

Chapter 14

COORDINATION INFRASTRUCTURES IN THE ENGINEERING OF MULTIAGENT SYSTEMS

Andrea Omicini, Sascha Ossowski and Alessandro Ricci

Abstract On the theoretical side, coordination is a critical issue for MAS engineering, since it deals with modelling and managing the ever growing complexity of the agent interplay within a MAS. On the practical side, the availability of powerful and robust infrastructures is a key factor to enable and promote MAS as a mainstream software engineering technology. By adopting Activity Theory as a unifying framework for the many existing approaches to MAS coordination, we put forward the notion of *artifact* as a key concept for infrastructures, from which we derive some distinctive properties that a coordination infrastructure should feature. Finally, we discuss how a principled approach to MAS engineering based on coordination infrastructures could be built around such a notion.

1. Introduction

Coordination is one of the key issues in the modelling and engineering of complex systems, and has been the subject of numerous investigations in areas such as Sociology, Economics and Organisational Theory. From an engineering point of view, the question of how to *design* computational mechanisms that allow for efficient coordination is foremost: coordination is conceived as a means to integrate various activities or processes in such a way that the resulting ensemble shows desired characteristics and functionalities. The design of coordination mechanisms is particularly challenging in the field of MAS, as they are usually embedded in highly dynamic environments, and neither the number nor the behaviour of agents can be directly controlled at design time.

In this chapter, we discuss how coordination infrastructures can be used as a means to instill coordination in open multiagent systems. In particular, we claim that additional high-level abstractions need to be integrated into agent-oriented design methodologies in order to exploit the full potential of coordi-

nation infrastructures, and to engineer coordinated MAS in open environments in an efficient and principled manner.

The chapter is organised as follows. Section 2 argues that the key problem of coordination engineering in MAS amounts to the governance of interaction from both the agents' and the designer's point of view. Subsequently (section 3) we outline the role of coordination infrastructures for this task and point to shortcomings in current approaches. Setting out from findings in Activity Theory, section 4 provides a uniform conceptual framework for many different approaches to coordination, and introduces the notion of *coordination artifact* as a key abstraction, that allows for a smooth conceptual integration of coordination infrastructures into MAS design. Section 5 provides clues on how to engineer MAS based on advanced coordination infrastructures within such an integrated conceptual framework. Final discussion concludes the chapter.

2. Coordination in MAS

2.1 Models of Coordination in MAS

Maybe the most widely accepted conceptualisation of coordination in the MAS field originates from work in the area of Organisational Science. (Malone and Crowston, 1994), define coordination as *the management of dependencies* between organisational activities. One of the many workflows in an organisation, for instance, may involve a secretary writing a letter, an official signing it, and another employee sending it to its final destination. The interrelation among these activities is modelled as a *producer/consumer* dependency, which can be managed by inserting additional *notification* and *transportation* actions into the workflow.

It is straightforward to generalise this approach to coordination problems in multiagent systems. Obviously, the subjects whose activities need to be coordinated (sometimes called *coordinables*) are the agents. The entities between which dependencies arise (or *objects of coordination*) are often termed quite differently, but usually come down to entities like goals, actions and plans. Depending on the characteristics of the MAS environment, a taxonomy of dependencies can be established, and a set of potential coordination actions assigned to each of them, e.g., (von Martial, 1992). Within this model, the *process* of coordination is to accomplish two major tasks: first, a *detection* of dependencies needs to be performed, and second, a *decision* respecting which coordination action to apply must be taken. A coordination *mechanism* shapes the way that agents perform these tasks (Ossowski, 1999).

The dependency model of coordination appears to be particularly well suited to *represent* relevant features of a coordination problem in MAS. The TAEMS framework presented by (Decker, 1996a), for instance, has been used to model coordination requirements in a variety of interesting MAS domains. It is also

useful to rationalise observed coordination behaviour along the lines of the knowledge-level perspective put forward by (Newell, 1993). Still, when *designing* coordination processes for real-world MAS, things are not as simple as the dependency model may suggest. Dependency detection may come to be a rather knowledge intensive task, which is further complicated by incomplete and potentially inconsistent local views of the agents. Moreover, making timely decisions that lead to efficient coordination actions is also everything but trivial. The problem becomes even more difficult when agents pursuing partially conflicting goals come into play. In all but the most simple MAS, the instrumentation of these tasks gives rise to complex patterns of interactions among agents. The set of possible interactions is often called the *interaction space* of coordination.

From a software engineering perspective, coordination is probably best conceived as the effort of *governing the space of interaction* of a MAS (Busi et al., 2001). When approaching coordination from a *design* stance, the basic challenge amounts to how to make agents converge on interaction patterns that adequately (i.e., instrumentally with respect to desired features of the agents and/or the MAS as a whole) solve the dependency detection and decision tasks.

2.2 Objective vs. Subjective Coordination in MAS

There are two ways of looking at the space of interaction: from the inside and from the outside of the interacting entities. In the context of multiagent systems, this amounts to say that we can look at interaction within a MAS from either the viewpoint of an agent, or from the viewpoint of an external observer not directly involved in the interaction. According to (Schumacher, 2001), and (Omicini and Ossowski, 2003), these are called, respectively, *subjective* and *objective* viewpoints over coordination.

From the subjective viewpoint of an agent, the space of interaction basically amounts to the observable behaviour of other agents and the evolution of the environment over time, filtered and interpreted according to the individual agent's perception and understanding. From the objective viewpoint, the space of agent interaction is roughly given by the observable behaviour of all the agents of a MAS and of the agent environment as well, and by their mutual interactions – more precisely, by all their *interaction histories* (Wegner, 1997). When adopting the acceptance of MAS coordination as the governance of the agent interaction space, then the two different viewpoints lead to two different ways of coordinating.

When looking at interaction from the individual viewpoint of an agent, *subjective coordination* roughly amounts to (i) monitoring all interactions that are perceivable and relevant to the agent, keeping track of their evolution over time; and (ii) finding out which (sequence of) actions would bring the over-

all state of the MAS (or, more generally, of the agent's world) to match the agent's own goals. So, in general, the acts of an agent that coordinates within a MAS are driven by its own perception and understanding of the behaviour of the other agents', capabilities and goals, as well as of the environment state and dynamics.

On the other hand, when taking an external viewpoint over interaction in a MAS – typically, the designer's viewpoint –, *objective coordination* means either directly or indirectly acting upon agent interaction so as to make the resulting evolution of a MAS accomplish one or more of the observer's (e.g., MAS designer's) goals. In general, the acts of external observers – whether they be MAS designers, developers, users, managers, or even agents working at the meta-level – are influenced not only by their perception and understanding of MAS agents and environment, but also by their a-priori knowledge of the agents' aims, capabilities and behaviour. Furthermore, some form of prediction of the global behaviour of the MAS and its environments is often desirable (Ossowski et al., 2002), so as to instill a coordination that is effective over time from the standpoint of the user.

2.3 Implications for MAS Engineering

Subjective and objective coordination have a different impact over MAS engineering. Subjective coordination affects the way in which individual agents behave and interact, whereas objective coordination affects the way in which interaction among the agent and the environment is enabled and ruled. So, whereas the main focus of subjective coordination is the behaviour of agents as (social) *individuals* immersed in a MAS, the emphasis of objective coordination lies more on the behaviour of a MAS as a whole.

When designing the architecture and the inner dynamics of single agents, the subjective viewpoint on coordination is clearly the most pertinent one. How to model other agents' mental states and to predict their actions, how to interpret and handle shared information in the agent system, when and why to move from an agent environment to another, and so on – all these questions concern subjective coordination, and affect the way in which the agents of a MAS are designed, developed and deployed as individual entities. So, the viability of approaches adopting a subjective coordination viewpoint to the engineering of MAS strictly depends not only on the mental (reasoning, planning and deliberation) capabilities of the agents, but also on their ability to foresee the effect of their actions on the environment, the behaviour of the other agents, and the overall dynamics of the environment as well.

On the other hand, in principle an external observer does not directly interact with the agents of a MAS. As a result, some capability to act on the space of MAS interaction without dealing directly with agents is obviously required

in order to enable any form of objective coordination. Given that agents are typically situated entities, acting on the agent environment makes it possible to affect the behaviour of an agent system without having to alter the agents themselves. Under this acceptance, then, objective coordination deals with the agent environment: modifying the virtual machine supporting agent functioning, changing resource availability and access policies, altering the behaviour of the agent communication channel, be it virtual or physical, and so on – all these are possible ways to influence and possibly harness the behaviour of a MAS without directly intervening on individual agents and undermine the basic assumption of agent autonomy. The viability of objective coordination in the engineering of agent systems depends then on the availability of suitable models of the agent environment, and on their proper embodiment within agent infrastructures. There, objective coordination would conceivably take on the form of a collection of suitably expressive coordination abstractions, provided as run-time coordination services by the agent infrastructure.

As discussed by (Omicini and Ossowski, 2003), the engineering of a MAS requires that both subjective and objective coordination are blended together. On the one side, in fact, a purely subjective approach to coordination in the engineering of agent systems would endorse a mere reductionistic view, coming to say that agent systems are compositional, and their behaviour is nothing more than the sum of the individual's behaviour – an easily defeasible argument, indeed. Among the many consequences, this would require global properties of the agent system to be “distributed” among individuals, providing neither abstractions nor mechanisms to encapsulate such properties. As a result, the purely subjective approach would directly entail lack of support for design, development, and, even more, deployment of agent systems' global properties – which would result in substantial difficulties for incremental development, impractical run-time modification, and so on. On the other side, a purely objective approach to coordination in the engineering of agent systems would endorse a rough holistic view – where only inter-agent dependencies and interactions count, and individuals' behaviour has no relevance for global system behaviour. Among the many consequences, this would stand in stark contrast with any notion of agent autonomy, and would prevent agents from featuring any ability to affect the environment for their own individual purposes – no space for anything resembling an agent left, in short.

In the end, all the above considerations suggest that any principled approach to the engineering of agent systems should necessarily provide support for both subjective and objective models of coordination, possibly integrating them in a coherent conceptual framework, and providing at the same time a suitable support for all the phases of the engineering processes – in terms of coordination languages, development tools, and run-time environments.

3. Infrastructures for MAS Engineering

3.1 On the Notion of Infrastructure

Today, infrastructure is a fundamental notion for complex systems in general, not only in computer science and engineering, but also in the context of organisational, political, economical and social sciences. In its most general acceptation, an infrastructure is defined as:

(*Merriam-Webster*) | (1) the underlying foundation or basic framework (as of a system or organisation) (2) the permanent installations required for military purposes; (3) the system of public works of a country, state, or region; also: the resources (as personnel, buildings, or equipment) required for an activity;

(*Cambridge*) | (4) the basic systems and services, such as transport and power supplies, that a country or organisation uses in order to work effectively;

(*The American Heritage*) | (5) the basic facilities, services, and installations needed for the functioning of a community or society, such as transportation and communications systems, water and power lines, and public institutions including schools, post offices, and prisons.

Every definition underlines the role of infrastructure as (part of) the environment that provides basic resources and critical services to complex systems (such as organisations, communities, societies, countries) living on top of it. In particular, definition (2) remarks the fact that an infrastructure is a *persistent* entity: once installed, an infrastructure typically survives the many systems it supports. Also, definitions (4) and (5) remark the key role of infrastructures: their services typically cover critical system issues, and provide features that individual system components could not afford to provide or obtain elsewhere.

In the context of MAS, infrastructure obviously plays a key role, given the potential complexity of both the system components (agents) and the component interplay (agent societies).

(Gasser, 2001), defines an infrastructure as

“a technical and social substrate that stabilises and rapidly enables instrumental (domain-centric, intentional) activity in a given domain... (solving) typical, costly, commonly accepted community (technical) problems in a systematic and appropriate ways”

Here, it is important to emphasise the notion of infrastructure as a *social, enabling* support for providing MAS with cheap and systematic solutions to common problems.

Another interesting definition is provided by (Sycara et al., 2003):

“Agents in a MAS are expected to coordinate by exchanging services and information, to be able to follow complex negotiation protocols, to agree on commit-

ments and to perform other socially complex operations. We define the infrastructure of a MAS as the set of services, conventions, and knowledge that support such complex interactions."

The stress is here on the support of complex agent (social) interplay, which is expressed in terms of services, convention and knowledge.

3.2 The Role of MAS Infrastructure

In a more abstract acceptance than the ones above, the main role of infrastructures in MAS is to model and shape the *agent environment*, from the two points of view (i) of the agents living in the MAS; and (ii) of MAS designers. From the inner viewpoint of an individual agent, the infrastructure typically provides the means to deal with the agent environment: to perceive and affect its state and dynamics (in general), to access resources and services, to obtain and store information, to interact with other agents (in particular). Typically, a suitably expressive and well-engineered infrastructure allows agents to represent their environment only through the runtime abstractions provided by the infrastructure, and to modify the agent environment according to the agent's needs and goals through infrastructure services. From the external viewpoint of a human designer, MAS are typically open systems, both in terms of the unpredictability of their environment (due to components and interactions not under the control of MAS designers), and of the dynamism of both MAS structures (e.g., the set of agents in a MAS) and MAS processes as well (e.g., the coordination activities within a MAS). Infrastructures are then the suitable place for designers to embed elements of control of MAS despite their inherent openness: such control can be exerted by means of runtime abstractions provided by the infrastructure that can embody and enforce interaction constraints, coordination laws and social norms. Even more, once they are suitably described and made accessible to agents, the same runtime abstractions can be exploited by intelligent agents in order to represent coercive structures of a MAS, and to act upon its global behaviour by introducing and/or modifying constraints, laws and norms (Omicini and Ricci, 2003).

Infrastructures play then a key role in the engineering of MAS, too. This is quite obvious when considering the last stages of the engineering process, that is, the development and deployment of MAS. Nevertheless this also holds when taking the early stages into account, that is, the modelling and design of MAS: the abstractions provided by the infrastructure are the most natural candidates to be adopted and exploited in the design of MAS structures and activities, which are then to be engineered on top of such abstractions. So, runtime abstractions should be flexible enough to support the engineering of heterogeneous systems, and – at the same time – effective in minimising the gap between the design and development / deployment / runtime of systems.

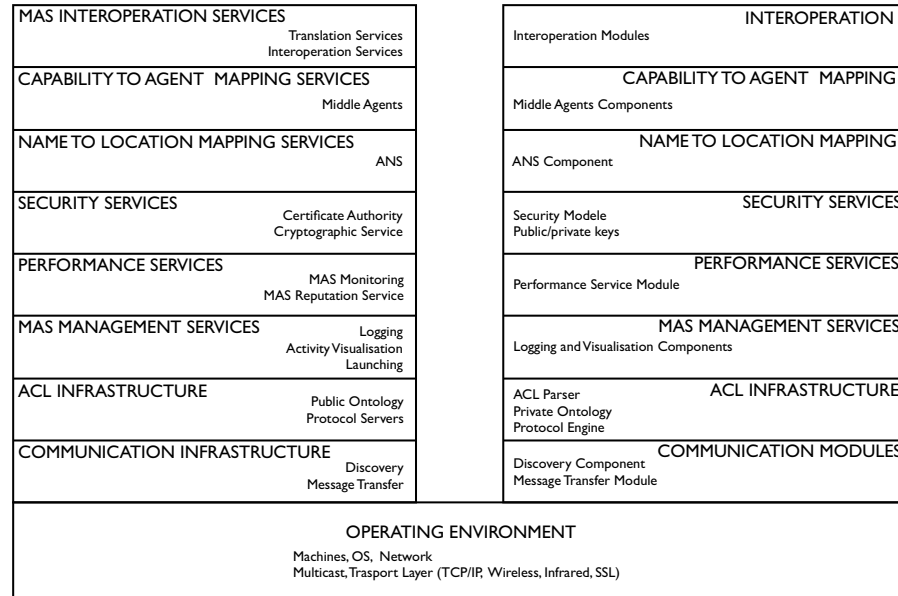


Figure 14.1. MAS infrastructure levels, according to (Sycara et al., 2003)

In this context, the tools provided by an infrastructure are fundamental to enable the manipulation of the abstractions through all the engineering stages, in particular at runtime. The definition of the engineering tools is a primary issue, that should be necessarily inspired and driven by the model embodied by the MAS infrastructure itself (Denti et al., 2002).

In the end, MAS infrastructures and tools play an essential engineering role by *keeping abstractions alive* through the whole engineering process, thus enabling software engineers to first design and then observe and act on MAS structures and processes at runtime, working upon abstractions adopted and exploited for the design of a MAS. This feature is particularly important to support forms of *online engineering* (see chapter 18), i.e., the capability of supporting system design / development / evolution while the systems are running – a particularly relevant feature in the context of MAS, given their intrinsic complexity and openness.

3.3 Enabling vs. Governing Infrastructures

As discussed above, infrastructures are useful to encapsulate and support critical features and properties of MAS; these properties typically concern the *interaction* dimension. For this extent, current MAS infrastructures can be

considered *enabling infrastructure*, since they provide abstractions that basically enable agent interaction at different levels: from communication to interoperability, to basic interaction services. This is apparent when considering the abstract architecture of two of the most important infrastructures currently adopted for MAS development and deployment: RETSINA (Sycara et al., 2003) (bottom of Figure 14.2) and JADE (Bellifemine et al., 2001) (top of Figure 14.2). There, in fact, services like agent communication, inter-operation, security, naming, location, etc., are necessary preconditions that make it possible for agents to live, coexist and interact within a MAS. Enabling infrastructures, then, basically define the nature of the agent interaction space within a MAS.

However, the increasing complexity and articulation of MAS for today's application scenarios call for a most effective engineering support from infrastructure, beyond the mere enabling of agent interaction. A well known example are Electronic Institutions (Noriega and Sierra, 2002): the social and normative capabilities required to infrastructures supporting eInstitutions goes far beyond the services provided by general purpose MAS enabling infrastructures, and cannot be straightforwardly engineered on top of it. Another example comes from team-oriented coordination: in order to be independent from the specific agent model, the TEAMCORE approach introduces the PROXY abstraction, an infrastructure component provided to agents for managing automatically all coordination dependencies with respect to the teams that agents belong to (Tambe et al., 2000). Similar team-oriented capability has been added to RETSINA by enhancing its Individual Agent Architecture (Giampapa and Sycara, 2002): in this way, contrary to the TEAMCORE approach, no real infrastructure support is provided from the infrastructure to team-oriented coordination, since the team-oriented capability is obtained by relying on augmented capabilities of the individual agents.

