in the content expression.

If any of these constraints are not preserved across levels in the semantic communication stack, then the message can become meaningless since the meaning for the whole cannot be derived. Secondly, in principle, both communicating entities need to be coded in such a way that they either explicitly or implicitly can derive the semantics of each communication at all levels. These types of constraints are normally defined in each language or formalism that is being used, for example, scoping rules, rules for delimiting opaque constructs, etc.

4.2 Reasoning Problem

In determining the meaning of a particular message there are two linked problems:

- **Problem 1**: Evaluating the meaning of *every item at each level* in terms of its defined semantics and then correctly accounting for the meaning of each individual instance.
- **Problem 2**: Extracting the meaning of the *whole communication*.

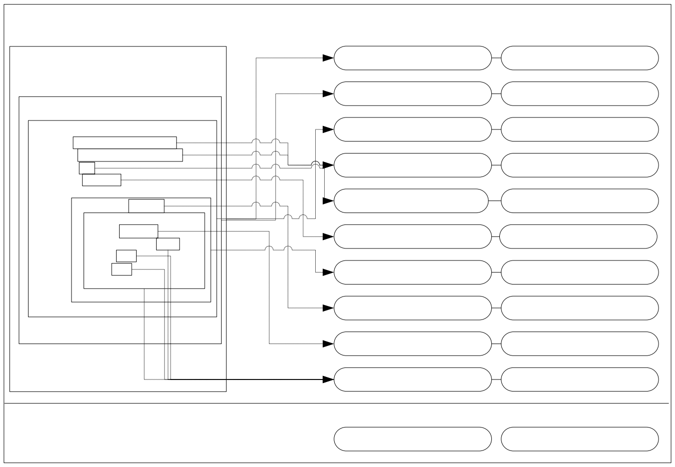

Figure 1: Different formalisms for different aspects of a message generate a layered reasoning problems which can potentially be very complex (or unsolvable) depending on the combination of formalisms.

When considering the problems, the following issues may also arise:

- Some statements may not make sense out of the global context, that is, an individual item cannot be evaluated on its own.
- The meaning of the whole communication may be more than the sum of the meaning of the individual items.

Therefore, the reasoning problems generated by interpreting the semantics of a particular message can become very challenging very quickly.

4.3 Reasoning Strategies

While systems have been built which use some of the existing semantic formalisms (see [Mayfield 96] for example), these have rarely been used together at different levels in the semantic communication stack and the resulting reasoning problems have often not been addressed.

A number of standard strategies to tackling the problem have been proposed, exist implicitly in existing work or can be derived by analogy from well-known AI reasoning strategies for complex problems. The most obvious of these strategies discussed in the next sections.

4.3.1 Unifying Semantic Framework

One of the main obstacles to reasoning with the kind of problem shown in Figure 2 is that there are many different formalisms involved at each level in the semantic communication stack; many of which cannot easily be reconciled with each other. One way to deal with this is to restrict all semantic definitions (and potentially message instances) to be expressed in on particular formalism, usually a logic.

If the formalism were to contain a large range of application, it would most probably need to contain constructs for each of the following:

- World model state (fact base),
- Time,
- Action,
- Sufficient semantics for a wide range of message and content expressions (potentially modes of belief, desire, intention, uncertainty as well was existential and universal quantifiers), and,
- Class/frame definitions and relations between them.

This approach of mapping all aspects of the problem into a single formalism is reminiscent of Agent Oriented Programming [Shoham 93], but other base technologies for this approach might include Prolog, KIF or logics such as LORA [Wooldridge 00].

The list of constructs needed would likely generate intractable reasoning problems (see FIPA-SL, for example). Further problems with this approach are:

- The real world is highly heterogeneous and it may not always be possible to map all aspects of a communication into a single formalism.
- The generic reasoning mechanism required would likely be sub-optimal for many specific problems (planning formalisms, for example, have evolved over years of work and in many domains, representation is key and often the difference is between being usable and unusable).
- Many applications may be very simple and understanding the message can be implicitly built into the application if the designer is able to determine the meaning at each of the various levels. For example, a system might handle only one conversation protocol, two performatives, a very limited content language, etc. The apparatus required for the complete semantic communication stack would be unnecessary and burdensome in this case.

