Flexible Multi-Agent Protocols

Citation for published version:
Walton, C & Robertson, D 2002, Flexible Multi-Agent Protocols. in Proceedings of UKMAS 2002.

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Proceedings of UKMAS 2002

School of Informatics, University of Edinburgh

Centre for Intelligent Systems and their Applications

Flexible Multi-Agent Protocols

by

Christopher Walton, Dave Robertson

Informatics Research Report EDI-INF-RR-0164

School of Informatics
http://www.informatics.ed.ac.uk/

November 2002

Flexible Multi-Agent Protocols

Christopher Walton, Dave Robertson

Informatics Research Report EDI-INF-RR-0164

SCHOOL *of* INFORMATICS
Centre for Intelligent Systems and their Applications
November 2002

Abstract :
In this paper we define a novel technique for the specification of agent protocols in Multi-Agent-Systems. This technique addresses a number of shortcomings of previous Electronic Institution based specifications. In particular, we relax the static specification of agent protocols as state-based diagrams and allow protocols to be defined and disseminated in a flexible manner during agent interaction. Our flexible specification is derived from process algebra and thus forms a sound basis for the verification of such systems.

Keywords : multi-agent systems, agent communication languages, electronic institutions, flexible agent protocols

Flexible Multi-Agent Protocols*

Christopher Walton (cdw@inf.ed.ac.uk) and
David Robertson (dr@inf.ed.ac.uk)
Centre for Intelligent Systems and their Applications (CISA),
Division of Informatics, The University of Edinburgh,
80 South Bridge, Edinburgh EH1 1HN

8 November 2002

Abstract

In this paper we define a novel technique for the specification of agent protocols in Multi-Agent-Systems. This technique addresses a number of shortcomings of previous Electronic Institution based specifications. In particular, we relax the static specification of agent protocols as state-based diagrams and allow protocols to be defined and disseminated in a flexible manner during agent interaction. Our flexible specification is derived from process algebra and thus forms a sound basis for the verification of such systems.

1 Introduction

A Multi-Agent-System (MAS) is defined as a collection of *agents*, i.e. independent, autonomous, and rational components, each with a separate identity and which interact within an environment [FG97, Jen00]. We equate the term rational with possessing reasoning powers, that is, an agent should be able to perform deductions using a representation of aspects from the environment [Woo00]. Each individual agent of a MAS exhibits intelligent behaviour based on its interactions with other agents, its environment and its internal reasoning [Nwa96].

A MAS is often built as a *society* in which individual agents, although potentially autonomous, observe conventions which allow them to co-operate. For example, agents participating in an auction must observer certain conventions for making bids and interacting with the auctioneer. In certain settings these conventions are called the *social norms* of the system. A collection of these conventions, relating to a particular task or function, is termed the *agent protocol* of the MAS.

In the real world, such specifications are commonplace, for example the rules of an auction can be specified in natural language. However, when applied to a MAS there are a number of additional factors which must be considered. These factors primarily relate to design issues of the MAS. One such issue concerns the distribution of the agent protocol, i.e. how do numerous distributed agents,

[*] This research has been sponsored by the European Union (contract IST-1999-10208) Sustainable Life-cycles in Information Ecosystems (SLIE) project

all participating in the protocol, know the global state of the protocol and where they fit in at any given time. A possible solution is to impose some form of global control, such as having an administrative agent whose sole responsibility is to enforce each step of the protocol [VSE02]. However, this is arguably contrary to the underlying principles of self-contained autonomous agents and also raises issues of availability and reliability of the administrator. For example, a problem in the administrative agent could lead to the collapse of the entire MAS.

It should be noted that there is already a body of work on the specification of such protocols for multi-agent systems. Two of these approaches are Electronic Institutions [RMN$^+$00] and Conversation Policies [GHB99]. We have been involved closely with the former [VRA$^+$01] and believe that it solves many of the issues which challenge MAS designers. Nonetheless, there are a number of remaining issues which we will describe in this paper, primarily relating to the static nature of the specifications in these approaches. In the latter half of this paper we will outline a novel technique which we believe overcomes these limitations. This technique permits agent protocols to be defined in a flexible manner during the interaction of agents.