In the end, current general purpose MAS infrastructures typically lack suitably abstractions to *govern* agent interaction. This seems instead a fundamental feature for enabling the specification and enactment of social norms, but also – more generally – for defining and executing social activities, such as agent coordination. In other words, complex system engineering calls for *governing infrastructures*, providing flexible and robust abstractions to model and shape the agent interaction space, in accordance with the social and normative objectives of systems.

Governing infrastructures become the natural *loci* where to embody a conceptual framework that uniformly accounts for organisation, coordination and security of MAS altogether (Omicini et al., 2003). From the organisational point of view, infrastructures are to provide explicit abstractions for modelling the structure of an organisation and its rules – e.g., using the notion of role and related permissions to access to resources. This is the case, for instance, of

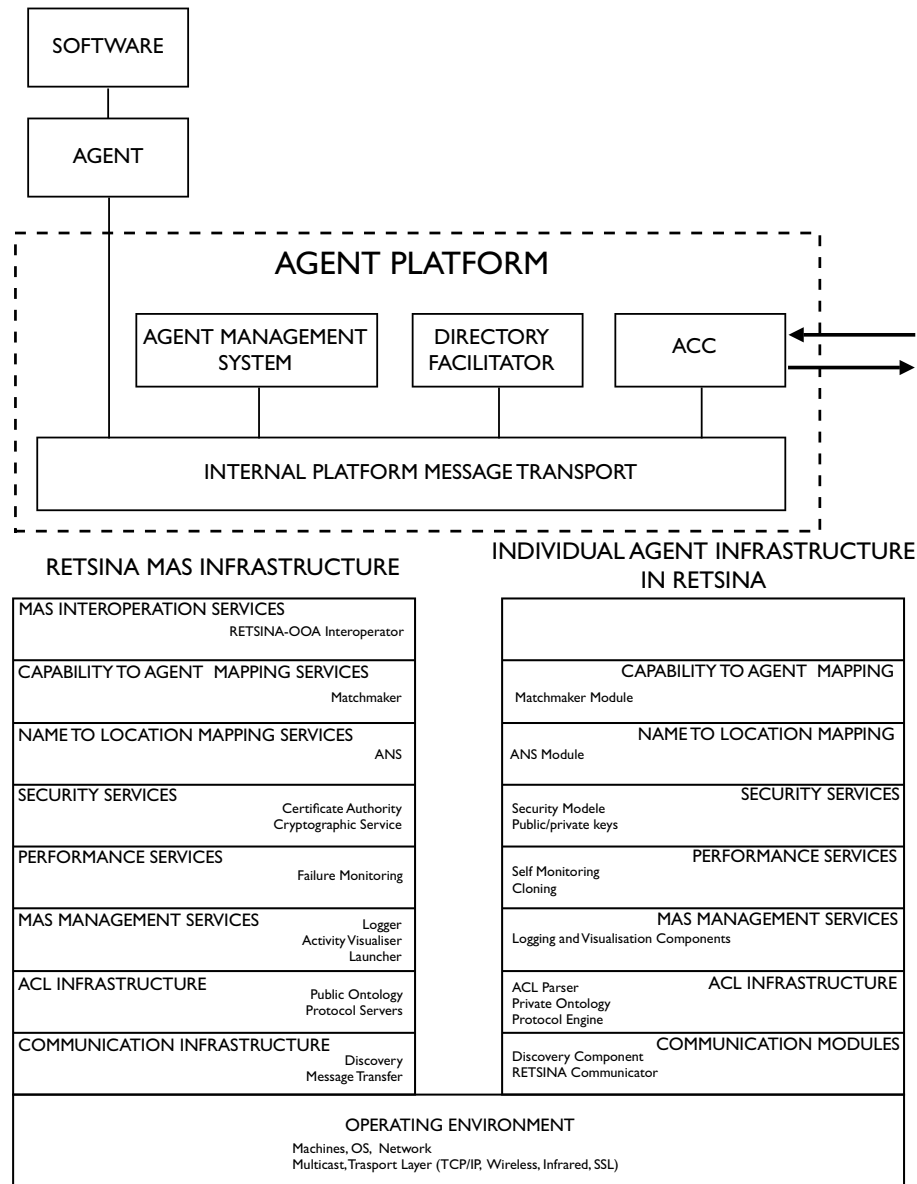


Figure 14.2. (top) FIPA reference model for Agent Platform, adopted by JADE (see chapter 13) – where the ACC is the Agent Communication Channel Component – and (bottom) RETSINA functional levels (Sycara et al., 2003)

the information system infrastructure that support the RBAC model (Sandhu et al., 1996), which is attracting attention also in the context of MAS. From the coordination point of view, infrastructure support can be described effectively by adopting the notion of *coordination as a service* (Omicini and Ossowski, 2003; Viroli and Omicini, 2003): according to this vision, the infrastructure itself is the provider of runtime (coordination) abstractions designed for specifically supporting the specification, execution and maintenance of MAS social activities. These abstractions become a fundamental tool to face the engineering complexity of coordination in MAS: both from the designer's and the agents' point of view, the coordination burden is distributed between agents and the specialised services provided by the infrastructure. The expressiveness and flexibility of coordination abstractions strongly influence the engineering of social activities, and, consequently, the complexity of the solutions adopted for the challenging application scenarios. Since they are part of the infrastructure, these coordination abstractions are typically expected to be robust and reliable, and specifically designed to support a critical activity as coordination is.

Two observations are worthwhile here. Firstly, the evolution from enabling to governing infrastructures can be devised also in other computer science fields, characterised as well by complex organisations and collaboration activities: CSCW and Workflow Management are relevant examples. Especially in the CSCW context the need for suitable infrastructure support for coordination has already emerged as a fundamental issue. (Schmidt and Simone, 1996), for instance, identify basic properties that coordination abstractions provided by an infrastructure should feature. Secondly, the approach of coordination as a service has also a deep impact on AOSE methodologies, since coordination abstractions – as they embody the *social* aspect of MAS – are meant to become explicitly subject of all the engineering stages, as it happens in SODA methodology (Omicini, 2001).

4. Modelling Coordination Infrastructures with Activity Theory

The research on coordination infrastructures is a primary issue also in other disciplines focusing on complex collaborative works in articulated organisation, such as CSCW and organisational science. The models and theories adopted and developed in those contexts can provide then useful insight for the MAS context. Accordingly, we considered Activity Theory very effective to frame and analyse coordination activities inside an organisation context, and the infrastructure support they require.

4.1 Activity Theory as a Framework for MAS Coordination

Once the many different coordination approaches have been properly understood and classified, a uniform conceptual framework is required that suitably reconciles both objective and subjective coordination, and helps putting them in the best perspective in the context of MAS engineering. To this end, (Ricci et al., 2003) adopt *Activity Theory* in order to shed some light on the role of subjective and objective approaches to coordination engineering, and their mutual relationship.

Activity Theory (AT henceforth) is a social psychological theory about the developmental transformation and dynamics in collective human work activity (Leontjev, 1978; Vygotskij, 1978). AT focuses on *human activities*, which are distinguished by their respective (physical and ideal) *objects*, that give them their specific directions, i.e., the *objectives* of the activities. Cooperation is understood as a *collaborative activity*, with one objective, but distributed onto several actors, each performing *actions* accordingly to the shared objective. Explicit norms and rules regulate the relationships among the individual participants' work.

Central to AT is the notion of *artifact* as a mediator for any sort of interaction in human activities: artifacts can be either physical or cognitive, such as operating procedures, heuristics, scripts, individual and collective experiences, and languages. Artifacts embody a set of social practise: their design reflects a history of particular use. As mediating tools, they have both an *enabling* and a *constraining* function: on the one hand, artifacts expand out possibilities to manipulate and transform different objects, but on the other hand the object is perceived and manipulated not 'as such' but within the limitations set by the tool. (Ricci et al., 2003) define the notion of *coordination artifact* to identify artifacts that are used in the context of collaborative activities in particular, mediating the interaction among actors involved in the same social context. Coordination artifacts can be *embodied* or *disembodied*, referring to respectively physically or cognitive/psychological artifacts. A similar concept can be found also in the CSCW context, with the notion of coordinative artifacts (Schmidt and Simone, 2000). It is worth noting the different acceptance of the term artifact as used in AT and CSCW with respect to the traditional software engineering context (Barthelmess and Anderson, 2002): in the latter, the term artifact is typically used to refer to documents (or deliverables) that are produced throughout a process, and the term *tool* is used to identify the means to perform operation on artifacts.

As far as collaborative activities are concerned, AT identifies three hierarchical levels defining their structure: *co-ordinated*, *co-operative*, and *co-constructive* (Bardram, 1998; Engeström et al., 1997).

- The *co-ordinated* aspect of work captures the normal and routine flow of interaction. Participants follow their *scripted roles*, each focusing on the successful performance of their actions, implicitly or explicitly assigned to them; they share and act upon a common object, but their individual actions are only externally related to each other. *Scripts* coordinating participants' actions are not questioned or discussed, neither known and understood in all their complexity: in this stage actors act as "wheels in the organisational machinery" (Kuutti, 1991), and co-ordination ensures that an activity is working in harmony with surrounding activities.
- The *co-operative* aspect of work concerns the mode of interactions in which actors focus on a common object and thus share the objective of the activity; unlike the previous case, actors do not have actions or roles explicitly assigned to them: with regard to the common object, each actor has to balance his/her own actions with other agent actions, possibly influencing them to achieve the common task. So, in this case the object of the activity is stable and agreed upon: however the means for realising the activity is not yet defined.
- The *co-constructive* aspect of work concerns interactions in which actors focus on re-conceptualising their own organisation and interaction in relation to their shared objects. Neither the object of work, nor the scripts are stable, and must be collectively constructed, i.e., *co-constructed*.

It is worth here to notice that in the analysis of collaborative activities, AT emphasises that a collaborative activity is not to be seen in general at one single level: co-ordination, co-operation, and co-construction are instead to be interpreted as *analytical* distinctions of the same collaborative activity, concurring in different times and modes to its development.

In the context of MAS coordination, the three levels identified by AT can be re-interpreted as follows (Figure 14.3):

- **Co-construction** – agents understand and reason about the (social) objectives (goals) of the MAS, and define a model of the social tasks required to reach them. This implies also identifying the interdependencies and the interactions to be faced and managed;
- **Co-operation** – agents design and define the coordination artifacts – either embodied (coordination media) or disembodied (plans, interaction protocols, etc.) – useful to carry on the social tasks and to manage the interdependencies and interactions devised out at the previous (co-construction) stage; and

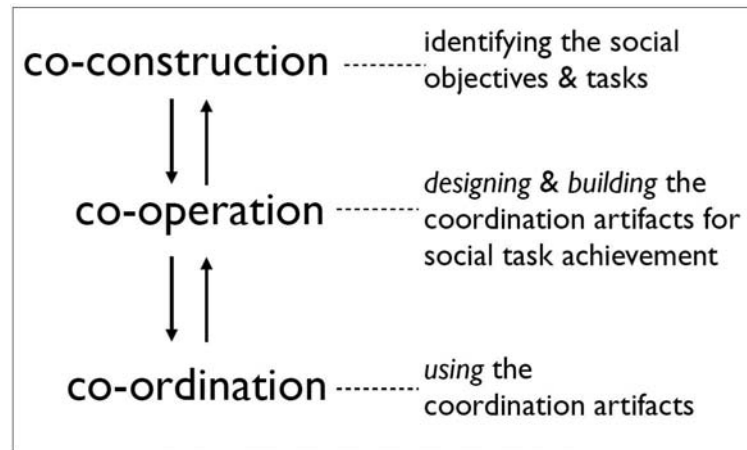


Figure 14.3. Levels of a collaborative activities as identified by Activity Theory and their relationship with coordination artifacts

- **Co-ordination** – agents exploit the coordination artifacts, and then the activities to manage interdependencies and interactions, either designed a-priori or planned at the co-operation stage, are enforced/automated.

At every level both top-down and bottom-up approaches are present: the former in modelling/designing/enacting social tasks, and the latter in identifying and managing dependencies and interactions. Both approaches rely on the engineering of coordination artifacts, be it embodied or disembodied.

Subjective approaches are fundamental for the co-construction and, in particular, the co-operation stage. Here it is necessary to reason about what kind of coordination is required, what kind of coordination laws must be developed to manage interactions and fulfil the social tasks identified in co-construction stage. Agent intelligence is useful to cooperatively build – by means of negotiation and high level (semantics driven) interaction protocols – effective co-ordination artifacts to be used in the co-ordination stage, be they disembodied such as interaction protocols or embodied such as coordination media. Instead, objective coordination is fundamental for the co-ordination stage, where coordination must be enacted in the most automated, fluid, and possibly optimised manner. The coordination medium abstraction (and coordination laws defining its behaviour) represents effectively the concept of embodied coordination artifact (and related mediating tools), embedding and enacting in the co-ordination stage the social laws and interaction constraints established in the co-operation stage.

Drawing a parallel between AT artifacts and coordination media may help to better recognise the role of the media inside MAS: as the artifacts, coordi-

nation media first are used to *enable* the interactions among the agents, and then to *mediate* them in the most automated manner. As the artifacts, media become *the* place where the *coordination knowledge* of the MAS is explicitly represented (Ossowski and Omicini, 2002), where it is enacted and can be further inspected. So, media become the source of the “social intelligence” that actually characterises the systemic/synergistic (as opposed to compositional) vision of MAS (Ciancarini et al., 2000). In this context, coordination laws become the coercive structures that can be used to tune and adapt dynamically such a collective intelligence (Ossowski, 1999).

4.2 Artifacts and Coordination Infrastructures

Quite frequently in the context of MAS, agents are the *only* abstraction used for system engineering – especially at the development and deployment stage. The matchmaking and brokering services required by any open MAS, for instance, are usually provided by *middle-agents* (Klusck and Sycara, 2001). Accordingly, these agents constitute a suitable way to embody AT artifacts at the co-ordination stage. So, in principle, they may take over the role of coordination media in the mediation of agent interactions.

However, AT clearly distinguishes between ontological properties of the artifacts (as well as related mediating tools) and the actors designing/developing (co-operation stage) and exploiting (co-ordination stage) the artifacts. This suggests to draw a similar distinction between agents and coordination media. As opposed to agents, the main properties that a coordination medium is expected to exhibit are the following:

- *Inspectability* – the behaviour of a coordination medium should be inspectable, both for human and artificial agents. Moreover, coordination specifications should be described in a declarative way, possibly with a formally defined semantics, to allow for their interpretation at run-time.
- *Efficiency/specificity* – a coordination medium should be specialised in the management of interactions, in order to maximise performance in the application of the coordination rules. Moreover, a medium should be specialised to support the concurrent actions (communications) of multiple agents, possibly providing security, reliability and fault tolerance capabilities.
- *Predictability* – the behaviour of a coordination medium should exactly reflect the coordination laws upon which it has been forged (autonomous, unpredictable behaviour is typically *not* desired). A formal semantics should be defined for the coordination model to precisely define the effect of the coordination laws on the state of the medium and, more generally, on the agent interaction space.

- *Malleability* – a coordination medium should be malleable, i.e., it should allow its behaviour to be forged and changed dynamically at execution time, according to the need. This property is fundamental for facing the openness of MAS environment, in terms of unpredictable events causing coordination breakdowns or the support of coordination service improvement or enhancement. A similar concept is defined for coordination mechanisms in the context of CSCW (Schmidt and Simone, 1996).

Most of the above properties are typically not featured by middle-agents, as they are not featured by agents in general. In fact, agents are generally supposed to be autonomous, pro-active, situated entities that interact by means of a general-purpose and high-level communication language (Wooldridge and Jennings, 1995a). As a result, for instance, an agent cannot be supposed to be inspectable. In addition, the general purpose acceptance of the agent notion typically puts limits to predictability, specificity and efficiency.

So, the most obvious embodiment of the notion of artifact for agent coordination is represented by a dedicated abstraction, provided at design time by the coordination model and enacted at runtime by the corresponding coordination infrastructure: that is, an inspectable, efficient, specific, predictable and dynamically forgeable coordination medium that could be used by the agent designer to govern the agent interaction space, but also by intelligent agents to perceive, understand and possibly change the overall MAS behaviour.

4.3 Balancing Coordination in MAS Engineering

Another central notion in AT is the dynamic transformation between levels in collaborative activities. Correspondingly, central to MAS coordination is the support for dynamic transformation from co-operation – that is, the subjective coordination level – to co-ordination (that is the objective coordination level), and vice versa. This is particularly relevant in the context of open and dynamic systems, where the environment is frequently subject to change, and collective goals, norms, and organisational rules should adapt accordingly. This form of dynamism is captured by two basic transitions, the *reflection* and the *reification* of coordination, which must be supported dynamically during system execution. These transitions are strictly related to the transformations seen in the AT, and account for:

- **Reification** – in this transition, coordination laws that have been designed and developed in the co-operation stage are reified in coordination media: intelligent agents forge the behaviour of coordination media in order to reflect the social rules established in the co-operation stage, and to be used as artifacts in the co-ordination stage. It is worth noting that coordination media are meant to embed not only the rules promoting cooperation among agents, but also the laws ruling interactions,

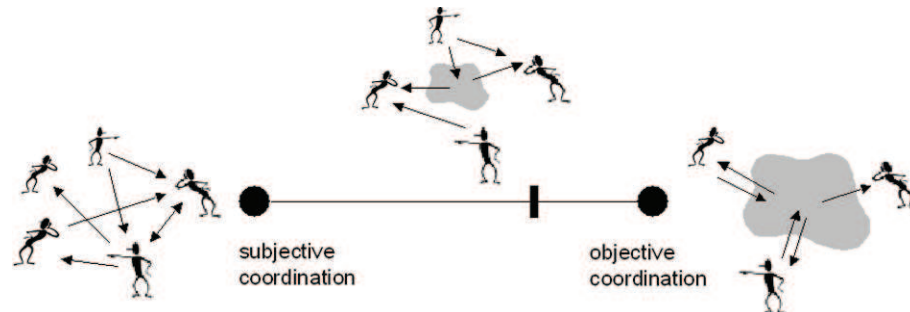


Figure 14.4. The coordination engineering segment: all coordination in agents (purely subjective, left end), all coordination in media (purely objective, right end)

useful to represent also norms and environment constraints, either mediating agent competitive (non cooperative) behaviour, or harnessing self-interested agent behaviours so as to achieve global MAS goals without affecting agent autonomy.

