Three Approaches to the Coordination of Multiagent Systems

Federico Bergenti
DII
Università degli Studi di Parma
Parco Area delle Scienze, 181/A
43100, Parma, Italy
bergenti@ce.unipr.it

Alessandro Ricci
DEIS
Università degli Studi di Bologna
Via Rasi e Spinelli, 176
47023, Cesena (FO), Italy
aricci@deis.unibo.it

ABSTRACT

The engineering of the social aspects has been acknowledged as one of the principal issues in the realization of real-world multiagent systems. The literature proposes a number of solutions to this problem and in this paper we consider three of them and we discuss their relationships. First, we take into account hybrid coordination models based on tuple centres. Then, we consider interaction protocols as a means to coordinate multiagent systems. Finally, we address the implicit coordination that the semantics of classic agent communication languages propose. We base our discussion on the common ground of coordination models and we sketch a comparison between such approaches.

Keywords

Multiagent System Coordination, Coordination Models, Multiagent System Engineering

1. INTRODUCTION

Multiagent systems (MASs) root their origins in artificial intelligence (AI) and the multidisciplinary nature of AI justifies the number of different point of views that researchers adopt to deal with MASs. This is the case also for the specific issues of coordination: they have been addressed by several research communities and count a number of different approaches [22, 21]. These different research paths share the same goal: face social aspects of MASs as first class issues, providing not only mechanisms, but also models of coordination. Typically, these social aspects concern inter-agent dependencies, namely the configuration of the system in terms of the basic interaction means, agent generation/destruction, and organisation of the environment [27]. The introduction of such models provide a base to treat coordination in the scope of agent oriented software engineering (AOSE) thus enabling the realization of real-world MASs [32].

Different approaches to coordination impact differently on the engineering of MASs and it would be important to have a conceptual framework (or a meta-model) to classify and to compare these approaches. The literature proposes some frameworks for the comparison of these approaches, however we believe that they are not sufficient to deal with the peculiar problem of managing the coordination of MASs. In particular, these frameworks (*i*) focus on mechanisms and strategies, not considering models [10], or (*ii*) they are focused on a specific research area and unable to capture the features of such heterogeneous models [14], or (*iii*) they are not enough specific to be effective in the engineering of open and dynamic MASs [29, 22]

We need a novel framework for comparing the various approaches to coordination that are available in the literature, but the introduction of such a framework and the comprehensive comparison of the available approaches would require a book on its own. In this paper we restrict our focus considering three approaches that we believe represent well the state of the art of three different communities. Hybrid models based on tuple centers are among the more recent proposals from the coordination community for the coordination of MAS. Interaction protocols are the one and only choice of the agent community. The implicit coordination that classic agent communication languages (ACLs) provide is an upcoming proposal from the more formally inclined part of the agent community. This proposal is meant to overcome many of the well-known limitations of interaction protocols.

We discuss such approaches framing their concepts within the classic ideas of a coordination models [4]. A coordination model consists of three elements: *(i)* the *coordinables*, i.e., the objects of the coordination; *(ii)* the *coordination media*, i.e., what enables the interaction between the coordinables; *(iii)* the *coordination laws*, i.e., the laws that govern the interaction between the coordination media and the coordinables, and the rules that the coordination media employs.

The rest of this paper uses the terminology of coordination models to analyse and then compare three different approaches to the coordination of MASs. In particular, section 2 introduces tuple centres. Section 3 consider the approach that the agent community is promoting for coordinating agents, i.e., interaction protocols. Section 4 shows that some classic work on the semantics of ACLs implicitly

SAC 2002, Madrid, Spain
Copyright 2002 ACM 1-58113-445-2/02/03 ...$5.00.

provides a coordination model.