4.3.2 Problem Decomposition

The levels in the semantic communication stack shown in Figure 1 might be considered somewhat arbitrary and it may often be difficult to choose at which level an item should be represented, for example, is the notion “to subscribe” part of a general communication language at the performative level or is it an action defined at the domain level, or both? This raises the question of the value of such a level decomposition in the first place.

However, such structure does provide a way of decomposing the whole reasoning problem down into smaller parts, that is, manipulating the problem to determine disjoint or nearly disjoint sub-problems and solving these individually (the standard AI divide and conquer strategy). This can be highly effective since complexity versus size in terms of evaluation can be additive across disjoint sub-problems but multiplicative inside a problem.

Interesting questions that arise from this are:

- How disjoint are the sub-problems in any particular semantic communication stack of languages?
- How dependent is this on the choice of languages, formalisms and domains?
- Can problem decomposition be performed dynamically through problem reformulation?[10]

Note that the argument for decomposition is orthogonal to the question of how these sub-problems are expressed.

4.3.3 Expressivity Restrictions

A third strategy is to constrain the expressivity of the languages and semantic formalisms used by a system. Examples of this are:

- Different levels of FIPA-SL and KIF are defined to enable designers to choose the version with appropriate computational costs.
- Some languages (CCL [Willmott 00b] for example) map the communication language into problem solving formalisms with known properties.
- Limited expressivity of ontology frameworks reduce the types of relations which can be used in domain description (description logic for example [Calvanese 98]).
- The domain descriptions (ontologies) that a particular system deals with can be severely restricted to reduce the size of the reasoning problem.
- It may also be possible to generate application-specific content languages [Cranefield 01].

The final consideration should take into account the computational properties of the whole system, and hence, the expressivity of an individual formalism needs to be considered in the context of other formalisms being used as well as the interdependencies between levels.

4.3.4 Limiting Reasoning Required

A final strategy is to severely limit the aims of reasoning being performed by, for example:

- Considering only a small subset of all possible messages and limiting the reasoning to determining whether an incoming message matches one of the finite set[11].

[10] This is often done in Constraint Satisfaction Problem Solving, for example.

[11] In general, this decision problem could be just as complex as the full reasoning problem since statements in some languages may be arbitrary logic expressions; deciding if two statements are equivalent could be very complex, comparing at the parse tree or string level would be more tractable but less flexible.

- Reasoning only over a small part of the message in full generality (see section 4.2 for example) whilst ignoring all other levels.

In the case where a message cannot be interpreted or does not fall into the small class that can be treated, an agent will usually reply that the message could not be understood. Almost all communicating agent systems currently do this to an extent since they rarely consider all levels of the semantic communication stack.

5. SEMANTICS IN AGENTCITIES

Whilst there are still many open questions in semantic communication research, the Agentcities initiative is attempting to use existing semantic frameworks in a large-scale initiative.

5.1 Semantic Communication Stack in Agentcities

The choices made for the first generation of test agents and services for the Agentcities Network are as shown in Table 2.

Level	Choices
Context	There is currently no prescribed formalism for this level since we do not assume a common worldwide context. Local context will, however, emerge through interactions, for example, in market places[12].
Conversation	AUML interaction diagrams
Message	Standard FIPA-ACL performative semantics and the FIPA-ACL string syntax
Content	FIPA-SL or first-order logic KIF
Domain Description	DAML+OIL

Table 2: Semantic framework choices

In the Agentcities initiative, descriptions for particular agents and services are currently being developed independently at each level, but have not yet been formed into a coherent top-to-bottom framework.