2 Agent Protocols

In the preceding section we described an *agent protocol* as a specification of the conventions enabling co-operation among agents. Is is clear that communication will be a crucial requirement of any agent protocol, as agents will be unable to co-operate if they cannot communicate. However, the actual mechanisms by which agents communicate are not part of the agent protocol itself. An agent protocol requires that a communication mechanism exists, but does not attempt to define the actual mechanism itself.

Communication among agents is performed by a separate *communication protocol.* The MAS community has largely adopted two standard protocols, namely KQML/KIF [MLF96] and FIPA-ACL [FIP99]. These communication protocols are similar in that the semantics of both are based on an underlying theory of speech acts [CL90]. These protocols perform communication through the passing of units of information between agents in the form of messages. Both standards define languages which ensure that agents can communicate regardless of their implementation. The differences between these standards are primarily in the *performatives* (message types) that they provide.

We do not intend to debate the merits of these standards here. Instead we note that the agent protocols, which are the subject of this paper, form a layer above that provided by these communication protocols. Intuitively, the communication protocols define *how* agents can communicate, while agent protocols define *if and when* agents should communicate. Communication protocols define the set of all possible illocutions, while agent protocols define the order and kind of messages relating to a task.

It should also be noted that agent protocols do not attempt to define how the agents rationalise internally. There are many differing methods by which agents can exhibit rational behaviour. In the MAS community, one of the most popular models of rational agency is the *Belief-Desire-Intention* (BDI) model [Bra87] based on modal logics. However, there are other models based on logics such as Concurrent MetateM [BFG$^+$95], and also models based on

Decision Theory and Game Semantics [PGW02].

Agent protocols, as described in this paper, can be seen as a middle layer between low-level communication issues and high-level rational issues. The need for agent protocols was originally thought to be unnecessary, as agent protocols can be encoded as part of the communication and rationalisation process. Agent protocols have also been considered to be undesirable as they may be thought of as restricting or even stifling emergent behaviours within such systems. However, we argue that while the absence of agent protocols may be acceptable in small systems (< 100 agents), their absence would be a serious hindrance in deployment of large MAS (≥ 100 agents). Attempting to deploy a large MAS without agent protocols, would result in a large "soup" of rational communicating agents. In order for such agents to co-operate it would be necessary for each agent to perform a large number of tasks, for example:

1. Individually identify relevant agents from the entire group.
2. Identify groups of agents with particular capabilities.
3. Negotiate agreements among groups of agents.
4. Handle the possibility of unavailable and unreliable agents.
5. Compensate for the asynchronous nature of message passing.

Agent protocols greatly assist in the design of large MAS as they impose structure on the agents, co-ordinate tasks between agents, and define commitments which agents must satisfy. They also simplify the design of individual agents as they separate the task of defining the co-ordination of the agents from the definition of agent behaviours. This separation also permits the refinement of the agent protocol independently from the design of individual agents.

3 Electronic Institutions

The Electronic Institution (EI) framework [RMN$^+$00] is a popular technique for specifying and deploying agent protocols. The underlying concept of this framework is that human interactions are always guided by formal and informal conventions. Human interactions are never completely unconstrained, rather they are controlled by such notions as conventions, customs, etiquette, and laws. Institutions are a means of representing these conventions (i.e. social norms). The EI framework provides a method for controlling the interaction of agents in a MAS using formally defined Institutions.

The terminology used in the formal definition of an EI is derived by considering an Institution as analogous to a theatre production. The agents that are co-ordinated by the Institution are analogous to the *actors*, and each agent takes one (or more) *roles* in the Institution. The interactions are articulated through the use of *scenes* in which groups of agents directly interact. Within a scene, all the participating agents follow a single script (modelled as a finite-state system) which guides their interactions. Scenes are composed as a network called a *performative structure*. Agents move through the Institution, participating in different scenes, with the possibility of assuming different roles in each. Because the actors are agents and not real humans, some additional terminology is also required. The agents interact through the use of speech acts. Institutions define the acceptable vocabulary (i.e. ontology) through the use of a *dialogic framework*. The actions of an agent in the context of an Institution may have consequences that either limit or enlarge its subsequent acting possibilities. The possible paths for an agent within the performative structure are thus defined by a set of *normative rules* containing obligations and commitments.