2. COORDINATION THROUGH TUPLE CENTRES

The models of coordination that rely on tuple centres origin from the community of programming languages for parallel and distributed systems. Starting from the work of Gelernter and Carriero [13] and of Wegner [31], coordination emerged as a fundamental dimension to be placed orthogonally, side by side, to computation in programming and developing concurrent and distributed systems. The aim was to promote a clear separation between the specification of components of the computation and the specification of their interactions or dependencies. This separation enables some important engineering properties, among the other the reuse of components and reuse of coordination specifications.

The first coordination models and languages developed – such as Linda, Gamma, Manifold to cite the representatives of large families – were focused on basic coordination mechanisms and activities, such as synchronization, exchange of simple information, notification and reaction (see [23, 22] for comprehensive surveys).

The evolution of these first models has been toward three main directions:

- more complex communication languages (XML documents in MARS-X, objects in Objective Linda, or forms used in Laura);
- more complex topologies used to organize coordination media (such as multiple space such as in TuCSoN or nested spaces in Bauhaus Linda and PageSpace);
- more expressive and powerful coordination media – starting from providing more expressive coordination languages (with new primitives such as in Bonita), toward a more general and comprehensive approach: making the coordination medium *programmable* [6]; important examples are, among the others, TuCSoN, MARS, PoliS, Law-Governed Linda.

The programmability of the coordination medium is the key characteristic of *hybrid coordination models* [16, 27]. Hybrid models provide the benefits of both information oriented models, in term of cleanness and elegance – agent interaction protocols can designed as simply and expressively as in data-driven models –, and control oriented models, in term of flexibility and power – the agent interaction space can be easily managed by the control-driven mechanisms provided by the coordination medium [4].

These characteristics make hybrid models suitable for multiagent systems [8].

In particular, hybrid models based on the *tuple centre* abstraction [18], such as TuCSoN [19] and LuCe [9], seem to be flexible enough to provide support for engineering of the social aspects of MASs as first class issues. Tuple centres can be understood as general purpose *coordination virtual machines* – with a formally-defined operational semantic [18] – which can be programmed in order to react to communication events, executing computations whose nature and properties depend on the language used to specify the behaviour. Consequently, social tasks, social rules and global constraints can be specified as coordination laws and embodied as behaviour specification of tuple centres.

The flexibility of the tuple centre models comes from clearly identifying as separate abstractions the communication languages and the language used to specify the behaviour of the coordination media. TuCSoN and LuCe models, for instance, use logic tuples as communication language and the ReSpecT language to program the behaviour of tuple centres [7]. In principle multiple programming languages providing different features – logic-based, low-level with formal semantic foundation such as ReSpecT, or high level, but without strong formal foundation such as Java as happens in MARS – could be used to program tuple centres, according to the needs of the application domain.

By virtue of these characteristics, tuple centre models support *objective* [27] or *uncoupled* [29] coordination: they effectively provide the separation of the individual perception of coordination and the global/social coordination issues (tasks, constraints, rules), which can be described with proper languages (such as ReSpecT) and embedded inside tuple centre as *social abstraction*, *outside* the coordinated agents.

Summing up, in tuple-centre based models the coordinables are agents using Linda-like coordination language to interact, tuple centres are the coordination medium and the programs used to specify tuple centre behaviour constitute the (not fixed, programmable) coordination laws. The properties of the coordination laws (complexity, specificity, general purpose-ness) depend on the language used to program tuple centre; for instance, in TuCSoN and LuCe models – by virtue of the Turing-equivalence of the ReSpecT language – general purpose coordination laws can be expressed, specified as reactions with transactional semantic.