- **Reflection** – in this transition, the behaviour of the coordination media deployed in co-ordination stage is inspected and possibly understood. Agents can retrieve the coordination laws underlying medium behaviour, and relate them to the history of MAS evolution, in order to either evolve them according to changes in coordination policies or in environmental conditions, or learn how to exploit the artifacts in a more effective and efficient way.

The role of coordination artifacts (and correspondingly of coordination infrastructures enacting them) is then central to the engineering of MAS, since they make it possible to balance dynamically subjective and objective coordination, providing the tools to establish at runtime the distribution of the burden coordination between media and agents. As a useful picture, we can draw an imaginary “coordination engineering segment” (see Figure 14.4), whose extremes represent the two opposite situations where on the one side (left in Figure 14.4) all coordination activity are carried on by agents, and on the other side (right in Figure 14.4) all the coordination burden is charged upon the media provided by the infrastructure. Conceptually, the main issue here is to provide the means to devise out at design time where the best “coordination engineering point” of the MAS lays in the coordination segment, that is, the best distribution of coordination activities between subjective to objective orientation, and then to move the coordination point of MAS at execution time, possibly in the follow-up of changes in the MAS environment, to tune the system’s performance, or to modify its behaviour. The position of the point depends on both, the co-

ordination scenario taken into account and the dynamics inside that scenario: the more automation/prescription is required and the more well-defined social rules are (such as for workflow systems), the more coordination knowledge can be represented and enacted through the coordination media. Accordingly, the less possible (or feasible) it is to clearly identify collective rules and constraints in the coordination context, the more the individual agents are to be charged with the coordination burden. As a result, automatable activities are carried out by specialised and efficient coordination artifacts provided by the infrastructure, whilst activities requiring intelligent deliberative capabilities are assigned to intelligent agents. Indeed, the capability of balancing task automation and cooperation flexibly is among the most important requirements for state-of-the-art systems for workflow management, supply chain management, and CSCW (Dayal et al., 2001; Nutt, 1996). The ability to change the “engineering point” of coordination dynamically is also of special importance for open MAS, where the environment can unpredictably change, and the overall structure and functionality of the system may evolve in time.

The above considerations lead to some additional requirements for coordination infrastructures. In particular, in order to support these capabilities, coordination infrastructures should provide the means (languages and tools) for enabling coordination reflection (from objective to subjective transition), to inspect the coordination laws defining medium behaviour, and coordination reification (from subjective to objective transition), defining/programming the behaviour of the coordination media.

5. Engineering MAS with Coordination Infrastructures

5.1 Impact on AOSE Methodologies

The availability of coordination infrastructures has a considerable impact on the process of MAS engineering, and therefore should play a significant role within agent-oriented methodologies. As already mentioned, infrastructures impact on both the final stages of the engineering process (development and deployment) as well as on the analysis and design stages, by means of the abstractions provided by the infrastructure model to represent the environment and to support coordination and organisation.

It is not without reason that AT, which we use as a meta-model to frame the basic elements of a MAS infrastructure model, is primarily used as an analytical tool for understanding collaborative work in complex organisational contexts, and as a design tool to improve them. In such contexts, AT makes it possible to face the complexity of the social activities by clearly separating individual and collective activities, and then by clearly identifying and designing the artifacts required to support both of them. Here we are interested in partic-

ular in artifacts supporting social activities, which we denoted as coordination artifacts.

Along this line, we can devise a correspondence between the levels identified by AT for collaborative activities – co-construction, co-operation and co-ordination – and the engineering stages as typically found in (agent-oriented) software engineering methodologies, i.e., analysis, design, development and runtime. Generally speaking, individual and social tasks are identified and described in the analysis and design stages of these methodologies (Omicini, 2001; Zambonelli et al., 2001a). Individual tasks are typically associated with one specific competence of the system, related to the need to affect a specific portion of the environment and carry out some simple task. Each agent in the system is assigned to one or more individual tasks, and assumes full responsibility for their correct and timely completion. From an organisational perspective, this corresponds to assigning each agent a specific role in the organisation. Conversely, social tasks represent the global responsibilities of the agent system. In order to carry out such tasks, several possibly heterogeneous competences usually need to be combined. The design of social tasks leads to the identification of global *social laws* that have to be respected and/or enforced by the society of agents, to enable the society itself to function properly and in accordance with the expected global behaviour (Zambonelli et al., 2001a).

Given this picture, it is possible to identify a correspondence between the analysis stage (where individual and, in particular, social tasks are identified) and the co-construction level, where the social objectives of the activities are shaped. Then, the identification of the social laws required to achieve the social tasks can be seen as a first step in the co-operation level. This level roughly corresponds to the design and development stages of the engineering process: coordination artifacts are the abstractions which make it possible to design and develop social tasks. At the co-operation level such artifacts are designed and developed to embody and enact – as governing abstractions provided by the infrastructure – the social laws and norms previously identified. Finally, the deployment and runtime stages correspond to the co-ordination level, when the coordination artifacts are instantiated and exploited.

A relevant aspect that it is worth to be pointed out here is that, in the case of AT, the three levels are distinct analytical moments that can be applied continuously, since a collaboration activity is considered to be *continuously* under development, given the intrinsic openness of the environment and the dynamism of organisations. Then, the infrastructure can play a fundamental role not only in providing abstractions and means for the individual engineering stages, but also to support the dynamism between these stages, *continuously*, promoting a form of *online engineering* – a process that appears as unavoidable for the engineering of complex open system (Fredriksson et al., 2003) (see chapter 18).

5.2 Coordination, Organisation and Security in the Same Engineering Context

In the context of MAS, organisation and coordination are strictly related and interdependent issues, and so MAS coordination infrastructures have a fundamental engineering role also in MAS organisation (Omicini and Ricci, 2003),

Generally speaking, *organisation* mainly deals with the structure and the long-term relationships between the components of a system, while coordination mainly concerns the processes and the dynamic interactions between the components of a system – often related to roles that usually frame agents in the structure/pattern of system organisation. In any case, both organisation and coordination concern and affect the way in which agents interact with each other, so that conceiving and representing them in the same framework is likely to provide several advantages. Conceptual economy is obviously the first benefit: for instance, the notion of role, usually introduced by organisational models, typically constrains agent actions, which is one of the corner-stones of coordination. Also, a common framework is the most obvious way to consistently support adaptation and evolution of organisation and coordination within an agent society: for instance, by managing explicitly the dependencies between the changes in the organisational settings (such as removal of a role, or changes in its capabilities in terms of interaction protocols) and the related effects on coordination activities. Even more, there are system aspects that can be modelled and engineered in their complex articulation only by considering organisation and coordination settings at the same time: security and electronic institutions are well-known examples. In particular, the multiple aspects related to the security issue in MAS can be tackled in a coherent and satisfactory framework only by covering the whole spectrum that ranges from organisation – with issues related to system structures and relations among the components – to coordination – with issues related to collective processes. Facing security modelling and engineering within this range increases system conceptual integrity, by promoting the reuse of abstractions such as roles, permissions, and societies – which have already proved to be effective in the context of organisation and coordination – in order to enforce complex and dynamic security policies.

Even though the need for run-time liveness of design abstractions supported by the MAS infrastructure follows from basic system engineering considerations, it has an impact on the engineering of intelligent systems (Omicini, 2001). When dealing with MAS organisation abstractions, their liveness allows in principle to dynamically inspect and, possibly, change or adapt it. This is obviously useful for promoting human activities over systems such as monitoring and incremental evolution: however, when dealing with intelligent

systems, the liveness of (organisation/coordination) abstractions is particularly relevant since the properties they embody can be in principle made available not only to humans, but also to intelligent agents. This clearly promotes self-reconfiguration and self-adaptation of intelligent systems: in fact, once an intelligent agent is enabled to inspect the social structure, and allowed to change it, it may reason about the organisation, make inferences, and possibly plan its evolution, for instance to fix some undesired behaviour, or to adapt to environmental changes (Omicini and Ricci, 2003).

Summing up, it is both possible and useful to conceive a MAS infrastructure that supports the modelling and enactment of organisation aspects in synergy with the coordination ones, by keeping the abstractions alive throughout the whole engineering process: that is, by providing MAS engineers with design abstractions also suitable for organisations (such as the notions of role, society, group) and then enabling their management (construction, inspection, adaptation) at both development and execution time. This synergy makes it possible to model and enact coordination activities taking into account the organisation context where they take place, characterised by some structure – in terms of roles, groups, or societies – and organisation rules, such as access control policies. Agents participate to social activities always by virtue of their position (roles) inside the organisation, which define what kind of coordination artifacts they can access and use, and what kind of actions they are allowed (or forbidden) to do on them.

As an example, introduced in (Omicini, 2002), the *Agent Coordination Context* (ACC) abstraction is an infrastructural notion suitable for the integration of organisation issues in a coordination context, especially in the case of artifact-based coordination infrastructure. The ACC notion is meant to model and enact agent position inside an organisational context acting as its environment, so as to define and constrain the agent actions on resources, in this case coordination artifacts (Omicini et al., 2003). Therefore, it is possible to conceive a MAS infrastructure which fruitfully adopts ACC to model and rule agent presence inside the organisation, and, more specifically, agent participation to social activities; this participation includes accessing and using the coordination artifacts as part of organisation resources.

6. An Example of a Coordination Infrastructure

TuCSon is an example of coordination infrastructure for MAS designed according the principles described in previous sections. Figure 14.5 gives a layered perspective of the infrastructure architecture, with organisation and coordination layer in evidence.

TuCSon provides services for the specification and enactment of coordination in MAS (Omicini and Zambonelli, 1999), according to the coordination

as a service approach. Coordination services are embodied in *tuple centres*, that are design/runtime coordination abstractions provided to agents by the infrastructure in order to enable and govern their interaction (Omicini and Denti, 2001). More precisely, tuple centres are *programmable* tuple spaces (Omicini and Denti, 2001), that is, sort of reactive logic-based blackboards; agents interact by writing, reading, and consuming *tuples* – ordered collections of heterogeneous information chunks – to/from tuple centres via simple communication operations (*out*, *rd*, *in*) which access tuples associatively. While the behaviour of a tuple space in response to communication events is fixed and pre-defined by the model, the behaviour of a tuple centre can be tailored to the application needs by defining a suitable set of *specification tuples*, which define how a tuple centre should react to incoming/outgoing communication events, and determine the coordination laws embodied by tuple centres. Tuple centres then can be seen as general-purpose customisable coordination artifacts, whose behaviour can be dynamically specified, forged and adapted so as to automate the co-ordination stage among agents using such artifacts.

The basic infrastructure model is currently being extended to support a role-based organisation model (Omicini and Ricci, 2003). This extension is realised by embodying the ACC notion as first class runtime abstraction (Omicini, 2002). In order to join dynamically a specific organisation, an agent must negotiate and obtain an ACC, as a private interface to access and use the tuple centres of the organisation. The actions enabled by the ACC depend on the active roles the agent is playing inside the organisation.

6.1 Balancing Coordination with TuCSoN

In the case of TuCSoN, the capability of balancing coordination between subjective and objective as discussed in section 4 is achieved by means of the tuple centre model, and the tools provided by the infrastructure. The coordination laws that define the behaviour of the coordination media (tuple centres) expressed as specification tuples can be inspected and changed dynamically by human and artificial agents by means of specific tools. We are verifying the effectiveness of this approach in scenarios such as pervasive computing – to engineer the social intelligence as required by smart environments – and inter-organisational workflow management systems (Ricci et al., 2002). In the last context, for instance, tuple centres have been used to play the role of the workflow engines, and workflow rules have been expressed as coordination laws embedded within tuple centres. Each workflow engine (mapped onto a tuple centre) acts then as a coordination artifact providing fluid coordination of the individual tasks executed autonomously by human and artificial agents. So, (i) workflow rules are inspectable by accessing the specification tuples embedded in tuple centres (reflection stage); (ii) workflow rules are modifiable at runtime

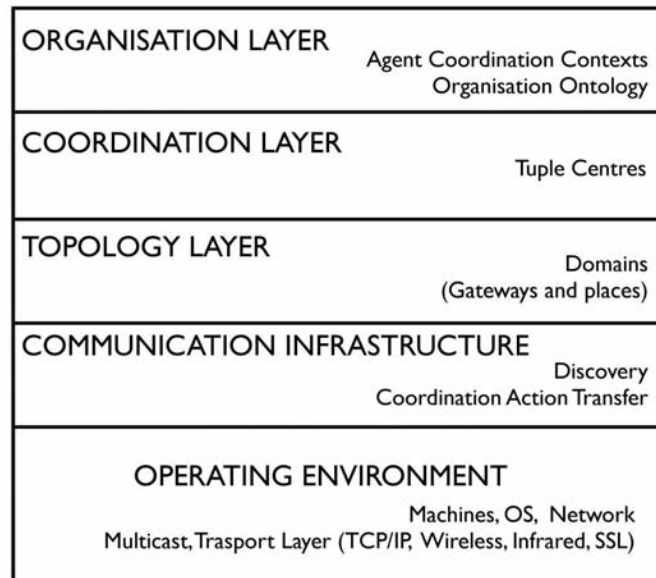


Figure 14.5. The layered architecture of a coordination infrastructure: the TuCSon case

– as a consequence of unexpected exceptions, or changes in the business environment – by changing the specification tuples within tuple centres (reification stage); and *(iii)* multiple workflow engines (tuple centres) can be exploited, spread over the infrastructure nodes, so as to distribute the coordination workload reflecting a multi-centric view of coordination (Omicini and Ossowski, 2003).

7. Discussion

In this chapter, we provided a brief overview over current conceptualisations, models and support infrastructures for coordination in MAS. We motivated that today's enabling infrastructures need to be extended so as to allow MAS designers to effectively govern the agent interaction space. Such coordination infrastructures may then become the natural loci for modelling and enacting mechanisms that bias autonomous agent (inter-)action and achieve instrumental behaviour. Drawing from findings in Activity Theory, we put forward the notion of artifact as a step toward a unified framework for coordination, and derived some distinctive properties that a coordination infrastructure should feature. Finally, we provided clues on how these notions can support a principled design process for MAS.

Although the ideas presented in this chapter tackle the problem of coordination infrastructures for MAS engineering mainly at a conceptual level, software

frameworks that adequately support the abstractions that we have put forward will soon be a reality. This, in turn, will facilitate a smooth integration with modern AOSE methodologies and thus allow the full exploitation of the potential of coordination infrastructures in all stages of MAS engineering.

References

- [Abelson, 1996] Abelson, S. (1996). *Structure and Interpretation of Computer Programs*. The MIT Press.
- [Abelson et al., 2000] Abelson, H., Allen, D., Coore, D., Hanson, C., Homsy, G., Knight, T., Napal, R., Rauch, E., Sussmann, G., and Weiss, R. (2000). Amorphous Computing. *Communications of the ACM*, 43(5), pages 43–50.
- [Abowd, 1999] Abowd, G. D. (1999). Software Engineering Issues for Ubiquitous Computing. *Proceedings of the 21st International Conference on Software Engineering (ICSE'99)*, Los Angeles, USA.
- [Adorni et al., 2001] Adorni, G., Bergenti, F., Poggi, A., and Rimassa, G. (2001). Enabling FIPA Agents on Small Devices. *Proceedings of the 5th International Workshop on Cooperative Information Agents (CIA 2001)*, Rome, Italy.
- [Aksit et al., 1993] Aksit, M., Wakita, K., Bosch, J., Bergmans, L., and Yonezawa, A. (1993). Abstracting Object-Interactions using Composition-Filters. Guerraoui, R., Nierstrasz, O., Riveill, M. (Eds.) *Object-based Distributed Processing*, Springer-Verlag.
- [Alberts et al., 2001] Alberts, D., Garstka, J., Hayes, R., and Signori, D. (2001) Understanding Information Age Warfare. *CCRP Publication Series*.
- [Amant, 2003] Amant, R. S. (2003). Planning Resources at the North Carolina State University. Available at <http://www.csc.ncsu.edu/faculty/stamant/planning-resources.html>.
- [Ankolekar et al., 2002] Ankolekar, A., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., Martin, D., McIlraith, S., Narayanan, S., Paolucci, M., Payne, T., and Sycara, K. (2002). DAML-S: Web Service Description for the Semantic Web. *Proceedings of the 1st International Semantic Web Conference (ISWC)*, pages 348–363.

- [Aridor and Lange, 1998] Aridor, Y., and Lange, D.B. (1998). Agent Design Patterns: Elements of Agent Application Design. *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*, pages 108–115, St. Paul, USA.
- [Ark and Selker, 1999] Ark, W. S., and Selker, T. (1999). A Look at Human Interaction with Pervasive Computers. *IBM Systems Journal*, 38(4).
- [Arnold et al., 1999] Arnold, K., O'Sullivan, B., Scheifler, R., Waldo, J., and Wollrath, A. (1999). *The Jini Specification*. Sun Microsystems.
- [Arquilla and Ronfeldt, 2000] Arquilla, J., and Ronfeldt, D. (2000). Swarming and the Future of Conflict. Available at <http://www.rand.org/publications/DB/DB311>.
- [Avila-Rosas et al., 2002] Avila-Rosas, A., Moreau, L., Dialani, V., Miles, S., and Liu, X. (2002). Agents for the Grid: A Comparison with Web Services (Part II: Service Discovery). *Proceedings of the Workshop on Challenges in Open Agent Systems*, Bologna, Italy.
- [Avouris and Gasser, 1992] Avouris, N. A., and Gasser, L. (Eds.) (1992). *Distributed Artificial Intelligence: Theory and Praxis*. Kluwer Academic Publishers.
- [Axelrod, 1984] Axelrod, R. (1984). *The Evolution of Cooperation*. Basic Books.
- [Baecker et al., 1997] Baecker, R., DiGiano, C., and Marcus, A. (1997). Software Visualization for Debugging. *Communications of the ACM*, 40(4), pages 44–54.
- [Ball, 1996] Ball, P. (1996). *The Self-Made Tapestry: Pattern Formation in Nature*. Princeton University Press.
- [Bardram, 1998] Bardram, J. (1998). Designing for the Dynamics of Cooperative Work Activities. *Proceedings of the ACA4 Conference on Computer Supported Cooperative Work*.
- [Barthelmess and Anderson, 2002] Barthelmess, P., and Anderson, K. M. (2002). A View of Software Development Environments based on Activity Theory. *Computer Supported Cooperative Work (CSCW)*, 11(1–2), pages 13–37.
- [Bauer, 1972] Bauer, F. L. (1972). Software Engineering. *Information Processing 71*.