5.2 Issues

Many of the choices listed in the previous section are the subject of considerable and ongoing debate. In particular:

- **Content language**: Five possible content language technologies have been evaluated in the context of the EU Agentcities.RTD project[13]: Prolog base, KIF base, FIPA-SL base, ebXML base and DAML+OIL base.
- **Ontology representation**: There is ongoing debate regarding the suitability of DAML+OIL for modeling functions and predicates that are often needed for logical languages.

Furthermore, there are known issues with some of the technology decisions:

- The semantics of performatives used in the FIPA interaction protocols which are modeled in AUML are often not defined in the same way as they are when the performatives are defined individually [Pitt 99].
- FIPA-ACL semantics are defined on the basis of non-observable properties of an agent, that is, its mental attitudes [Pitt 99].
- No group communication semantics are defined which limits interactions to binary pairs of agents.
- FIPA-SL is not well tested and in its full form presents intractable reasoning problems.

Despite these potential problems, the proposed semantic communication stack for Agentcities does form a useful first step since all levels in the stack defined can be handled. Furthermore, many of the components are supported by existing software toolkits; FIPA components by FIPA Agent platforms[14] and DAML+OIL by a number of editors.

5.3 Future Developments

Up until now, the work in the Agentcities initiative has been focused on deploying a network infrastructure of agent platforms, agents and services. In recent months, however, the work has turned towards service development that will provide the first real use of the chosen technologies in the semantic communication stack. We expect to learn much about the strengths and weaknesses of these technologies over the coming months.

An important consideration is that semantic frameworks will be used throughout the Network even as they are being adapted and developed. This has lead to the creation of the ACTF Communication Working Group[15] which will act as focal point for discussion on communication issues in the Network. The objectives of this group are to:

- Compose a set of existing communication technologies (such as FIPA-ACL, FIPA-SL, KIF, DAML+OIL) into a coherent communication framework by providing a draft solution by July, 2002 which will then be refined over the following 12 months.
- Gather communication requirements from activities across the Agentcities Network.
- Provide user-guide style documentation for communication in the Agentcities Network.

Results from the working group will be made publicly available and communicated to other research and industry activities (such as FIPA) through liaison activities.

6. CONCLUSIONS

This paper has presented some of the challenges raised in the area of semantic communication by the Agentcities initiative which are also relevant more generally to developing on-line environments such as Web Services and GRID computing.

[12] Frameworks such as Upper Ontologies [SUO] and Social laws [Shoham 95] may form part of the context.

[13] See http://www.agentcities.org/EURTD/

[14] See http://www.agentcities.org/ for a list currently used in the Agentcities Network.

[15] See http://www.agentcities.org

We expect that semantic communication in the Agentcities Network will initially be fragmented, inconsistent and require a considerable amount of human intervention to function. However, we hope that this process will provide a focal point for future research and discussion and eventually lead to more automated agent-to-agent interaction.

Above all, we believe that a holistic (top-to-bottom) view which considers all levels of the semantic communication stack is required to reach the end goal of semantic interoperability and that live tests are an important way forward. We hope also that the problem descriptions, definitions and breakdown presented will be generally useful for other researchers in the field.

7. ACKNOWLEDGMENTS

The research described in this paper is partly supported by the EC project Agentcities.RTD, reference IST-2000-28385. The opinions expressed in this paper are those of the authors and are not necessarily those of the Agentcities.RTD project partners.

8. REFERENCES

[AUML] Agent UML Modeling Language. http://www.auml.org/

[Calvanese 98] "Description Logics for Conceptual Data Modeling",] D. Calvanese, M. Lenzerini, D. Nardi. In collection: *Logics for Databases and Information Systems*, J. Chomicki and G. Saake eds., Kluwer, 1998.

[Cohen 97] Cohen, Philip. R., Cheyer, Adam., Wang, Michele. and Baeg, Soon. Cheol. "An Open Agent Architecture" In: Readings in Agents, Eds. Michael N. Huhns & Munindar P. Singh, pages 197 –204, 1997.