3. COORDINATION THROUGH INTERACTION PROTOCOLS

A large portion of the agent community is currently working on the problem of coordination using interaction protocols, i.e., predetermined patterns of interaction. This approach is based on considering coordination essentially as a problem of communication. Communication plays a fundamental role for agents and many researches consider the so called *social ability* a characterizing feature of agents [33]. Some ACL have been designed to support the social nature of agents. The most important of them, e.g., KQML [11] and the FIPA ACL [12], rely on agents communicating through asynchronous messages that are structured in terms of (*i*) a performative verb, or simply performative, and (*ii*) a content. The performative expresses the intent of the agent in sending the message. For example, a FIPA agent, i.e., an agent that communicate using the FIPA ACL, employs a performative called `inform` if its intent in sending the message is informing another agent of something that it believes true. Each ACL provides its own set of performatives, e.g., the FIPA ACL provides a set of performatives that contains `inform`, `request`, `propose` and many others. The content together with the performative associate a meaning with a message. For example, if the content of an `inform` message is the proposition `fatherOf(Mary, John)`, then the sender utters the message to inform the receiver that it believes that John is Mary's father.

Structuring messages in terms of a performative and a content roots to the speech-act theory [28] and it has been recognized recently as a means to achieve an higher-level of interoperability between agents [1]. This particular structure suggested a way for describing interaction protocols. Classically, an interaction protocol relies on a set of roles that the agents in the protocol play and it is described as a finite state machine where:

- states identify global states of the protocol;
- transitions represent messages and they are labelled with a role identifier and a performative. For any transition, agents playing the associated role can send a message that uses the associated performative.

A typical example of interaction protocol is shown in figure 1. This protocol, called FIPA contract net, encompasses two roles: the manager and the contractor. The manager sends a `cfp` (call for proposal) to all contractors and waits for contractors to `propose`. Then, it decides to accept one proposal, sends an `accept_proposal` to one contractor, while sending a `refuse_proposal` to the others. The protocols end when the accepted contractor `inform` the manager that the negotiated activity has been performed.

Interaction protocols are a coordination model. The coordinables are, obviously, agents. The coordination medium is the ACL, while the coordination laws are expressed through the finite state machine that describes the protocol.

4. COORDINATION THROUGH THE SEMANTICS OF ACLS

ACLs have long been criticized for their lack of formal semantics. A formal semantics for KQML have been proposed [15] and when FIPA chose to define its own ACL it also introduced a formal semantics. FIPA borrowed such a semantics from Sadek's ARCOL [26]. Anyway, the debate on an accepted semantics for ACLs is far from concluded and researchers find no agreement on this topic beyond that a formal semantics is needed. For our purpose of showing how the semantics of an ACL implies a coordination model, the semantics of the FIPA ACL is sufficient and we prefer to avoid this debate.

FIPA associates a meaning to a message in terms of a *feasibility precondition* (FP) and a *rational effect* (RE). The feasibility precondition states what must be true for an agent to send a message, while the rational effect shows what the agent is intending to achieve with such a message. For example, a simplified version of the semantics of `inform` messages is [12]:

$< s, \texttt{inform}\ (r, \phi) >$
$FP(\phi) : B_s\phi$
$RE(\phi) : B_r\phi$

Which means that agent s informs agent r of ϕ only if it believes that ϕ is true ($FP(\phi) = B_s\phi$) and its objective in sending such a message is that agent r comes to know that ϕ is true ($RE(\phi) = B_r\phi$). In this example, only the performative is fixed, i.e., `inform`, while the content of the message ϕ is any.

Another example is the following simplified semantics for `request` messages:

$< s, \texttt{request}\ (r, \alpha) >$
$FP(\alpha) : B_s(Feasible(\alpha))$
$RE(\alpha) : Done(\alpha)$

Which means that agent s request agent r to perform action α only if it believes that α is feasible ($FP(\alpha)$) and its objective in sending such a message is that action α is ultimately done by some agent ($RE(\alpha)$).

The availability of feasibility preconditions and rational effects is sufficient for an agent to decide autonomously when to send which message. If the agent has a goal g that matches the rational effect of a message for a given content γ, i.e., $g = RE(\gamma)$, then the agent might decide to send such a message. In order for this to done, the message must be feasible and the agent might decide to bring about a new subgoal $g' = FP(\gamma)$. This process is a classic, and trivial, planning technique that leads to the dynamic construction of a plan.