- [Bauer, 1999] Bauer, B. (1999). Extending UML for the Specification of Interaction Protocols. Submission for the 6th Call for Proposal of FIPA and revised version of FIPA-99.
- [Bauer, 2001] Bauer, B. (2001). UML Class Diagrams Revisited in the Context of Agent-Based Systems. Wooldridge, M., Ciancarini, P., and Weiss, G. (Eds.) *Proceedings of Agent-Oriented Software Engineering (AOSE'01)*, pages 1–8.
- [Bauer et al., 2001] Bauer, B., Müller, J. P., and Odell, J. (2001). Agent UML: A Formalism for Specifying Multiagent Software Systems. *International Journal of Software Engineering and Knowledge Engineering*, 11(3), pages 207–230.
- [Bellavista et al., 2001] Bellavista, P., Corradi, A., and Stefanelli, C. (2001). Mobile Agent Middleware for Mobile Computing. *IEEE Computer*, 34(3).
- [Bellifemine et al., 2001] Bellifemine, F., Poggi, A., and Rimassa, G. (2001). Developing Multi-agent Systems with a FIPA-Compliant Agent Framework. *Software Practice and Experience*, 31, pages 103–128.
- [Benni, 1988] Beni, G. (1988). The Concept of Cellular Robotic System. *Proceedings of the IEEE International Symposium on Intelligent Control*, Los Alamitos, CA, pages 57–62.
- [Benni and Hackwood, 1992] Beni, G., and Hackwood, S. (1992). Stationary Waves in Cyclic Swarms. *Proceedings of the IEEE International Symposium on Intelligent Control*, Los Alamitos, CA, pages 234–242.
- [Benni and Wang, 1989] Beni, G., and Wang, J. (1989). Swarm Intelligence. *Proceedings of 7th Annual Meeting of the Robotics Society of Japan*, Tokyo, pages 425–428.
- [Benni and Wang, 1991] Beni, G., and Wang, J., 1991. Theoretical Problems for the Realization of Distributed Robotic Systems. *Proceedings of the IEEE International Conference on Robotic and Automation*, Los Alamitos, CA, pages 1914–1920.
- [Bergenti, 2003] Bergenti, F. (2003). A Discussion of Two Major Benefits of Using Agents in Software Development. *Engineering Societies in the Agents World III*, pages 1–12, Springer-Verlag.
- [Bergenti and Poggi, 2001] Bergenti, F., and Poggi, A. (2001). A Development Toolkit to Realize Autonomous and Interoperable Agents. *Proceedings of the 5th International Conference on Autonomous Agents*, pages 632–639.

- [Bergenti and Ricci, 2002] Bergenti, F., and Ricci, A. (2002). Three Approaches to the Coordination of Multiagent Systems. *Proceedings of the 2002 ACM Symposium on Applied Computing*, pages 367–372.
- [Berlin, 1994] Berlin A. (1994) *Towards Intelligent Structures: Active Control of Buckling*. Ph.D. Thesis, MIT.
- [Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. *Scientific American*, 284(5), pages 34–43.
- [Bernon et al., 2003] Bernon, C., Camps, V., Gleizes, M.-P., and Picard, G. (2003). Tools for Self-Organizing Applications Engineering. Di Marzo Serugendo, G., Karageorgos, A., Rana, O. F., and Zambonelli, F. (Eds.) *Proceedings of the 1st International Workshop on Engineering Self-Organizing Applications (ESOA 2003)*, Melbourne, Australia.
- [Bernon et al., 2002] Bernon, C., Gleizes, M.-P., Peyruqueou, S., and Picard, G. (2002). ADELFE: A Methodology for Adaptive Multi-Agent Systems Engineering. Petta, P., Tolksdorf, R., and Zambonelli, F. (Eds.) *Proceedings of the 3rd International Workshop on Engineering Societies in the Agents World (ESAW 2002)*, pages 156–169.
- [Biebricher, C. K., 1995] Biebricher, C. K., Nicolis, G., and Schuster, P. (1995). Self-Organization in the Physico-Chemical and Life Sciences. European Union, 1995.
- [Bieszczad et al., 1998] Bieszczad, A., Pagurek, B., and White, T. (1998). Mobile Agents for Network Management. *IEEE Communications Surveys*, 1(1), pages 2–9.
- [Bigus et al., 2002] Bigus, J. P., Schlönagle, D. A., Pilgrim, J. R., Mills III, W. N., and Diao, Y. (2002). ABLE: A Toolkit for Building Multiagent Autonomic Systems. *IBM Systems Journal*, 41(2), pages 350–371.
- [Binder and Lichtl, 2002] Binder, W., and Lichtl, B. (2002). Using a Secure Mobile Object Kernel as Operating System on Embedded Devices to Support the Dynamic Upload of Applications. *Proceedings of the 6th IEEE International Conference on Mobile Agents (MA'2002)*, Barcelona, Spain.
- [Bledsoe, 1985] Bledsoe, L. J., and Henschen W. W. (1985). What is Automated Theorem Proving? *Journal of Automated Reasoning*, 1(1), pages 23–28.
- [Boehm, 1984] Boehm, B. W. (1984). Verifying and Validating Software Requirements and Design Specifications. *IEEE Software*, 1(1), pages 75–84.

- [Boissier, 2003] Boissier, O. (2003). Master Web Intelligence: Organizations. Available at <http://www.emse.fr/~boissier>.
- [Bonabeau, 2003] Bonabeau, E. (2003). Swarm Intelligence. *Proceedings of Swarming: Network Enabled C4ISR*, Tysons Corner, VA.
- [Bonabeau et al., 1999] Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence*. Oxford University Press.
- [Booch, 1994] Booch, G. (1994). *Object-Oriented Analysis and Design with Applications*. Addison-Wesley.
- [Booch et al., 1999] Booch, G., Rumbaugh, J., and Jacobson, I. (1999). *The Unified Modeling Language User Guide*. Addison-Wesley.
- [Bordini et al., 2002] Bordini, R. H., Bazzan, A. L. C., de O. Jannone, R., Basso, D. M., Vicari, R. M., and Lesser, V. R. (2002). Agentspeak(XL): Efficient Intention Selection in BDI Agents via Decision-Theoretic Task Scheduling. *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1294–1302.
- [Borriello, 2002] Borriello, G. (2002). Key Challenges in Communication for Ubiquitous Computing. *IEEE Communications Magazine*.
- [Bowen, 2003] Bowen, Jonathan (2003). Formal Methods Resources. Available at <http://www.afm.sbu.ac.uk>.
- [Box et al., 2000] Box, D., Skonnard, A., and Lam, J. (2000). *Essential XML: Beyond Markup*. DevelopMentor Series. Addison-Wesley.
- [Bratman, 1987] Bratman, M. E. (1987). *Intentions, Plans, and Practical Reason*. Harvard University Press.
- [Bratman et al., 1988] Bratman, M. E., Israel, D., and Pollack, M. (1988). Plans and Resource-Bounded Practical Reasoning. *Journal of Computational Intelligence*, 4(4), pages 349–355.
- [Brazier et al., 1994] Brazier, F. M. T., van Langen, P., Treur, J., Wijngaards, N., and Willems, M. (1994). Modelling a Design Task in DESIRE: The VT Example. Technical Report IR-377, Universiteit Amsterdam.
- [Brazier et al., 1997] Brazier, F. M. T., Dunin-Keplicz, B. M., Jennings, N. R., and Treur, J. (1997). DESIRE: Modelling Multi-Agent Systems in a Compositional Formal Framework. *International Journal of Cooperative Information Systems*, 6(1), pages 67–94.

- [Brazier et al., 1999] Brazier, F. M. T., Jonker, C. M., Jungen, F. J., and Treur, J. (1999). Distributed Scheduling to Support a Call Centre: A Co-operative Multi-Agent Approach. *Applied Artificial Intelligence Journal*, 13.
- [Brazier et al., 2000] Brazier, F. M. T., Jonker, C. M., and Treur, J. (2000). Compositional Design and Reuse of a Generic Agent Model. *International Journal of Cooperative Information Systems*, 9(3), pages 171–207.
- [Brazier et al., 2002] Brazier, F. M. T., Jonker, C. M., and Treur, J. (2002). Principles of Component-Based Design of Intelligent Agents. *Data and Knowledge Engineering*, 41.
- [Bresciani et al., 2001] Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., and Mylopoulos, J. (2001). A Knowledge Level Software Engineering Methodology for Agent Oriented Programming. *Proceedings of the 5th International Conference on Autonomous Agents*, pages 648–655, Montreal, CA.
- [Bresciani et al., 2002] Bresciani, P., Giorgini, P., Giunchiglia, F., Mylopoulos, J., and Perini, A. (2002). Tropos: An Agent-Oriented Software Development Methodology. Technical Report DIT-02-0015, University of Trento.
- [Brézillon, 2003] Brézillon, P. (2003). Focusing on Context in Human-Centered Computing. *IEEE Intelligent Systems*, 18(3).
- [Brueckner, 2000] Brueckner, S. A. (2000). Return from the Ant: Synthetic Ecosystems for Manufacturing Control. Dr.rer.nat. Thesis. Humboldt University.
- [Brueckner and Parunak, 2002] Brueckner, S. A., and Parunak, H. V. D. (2002). Swarming Agents for Distributed Pattern Detection and Classification. *Proceedings of Workshop on Ubiquitous Computing, AAMAS 2002*, Bologna, Italy.
- [Brueckner and Parunak, 2003] Brueckner, S. A., and Parunak, H. V. D. (2003). Information-Driven Phase Changes in Multi-Agent Coordination. *Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS 2003)*, pages 950–951.
- [Bryson, 2001] Bryson, J. J. (2001). *Intelligence by Design: Principles of Modularity and Coordination for Engineering Complex Adaptive Agents*. Ph.D. Thesis, MIT.
- [Bryson et al., 2002] Bryson, K., Luck, M., Joy, M., and Jones, D. (2002). Agent Interaction for Bioinformatics Data Management. *Applied Artificial Intelligence*.

- [Burmeister, 1996] Burmeister, B. (1996). Models and Methodology for Agent-Oriented Analysis and Design. *Working Notes of the KI'96 Workshop on Agent Oriented Programming and Distributed Systems*.
- [Burrafato and Cossentino, 2002] Burrafato, P., and Cossentino, M. (2002). Designing a Multi-Agent Solution for a Bookstore with the PASSI Methodology. *Proceedings of the 4st International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2002)*, Toronto, CA.
- [Busetta et al., 1998] Busetta, P., Rönquist, R., Hodgson, A., and Lucas, A. (1998). JACK Intelligent Agents – Components for Intelligent Agents in Java. Technical Report, Agent Oriented Software Pty. Ltd. Available at <http://www.agent-software.com>.
- [Busetta et al., 2001] Busetta, P., Carman, M., Serafini, L., Stockinger, K., and Zini, F. (2001). Grid Query Optimisation in the Data Grid. Technical Report IRST 0109-01, Istituto Trentino di Cultura.
- [Bush et al., 2001] Bush, G., Craneffeld, S., and Purvis, M. (2001). The Styx Agent Methodology. The Information Science Discussion Paper Series 2001/02, University of Otago, New Zealand.
- [Busi et al., 2001] Busi, N., Ciancarini, P., Gorrieri, R., and Zavattaro, G. (2001). Coordination Models: A Guided Tour. Omicini, A., Zambonelli, F., Klusch, M., and Tolksdorf, R. (Eds.) *Coordination of Internet Agents: Models, Technologies, and Applications*, pages 6–24.
- [Bussmann, 1998] Bussmann, S. (1998). Agent-Oriented Programming of Manufacturing Control Tasks. *Proceedings of the 3rd International Conference on Multi-Agent Systems*, pages 57–63.
- [Butler et al., 2001] Butler, Z., Byrnes, S., and Rus, D. (2001). Distributed Motion Planning for Modular Robots with Unit-Compressible Modules. *Proceedings of the International Conference on Intelligent Robots and Systems*.
- [Butler et al., 2002] Butler, Z., Kotay, K., Rus, D., and Tomita, K. (2002). Generic Decentralized Control for a Class of Self-Reconfigurable Robots. *Proceedings of the IEEE International Conference on Robotics and Automation*.
- [Buyya et al., 2001] Buyya, R., Giddy, J., and Abramson, D. (2001). An Economy Grid Architecture for Service-Oriented Grid Computing. *Proceedings of the 10th IEEE International Heterogeneous Computing Workshop (HCW 2001)*, San Francisco, USA.

- [Caire et al., 2001a] Caire, G., Chainho, P., Evans, R., Garijo, F., Gomez Sanz, J., Kearney, P., Leal, F., Massonet, P., Pavon, J., and Stark, J. (2001a). Methodology for Agent Oriented Software Engineering. EURESCOM Project P907 Deliverable 3.
- [Caire et al., 2001b] Caire, G., Leal, F., Chainho, P., Evans, R., Garijo, F., Gomez-Sanz, J. J., Pavon, J., Kerney, P., Stark, J., and Massonet, P. (2001b). Agent Oriented Analysis using MESSAGE/UML. Weiss, G., Cianciarini, P., and Wooldridge, M. (Eds.) *Agent-Oriented Software Engineering II*, Springer-Verlag.
- [Camazine et al., 2001] Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraulaz, G., and Bonabeau, E. (2001). *Self-Organization in Biological Systems*. Princeton University Press.
- [Camps et al., 1998] Camps, V., Gleizes, M.-P., and Glize, P. (1998). Une Théorie des Phénomènes Globaux Fondée sur des Interactions Locales. *Actes des Sixième Journées Francophones IAD&SMA (JFIADSMA98)*, Editions Hermès.
- [Capera et al., 2003a] Capera, D., Georgé, J.-P., Gleizes, M.-P., and Glize, P. (2003a). The AMAS Theory for Complex Problem Solving based on Self-Organizing Cooperative Agents. *Proceedings of the 1st International Workshop on Theory and Practice of Open Computational Systems (TAPOCS)*, pages 383–388.
- [Capera et al., 2003b] Capera, D., Georgé, J.-P., Gleizes, M.-P., and Glize, P. (2003b). Emergence of organisations, emergence of functions. *Proceedings of the FAISB'03 Symposium on Adaptive Agents and Multi-Agent Systems*.
- [Carriero and Gelernter, 1989] Carriero, N., and Gelernter, D. (1989). Linda in Context. *Communications of the ACM*, 32(4), pages 444–458.
- [Castelfranchi, 1998] Castelfranchi, C. (1998). Modelling Social Action for AI Agents. *Artificial Intelligence*, 103(1).
- [Castelfranchi, 2000] Castelfranchi, C. (2000). Founding Agent's 'Autonomy' on Dependence Theory. *Proceedings of 14th European Conference on Artificial Intelligence*, pages 353–357, Berlin, Germany.
- [Castelfranchi and Falcone, 1998] Castelfranchi, C., and Falcone, R. (1998). Towards a Theory of Delegation for Agent-Based Systems. *Robotics and Autonomous Systems*, 24:141.
- [Castro et al., 2001] Castro, J., Kolp, M., and Mylopoulos, J. (2001). A requirements-driven development methodology. Dittrich, K.R., Geppert, A.

- and Norrie, M.C., editors, *Proceedings of the 13th International Conference on Advanced Information Systems Engineering (CAiSE'01)*, volume 2068, pages 108–123, Interlaken, Switzerland. Springer-Verlag.
- [Castro et al., 2002] Castro, J., Kolp, M., and Mylopoulos, J. (2002). Towards Requirements-Driven Information Systems Engineering: The Tropos Project. *Information Systems*. Elsevier.
- [Cernuzzi and Rossi, 2002] Cernuzzi, L., and Rossi, G. (2002). On the Evaluation of Agent Oriented Modeling Methods. *Proceedings of the OOPSLA 2002 Workshop on Agent-Oriented Methodologies*, pages 21–30, Seattle, USA.
- [Chen and Finin, 2002] Chen, H., and Finin, T. (2002). Beyond Distributed AI – Agent Teamwork in Ubiquitous Computing. *Proceedings of the 1st International Workshop on Ubiquitous Agents on Embedded, Wearable, and Mobile Devices*, Bologna, Italy.
- [Cheung et al., 1997] Cheung, P., Berlin, A., Biegelsen, D. K., and Jackson, W. B. (1997). Batch Fabrication of Pneumatic Valve Arrays by Combining MEMS with Printed Circuit Board Technology. *Proceedings of the Symposium on Micro-Mechanical Systems*.
- [Chopra and Singh, 2003] Chopra, A., and Singh, M. P. (2003). Nonmonotonic Commitment Machines. Dignum, F. (Ed.) *Proceedings of the 2003 AAMAS Workshop on Agent Communication Languages*, Springer-Verlag.
- [Chung et al., 2000] Chung, L., Nixon, B. A., Yu, E., and Mylopoulos, J. (2000). *Non-Functional Requirements in Software Engineering*. Kluwer Academic Publishers.
- [Ciancarini et al., 2000] Ciancarini, P., Omicini, A., and Zambonelli, F. (2000). Multiagent System Engineering: The Coordination Viewpoint. Jennings, N. R., and Lespérance, Y. (Eds.) *Intelligent Agents VI. Agent Theories, Architectures, and Languages*, pages 250–259. Springer-Verlag.
- [Cimatti et al., 2002] Cimatti, A., Clarke, E. M., Giunchiglia, E., Giunchiglia, F., Pistore, M., Roveri, M., Sebastiani, R., and Tacchella, A. (2002). NUSMV 2: An Opensource Tool for Symbolic Model Checking. *Computer Aided Verification*, Springer-Verlag.
- [Clarke et al., 1999] Clarke, E., Grumberg, O., and Peled, D. (1999). *Model Checking*. The MIT Press.
- [Clarke et al., 2001] Clarke, D., Elien, J.-E., Ellison, C., Fredette, M., Morcos, A., and Rivest, R. L. (2001). Certificate Chain Discovery in SPKI/SDSI. *Journal of Computer Security*, 9(4), pages 285–338.