It is helpful to consider an example of an Institution in order to illustrate these concepts. In Figure 1 we present an example scene which would form part of a larger Institution. This scene defines an interaction protocol between doctor and patient agents. This is scene is intended to represent a patient visiting a general practitioner to obtain a diagnosis of some symptoms.

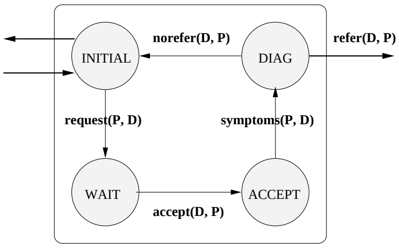

Figure 1: General Practitioner Scene

There are two roles in this scene: doctor and patient. Any number of agents (i.e. actors) satisfying these roles are permitted. For convenience we assume that all agents use the same dialogic framework (i.e. they know how to communicate) and that there are no normative rules. The scene begins with all the agents entering the INITIAL state. A patient agent then sends a request message to a doctor agent indicated by **request(P, D)**, where P is a patient and D is a doctor. This message is intended to represent the patient making an appointment to see a doctor. The patient then enters the WAIT state until an **accept(D, P)** message is received from the doctor. At this point the agent enters the ACCEPT state and proceeds to send a message **symptoms(P, D)** to the doctor. The doctor then performs a diagnosis of the patient in the DIAG state and the result is that the agent is referred **refer(D, P)** for further diagnosis, or no-referral **norefer(D, P)** is made in which case the patient leaves the scene.

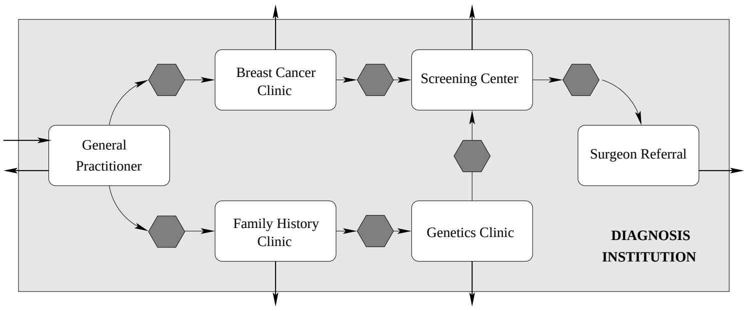

Figure 2: Institution for the Diagnosis of Breast Cancer

We have considerably simplified this example scene for the purpose of this paper. Nonetheless, this example adequately demonstrates the specification of a protocol in this framework. The full EI would consist of a number of inter-linked scenes as illustrated in Figure 2. The rectangles represent scenes, and the inter-scene connectives represent the performative structure. A tool for constructing these Institution specifications is available [ECS02].

We will not go further into the into the formal definition of EI here as these details are presented in [ERS$^+$01]. We note that Electronic Institutions are defined as instances of state-charts [Har87]. The finite-state nature of the EI is a useful property which enables a range of automated verification techniques, such as model checking [BGS98, HW02], to be performed. The utility of finite-state models in the specification of protocols has been amply demonstrated in the specification of hardware systems [CGP99]. Institutions can be readily extended to infinite-state systems, but we consider that the need for such extensions has yet to be adequately demonstrated.

EI provide a controlled environment in which groups of agents can successfully co-operate. The issues outlined in Section 2, for which agent protocols were proposed, are resolved as follows:

1. Only agents which are relevant to a particular task are admitted into the Institution or scene by the performative structure.