The semantics of the FIPA ACL implicitly defines a coordination model because the dynamic construction of a plan takes into account other agents and implicitly coordinates the activities between agents. Needless to say, the coordinables are the agents. The medium is the ACL, while the coordination laws are implicitly expressed through the semantics of the ACL. In particular, the coordination laws emerge from the interaction of the goals the agents have, i.e., the rational effects of sending messages and performing actions, and any constraint on the achievement of such goals, i.e., the feasibility preconditions on messages and actions.

5. DISCUSSION

In this section we focus on some important properties that impact on the engineering of coordination in MASs.

Tuple centre based models are particularly suitable for coordination of mobile agents in open and dynamic environments like the Internet [3]. Essentially, two are the main reasons: first, the temporal and spatial uncoupling properties of generative communication – on which these models are based – make more agile the communication among autonomous (mobile) agents, which can interact without knowing each others or without establishing temporal meeting points. Second, objective coordination supported by these models allows to enable effective coordination despite of the openness and dynamism of the environment, without needing an agreement among the coordinated agents.

Objective coordination makes these models suitable for heterogeneous MAS: tuple centre based models don't require any specific computational model for agents – which could be high level like BDI agents as well as low level like simple threads. Therefore, they can be used with different agent models, providing a uniform conceptual framework for interaction and coordination.

The main critic coming from the agent community to tuple centre models is that they induce some form of centralization in MAS. Actually, these models conceive the global agent interaction space as a *multiplicity* of independent interaction contexts, organized according topologies that depends on the specific model: in TuCSoN, for instance, tuple centres belong to (network) nodes, organized in trees [5]. Typically, agent societies are build upon a dynamic set of decentralized tuple centres, according to the social aspects to be represented and enforced. This *context-oriented* vision has been promoted also by other approaches, in particular focusing on mobility and its relationship with coordination [2, 25].

Part of the agent community (in particular FIPA groups) also criticizes the absence of a (high level) semantic for the (Linda-like) coordination language used by tuple centre models. However, these coordination models have been

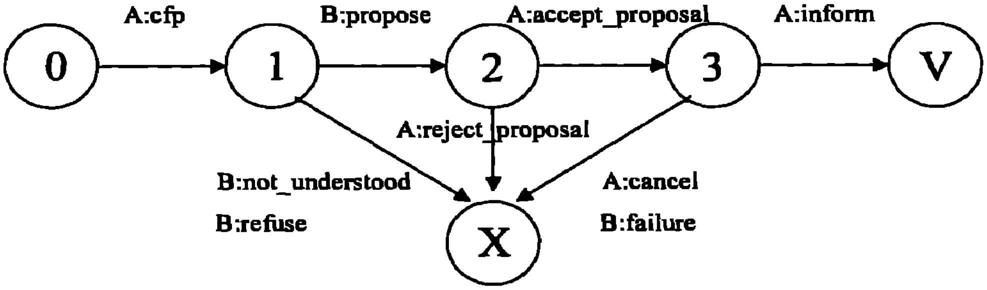

Figure 1: The FIPA contract net protocol.

designed explicitly to enable coordination relying only on *observable behaviour* of agents, without any assumption about their mental attitudes. Again, this choice seems to be particularly effective to enable coordination in open environments, where no assumption about benevolence or veracity of agents can be made.

Interaction protocols are a coordination model that has been particularly criticized both from the agent community and the coordination community. The agent community does not criticize the idea of interaction protocol, i.e., of a predefined pattern of interaction, but it heavily criticize the description in terms of finite state machines with the characteristics showed in section 3. The problems of using such a description are:

- it does not consider formally the contents of messages, e.g., the specification of the FIPA contract net does not state what a contractor might include in its proposal;
- it does not provide mechanisms for shortcutting the protocol in favourable situations, e.g., in the FIPA contract net when only one contractor is available, the manager cannot decide to hire it without engaging the whole protocol;
- it is hard to verify that messages comply with the semantics of the ACL, e.g., for the case of the FIPA contract net, no formal proof is available that sending a propose in response to a cfp is coherent with the semantics of such performatives.