- [Clearwater, 1996] Clearwater, S. H. (Ed.) (1996). *Market-Based Control. A Paradigm for Distributed Resource Allocation*. World Scientific Publishing.
- [Clements, 1996] Clements, P. C. (1996). A Survey of Architecture Description Languages. *Proceedings of the 8th International Workshop on Software Specification and Design*.
- [Clough, 2003] Clough, B. (2003). Emergent Behavior (Swarming): Tool Kit for Building UAV Autonomy. *Proceedings of Swarming: Network Enabled C4ISR*, Tysons Corner, VA.
- [Coffman et al., 1971] Coffman, E. G., Elphick, M., and Shoshani, A. (1971). System Deadlocks. *ACM Computing Surveys*, 3(2), pages 67–78.
- [Cohen, 1988] Cohen, J. (1988). A View of the Origins and Development of Prolog. *Communications of the ACM*, 31(1), pages 26–36.
- [Cohen and Levesque, 1990] Cohen, P. R., and Levesque, H. J. (1990). Intention is Choice with Commitment. *Artificial Intelligence*, 42, pages 213–261.
- [Cohen and Levesque, 1991] Cohen, P. R., and Levesque, H. J. (1991). Teamwork. *Nous*, 25(4), pages 487–512.
- [Coleman et al., 1994] Coleman, D., Arnold, P., Bodoff, S., Dollin, D., Gilchrist, H., Hayes, F., and Jeremas, P. (1994). *Object-Oriented Development: The FUSION Method*. Prentice-Hall International.
- [Collinot et al., 1996] Collinot, A., Drogoul, A., and Benhamou, P. (1996). Agent Oriented Design of a Soccer Robot Team. *Proceedings of ICMAS'96*.
- [Collis and Ndumu, 1999] Collis, J. C., and Ndumu, D. T. (1999). *The Role Modelling Guide*. Applied Research and Technology, BT Labs.
- [Conte et al., 1998] Conte, R., Gilbert, N., and Simao Sichman, J. (1998). MAS and Social Simulation: A Suitable Commitment. Sichman, J. S., Conte, R., and Gilbert, N. (Eds.) *Proceedings of the 1st International Workshop on Multi Agent Based Simulation*, pages 1–9. Springer-Verlag.
- [Conte and Sichman, 1995] Conte, R., and Sichman, J. S. (1995). Depnet: How to Benefit from Social Dependence. *Journal of Mathematical Sociology*, 20(2-3), pages 161–177.
- [Coore, 1999] Coore, D. (1999). Botanical Computing: A Developmental Approach to Generating Interconnect Topologies on an Amorphous Computer. Ph.D. Thesis, MIT.

- [Cossentino, 2001] Cossentino, M. (2001). Different Perspectives in Designing Multi-Agent System. *Proceedings of the Workshop on Agent Technology and Software Engineering*, Erfurt, Germany.
- [Cossentino and Potts, 2002] Cossentino, M., and Potts, C. (2002). A CASE Tool Supported Methodology for the Design of Multi-Agent Systems. *Proceedings of the 2002 International Conference on Software Engineering Research and Practice*, Las Vegas, USA.
- [Cossentino et al., 2003] Cossentino, M., Hopmans, G., and Odell, J. (2003). FIPA standardization Activities in the Software Engineering Area. *Proceedings of the 2003 Workshop on Objects and Agents (WOA03)*, Cagliari, Italy.
- [Cox et al., 2001] Cox, S. J., Fairman, M. J., Xue, G., Wason, J. L., and Keane, A. J. (2001). The Grid: Computational and Data Resource Sharing in Engineering Optimisation and Design Search. *Proceedings of the 2001 ICPP Workshops*, pages 207–212.
- [Crutchfield, 1994] Crutchfield, J. P. (1994). The Calculi of Emergence: Computation, Dynamics, and Induction. *Physica D*, 75, pages 11–54.
- [Curbera et al., 2002a] Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., and Weerawarana, S. (2002a). Unraveling the Web Services Web. *IEEE Internet Computing*, 6, pages 86–93.
- [Curbera et al., 2002b] Curbera, F., Golland, Y., Klein, J., Leymann, F., Roller, D., Thatte, S., and Weerawarana, S. (2002b). Business Process Execution Language for Web Services. Available at <http://www.ibm.com/developerworks/library/ws-bpel>.
- [Dam, 2003] Dam, K. H. (2003). Evaluating Agent-Oriented Software Engineering Methodologies. Master's Thesis, RMIT University.
- [Dam and Winikoff, 2003] Dam, K. H., and Winikoff, M. (2003). Comparing Agent-Oriented Methodologies. *Proceedings of the 5th International Bi-Conference Workshop on Agent-Oriented Information Systems*.
- [Dardenne et al., 1993] Dardenne, A., van Lamsweerde, A., and Fickas, S. (1993). Goal-Directed Requirements Acquisition. *Science of Computer Programming*, 20(1–2), pages 3–50.
- [Darimont et al., 1997] Darimont, R., Delor, E., Massonet, P., and van Lamsweerde, A. (1997). GRAIL/KAOS: An Environment for Goal-Driven Requirements Engineering. *Proceedings of the 19th International Conference on Software Engineering*, pages 612–613.

- [Dastani et al., 2003] Dastani, M., van Riemsdijk, B., Dignum, F., and Meyer, J. J. (2003). A Programming Language for Cognitive Agents: Goal Directed 3APL. *Proceedings of the 1st Workshop on Programming Multiagent Systems: Languages, Frameworks, Techniques, and Tools (ProMAS03)*.
- [Davis and Smith, 1983] Davis, R., and Smith, R. G. (1983). Negotiation as a Metaphor for Distributed Problem-Solving. *Artificial Intelligence*, 20, pages 63–109.
- [Day and Lawrence, 2000] Day, S. J., and Lawrence, P. A. (2000). Measuring Dimensions: The Regulation of Size and Shape. *Development*, 127, 2977–2987.
- [Dayal et al., 2001] Umeshwar, D., Meichun, H., and Ladin, R. (2001). Business Process Coordination: State of the Art, Trends and Open Issues. Apers, P. M. G., Atzeni, P., Ceri, S., Paraboschi, S., Ramamohanarao, K., and Snodgrass, R. T. (Eds.) *Proceedings of the 27th International Conference on Very Large Data Bases (VLDB 2001)*, pages 3–13.
- [De Giacomo et al., 2000] De Giacomo, G., Lesperance, Y., and Levesque, H. J. (2000). Congolog, A Concurrent Programming Language based on the Situation Calculus. *Artificial Intelligence*, 121, pages 109–169.
- [De Roure et al., 2003] De Roure, D., Jennings, N. R., and Shadbolt, N. (2003). The Semantic Grid: A Future e-Science Infrastructure. Berman, F., Fox, G., and Hey, A. J. G. (Eds.) *Grid Computing: Making the Global Infrastructure a Reality*, pages 437–470, John Wiley & Sons.
- [Debenham and Henderson-Sellers, 2002] Debenham, J., and Henderson-Sellers, B. (2002). Full Lifecycle Methodologies for Agent-Oriented Systems – The Extended OPEN Process Framework. *Proceedings of Agent-Oriented Information Systems (AOIS-2002)*, Toronto, CA.
- [Decker, 1995] Decker, K. S. (1995). *Environment Centered Analysis and Design of Coordination Mechanisms*. Ph.D. Thesis, University of Massachusetts.
- [Decker, 1996a] Decker, K. S. (1996a). TÆMS: A Framework for Environment Centered Analysis and Design of Coordination Mechanisms. O'Hare, G. M. P., and Jennings, N. R., (Eds.) *Foundations of Distributed Artificial Intelligence*, pages 429–448, John Wiley & Sons.
- [Decker, 1996b] Decker, K. S. (1996b). Task Environment Centered Simulation. *Simulating Organizations: Computational Models of Institutions and Groups*. AAAI Press/The MIT Press.

- [Decker et al., 1989] Decker, K. S., Durfee, E. H., and Lesser, V. R. (1989). Evaluating Research in Cooperative Distributed Problem Solving. Huhns, M. N., and Gasser, L. (Eds.) *Distributed Artificial Intelligence, Volume 2*, pages 487–519. Pitman/Morgan Kaufmann.
- [Decker et al., 1997] Decker, K. S., Sycara, K., and Williamson, M. (1997). Middle-Agents for the Internet. *Proceedings of IJCAI'97*.
- [Decker et al., 2001] Decker, K. S., Zheng, X., and Schmidt, C. (2001). A Multi-Agent System for Automated Genetic Annotation. *Proceedings of the 5th ACM International Conference on Autonomous Agents*, Montreal, Canada.
- [DeLoach, 2001] DeLoach, S. A. (2001). Analysis and Design using MaSE and agentTool. *Proceedings of the 12th Midwest Artificial Intelligence and Cognitive Science Conference (MAICS)*, Miami University Press.
- [DeLoach, 2002] DeLoach, S. A. (2002). Modeling Organizational Rules in the Multiagent Systems Engineering Methodology. *Proceedings of the 15th Canadian Conference on Artificial Intelligence*.
- [DeLoach and Wood, 2001] DeLoach, S. A., and Wood, M. (2001). Developing Multiagent Systems with agentTool. Castelfranchi, C., and Lesperance, Y. (Eds.) *Intelligent Agents VII.*, pages 46–60, Springer-Verlag.
- [DeLoach et al., 2001] DeLoach, S. A., Wood, M. F., and Sparkman, C. H. (2001). Multiagent Systems Engineering. *International Journal of Software Engineering and Knowledge Engineering*, 11(3), pages 231–258.
- [DeLoach et al., 2003] DeLoach, S. A., Matson, E. T., and Li, Y. (2003). Exploiting Agent Oriented Software Engineering in Cooperative Robotics Search and Rescue. *International Journal of Pattern Recognition and Artificial Intelligence*.
- [Demazeau, 1995] Demazeau, Y. (1995). From Cognitive Interactions to Collective Behaviour in Agent-Based Systems. *Proceedings of the European Conference on Cognitive Science*, pages 117–132, Saint-Malo, France.
- [Denti et al., 2002] Denti, E., Omicini, A., and Ricci, A. (2002). Coordination Tools for MAS Development and Deployment. *Applied Artificial Intelligence*, 16(9-10), pages 721–752.
- [Depke et al., 2001] Depke, R., Heckel, R., and Kuster, J. M. (2001). Improving the Agent-Oriented Modeling Process by Roles. *Proceedings of the 5th International Conference on Autonomous Agents*, pages 640–647.

- [Devedzic, 1999] Devedzic, V. (1999). A Survey of Modern Knowledge Modeling Techniques. *Expert Systems with Applications*, 17(275).
- [Devlin, 2003] Devlin, K. (2003). Why Universities Require Computer Science Students to take Math. *Communications of the ACM*, 46(9), pages 37–39.
- [Dey, 2001] Dey, A. K. (2001). Evaluation of Ubiquitous Computing Systems: Evaluating the Predictability of Systems. Abowd, G. D., Brumitt, B., and Shafer, S. (Eds.) *Proceedings of Evaluation Methods for Ubiquitous Computing Workshop*, Springer-Verlag.
- [Dietterich, 1998] Dietterich, T. G. (1998). Machine-Learning Research: Four Current Directions. *AI Magazine*, 18(4), pages 97–136.
- [DiLeo et al., 2002] DiLeo, J., Jacobs, T., and DeLoach, S. (2002). Integrating Ontologies into Multiagent Systems Engineering. *Proceedings of the 4th International Bi-Conference Workshop on Agent-Oriented Information Systems*.
- [Do et al., 2003] Do, T., Kolp, M., and Pirotte, A. (2003). Social Patterns for Designing Multi-agent Systems. *Proceedings of the 15th International Conference on Software Engineering and Knowledge Engineering (SEKE'03)*.
- [Dorigo and Di Caro, 1999] Dorigo, M., and Di Caro, G. (1999). The Ant Colony Optimization Meta-Heuristic. Corne, D., Dorigo, M., and Glover, F. (Eds.) *New Ideas in Optimization*, McGraw-Hill.
- [Dorigo et al., 1996] Dorigo, M., Maniezzo, V., and Colorni, A. (1996). The Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics*, 26(1), pages 1–13.
- [Drogoul and Zucker, 1998] Drogoul, A., and Zucker, J. (1998). Methodological Issues for Designing Multi-Agent Systems with Machine Learning Techniques: Capitalizing Experiences from the Robocup Challenge. Technical Report LIP6 1998/041, Laboratoire d'Informatique de Paris 6.
- [Du Bois, 1997] Du Bois, P. (1997). The ALBERT II Reference Manual. Technical Report RR-97-002, University of Namur.
- [Dubois, 1998] Dubois, P., and Heymans E. (1998). Scenario-Based Techniques for Supporting the Elaboration and the Validation of Formal Requirements. Technical Report CREWS 98-15, Université de Namur.
- [Dubois et al., 1994] Dubois, E., Du Bois, P., Dubru, F., and Petit, M. (1994). Agent-Oriented Requirements Engineering: A Case Study using the ALBERT Language. *Proceedings of the 4th International Working Conference*

- on *Dynamic Modelling and Information Systems (DYNMOD'94)*, pages 205–238.
- [Dulay et al., 2001] Dulay, N., Damianou, N., Lupu, E., and Sloman, M. (2001). A Policy Language for the Management of Distributed Agents. Wooldridge, M. J., Weiss, G., and Ciancarini, P. (Eds.) *Agent-Oriented Software Engineering II*, pages 84–100, Springer-Verlag.
- [Durfee and Lesser, 1991] Durfee, E. H., and Lesser, V. R. (1991). Partial Global Planning: A Coordination Framework for Distributed Hypothesis Formation. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(5), pages 1167–1183.
- [Durfee et al., 1989] Durfee, E. H., Lesser, V. R., and Corkill, D. D. (1989). Trends in Cooperative Distributed Problem Solving. *IEEE Transactions on Knowledge and Data Engineering*, 1(1), pages 63–83.
- [Edmonds, 1998] Edmonds, B. (1998). Social Embeddedness and Agent Development. *Proceedings of UKMAS'98*.
- [Edmonds, 2003a] Edmonds, B. (2003a). Simulation and Complexity – How They can Relate. Feldmann, V., and Mühlfeld, K. (Eds.) *Virtual Worlds of Precision*, Lit Verlag.
- [Edmonds, 2003b] Edmonds, B. (2003b). Against: A Priori Theory, For: Descriptively Adequate Computational Modelling. *The Crisis in Economics: The Post-Autistic Economics Movement: The first 600 days*, pages 175–179, Routledge.
- [Edwards, 2000] Edwards, S. J. A. (2000). Swarming on the Battlefield: Past, Present, and Future. Technical Report MR-1100-OSD, RAND.
- [Edwards, 2003] Edwards, S. J. A. (2003). Military History of Swarming. *Proceedings of Swarming: Network Enabled C4ISR*, Tysons Corner, VA.
- [Edwards and Grinter, 2001] Edwards, W. K., and Grinter, R. E. (2001). At Home with Ubiquitous Computing: Seven Challenges. Abowd, G. D., Brummitt, B., and Shafer, S. (Eds.) *Proceedings of the 3rd International Conference on Ubiquitous Computing (UBICOMP'2001)*, Springer-Verlag.
- [Ekudden et al., 2001] Ekudden, E., Horn, H., Melander, M., and Olin, J. (2001). On-Demand Mobile Media – A Rich Service Experience for Mobile Users. *Ericsson Review, The Telecommunications Technology Journal*, 4.
- [Elammari and Lalonde, 1999] Elammari, M., and Lalonde, W. (1999). An Agent-Oriented Methodology: High-Level and Intermediate Models. Wagner, G., and Yu, E. (Eds.) *Proceedings of the 1st International Workshop on Agent-Oriented Information Systems*.

- [Engeström et al., 1997] Engeström, Y., Brown, K., Christopher, C. L., and Gregory, J. (1997). Coordination, Cooperation, and Communication in the Courts: Expansive Transitions in Legal Work. Cole, M., Engeström, Y., and Vasquez, O. (Eds.) *Mind, Culture, and Activity*, Cambridge University Press.
- [Esler et al., 1999] Esler, M., Hightower, J., Anderson, T., and Borriello, G. (1999). Next Century Challenges: Data-Centric Networking for Invisible Computing – The Portolano Project at the University of Washington. *Proceedings of the ACM International Conference on Mobile Computing (MobiCOM'1999)*, Seattle, USA.
- [Esteva et al., 2002] Esteva, M., de la Cruz, D., and Sierra, C. (2002). Islander: An Electronic Institutions Editor. *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1045–1052.
- [Fallah-Seghrouchni and Suna, 2003] Fallah-Seghrouchni, A., and Suna, A. (2003). A Programming Language for Autonomous and Mobile Agents. *Proceedings of IAT 2003*.
- [Fankhauser et al., 1991] Fankhauser, P., Kracker, M., and Neuhold, E. J. (1991). Semantic vs. Structural Resemblance of Classes. *ACM SIGMOD RECORD* 20(4), pages 59–63.
- [Fano and Gershman, 2002] Fano, A., and Gershman, A. (2002). The Future of Business Services in the Age of Ubiquitous Computing. *Communications of the ACM*, 45(12).
- [Fensel and Motta, 2001] Fensel, D., and Motta, E. (2001). Structured Development of Problem Solving Methods. *Knowledge and Data Engineering*, 13(6), pages 913–932.
- [Fenster et al., 1995] Fenster, M., Kraus, S., and Rosenschein, J. S. (1995). Coordination without Communication: Experimental Validation of Focal Point Techniques. *Proceedings of International Conference on Multi-Agent Systems (ICMAS'95)*, pages 102–108, San Francisco, USA.
- [Ferber, 1999] Ferber, J. (1999). *Multi-Agent Systems*. Addison-Wesley.
- [Ferber and Gutknecht, 1998] Ferber, J., and Gutknecht, O. (1998). A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems. *Proceedings of the 3rd International Conference on Multi-Agent Systems (ICMAS-98)*, pages 128–135.

- [Ferguson, 1992] Ferguson, I. A. (1992). *TouringMachines: An Architecture for Dynamic, Rational, Mobile Agents*. Ph.D. Thesis, University of Cambridge.
- [Fikes and Nilsson, 1971] Fikes, R., and Nilsson, J. (1971). STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence*, 2(3–4).
- [Fisher, 1994] Fisher, M. (1994). A Survey of Concurrent METATEM – The Language and its Applications. Gabbay, D. M., and Ohlbach, H. J. (Eds.) *Proceedings of the 1st International Conference on Temporal Logic*, pages 480–505, Springer-Verlag.
- [Fisher, 1995] Fisher, M. (1995). Representing and Executing Agent-Based Systems. Wooldridge, M., and Jennings, N. R. (Eds.) *Intelligent Agents: Theories, Architectures, and Languages*, pages 307–323. Springer-Verlag.
- [Fisher and Wooldridge, 1997] Fisher, M., and Wooldridge, M. J. (1997) On the Formal Specification and Verification of Multi-Agent Systems. *International Journal of Cooperative Information Systems*, 6:1, pages 37–65.
- [Fornara and Colombetti, 2002] Fornara, N., and Colombetti, M. (2002). Operational Specification of a Commitment-Based Agent Communication Language. *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 535–542.
- [Forrest, 1991] Forrest, S. (Ed.) (1991). Emergent Computation: Self-Organizing, Collective, and Cooperative Phenomena in Natural and Artificial Computing Networks. Special issue of *Physica D*, The MIT Press.
- [Foster, 2002] Foster, I. (2002). What is the Grid? A Three Point Checklist. Available at <http://www-fp.mcs.anl.gov/~foster>.
- [Foster and Kesselman, 1999] Foster, I., and Kesselman, C. (Eds.) (1999). *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufman.
- [Foster et al., 2001] Foster, I., Kesselman, C., and Tuecke, S. (2001). The Anatomy of the Grid. Enabling Scalable Virtual Organizations. *International Journal of Supercomputer Applications*.
- [Foster et al., 2002] Foster, I., Kesselman, C., Nick, J., and Tucke, S. (2002). The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. *Open Grid Service Infrastructure WG*, Global Grid Forum.
- [Fox and Gruninger, 1998] Fox, M. S., and Gruninger, M. (1998). Enterprise Modelling. *AI Magazine*, 19(3), pages 109–121.