2. Agents assume distinct roles which identifies their capabilities.

3. Negotiation between agents is controlled inside scenes.

4. Normative rules define the obligations and commitments for agents.

5. Asynchrony in the system is controlled by the use of a state-based model.

The use of the EI framework addresses a large number of issues in the design of MAS. However, our experience in the implementation of such models has highlighted a number of issues which have yet to be addressed in this framework. There are several ambiguities in the specification, as defined in [ERS$^+$01], such as how agents move between Institutions. However, such issues can readily be addressed and therefore are not of importance here. Rather, there are three pertinent issues which we consider to be real obstacles in the deployment of Institution-based MAS:

1. There is no mechanism for disseminating Institution protocols.

2. The topology of the Institutions are static.

3. Institutions need synchronisation to ensure that they operate smoothly.

The dissemination of Institution protocols to individual agents appears to be a significant problem. This problem arises when a new agent wishes to participate in an Institution. The agent must follow the design of the Institution, but this requires that the agent knows the internal details of the Institution. The current approach used to design agents which will operate inside an Institution is to construct a plan for the agent with respect to a particular Institution, and then synthesise an agent description from this plan [VSSQ02]. The assumption in this approach is that it will be known in advance which scenes the agent will participate in. This assumption is valid for a number of simple Institutions, but breaks down when used with more complex models. For example, if we consider our medical diagnosis scenario, the patient will not be aware of which scenes are required until a diagnosis is performed by the doctor. Thus, in order to synthesise such an agent it is necessary to consider all possible outcomes of the diagnosis and design the agent to cope with all such possibilities. If we consider all possible medical diagnoses then the problem quickly becomes intractable.

The second issue concerning the static topology of Institutions is closely related to the dissemination problem. The synthesis of plans for particular Institutions requires that the Institution descriptions are known and do not change. If the Institution definition changes then the corresponding plans will also change and the agents must be re-synthesised. The synthesis process is not fully automated and thus the agents may need substantial reworking even for minor changes. This will not be an issue if the Institution definitions are guaranteed to be immutable, but one can imagine many scenarios where this is undesirable. For example, if one wishes to streamline some medical procedure, or compare the performance of competing medical procedures then this cannot be easily achieved with the current framework.

The final issue concerns the synchronisation of agents within an Institution. The performative structures and normative rules contain conditions, which restrict the behaviours of the agents. Similarly, the agent protocols rely on agents being aware of the current state of the Institution, and where they are expected to interact. However, the enforcement of this synchronisation raises a number of questions when implementing these systems. The enforcement techniques which have been proposed [VSE02] rely on the use of administrative agents, or agent proxies to ensure the smooth running of the Institutions. However, as stated in the introduction, this is arguably contrary to the underlying principles of self-contained autonomous agents.

4 Flexible Protocols

The apparent shortcomings of the EI framework discussed above can be addressed by adopting a more dynamic approach to the specification of agent protocols. Our approach relaxes the requirement that all agent protocols must be defined before evaluation can take place. Rather, the agent protocol is built dynamically during evaluation, as the agents communicate. It should be noted that our approach is not a complete replacement for the EI framework, rather it is a replacement for the specification of scene descriptions. We retain the notion of an Institution, the partitioning of Institutions into scenes, and the associated normative rules and dialogic framework.

In order to understand our approach, it is helpful to consider another analogy. As stated earlier, the Electronic Institutions framework is based on the analogy of actors performing in a production. Within a scene, the protocol is described by a global script, and the actors (i.e. agents) stick rigidly to this script. Our approach adopts a different analogy, that of agents involved in a *conversation.* The protocol is delivered in a piecemeal manner as the conversation progresses. Mixing the analogies, our approach can be seen as a modification such that the script is passed between agents. However, this script can be extended and amended as the conversation progresses.

In our medical example, a patient comes to a doctor for diagnosis. The outcome of this diagnosis determines the resulting actions of the patient. The patient does not know in advance what these actions will be. For example, the doctor may say *"Go to the screening clinic for further diagnosis"* or *"You just need a good night sleep"*. In effect, the diagnosis presented by the doctor informs the patient of the next steps in the protocol. In the EI framework, the definition would require at this point that all possible actions were made explicit, and the patient would make a choice among these actions. However, this does not present a natural fit, where the space of choices may be large and may change over time.

In general, a great deal of conversation is an expression of protocol. For example, the question *"Can you tell me the time?"* implies that the next step in the protocol is for the other party to respond with the time (or an apology). Similarly, the statement *"Let's meet for drinks at 8:30pm in the Peartree"* implies a sequence of protocol steps which will be followed. In real conversation, these steps are usually implicit, determined by social norms. Thus, to make this applicable to MAS, all these steps are made explicit in the conversation.