A small group of researchers is facing the problem of finding alternative means for specifying interaction protocols. For example, Singh proposes to formalize interaction protocols in terms of commitments between agents playing different roles [30]. Pitt and Mandhani propose interaction protocols as a means to define the semantics of ACLs [24]. Such proposals are not yet accepted because they require to change the semantics of the ACL.

Parts of the coordination community criticize interaction protocols because they couple coordination and computation. The useful part of the code of an agent is interleaved with code that is there only to support coordination.

The study of the coordination capabilities of the semantics of ACLs is a rather novel theme and no previous work with a reasonable importance is available. The problem of this approach is that it heavily demands on the agent's capabilities. Agents must integrate a full-featured planner to exploit such an approach completely. Anyway, it provides a purely goal-oriented coordination, i.e., interaction is performed only to achieve goals and it is not an activity on its own. Notably, this approach subsumes that of interaction protocols because agents might use the semantics of the ACL to assemble interaction protocols on the fly. This suggests that the use of both approaches can relax the requirements on the agent's capabilities. For example, a trade off between these approaches is proposed in the PARADE framework [1].

We conclude the discussion comparing models features according to some basic criteria / properties, which we found interesting and useful from an engineering point of view.

Design Focus

Tuple centre modes enhance design focus because the interaction between agents can be designed by focusing on their individual tasks (e.g., concentrating on the information needed and produced / inferred by the agent in the process of achieving its tasks), in many sense disregarding the social tasks of the groups the agent belongs to [4]. This is the case also for coordination based on the semantics of the ACL: the interaction is not designed at all and the focus is only on agent's tasks.

Coordination Design

Tuple centre models allow social rules to be represented in term of coordination laws, changed upon the coordination media, and can be designed and implemented independently, with no concern for the internal structure of the agents [4]. This is the case also for the coordination based on the semantics of the ACL because the coordination laws are expressed through the semantics of the performatives.

Modularity and Reusability

Autonomous agents focusing on their tasks, with no concern with coordination issues and designed around very straightforward interaction protocols, are a natural source of modularity of multiagent systems. They can be reused wherever their capabilities are needed, independently of the social rules. Dually, coordination rules can be exploited to achieve social tasks simply given the agent goals and protocols, independently of the internal structure of the agents [4]. In its more pure form, only coordination based on the semantics of the ACL provides support this characteristic. Agents take such semantics form a library and they are not concerned at all about the coordination issues. Also tuple centre models might exhibit such a feature to some extend, but it depends how the developer decide to use the features they provide.

Incremental Design and Implementation

In tuple centre based models, once the social rules are designed, individual capabilities can be refined so as to improve the agent ability to achieve its task independently of the rest of the systems. Analogously, once that agents tasks and interaction protocols are designed and implemented, coordination rules can be independently refined so as to improve the capability of the multiagent system to achieve its social goals [4]. Basically also the other two models exhibit to some extent this characteristic, even if in tuple centre based models this feature is more evident, thanks to the complete separation between coordinated agents and coordination rules, encapsulated in tuple centres.

Support for Prescriptive Coordination

Prescriptive coordination is the possibility to constrain agent interactions to reflect sound behaviours according to the social goals defined for the agent society, balancing agents' autonomy and the enforcing of coordination activities. This can be achieved with all three models we analysed; however *(i)* it is harder to achieve it practically through interaction protocols because of their possible lack of semantics; *(ii)* models supporting objective coordination, such as tuple centre based ones, can be prescriptive without relying on any assumption on agents' veracity, skills and knowledge about coordination activities, which are required by the other two approaches.