- [Fredriksson and Gustavsson, 2003] Fredriksson, M., and Gustavsson, R. (2003). Articulation of an Open Computational System for Network-Centric Warfare. Bubenko Jr., J. (Ed.) *Conference for the Promotion of Research in Information Technology at New Universities and at University Colleges (ITL)*, Visby, Sweden.
- [Fredriksson et al., 2003] Fredriksson, M., Gustavsson, R., and Ricci, A. (2003). Sustainable Coordination. Klusch, M., Bergamaschi, S., Edwards, P., and Petta, P. (Eds.) *Intelligent Information Agents: An AgentLink Perspective*, pages 203–233. Springer-Verlag.
- [Friedman-Hill, 2003] Friedman-Hill, E. (2003). Java Expert System Shell (JESS). Available at <http://herzberg.ca.sandia.gov/jess>.
- [Furmento et al., 2001] Furmento, N., Newhouse, S., and Darlington, J. (2001). Building Computational Communities from Federated Resources. *Proceedings of the 7th International Euro-Par Conference (Euro-Par 2001)*, pages 855–863.
- [Fuxman, 2001] Fuxman, A. (2001). Formal Analysis of Early Requirements Specifications. Master's Thesis, University of Toronto.
- [Fuxman et al., 2001] Fuxman, A., Pistore, M., Mylopoulos, J., and Traverso, P. (2001). Model Checking Early Requirements Specifications in Tropos. *Proceedings of the IEEE International Symposium on Requirements Engineering*, pages 174–181, Toronto, CA.
- [Fuxman et al., 2003] Fuxman, A., Liu, L., Pistore, M., Roveri, M., and Mylopoulos, J. (2003). Specifying and Analyzing Early Requirements in Tropos: Some Experimental Results. *Proceedings of the 11th IEEE International Requirements Engineering Conference*, Monterey Bay, USA.
- [Gamma et al., 1995] Gamma, E., Helm, R., Johnson, J., and Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- [Garcia-Molina and Germano, 1984] Garcia-Molina, H., Germano, F. (1984). Debugging a Distributed Computer System. *IEEE Transactions on Software Engineering*, 10(2), pages 210–219.
- [Garijo et al., 1998] Garijo, F., Tous, J., Matias, J. M., Corley, S., and Tessa-laar, M. (1998). Development of a Multi-Agent System for Cooperative Work with Network Negotiation Capabilities. Albayrak, S. (Ed.) *Intelligent Agents for Telecommunication Applications*, pages 204–219, Springer-Verlag.

- [Garijo et al, 2001] Garijo, F., Gómez-Sanz, J., Pavón, J., Massonet, P. Multi-Agent System Organisation. An Engineering Perspective. Proceedings of MAAMAW 2001, Springer-Verlag.
- [Gasser, 2001] Gasser, L. (2001). MAS Infrastructure Definitions, Needs, Prospects. Wagner, T., and Rana, O. (Eds.) *Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems*, pages 1–11. Springer-Verlag.
- [Gasser and Briot, 1992] Gasser, L., and Briot, J.-P. (1992). Distributed Artificial Intelligence: Theory and Praxis. *Object-Oriented Concurrent Programming and Distributed Artificial Intelligence*, pages 81–108. Kluwer Academic Publishers.
- [Gelernter and Carriero, 1992] Gelernter, D., and Carriero, N. (1992). Coordination Languages and Their Significance. *Communications of the ACM*, 25(2), pages 97–107.
- [Genesereth and Ketchpel, 1997] Genesereth, M. R., and Ketchpel, S. P. (1997). Software Agents. *Communications of the ACM*, 37(7).
- [Genesereth and Nilsson, 1987] Genesereth, M. R., and Nilsson, N. J. (1987). *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann Publisher.
- [Genesereth et al., 1986] Genesereth, M. R., Ginsburg, M., and Rosenschein, J. S. (1986). Cooperation without Communication. *Proceedings of the National Conference on Artificial Intelligence (AAAI'86)*, pages 51–57, AAAI Press.
- [Georgé et al., 2003] Georgé, J.-P., Gleizes, M.-P., Glize, P., and Régis, C. (2003). Real-Time Simulation for Flood Forecast: An Adaptive Multi-Agent System STAFF. Kazakov, D., Kudenko, D., and Alonso, E. (Eds.) *Proceedings of the AISB'03 Symposium on Adaptive Agents and Multi-Agent Systems (AAMAS'03)*, pages 109–114.
- [Gervais, 2003] Gervais, M.-P. (2003). ODAC: An Agent-Oriented Methodology based on ODP. *Journal of Autonomous Agents and Multi-Agent Systems*, 7(3), pages 199–228.
- [Gervais and Muscutariu, 2001] Gervais, M.-P., and Muscutariu, F. (2001). Towards an ADL for Designing Agent-Based Systems. Wooldridge, M. J., Weiss, G., and Cianciarini, P. (Eds.) *Agent-Oriented Software Engineering II*, pages 263–277. Springer-Verlag.
- [Giampapa and Sycara, 2002] Giampapa, J. A., and Sycara, K. (2002). Team-Oriented Agent Coordination in the RETSINA Multi-Agent System. Technical Report CMU-RI-TR-02-34, Carnegie Mellon University.

- [Gilber, 1995] Gilbert, G. N., 1995. Emergence in Social Simulation. Gilbert, G. N., and Conte, R. (Eds.) *Artificial Societies: The Computer Simulation of Social Life*, UCL Press.
- [Giorgini et al., 2002] Giorgini, P., Nicchiarelli, E., Mylopoulos, J., and Sebastiani, R. (2002). Reasoning with Goal Models. *Proceedings of the International Conference of Conceptual Modeling*, Springer-Verlag.
- [Giorgini et al., 2003] Giorgini, P., Mueller, J., and Odell, J. (Eds.) (2003). *Agent-Oriented Software Engineering III*, Springer-Verlag.
- [Giunchiglia et al., 2002] Giunchiglia, F., Mylopoulos, J., and Perini, A. (2002). The Tropos Software Development Methodology: Processes, Models and Diagrams. *Proceedings of Agent-Oriented Software Engineering*, Springer-Verlag.
- [Giunchiglia et al., 2003] Giunchiglia, F., Odell, J., and Weiss, G. (Eds.) (2003). *Agent-Oriented Software Engineering IV*, Springer-Verlag.
- [Glaser, 1996] Glaser, N. (1996). The CoMoMAS Methodology and Environment for Multi-Agent System Development. Zhang, C., and Lukose, D. (Eds.) *Multi-Agent Systems Methodologies and Applications*, pages 1–16. Springer-Verlag.
- [Gleizes et al., 1999] Gleizes, P.-P., Camps, V., and Glize, P. (1999). A Theory of Emergent Computation Based on Cooperative Self-Organization for Adaptive Artificial Systems. *Proceedings of the 4th European Congress of Systems Science*, Valencia, Spain.
- [Goldstein, 1999] Goldstein, J. (1999). Emergence as a Construct: History and Issues. *Emergence*, 1-1, pages 49–72.
- [Gomez-Sanz and Fuentes, 2002] Gomez-Sanz, J., and Fuentes, R. (2002). Agent Oriented System Engineering with INGENIAS. *Proceedings of the 4th Iberoamerican Workshop on Multi-Agent Systems*.
- [Gomez-Sanz and Pavon, 2003] Gomez-Sanz, J., and Pavon, J. (2003). Agent Oriented Software Engineering with INGENIAS. Marík, V., Müller, J., and Pechoucek, M. (Eds.) *Multi-Agent Systems and Applications III*, pages 394–403, Springer-Verlag.
- [Gomez-Sanz et al., 2002] Gomez-Sanz, J., Pavon, J., and Garijo, F. (2002). Meta-Models for Building Multi-Agent Systems. *Proceedings of the 2002 ACM symposium on Applied computing*, pages 37–41. ACM Press.
- [Graham et al., 1997] Graham, I., Hederson-Sellers, B., and Younessi, H. (1997). *The OPEN Process Specification*. Addison-Wesley.

- [Grassé, 1959] Grassé, P.-P. (1959). La Reconstruction du Nid et les Coordinations Inter-Individuelles chez *Bellicositermes Natalensis* et *Cubitermes* sp. La Théorie de la Stigmergie: Essai d'Interprétation du Comportement des Termites Constructeurs. *Insectes Sociaux*, 6, pages 41–84.
- [Gray and Reuter, 1993] Gray, J., and Reuter, A. (1993). *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann.
- [Gruber, 1991] Gruber, T. R. (1991). The Role of a Common Ontology in Achieving Sharable, Reusable Knowledge Bases. *Proceedings of the Knowledge Representation and Reasoning Conference*, pages 601–602.
- [Guerraoui and Schiper, 1997] Guerraoui, R., and Schiper, A. (1997). Software-Based Replication for Fault Tolerance. *IEEE Computer*, 30(4), pages 68–74.
- [Guessoum and Briot, 1999] Guessoum, Z., and Briot, J.-P. (1999). From Active Objects to Autonomous Agents. *IEEE Concurrency*, 7(3), pages 68–76.
- [Guessoum et al., 2002] Guessoum, Z., Briot, J.-P., and Charpentier, S. (2002). Dynamic and Adaptive Replication for Large-Scale Reliable Multi-Agent Systems. *Proceedings of the 1st International Workshop on Software Engineering for Large-Scale Multi-Agent Systems (SELMAS'02)*, Orlando, USA, ACM Press.
- [Gulyas and Corliss, 1999] Gulyas, T. Kozsik, L., and Corliss, J. B. (1999). The Multi-Agent Modelling Language and the Model Design Interface. *The Journal of Artificial Societies and Social Simulation*, 2(4).
- [Gurevich, 1984] Gurevich, Y. (1984). Toward Logic Tailored for Computational Complexity. *Computation and Proof Theory*, 1104, 175–216.
- [Gustavsson and Fredriksson, 2003] Gustavsson, R., and Fredriksson, M. (2003). Sustainable Information Ecosystems. Garcia, A., Lucena, C., Zambonelli, F., Omicini, A., and Castro, J. (Eds.) *Software Engineering for Large-Scale Multi-Agent Systems*, pp. 127-142, Springer-Verlag.
- [Gutknecht and Ferber, 2001] Gutknecht, O., and Ferber M. (2001). Integrating Tools and Infrastructures for Generic Multi-Agent Systems. *Proceedings of Agents'01*, pages 441–448.
- [Hackwood and Beni, 1991] Hackwood, S., and Beni, G. (1991). Self-Organizing Sensors by Deterministic Annealing. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robot and Systems*, pages 1177–1183.

- [Hackwood and Beni, 1992] Hackwood, S., and Beni, G. (1992). Self-Organization of Sensors for Swarm Intelligence. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 819–29.
- [Halpern et al., 1995] Halpern, J. Y., Fagin, R., Moses, Y., and Vardi, M. Y. (1995). *Reasoning About Knowledge*. The MIT Press.
- [Hayden et al., 1999] Hayden, S., Carrick, C., and Yang, Q. (1999). Architectural Design Patterns for Multiagent Coordination. *Proceedings of the 3rd International Conference on Autonomous Agents (Agents'99)*, Seattle, USA.
- [Heiler, 1995] Heiler, S. (1995). Semantic Interoperability. *ACM Computing Surveys*, 27(2), pages 271–273.
- [Heusse et al., 1998] Heusse, M., Guérin, S., Snyers, D., and Kuntz, P. (1998). Adaptive Agent-Driven Routing and Load Balancing in Communication Networks. *Advances in Complex Systems*, 1, pages 234–257.
- [Hexmoor, 2001] Hexmoor, H. (2001). A Cognitive Model of Situated Autonomy. *Advances in Artificial Intelligence*, pages 325–334. Springer-Verlag.
- [Hexmoor et al., 2003] Hexmoor, H., Castelfranchi, C., and Falcone, R. (2003). A Prospectus on Agent Autonomy. *Agent Autonomy*, 1(1), pages 1–8.
- [Heylighen, 1992] Heylighen, F. (1992). Evolution, Selfishness and Cooperation; Selfish Memes and the Evolution of Cooperation. *Journal of Ideas*, 2-4, pages 70–84.
- [Hindriks et al., 1999] Hindriks, K. V., Boer, F. S., der Hoek, W. V., and Meyer, J.-J. (1999). Agent Programming in 3APL. *Autonomous Agents and Multi-Agent Systems*, 2(4), pages 357–401.
- [Holland, 1997] Holland, J. H. (1997). *Emergence: From Order to Chaos*. Oxford University Press.
- [Holzmann, 1991] Holzmann, G. J. (1991). *Design and Validation of Computer Protocols*. Prentice-Hall International.
- [Holzmann, 1997] Holzmann, G. J. (1997). The Model Checker Spin. *IEEE Transactions on Software Engineering*, 23(5), pages 279–295.
- [Horn, 2001] Horn, P. M. (2001). Autonomic Computing – IBM's Perspective on the State of Information Technology. Available at <http://www.ibm.com/research/autonomic>.

- [Hornung and Bryan, 2002] Hornung, E., and Bryan, B. M. (Eds.) (2002). *The Quest for Immortality: Treasures of Ancient Egypt*. National Gallery of Art.
- [Howden et al., 2001] Howden, N., Ronnquist, R., Hodgson, A., and Lucas, A. (2001). JACK Intelligent Agents – Summary of an Agent Infrastructure. *Proceedings of Agents'01*.
- [Huang et al., 1999] Huang, A. C., Ling, B. C., and Ponnekanti, S. (1999). Pervasive Computing: What is it Good For? *Proceedings of The ACM International Workshop on Data Engineering for Wireless and Mobile Access (MOBIDE'1999)*, Seattle, USA.
- [Huberman, 1991] Huberman, B. A. (1991). The Performance of Cooperative Processes. Forrest, S. (Ed.) *Emergent Computation: Self-Organizing, Collective, and Cooperative Phenomena in Natural and Artificial Computing Networks*, The MIT Press.
- [Huget, 2002a] Huget, M.-P. (2002a). Agent UML Class Diagrams Revisited. Bauer, B., Fischer, K., Muller, J., and Rumpe, B. (Eds.) *Proceedings of Agent Technology and Software Engineering (AgeS)*.
- [Huget, 2002b] Huget, M.-P. (2002b). Generating Code for Agent UML Sequence Diagrams. Bauer, B., Fischer, K., Muller, J., and Rumpe, B. (Eds.) *Proceedings of Agent Technology and Software Engineering (AgeS)*.
- [Huget, 2002c] Huget, M.-P. (2002c). Model Checking Agent UML Protocol Diagrams. *Proceedings of the Workshop on Model Checking Artificial Intelligence (MoChArt)*.
- [Huget, 2002d] Huget, M.-P. (2002d). Nemo: An Agent-Oriented Software Engineering Methodology. *Proceedings of the Workshop on Agent-Oriented Methodologies*, pages 41–53, Seattle, USA.
- [Huhns and Singh, 1998] Huhns, M. N., and Singh, M. P. (1998). Agents and Multiagent Systems: Themes, Approaches, and Challenges. Huhns, M. N., and Singh, M. P. (Eds.) *Reading in Agents*, pages 1–23.
- [Huhns and Singh, 1999] Huhns, M. N., and Singh, M. P. (1999). A Multiagent Treatment of Agenthood. *Applied Artificial Intelligence*, 13(1–2), pages 3–10.
- [Huhns and Stephens, 1999] Huhns, M. N., and Stephens, L. M. (1999). Multiagent Systems and Societies of Agents. Weiss, G. (Ed.) *Multiagent Systems*, pages 79–120, The MIT Press.

- [Huhns et al., 2002] Huhns, M. N., Stephens, L. M., and Ivezic, N. (2002). Automating Supply-Chain Management. *Proceedings of the 1st International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 1017–1024, ACM Press.
- [Huzita and Scimemi, 1989] Huzita, H., and Scimemi, B. (1989). The Algebra of Paper-Folding. *Proceedings of the 1st International Meeting of Origami Science and Technology*.
- [Iglesias, 1998] Iglesias, C. (1998). *Definicion de una Metodologia para el Desarrollo de Sistemas Multi-Agente*. Ph.D. Thesis, Universidad Politécnica de Madrid.
- [Iglesias et al., 1998a] Iglesias, C. A., Garrijo, M., González, J. (1998a). A Survey of Agent-Oriented Methodologies. *Agent Theories, Architectures and Languages*.
- [Iglesias et al., 1998b] Iglesias, C. A., Garrijo, M., Gonzales, J. C., and Velasco, J. R. (1998b). Analysis and Design of Multi-Agent Systems Using MAS-CommonKADS. Singh, M. P., Rao, A., and Wooldridge, M. J. (Eds.) *Intelligent Agents IV*, pages 313–326. Springer-Verlag.
- [Inbody, 2003] Inbody, D. (2003). Swarming: Historical Observations and Conclusions. *Proceedings of Swarming: Network Enabled C4ISR*, Tysons Corner, VA.
- [Intanagonwiwat et al., 2000] Intanagonwiwat, C., Govindan, R., Estrin, D. (2000). Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. *Proceedings of MobiCom 2000*.
- [d’Inverno and Luck, 1996] d’Inverno, M., and Luck, M. (1996). A Formal View of Social Dependence Networks. *Proceedings of the 1st Australian Workshop on Distributed Artificial Intelligence*, pages 115–129, Springer-Verlag.
- [d’Inverno et al., 2000] d’Inverno, M., Hindriks, K., and Luck, M. (2000). A Formal Architecture for the 3APL Agent Programming Language. *Proceedings of the 1st International Conference of B and Z Users*, pages 168–187, Springer-Verlag.
- [ITU, 1999] International Telecommunications Union (1999). Languages for Telecommunication Applications, Formal description techniques (FDT): Message Sequence Charts (MSC). International Telecommunications Union.
- [Jacob, 2001] Jacob, C. (2001). *Illustrating Evolutionary Computation with Mathematica*. Morgan Kaufmann.