[Cranefield 01] "Generating Ontology-Specific Content Languages", Stephen Cranefield, Martin Purvis. First International workshop on Ontologies in Agent Systems (OAS2001). http://www.ceur-ws.org/Vol-52

[DAML+OIL] DARPA Agent Markup Language Activity. http://www.daml.org/.

[ebXML] XML Based electronic business specifications. UN/CEFACT and OASIS. http://www.ebxml.org/

[Fensel 01] "Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce", Dieter Fensel, Springer Verlag 2001.

[FIPA-ACL] "FIPA ACL Message Structure Specification", FIPA00061, Foundation for Intelligent Physical Agents. http://www.fipa.org/specs/fipa00061

[FIPA-SL] "FIPA SL Content Language Specification", FIPA00008, Foundation for Intelligent Physical Agents. http://www.fipa.org/specs/fipa00008/

[GRID] Global Grid Forum. http://www.gridforum.org/

[Jackendorf 99] "Semantics and Cognition", Ray Jackendorf, 1999, MIT Press.

[KIF] "Knowledge Interchange Format". http://logic.stanford.edu/kif/kif.html

[Kimur 00] Kimur, S. Huber, M, McGee, D., Cohen, P., Levesque, H. Designing the Semantics of Agent Communication Languages for Group Interaction, in The Seventeenth National Conference on Artificial Intelligence (AAAI 2000), AAIT Press/The MIT Press, pp. 42-47, Austin, Texas, July 30-August 3, 2000.

[KQML] "Specification of the KQML Agent-Communication Language -- plus example agent policies and Architectures", Tim Finin et. Al. The DARPA Knowledge Sharing Initiative External Interfaces Working Group Technical Report 1993. http://www.cs.umbc.edu/kqml/papers/kqmlspec.ps

[Mayfield 96] "Evaluating KQML as an agent communication language*",*] J. Mayeld, Y. Labrou, and T. Finin.. In M. J. Wooldridge, J. P. M□uller, and M. Tambe, editors, "*Intelligent Agents II*", volume 1039 of Lecture Notes in Articial Intelligence, pages 347-360, Berlin, Germany, 1996. Springer

[ONTOL] "The Ontolingua Server: A Tool for Collaborative Ontology Construction", A. Farquhar, R. Fikes, & J. Rice, Technical Report, Knowledge Systems Laboratory, Stanford University, September, 1996.

[Pitt 99] "A Protocol-Based Semantics for an Agent Communication Language", J.V. Pitt and A. Mamdani. *Proceedings 16th International Joint Conference on Artificial Intelligence IJCAI'99*, Stockholm; Morgan-Kaufmann Publishers, pp486–491, 1999.

[Shoham 93] "Agent-Oriented Programming", Y. Shoham Artificial Intelligence (60) pp. 51-92, 1993.

[Shoham 95] "On Social Laws for Artificial Agent Societies: Off-Line Design", Shoham, Y. and Tennenholz, M. Artificial Intelligence 73, pages 231-252. 1995.

[SUO] "IEEE P1600.1 Standard Upper Ontology (SUO) Working Group". http://suo.ieee.org/

[Uschold 02] "The Semantic Continuum" Mike Uschold, Knowledge Engineering Review (to appear 2002).

[Web Services] W3C Web Services Activity. http://www.w3.org/2002/ws/

[Willmott 00] "Agentcities: A Worldwide Open Agent Network" Steven Willmott, Jonathan Dale, Bernard Burg, Patricia Charlton and Paul O'brien. Agentlink News Issue 8, November 2001.

[Willmott 00b] "CCL: Expressions of Choice in Agent Communication" S. Willmott, M. Calisti, B. Faltings, S. Macho-Gonzalez, O. Belakhdar, M. Torrens. The Fourth International Conference on MultiAgent Systems (ICMAS-2000), July 2000. pages 325-332.

[Wooldridge 00] "Reasoning about Rational Agents" MIT Press, July 2000.