Support for Formal Approaches to Interaction and Coordination

Tuple centres are interaction contexts which defines, by virtue of their operational semantic, a total order of the communication events (and reactions) managed, and a local notion of time [18, 17]. This feature – which is provided by the model – enables consistent observation of interactions happening in the social context defined by a single tuple centre. Then, in principle, formal approaches and tools (such as temporal logics, model checkers) can be exploited to reason about social context evolution, to verify properties of coordination activities. The same features can be provided by the other two approaches, but as mechanisms or implementation strategies, and not as feature of their models.

6. CONCLUSIONS

We discussed three representative approaches for the coordination in multiagent systems, coming from different research areas (distributed artificial intelligence and parallel computing in primis), and we provided a comparison, focusing on their impact on the engineering of MASs.

Actually, some comparative surveys about coordination models can be found in literature [23, 22, 29], but either they do not cover the heterogeneity of the coordination approaches found in MASs, or they do not account for the engineering impact of the models on the design and development.

This paper is not meant to propose a final choice between the coordination models that we analysed. We believe that such a choice is not possible until we focus on the engineering properties of the system we want to build. Nevertheless, the comparison between such models helps understanding them better and provides some guidelines for choosing the right model for each system.

7. REFERENCES

[1] F. Bergenti and A. Poggi. A development toolkit to realize autonomous and inter-operable agents. In *Proceedings of the 5th International Conference on Autonomous Agents (Agents 2001)*, pages 632–639, Gaithersburg, Maryland, 2001. ACM Press.

[2] G. Cabri, L. Leonardi, and F. Zambonelli. Mobile-agent coordination models for internet applications. *IEEE Computer*, 33(2):82–89, Feb. 2000.

[3] P. Ciancarini, A. Omicini, and F. Zambonelli. Coordination technologies for Internet agents. *Nordic Journal of Computing*, 6(3):215–240, Fall 1999.

[4] P. Ciancarini, A. Omicini, and F. Zambonelli. Multiagent system engineering: the coordination viewpoint. In N. R. Jennings and Y. Lespérance, editors, *Intelligent Agents VI — Agent Theories, Architectures, and Languages*, volume 1767 of *LNAI*, pages 250–259. Springer-Verlag, February 2000.

[5] M. Cremonini, A. Omicini, and F. Zambonelli. Coordination and access control in open distributed agent systems: The TuCSoN approach. In A. Porto and G.-C. Roman, editors, *Coordination Languages and Models*, volume 1906 of *LNCS*, pages 99–114. Springer-Verlag, 2000.

[6] E. Denti, A. Natali, and A. Omicini. Programmable coordination media. In D. Garlan and D. Le Métayer, editors, *Coordination Languages and Models*, volume 1282 of *LNCS*, pages 274–288. Springer-Verlag, 1997.

[7] E. Denti, A. Natali, and A. Omicini. On the expressive power of a language for programming coordination media. In *Proc. of the 1998 ACM Symposium on Applied Computing (SAC'98)*, pages 169–177. ACM, 1998. Track on Coordination Models, Languages and Applications.

[8] E. Denti and A. Omicini. Designing multi-agent systems around an extensible communication abstraction. In J.-J. C. Meyer and P.-Y. Schobbens, editors, *Formal Models of Agents*, volume 1760 of *LNAI*, pages 90–102. Springer-Verlag, 1999. ESPRIT Project ModelAge Final Report, Selected Papers.

[9] E. Denti and A. Omicini. Engineering multi-agent systems in LuCe. In S. Rochefort, F. Sadri, and F. Toni, editors, *ICLP'99 International Workshop on Multi-Agent Systems in Logic Programming (MAS'99)*, Las Cruces (NM), 30 Nov. 1999.

[10] E. H. Durfee. Scaling up agent coordination strategies. *IEEE Computer*, 34(7), July 2001.

[11] T. Finin, R. Fritzson, D. McKay, and R. McEntire. KQML as an agent communication language. In *Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM*, pages 456–463, Gaithersburg, Maryland, 1994. ACM Press.

[12] FIPA. FIPA 97 specification part 2: Agent communication, Oct. 1998. Version 2.0.