- [Jacobson et al., 1999] Jacobson, I., Booch, G., and Rumbaugh, J. (1999). *The Unified Software Development Process*. Addison-Wesley.
- [Jeng and Cheng, 1995] Jeng, J.-J., Cheng, B. H. C. (1995). Specification Matching for Software Reuse: A Foundation. Proceedings of the ACM SIGSOFT Symposium Software Reusability, ACM Press.
- [Jennings, 2000] Jennings, N. R. (2000). On Agent-Based Software Engineering. *Artificial Intelligence*, 117(2), pages 277–296.
- [Jennings, 2001] Jennings, N. R. (2001). An Agent-Based Approach for Building Complex Software Systems. *Communications of the ACM*, 44(4), pages 35–41.
- [Jennings et al., 1998] Jennings, N. R., Sycara, K., and Wooldridge, M. J. (1998). A Roadmap of Agent Research and Development. *Autonomous Agents and Multi-Agent Systems*, 1(1).
- [Jennings et al., 2000] Jennings, N. R., Faratin, P., Norman, T. J., O'Brien, P., and Odgers, B. (2000). Autonomous Agents for Business Process Management. *International Journal of Applied Artificial Intelligence*, 14(2).
- [Johansson and Saffiotti, 2001] Johansson, S. J., and Saffiotti, A. (2001). Using the Electric Field Approach in the RoboCup Domain. *Proceedings of RoboCup*.
- [Johnson and Zweig, 1991] Johnson R. E., Zweig, J. M. (1991). Delegation in C++. *The Journal of Object Oriented Programming*, 4(7), pages 31–34.
- [Joyce et al., 1987] Joyce, J., Lomow, G., Slind, K., and Unger, B. (1987). Monitoring Distributed Systems. *ACM Transactions on Computer Systems*, 5(2), pages 121–150.
- [Juan et al., 2002] Juan, T., Pierce, A., and Sterling, L. (2002). ROADMAP: Extending the Gaia methodology for Complex Open Systems. *Proceedings of the 1st ACM Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 3–10, ACM Press.
- [Judge et al., 1998] Judge, D. W., Odgers, B. R., Shepherdson, J. W., and Cui, Z. (1998). Agent-Enhanced Workflow. *BT Technology Journal*, 16(3), pages 79–85.
- [Julian and Botti, 2004] Julian, V., and Botti, V. (2004) Developing Real-Time Multi-Agent Systems. To appear in *Integrated Computer Aided Engineering Journal*, 2004.

- [Kagal et al., 2001] Kagal, L., Finin, T., and Joshi, A. (2001). Moving from Security to Distributed Trust in Ubiquitous Computing Environments. *IEEE Computer*, 34(12).
- [Kahn and Cicalese, 2001] Kahn, M. L., and Cicalese, C. (2001). CoABS Grid Scalability Experiments. *Proceedings of the 2nd International Workshop on Infrastructure for Scalable Multi-Agent systems at Autonomous Agents*, Montreal, CA.
- [Kahn et al., 1999] Kahn, J. M., Katz, R. H., and Pister, K. S. (1999). Mobile Networking for Smart Dust. *Proceedings of MobiCom'99*.
- [Karamcheti et al., 1996] Karamcheti, V., Plevyak, J., and Chien, A., (1996). Runtime Mechanisms for Efficient Dynamic Multithreading. *Journal of Parallel and Distributed Computing*, 37, pages 21–40.
- [Kawamura et al., 1999] Kawamura, T., Yoshioka, N., Hasegawa, T., Ohsuga, A., and Honiden, S., (1999). Bee-gent: Bonding and Encapsulation Enhancement Agent Framework for Development of Distributed Systems. *Proceedings of the 6th Asia-Pacific Software Engineering Conference*.
- [Kendall, 1998] Kendall, E. A. (1998). Agent Roles and Role Models: New Abstractions for Multiagent System Analysis and Design. *Proceedings of the International Workshop on Intelligent Agents in Information and Process Management*.
- [Kendall, 2000] Kendall, E. A. (2000). Agent Software Engineering with Role Modelling. *Proceedings of the Workshop on Agent-Oriented Software Engineering*, pages 163–169, Springer-Verlag.
- [Kendall and Malkoun, 1996] Kendall, E. A., and Malkoun, M. T. (1996). The Layered Agent Patterns. *Pattern Languages of Programs (PLoP'96)*.
- [Kendall et al., 1995] Kendall, E. A., Malkoun, M. T., and Jiang, C. H. (1995). A Methodology for Developing Agent Based Systems. Zhang, C., and Lukose, D. (Eds.) *Proceedings of the 1st Australian Workshop on Distributed Artificial Intelligence*.
- [Kennedy et al., 2001] Kennedy, J., Eberhart, R. C., and Shi, Y. (2001). *Swarm Intelligence*. Morgan Kaufmann.
- [Kephart, 2002] Kephart, J. (2002). Software Agents and the Route to the Information Economy. *Proceedings of the National Academy of Science*, 99(3), pages 7207–7213.
- [Kerr et al., 1998] Kerr, D., O'Sullivan, D., Evans, R., Richardson, R., and Somers, F., (1998). Experiences using Intelligent Agent Technologies as

- a Unifying Approach to Network and Service Management. *Proceedings of IS & N'98*, Antwerp, Belgium.
- [Kiczales et al., 1991] Kiczales, G., des Rivières, J., and Bobrow, D. G. (1991). *The Art of the Metaobject Protocol*. The MIT Press.
- [Kim, 1997] Kim, J. (1997). Explanation, Prediction, and Reduction in Emergentism. *Intellectica – Emergence and Explanation*, 2-25.
- [Kinny and Georgeff, 1996] Kinny, D., and Georgeff, M. (1996). Modelling and Design of Multi-Agent Systems. *Intelligent Agents III*, Springer-Verlag.
- [Kinny et al., 1996] Kinny, D., Georgeff, M., and Rao, A. (1996). A Methodology and Modelling Technique for Systems of BDI Agents. van der Velde, W., and Perram, J. (Eds.) *Agents Breaking Away*, pages 56–71. Springer-Verlag.
- [Klusch and Sycara, 2001] Klusch, M., and Sycara, K. (2001). Brokering and Matchmaking for Coordination of Agent Societies: A Survey. Omicini, A., Zambonelli, F., Klusch, M., and Tolksdorf, R. (Eds.) *Coordination of Internet Agents: Models, Technologies, and Applications*, pages 197–224.
- [Knublauch and Rose, 2002] Knublauch, H. H., and Rose, T. (2002). Tool-Supported Process Analysis and Design for the Development of Multi-Agent Systems. *Proceedings of AOSE 2002*, Springer-Verlag.
- [Kolp et al., 2001] Kolp, M., Giorgini, P., and Mylopoulos, J. (2001). A Goal-Based Organizational Perspective on Multi-Agents Architectures. *Proceedings of the 8th International Workshop on Agent Theories, Architectures, and Languages, ATAL'01*, Seattle, USA.
- [Koning and Romero-Hernandez, 2002] Koning, J.-L., and Romero-Hernandez, I. (2002). Generating Machine Processable Representations of Textual Representations of AUML. Giunchiglia, F., Odell, J., and Weiss, G. (Eds.) *Proceedings of the Workshop on Agent-Oriented Software Engineering (AOSE)*.
- [Kortuem and Segall, 2003] Kortuem, G., and Segall, Z. (2003). Wearable Communities: Augmenting Social Networks with Wearable Computers. *IEEE Pervasive Computing Magazine*, 2(1).
- [Kouadri Mostéfaoui, 2003] Kouadri Mostéfaoui, S. (2003). Towards a Context-Oriented Services Discovery and Composition Framework. *Proceedings of the Workshop on Artificial Intelligence, Information Access, and Mobile Computing*, Acapulco, Mexico.

- [Kruchten, 2000] Kruchten, P. (2000). The Rational Unified Process: An Introduction. Addison-Wesley.
- [Kumar, 2002] Kumar, M. (2002). Contrast and Comparison of Five Major Agent Oriented Software Engineering (AOSE) methodologies. Available at <http://students.jmc.ksu.edu/grad/madhukar/www/professional/aosepaper.pdf>.
- [Kuutti, 1991] Kuutti, K. (1991). The Concept of Activity as a Basic Unit of Analysis for CSCW Research. *Proceedings of the 2nd European Conference on Computer Supported Cooperative Work (ECSCW '91)*, pages 249–264, Kluwer Academic Publishers.
- [Kuwabara et al., 1996] Kuwabara, K., Ishida, T., Nishibe, Y., and Suda, T. (1996). An Equilibratory Market-Based Approach for Distributed Resource Allocation and Its Applications to Communication Network Control. *Market-Based Control: A Paradigm for Distributed Resource Allocation*, pages 53–73, World Scientific Publishing.
- [Labrou and Finin, 1997] Labrou, Y., and Finin, T. (1997). A Proposal for a New KQML Specification. Technical Report TR-CS-97-03, University of Maryland.
- [Labrou et al., 1999] Labrou, Y., Finin, T., and Peng, Y. (1999). The Current Landscape of Agent Communication Languages. *Intelligent Systems*, 14(2), pages 45–52.
- [Lacey et al., 2000] Lacey, T. H., and DeLoach, S. A. (2000). Automatic Verification of Multiagent Conversations. *Proceedings of the 11th Annual Midwest Artificial Intelligence and Cognitive Science Conference*, Fayetteville, USA.
- [Laird et al., 1987] Laird, J. E., Newell, A., and Rosenbloom, P. S. (1987). SOAR: An Architecture for General Intelligence. *Artificial Intelligence*, 33(1), pages 1–64.
- [Lange and Oshima, 1999] Lange, D., and Oshima, M. (1999). Dispatch Your Agents; Shut Off Your Machine. *Communications of the ACM*, 42(3).
- [Lavender and Schmidt, 1996] Lavender, G., and Schmidt, D., (1996). Active Object: An Object Behavioural Pattern for Concurrent Programming. Vlissides, J.M., Coplien, J.O., and Kerth, N.L. (Eds.) *Pattern Languages of Program Design*, Addison-Wesley.
- [Lawley et al., 2003] Lawley, R., Decker, K., Luck, M., Payne, T., and Moreau, L. (2003). Automated Negotiation for Grid Notification Services.

- Proceedings of the 9th International Europar Conference (EURO-PAR'03)*, Springer-Verlag.
- [Lawrence, 1992] Lawrence, P. A. (1992). *The Making of a Fly: the Genetics of Animal Design*. Blackwell Science.
- [Lea, 1997] Lea, D. (1997). *Concurrent Programming in Java: Design Principles and Patterns*. Addison-Wesley.
- [Leontjev, 1978] Leontjev, A. (1978). *Activity, Consciousness, and Personality*. Prentice-Hall International.
- [Lesperance et al., 1999] Lesperance, Y., Kelley, T. G., Mylopoulos, J., and Yu, E. (1999). Modeling Dynamic Domains with Congolog. *Proceedings of CAiSE'99*, pages 365–380, Springer-Verlag.
- [Leyman, 2001] Leyman, F. (2001). *Web Services Flow Language (WSFL)*. Technical report, IBM.
- [Lind, 2000] Lind, J. (2000). A Development Method for Multiagent systems. *Proceedings of the 15th European Meeting on Cybernetics and Systems Research*.
- [Lind, 2001] Lind, J. (2001). *Iterative Software Engineering for Multi-Agent Systems, The MASSIVE Method*. Springer-Verlag.
- [Loo, 2003] Loo, A. W. (2003). The Future of Peer-to-Peer Computing. *Communications of the ACM*, 46(9), pages 56–61, ACM Press.
- [Lord et al., 2003] Lord, P., Wroe, C., Stevens, R., Goble, C., Miles, S., Moreau, L., Decker, K., Payne, T., and Papay, J. (2003). Semantic and Personalised Service Discovery. Cheung, W. K., and Ye, Y. (Eds.) *Proceedings of the Workshop on Knowledge Grid and Grid Intelligence (KGGI'03)*, pages 100–107.
- [Luck and d'Inverno, 1995] Luck, M., and d'Inverno, M. (1995). A Formal Framework for Agency and Autonomy. Lesser, V., and Gasser, L. (Eds.) *Proceedings of the 1st International Conference on Multi-Agent Systems (ICMAS-95)*, pages 254–260, AAAI Press.
- [Luck and d'Inverno, 2001] Luck, M., and d'Inverno, M. (2001). *A Conceptual Framework for Agent Definition and Development*. *The Computer Journal* 44(1), pages 1–20.
- [Luck et al., 2003] Luck, M., McBurney, P., and Preist, C. (2003). *Agent Technology: Enabling Next Generation Computing*. AgentLink II.

- [Lynch, 1996] Lynch, N. (1996). *Distributed Algorithms*. Morgan Kaufmann.
- [Lyytinen and Yoo, 2002] Lyytinen, K., and Yoo, Y. (2002). Issues and Challenges in Ubiquitous Computing. *Communications of the ACM*, 45(12).
- [Maamar et al., 2001] Maamar, Z., Dorion, E., and Daigle, C. (2001). Towards Virtual Marketplaces for E-Commerce. *Communications of the ACM*, 44(12).
- [Maamar et al., 2004] Maamar, Z., Sheng, Q. Z., and Benatallah, B. (2004). On Composite Web Services Provisioning in an Environment of Fixed and Mobile Computing Resources. *Information Technology and Management Journal*, 5(3), Kluwer Academic Publishers.
- [Machado, 2003] Machado, R. (2003). SIM Speak. Available at http://www.inf.ufrgs.br/~bordini/SIM_Speak.
- [Machado and Bordini, 2001] Machado, R., and Bordini, R. H. (2001). Running AgentSpeak(L) Agents on SIM_AGENT. Meyer, J. J., and Tambe, M. (Eds.) *Intelligent Agents VIII*, pages 158–174.
- [Maes, 1994] Maes, P. (1994). Agents that Reduce Work and Information Overload. *Communications of the ACM*, 37(7), pages 31–40.
- [Malone and Crowston, 1994] Malone, T., and Crowston, K. (1994). The Interdisciplinary Study of Coordination. *ACM Computing Surveys*, 26(1), pages 87–119.
- [Mamei et al., 2003a] Mamei, M., Zambonelli, F., and Leonardi, L. (2003a). Tuples on the Air: a Middleware for Context-Aware Computing in Dynamic Networks. *Proceedings of the International ICDCS Workshop on Mobile Computing Middleware*.
- [Mamei et al., 2003b] Mamei, M., Zambonelli, F., and Leonardi, L. (2003b). Distributed Motion Coordination with Co-Fields: A Case Study in Urban Traffic Management. *Proceedings of the IEEE Symposium on Autonomous Decentralized Systems*.
- [Mangina, 2002] Mangina, E. (2002). Review of Software Products for Multi-Agent Systems. AgentLink II.
- [Marik et al., 2003] Marik, V., Pechoucek, M., Vrba, P., and Hrdonka, V., (2003). FIPA Standards and Holonic Manufacturing. Agent Based Manufacturing. Deen, S. M. (Ed.), *Advances in the Holonic Approach*, pages 89–121, Springer-Verlag.

- [Massonet et al., 2002] Massonet, P., Deville, Y., and Neve, C. (2002). From AOSE Methodology to Agent Implementation. *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02)*, Bologna, Italy.
- [Mathieson et al., 2004] Mathieson, I., Dance, S., Padgham, L., Gorman, M., and Winikoff, M. (2004). An Open Meteorological Alerting System: Issues and Solutions. *Proceedings of the 27th Australasian Computer Science Conference*, Dunedin, New Zealand.
- [Maximilien and Singh, 2002] Maximilien, E. M., and Singh, M. P. (2002). Reputation and Endorsement for Web Services. *ACM SIGEcom Exchanges*, 3(1), pages 24–31.
- [McCabe and Clark, 1995] McCabe, F. G., and Clark, K. L. (1995). April – Agent PROcess Interaction Language. Wooldridge, M. J., and Jennings, N. (Eds.) *Intelligent Agents*, Springer-Verlag.
- [McCarthy, 1978] McCarthy, J. (1978). History of LISP. *The 1st ACM SIG-PLAN Conference on History of Programming Languages*, pages 217–223.
- [McCarthy and Hayes, 1981] McCarthy, J., and Hayes, P. J. (1981). Some Philosophical Problems from the Standpoint of Artificial Intelligence. Weber, B. L. and Nilsson, N. J. (Eds.) *Readings in Artificial Intelligence*, pages 431–450. Morgan Kaufmann.
- [McIlraith and Martin, 2003] McIlraith, S., and Martin, D. (2003). Bringing Semantics to Web Services. *IEEE Intelligent Systems*, 18(1), pages 90–93.
- [Medvidovic and Taylor, 1997] Medvidovic, N., and Taylor, R. N. (1997). A Framework for Classifying and Comparing Architecture Description Languages. *Proceedings of the 5th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, pages 60–76, Springer-Verlag.
- [Meling et al., 2001] Meling, H., Montresor, A., and Babaoglu, O. (2001). Peer-to-Peer Document Sharing using the Ant Paradigm. *Proceedings of the Norsk Informatikkonferanse (NIK)*.
- [Mendez and Tolksdorf, 2003] Mendez, R., and Tolksdorf R. (2003). A New Approach to Scalable Linda-Systems Based on Swarms. *Proceedings of ACM SAC '03*.
- [Messer et al., 2002] Messer, A., Greeberg, I., Bernadat, P., and Milojevic, D. (2002). Towards a Distributed Platform for Resource-Constrained Devices. *Proceedings of the IEEE 22nd International Conference on Distributed Computing Systems (ICDCS'2002)*, Vienna, Austria.