The embedding of protocol inside conversations addresses all of the perceived shortcomings of the EI approach. Protocols no longer need to be distributed to all the agents in advance, as the protocol is transmitted during the conversation, i.e. as messages are exchanged between agents. Changes can easily be made to the protocols by revising or amending conversation patterns, e.g. by modifying the definition of an agent. Finally, the flow of the conversation synchronises the agents without the need for intervention by proxy or administrative agents. It should be noted that the related work on Conversation Policies [GHB99] does not adopt this approach, but is also based on static state-chart models of agent interactions.

$$
\begin{array}{rcllll}
P & \in & \text{Flexible Protocol} & ::= & A^{\{n\}} & \\
A & \in & \text{Agent Protocol} & ::= & \texttt{agent(id) ::}\ op & \\
op & \in & \text{Operation} & ::= & \phi & \\
 & & & \mid & \texttt{(}\ op\ \texttt{)} & \text{(Precedence)} \\
 & & & \mid & op_1\ \texttt{=>}\ op_2 & \text{(Send)} \\
 & & & \mid & op_1\ \texttt{<=}\ op_2 & \text{(Receive)} \\
 & & & \mid & op_1\ \texttt{then}\ op_2 & \text{(Sequence)} \\
 & & & \mid & op_1\ \texttt{or}\ op_2 & \text{(Choice)} \\
 & & & \mid & op_1\ \texttt{par}\ op_2 & \text{(Parallel Composition)} \\
\phi & \in & \text{Term} & ::= & \texttt{id} & \text{(Identifier)} \\
 & & & \mid & \rho & \text{(Performative)} \\
 & & & \mid & \epsilon & \text{(No Action)}
\end{array}
$$

Figure 3: Abstract Syntax for Flexible Protocols.

We will now present a formal description of our method, which we term *flexible protocols*. The first stage in this description is the definition of a language for expressing these protocols. This language will be used during the exchange of messages to express the protocol. A simplified description of this language appears in Figure 3. A flexible protocol P comprises a sequence of agent protocols A. An agent protocol comprises an agent definition together with terms

ϕ composed with operators *op*. Terms are either performatives (i.e. messages) ρ, agent identifiers `id`, or empty actions ϵ. The operators define the exchange of messages and the sequence of these exchanges. The operators are essentially those of the CCS process algebra [Mil89]. This similarity is intentional as it will assist with future work on the verification of these protocols.

```
agent(patient) ::
    request(appoint) => doctor then
    offer(symptoms) => doctor then ...

agent(doctor) ::
    accept(appoint) => patient then
    (refer(clinic) => patient or norefer => patient) then
    doctor
```

Figure 4: General Practitioner Protocol Example.

The flexible protocol language which we have defined can be used to encode state-machine protocol models such as those found in EI models. It should be noted that our language is more expressive than state-based evaluation as is permits parallel composition of actions. As an example, Figure 4 is an encoding of the simplified General Practitioner protocol previously presented as a state-based model in Figure 1. The description is relatively straightforward as it consists of a sequence of messages sent between the `doctor` and `patient` agents. Note that the language only specifies the protocol, it does not attempt to define any reasoning internal to the agents. For example, there are no conditions specified in the doctor agent to distinguish between a `refer` and `norefer` diagnosis. The final step in the doctor agent indicates that the agent should restart from the beginning.

	`agent(patient) ::`	`agent(doctor) ::`
1	`(request(appoint) => doctor,` ϵ`)`	

2		`(accept(appoint) => patient,`
		`offer(symptoms) => doctor)`

3	`(offer(symptoms) => doctor,` ϵ`)`	

4		`((refer(clinic) => patient or`
		`norefer => patient) then`
		`doctor , ...)`

5	`...`	

Figure 5: Flexible Protocol Exchange.

At this point, our definition is loosely equivalent to the description of a scene in the EI framework. In order to obtain the flexible nature of our protocol description we require an additional definition. We define a *message* between agents as a tuple operators (op_1, op_2). The first component of the message is the reply to a previous message and the second contains the steps of the protocol which the receiver should follow (i.e. an explicit representation of the commitments and social norms). Using this definition we can re-state the General Practitioner protocol as the series of message exchanges shown in Figure 5.