[13] D. Gelernter and N. Carriero. Coordination languages and their significance. *Communications of the ACM*, 35(2):97–107, Feb. 1992.

[14] N. Jennings. Coordination techniques for distributed artificial intelligence. In G. M. P. O'Hare and N. R. Jennings, editors, *Foundations of Distributed Artificial Intelligence*, pages 187–210. Wiley, 1996.

[15] Y. Labrou and T. Finin. Semantics for an agent communication language. *Lecture Notes in Computer Science*, 1365:209–214, 1998.

[16] A. Omicini. Hybrid coordination models for handling information exchange among Internet agents. In A. Bonarini, M. Colombetti, and P. L. Lanzi, editors, *AI*IA 2000 Workshop "Agenti intelligenti e Internet: teorie, strumenti e applicazioni"*, pages 1–4, Milano (I), 13 Sept. 2000.

[17] A. Omicini and E. Denti. Formal ReSpecT. In A. Dovier, M. C. Meo, and A. Omicini, editors, *Declarative Programming – Selected Papers from AGP'00*, volume 48 of *Electronic Notes in Theoretical Computer Science*, pages 179–196. Elsevier Science B. V., 2001.

[18] A. Omicini and E. Denti. From tuple spaces to tuple centres. *Science of Computer Programming*, 41(3):277–294, Nov. 2001.

[19] A. Omicini and F. Zambonelli. Coordination for Internet application development. *Journal of Autonomous Agents and Multi-Agent Systems*, 2(3):251–269, September 1999. Special Issue: Coordination Mechanisms for Web Agents.

[20] A. Omicini, F. Zambonelli, M. Klusch, and R. Tolksdorf, editors. *Coordination of Internet Agents: Models, Technologies, and Applications.* Springer-Verlag, Mar. 2001.

[21] S. Ossowski. *Co-ordination in Artificial Agent Societies: social structures and its implications for autonomous problem solving agents*, volume 1535 of *LNAI.* Springer-Verlag, 1998.

[22] G. A. Papadopoulos. Models and technologies for the coordination of Internet agents: A survey. In Omicini et al. [20], chapter 2, pages 25–56.

[23] G. A. Papadopoulos and F. Arbab. Coordination models and languages. *Advances in Computers*, 46:329–400, 1998.

[24] J. Pitt and E. Mamdani. A protocol based semantics for an agent communication language. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI99),*, pages 486–491, Stockholm Sweden, 1999.

[25] G.-C. Roman, A. L. Murphy, and G. P. Picco. Coordination and mobility. In Omicini et al. [20], chapter 10, pages 253–273.

[26] M. D. Sadek. Dialogue acts are rational plans. In *Proceedings of the ESCA/ETRW Workshop on the Structure of Multimodal Dialogue*, pages 1–29, Maratea, Italy, 1991.

[27] M. Schumacher. *Objective Coordination in Multi-Agent System Engineering – Design and Implementation*, volume 2039 of *LNAI.* Springer-Verlag, Apr. 2001.

[28] J. R. Searle. *Speech Acts: An Essay in the Philosophy of Language.* Cambridge University Press, Cambridge, England, 1969.

[29] R. Tolksdorf. Models of coordination. In A. Omicini, R. Tolksdorf, and F. Zambonelli, editors, *Engineering Societies in the Agents World*, volume 1972 of *LNAI*, pages 78–92. Springer-Verlag, Dec. 2000.

[30] M. Venkatraman and P. Singh, Munindar. Verifying compliance with commitment protocols: Enabling open web-based multiagent systems. In *Autonomous Agents and Multi-Agent Systems*, pages 632–639, Gaithersburg, Maryland, 2001. ACM Press.

[31] P. Wegner. Why interaction is more powerful than computing. *Communications of the ACM*, 40(5):80–91, May 1997.

[32] M. J. Wooldridge. Agent-based software engineering. *IEE Proceedings Software Engineering*, 144(1):26–37, 1997.

[33] M. J. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.