At Step 1 in the example, the patient requests an appointment with the doctor. However, at this stage in the exchange the patient does not know the remaining steps in the protocol and therefore ϵ is passed as the second component of the message. The doctor responds in Step 2 with an acceptance of the appointment and, through the use of the second argument, also informs the patient that a list of symptoms is required. The patient returns these symptoms in Step 3, but is again unaware of any future steps in the protocol. The doctor then makes a diagnosis and will return any future actions for the patient in the message.

This definition achieves our aim of providing a flexible protocol exchange between agents. Agents now inform each other of the protocol as evaluation proceeds, and the agents are synchronised by this exchange. We conclude our definition here for this paper, although there are a number of straightforward extensions which can be made. The protocol can be modified by introducing new agents, or through on-the-fly rewriting during the exchange of messages. Furthermore, the agents may participate in multiple protocol exchanges at the same time. All of these are benefits which could not be realised by static state-based models.

5 Conclusions and Further Work

In this paper we presented a novel technique for representing agent protocols in Multi-Agent Systems. We have termed this new style of representation *flexible protocols*. Our intention was to address specific issues relating to the implementation of static state-based models of agent protocols found in the definition of Electronic Institutions [ERS$^+$01].

The definition of our flexible protocols utilises the flow of messages between agents to synchronise and disseminate agent protocols. This technique appears to provide a good abstraction for the process of protocol exchange in real conversation. It allows the protocol to proceed naturally along with the patterns of interactions between agents. In particular, it allows the protocol to adapt to sequences of events, such as the phased diagnosis of a medical condition in our example. Previously, a static agent protocol had to anticipate all possible event sequences. Flexible protocols also permit protocol changes to be easily integrated into systems during evaluation.

Our definition is intended to assist with the implementation of agent protocols. To this end, we have implemented a proof-of-concept interpreter for our language in Prolog and are currently proceeding with an implementation in Java. An important issue relating to this implementation is the verification of our flexible protocols. With the previous static state-based models it was relatively straightforward to examine the protocols for errors or omissions, e.g. by using graph traversal algorithms. However, the flexible nature of our protocols significantly complicates the nature of this analysis. To address this issue we are currently investigating the use of model-checking techniques [CGP99] to perform automated verification. The advantage of this approach is that it leads to an exploration of the space of all possible conversations.

In this paper we have defined a simple language for expressing our flexible protocols. The operations of this language are derived from those of the CCS process algebra [Mil89]. In order to extend this language, an obvious direction would be to utilise more recent results from the use of process algebra in the specification of concurrent systems. In particular, we see the Ambient Calculus [CG98] as a good candidate for improving the expressiveness of our language. Adopting an Ambient-style semantics would enable us to express the boundaries between scenes and institutions in our language in addition to specifying the actual agent protocols. This would provide a unified approach to the specification of all the levels of an Institution, though this currently remains as future work.

References

[BFG$^+$95] H. Barringer, M. Fisher, D. Gabbay, G. Gough, and R. Owens. MetateM: an Imperative Approach to Temporal Logic Programming. *Formal Aspects of Computing*, 7(E):111–154, 1995.

[BGS98] Massimo Benerecetti, Fausto Giunchiglia, and Luciano Serafini. Model Checking Multiagent Systems. *Journal of Logic and Computation*, 8(3):401–423, June 1998.

[Bra87] M. E. Bratman. *Intention, Plans, and Practical Reason.* Harvard University Press, 1987.

[CG98] Luca Cardelli and Andrew D. Gordon. Mobile Ambients. In Maurice Nivat, editor, *Foundations of Software Science and Computational Structures*, number 1378 in Lecture Notes in Computer Science, pages 140–155. Springer-Verlag, 1998.

[CGP99] Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model Checking.* MIT Press, 1999.

[CL90] P. R. Cohen and H. J. Levesque. Rational interaction as the basis for communication. *Intentions in Communication*, pages 221–256, 1990.

[ECS02] Marc Esteva, David de la Cruz, and Carles Sierra. ISLANDER: an electronic institutions editor. In *Proceedings of the First International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS)*, pages 1045–1052, Bologna, Italy, July 2002. ACM press.

[ERS$^+$01] M. Esteva, J. A. Rodríguez, C. Sierra, P. Garcia, and J. L. Arcos. On the Formal Specification of Electronic Institutions. In *Agent-mediated Electronic Commerce (The European AgentLink Perspective)*, number 1991 in Lecture Notes in Artificial Intelligence, pages 126–147, 2001.

[FG97] Stan Franklin and Art Graesser. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In *Proceedings of the Third International Workshop on Agent Theories*, number 1193 in Lecture Notes on Artificial Intelligence, pages 21–36. Springer-Verlag, Budapest, Hungary, 1997.

[FIP99] FIPA Foundation for Intelligent Physical Agents. *FIPA Specification Part 2 - Agent Communication Language*, April 1999. Available at: `www.fipa.org`.

[GHB99] Mark Greaves, Heather Holmback, and Jeffrey Bradshaw. What is a Conversation Policy? In *Proceedings of the Workshop on Specifying and Implementing Conversation Policies, Autonomous Agents '99*, Seattle, Washington, May 1999.

[Har87] D. Harel. Statecharts: A Visual Formalism for Computer System. *Science of Computer Programming*, 8(3):231–274, 1987.

[HW02] Wiebe van der Hoeck and Michael Wooldridge. Model Checking Knowledge and Time. In *In Proceedings of the 9th SPIN Workshop*, number 2318 in Lecture Notes in Computer Science, pages 95–111, Grenoble, France, April 2002. Springer Verlag.

[Jen00] N. R. Jennings. On Agent-Based Software Engineering. *Artificial Intelligence*, 117(2):277–296, 2000.

[Mil89] Robin Milner. *Communication and Concurrency.* Prentice-Hall International, 1989.

[MLF96] J. Mayfield, Y. Labrou, and T. Finin. Evaluating KQML as an Agent Communication Language. In *Intelligent Agents II*, volume 1037 of *Lecture Notes on Artificial Intelligence*, pages 347–360. Springer-Verlag, 1996.

[Nwa96] Hyacinth S. Nwana. Software Agents: An Overview. In *Knowledge Engineering Review*, volume 11, chapter 3, pages 1–40. Cambridge University Press, September 1996.

[PGW02] S. Parsons, P. Gmytrasiewicz, and M. Wooldridge. *Game Theory and Decision Theory in Agent-based Systems*, volume 5 of *"Multi-agent Systems" International Book Series.* Kluwer Academic Publishers, June 2002.

[RMN$^+$00] J. A. Rodríguez, F. J. Martín, P. Noriega, P. Garcia, and C. Sierra. Towards a Formal Specification of Complex Social Structures in Multi-Agent Systems. In J. Padget, editor, *Collaboration between Human and Artificial Societies*, number 1624 in Lecture Notes in Artificial Intelligence, pages 284–300. Springer Verlag, 2000.

[VRA$^+$01] Wamberto Vasconcelos, David Robertson, Jaume Augustí, Carles Sierra, Michael Wooldridge, Simon Parsons, Chris Walton, and Jordi Sabater. A Lifecycle for Models of Large Multi-Agent Systems. In *Proceedings of the 2nd International Workshop on Agent-Oriented Software Engineering (AOSE-2001)*, Montreal, Canada, May 2001. ACM press.

[VSE02] Wamberto Vasconcelos, Carles Sierra, and Marc Esteva. An Approach to Rapid Prototyping of Large Multi-Agent Systems. In *Proceedings of the 17th IEEE International Conference on Automated Software Engineering (ASE-2002)*. IEEE, September 2002.

[VSSQ02] Wamberto Vasconcelos, Jordi Sabater, Carles Sierra, and Joaquim Querol. Skeleton-based agent development for Electronic Institutions. In *Proceedings of the First International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS)*, pages 696–703, Bologna, Italy, July 2002. ACM press.

[Woo00] Michael Wooldridge. *Reasoning about Rational Agents.* MIT Press, 2000.