- [Meyer, 1992] Meyer, B. (1992). Applying Design by Contract. *IEEE Computer*, 25(10), pages 40–51.
- [Meyer, 1997] Meyer, B. (1997). *Object-Oriented Software Construction*. Prentice-Hall International.
- [Miles et al., 2000] Miles, S., Joy, M., and Luck, M. (2000). Designing Agent-Oriented Systems by Analysing Agent Interactions. Ciancarini, P., and Wooldridge, M. J. (Eds.) *Proceedings of Agent-Oriented Software Engineering 2000 (AOSE 2000)*, pages 171–184.
- [Miles et al., 2003] Miles, S., Papay, J., Dialani, V., Luck, M., Decker, K., Payne, T., and Moreau, L. (2003). Personalised Grid Service Discovery. *IEE Proceedings of Software: Special Issue on Performance Engineering*, 150(4), pages 252–256.
- [Miller et al., 1996] Miller, M. S., Krieger, D., Hardy, N., Hibbert, C., and Tribble, D. E. (1996). An Automated Auction in ATM Network Bandwidth. *Market-Based Control. A Paradigm for Distributed Resource Allocation*, pages 96–125.
- [Milojicic et al., 1998] Milojevic, D., Breugst, M., Busse, I., Campbell, J., Covaci, S., Friedman, B., Kosaka, K., Lange, D., Ono, K., Oshima, M., Tham, C., Virdhagriswaran, S., and White J. (1998). MASIF – The OMG Mobile Agent System Interoperability Facility. Rothermel K., and Hohl, F. (Eds.) *Proceedings of the 2nd International Workshop Mobile Agents*, pages 50–67, Springer-Verlag.
- [Mintzberg, 1992] Mintzberg, H. (1992). *Structure in Fives: Designing Effective Organizations*. Prentice-Hall International.
- [Morabito et al., 1999] Morabito, J., Sack, I., and Bhate, A. (1999). *Organization Modeling: Innovative Architectures for the 21st Century*. Prentice-Hall International.
- [Moraitis et al., 2002] Moraitis, P., Petraki, E., and Spanoudakis, N. I. (2002). Engineering JADE Agents with the Gaia Methodology. *Proceedings of the International Workshop on Agents and Software Engineering*.
- [Moreau, 2002] Moreau, L. (2002). Agents for the Grid: A Comparison with Web Services (Part 1: The Transport Layer). Bal, H. E., Lohr, K.-P., and Reinefeld, A. (Eds.) *Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID 2002)*, pages 220–228, Berlin, Germany.
- [Moreau et al., 2003] Moreau, L., Miles, S., Goble, C., Greenwood, M., Dialani, V., Addis, M., Alpdemir, N., Cawley, R., De Roure, D., Ferris, J.,

- Gaizauskas, R., Glover, K., Greenhalgh, C., Li, P., Liu, X., Lord, P., Luck, M., Marvin, D., Oinn, T., Paton, N., Pettifer, S., Radenkovic, M. V., Roberts, A., Robinson, A., Rodden, T., Senger, M., Sharman, N., Stevens, R., Warboys, B., Wipat, A., and Wroe, C. (2003). On the Use of Agents in a BioInformatics Grid. *Proceedings of the Third IEEE/ACM CCGRID'2003 Workshop on Agent Based Cluster and Grid Computing*, pages 653–661, Tokyo, Japan.
- [Moss, 2000] Moss, S. (2000). *Editorial Introduction: Messy Systems – The Target for Multi Agent Based Simulation*, Springer-Verlag.
- [Moulin and Chaib-draa, 1996] Moulin, B., and Chaib-draa, B. (1996). An Overview of Distributed Artificial Intelligence. *Foundations of Distributed Artificial Intelligence*, John Wiley & Sons.
- [Muller, 1996] Muller, J. P. (1996). *The Design of Intelligent Agents: A Layered Approach*, Springer-Verlag.
- [Muller, 2003] Muller, J. P. (2003). The Right Agent (Architecture) to do the Right Thing. *Intelligent Agents V*, pages 105–112, Springer-Verlag.
- [Muller and Pischel, 1994] Muller, J., and Pischel, M. (1994). Modelling Reactive Behaviour in Vertically Layered Agent Architectures. Cohen, A. G. (Ed.) 11th *European Conference on Artificial Intelligence (ECAI'94)*, pages 709–713.
- [Mylopoulos and Castro, 2000] Mylopoulos, J., and Castro, J. (2000). Tropos: A Framework for Requirements-Driven Software Development. *Proceedings of 12th Conference on Advanced Information Systems Engineering (CAISE)*.
- [Nakajima et al., 2003] Nakajima, T., Awa, I., and Tokunaga, E. (2003). A Proactive Middleware Platform for Mobile Computing. *Proceedings of the 4th ACM International Middleware Conference (Middleware'2003)*, Rio de Janeiro, Brazil.
- [Nagpal, 2001] Nagpal, 2001. Programmable Self-Assembly: Constructing Global Shape using Biologically-Inspired Local Interactions and Origami Mathematics. Ph.D. Thesis, MIT, 2001.
- [Nagpal, 2002] Nagpal, R. (2002). Programmable Self-Assembly Using Biologically-Inspired Multiagent Control. *Proceedings of the Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.
- [Nagpal and Coore, 1998] Nagpal, R., and Coore, D. (1998). An Algorithm for Group Formation in an Amorphous Computer. *Proceedings of the In-*

ternational Conference on Parallel and Distributed Computing and Systems (PDCS).

- [Nagpal et al., 2003] Nagpal, R., Shrobe, H., and Bachrach, J. (2003). Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network. *Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks (IPSN)*.
- [Ndumu et al., 1999] Ndumu, D. T., Nwana, H. S., Lee, L. C., and Collis, J. C. (1999). Visualising and Debugging Distributed Multi-Agent Systems. *Proceedings of the 3rd Annual Conference on Autonomous Agents*, pages 326–333, ACM Press.
- [Neufeld, 1980] Neufeld, E. (1980). Insects as Warfare Agents in the Ancient Near East. *Orientalia*, 49(1), pages 30–57.
- [Newell, 1982] Newell, A. (1982). The Knowledge Level. *Artificial Intelligence*, 7(18), pages 87–127.
- [Newell, 1993] Newell, A. (1993). Reflections on the Knowledge Level. *Artificial Intelligence*, 59, pages 31–38.
- [Nilsson, 1971] Nilsson, N. (1971). *Problem Solving Methods in Artificial Intelligence*. McGraw Hill.
- [Noriega and Sierra, 1999] Noriega, P., and Sierra, C. (1999). Auctions and Multi-agent Systems. Klusch, M. (Ed.) *Information Agents*, pages 153–175, Springer-Verlag.
- [Noriega and Sierra, 2002] Noriega, P., and Sierra, C. (2002). Electronic Institutions: Future Trends and Challenges. Klusch, M., Ossowski, S., and Shehory, O. (Eds.) *Cooperative Information Agents VI*, Springer-Verlag.
- [Nuseibeh and Easterbrook, 2000] Nuseibeh, B. A., and Easterbrook, S. M. (2000). Requirements Engineering: A Roadmap. *Proceedings of the 22nd International Conference on Software Engineering (ICSE'00)*, pages 35–46.
- [Nutt, 1996] Nutt, G. (1996). The Evolution Toward Flexible Workflow Systems. *Distributed Systems Engineering*, 3(4), pages 276–294.
- [Nwana, 1996] Nwana, H. S. (1996). Software Agents: An Overview. *The Knowledge Engineering Review*, 11(3), pages 205–244.
- [Nwana and Ndumu, 1999] Nwana, H. S., and Ndumu, D. T. (1999). A Perspective on Software Agents Research. *The Knowledge Engineering Review*, 14(2), pages 1–18.

- [Nwana et al., 1999] Nwana, H. S., Ndumu, D. T., Lee, L. C., and Collis, J. C. (1999). Zeus: A Toolkit for Building Distributed Multi-agent Systems. *Applied Artificial Intelligence Journal*, 1(13), pages 129–185.
- [Oaks and Wong, 2000] Oaks, S., and Wong, H. (2000). *Jini in a Nutshell*. O'Reilly.
- [Obreiter et al., 2003] Obreiter, P., König-Ries, B., and Klein, M. (2003). Stimulating Cooperative Behavior of Autonomous Devices – An Analysis of Requirements and Existing Approaches. Technical Report 2003-1, University of Karlsruhe.
- [Odell, 2002] Odell, J. (2002). Objects and Agents Compared. *Journal of Object Computing*, 1:1.
- [Odell et al., 2000] Odell, J., Parunak, V. H., and Bauer, B. (2000). Extending UML for Agents. *Proceedings of the Agent Oriented Information Systems (AOIS) Workshop*.
- [Odell et al., 2001] Odell, J., Parunak, V. H., and Bauer, B. (2001). Representing Agent Interaction Protocols in UML. Ciancarini, P., and Wooldridge, M. J. (Eds.) *Proceedings of the First International Workshop on Agent Oriented Software Engineering (AOSE-2000)*, pages 121–140, Springer-Verlag.
- [OMG, 1999] Object Management Group (1999). Mobile Agent System Interoperability Facility (MASIF). Available at <http://www.fokus.gmd.de/research/cc/ecco/masif>.
- [OMG, 2000a] Object Management Group (2000a). CORBA 2.4.2 Specification. Available at <http://www.omg.org>.
- [OMG, 2000b] Object Management Group (2000b). Meta Object Facility (MOF). Available at <http://www.omg.org>.
- [OMG, 2000c] Object Management Group (2000c). Unified Modeling Language Specification. Version 1.3. Available at <http://www.omg.org>.
- [OMG, 2001] Object Management Group (2001). Model Driven Architecture (MDA). Technical report, OMG.
- [OMG, 2002] Object Management Group (2002). Software Process Engineering Metamodel. Version 1.0. Available at <http://www.omg.org/technology/documents/formal/spem.htm>.
- [OMG, 2003a] Object Management Group (2003a). XML Metadata Interchange – Version 1.1. Available at <http://www.omg.org>.

- [OMG, 2003b] Object Management Group (2003b). Unified Modeling Language: Superstructure – Version 2.0. Available at <http://www.omg.org>.
- [OMG, 2003c] Object Management Group (2003c). Response to the UML 2.0 OCL RfP. Available at <http://www.omg.org>.
- [Omicini, 2001] Omicini, A. (2001). SODA: Societies and Infrastructures in the Analysis and Design of Agent-Based Systems. Ciancarini, P., and Wooldridge, M. J. (Eds.) *Agent-Oriented Software Engineering*, pages 185–193. Springer-Verlag.
- [Omicini, 2002] Omicini, A. (2002). Towards a Notion of Agent Coordination Context. Marinescu, D. C., and Lee, C. (Eds.) *Process Coordination and Ubiquitous Computing*, pages 187–200. CRC Press.
- [Omicini and Denti, 2001] Omicini, A., and Denti, E. (2001). From Tuple Spaces to Tuple Centres. *Science of Computer Programming*, 41(3), pages 277–294.
- [Omicini and Ossowski, 2003] Omicini, A., and Ossowski, S. (2003). Objective versus Subjective Coordination in the Engineering of Agent Systems. Klusch, M., Bergamaschi, S., Edwards, P., and Petta, P. (Eds.) *Intelligent Information Agents: An AgentLink Perspective*, pages 179–202, Springer-Verlag.
- [Omicini and Ricci, 2003] Omicini, A., and Ricci, A. (2003). Reasoning about Organisation: Shaping the Infrastructure. *AI*IA Notizie*, XVI(2), pages 7–16.
- [Omicini and Zambonelli, 1999] Omicini, A., and Zambonelli, F. (1999). Coordination for Internet Application Development. *International Journal on Autonomous Agents and Multi-Agent Systems*, 2(3), pages 251–269.
- [Omicini et al., 2003] Omicini, A., Ricci, A., and Viroli, M. (2003). Formal Specification and Enactment of Security Policies through Agent Coordination Contexts. Focardi, R., and Zavattaro, G. (Eds.) *Security Issues in Coordination Models, Languages and Systems*, Elsevier Science.
- [Ossowski, 1999] Ossowski, S. (1999). *Coordination in Artificial Agent Societies – Social Structure and Its Implications for Autonomus Problem-solving Agents*, Springer-Verlag.
- [Ossowski and Omicini, 2002] Ossowski, S., and Omicini, A. (2002). Coordination Knowledge Engineering. *The Knowledge Engineering Review*, 17(4), pages 309–316.

- [Ossowski et al., 2002] Ossowski, S., Hernández, J. Z., Iglesias, C. A., and Fernández, A. (2002). Engineering Agent Systems for Decision Support. Petta, P., Tolksdorf, R., and Zambonelli, F. (Eds.) *Engineering Societies in an Agent World III*. Springer-Verlag.
- [Ott et al., 1990] Ott, E., Grebogi, C., and Yorke, J. A. (1990). Controlling Chaos. *Physical Review Letters*, 64(11), pages 1196–1199.
- [Padgham and Winikoff, 2002] Padgham, L., and Winikoff, M. (2002). Prometheus: A Pragmatic Methodology for Engineering Intelligent Agents. *Proceedings of the Workshop on Agent-Oriented Methodologies*, pages 97–108.
- [Paolucci et al., 2002] Paolucci, M., Kawamura, T., Payne, T. R., and Sycara, K. (2002). Importing the Semantic Web in UDDI. *Proceedings of the Workshop on Web Services, E-Business and Semantic Web*.
- [Papadopoulos and Arbab, 1998] Papadopoulos, G. A., and Arbab, F. (1998). Coordination Models and Languages. *The Engineering of Large Systems*, 46.
- [Papasimeon and Heinze, 2001] Papasimeon, M., and Heinze, C. (2001). Extending the UML for Designing JACK Agents. *Proceedings of the Australian Software Engineering Conference (ASWEC 01)*.
- [Parunak, 1990] Parunak, H. V. D. (1990). Distributed AI and Manufacturing Control: Some Issues and Insights. Demazeau, Y., and Muller, J.-P. (Eds.) *Decentralized AI*, pages 81–104.
- [Parunak, 1997] Parunak, H.V.D. (1997). Go to the Ant: Engineering Principles from Natural Agent Systems. *Annals of Operations Research*, 75, pages 69–101.
- [Parunak and Brueckner, 2001] Parunak, H. V. D., and Brueckner, S. (2001). Entropy and Self-Organization in Multi-Agent Systems. *Proceedings of the 5th International Conference on Autonomous Agents (Agents 2001)*, pages 124–130, ACM Press.
- [Parunak and Brueckner, 2003] Parunak, H. V. D., and Brueckner, S. (2003). Swarming Coordination of Multiple UAV's for Collaborative Sensing. *Proceedings of 2nd AIAA "Unmanned Unlimited" Systems, Technologies, and Operations Conference*, San Diego, USA.
- [Parunak and Odell, 2001] Parunak, H. V. D., and Odell, J. (2001). Representing Social Structures in UML. *Proceedings of the 5th International Conference on Autonomous Agents*, pages 100–101. ACM Press.

- [Parunak and Vanderbok, 1997] Parunak, H. V. D., and Vanderbok, R. S. (1997). Managing Emergent Behavior in Distributed Control Systems. *Proceedings of ISA-Tech'97*.
- [Parunak et al., 1998a] Parunak, H. V. D., Savit, R., and Riolo R. L. (1998). Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide. *Proceedings of the Workshop on Modeling Agent Based Systems*, pages 1–15.
- [Parunak et al., 1998b] Parunak, H. V. D., Sauter, J., and Clask, S. (1998). Toward the Specification and Design of Industrial Synthetic Ecosystems. Singh, M., Rao, A., and Wooldridge, M. J. (Eds.) *Agent Theories, Architectures, and Languages (ATAL)*, pp. 45–59, Springer-Verlag.
- [Parunak et al., 2000] Parunak, H. V. D., Brueckener, S. A., Sauter, J., and Matthews, R. (2000). Distinguishing Environmental Properties and Agent Dynamics: A Case Study in Abstraction and Alternate Modelling Technologies. Omicini, A., Tolksdorf, R., and Zambonelli, F. (Eds.) *Engineering Societies in the Agents' Eorld (ESAW)*, pp. 19–33, Springer-Verlag.
- [Parunak et al., 2002a] Parunak, H. V. D., Brueckner, S. A., and Sauter, J. (2002a). Digital Pheromone Mechanisms for Coordination of Unmanned Vehicles. *Proceedings of 1st International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002)*, pages 449–450.
- [Parunak et al., 2002b] Parunak, H. V. D., Purcell, M., and O'Connell, R. (2002b). Digital Pheromones for Autonomous Coordination of Swarming UAV's. *Proceedings of 1st AIAA "Unmanned Aerospace Vehicles" Systems, Technologies, and Operations Conference*, Norfolk, VA.
- [Pasquier and Chaib-draa, 2003] Pasquier, P., and Chaib-draa, B. (2003). The Cognitive Approach for Agent Communication Pragmatics. *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multi-agent Systems*, pages 544–551, ACM Press.
- [Patil et al., 1992] Patil, R. S., Fikes, R. E., Patel-Scheneider, P. F., McKay, D., Finin, T., Gruber, T., and Neches, R. (1992). The DARPA Knowledge Sharing Effort: Progress Report. *Proceedings of the 3rd Conference on Principles of Knowledge Representation and Reasoning*, pages 103–114. Cambridge, USA.
- [Pavón and Gómez-Sanz, 2003] Pavón, J., and Gómez-Sanz, J. (2003). Agent Oriented Software Engineering with INGENIAS. *Multi-Agent Systems and Applications III*, pages 394–403, Springer-Verlag.

- [Payton et al., 2002] Payton, D., Estkowski, E., and Howard, R. (2002). Progress in Pheromone Robotics. *Proceedings of the 7th International Conference on Intelligent Autonomous Systems*.
- [Pister, 2001] Pister, K. (2001). Smart Dust: Autonomous Sensing and Communication in a Cubic Millimeter. Available at <http://robotics.eecs.berkeley.edu/~pister/SmartDust>.
- [Plosch and Pichler, 1999] Plosch, R., and Pichler, J. (1999). Contracts: From Analysis to C++ Implementation. *IEEE Computer*, pages 248–257.
- [Poor, 2001] Poor, R. D. (2001). Embedded Networks: Pervasive, Low-Power, Wireless Connectivity. Ph.D. Thesis, MIT.
- [Popovici et al., 2003] Popovici, A., Frei, A., and Alonso, G. (2003). A Proactive Middleware Platform for Mobile Computing. *Proceedings of the 4th International Middleware Conference (Middleware'2003)*, Rio de Janeiro, Brazil.
- [Poslad et al., 2000] Poslad, S., Buckle, P., and Hadingham, R. (2000). The FIPA-OS Agent Platform: Open Source for Open Standards. Available at <http://fipa-os.sourceforge.net>.
- [Poutakidis et al., 2002] Poutakidis, D., Padgham, L., and Winikoff, M. (2002). Debugging Multi-Agent Systems using Design Artifacts: The Case of Interaction Protocols. *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 960–967, ACM Press.
- [Poutakidis et al., 2003] Poutakidis, D., Padgham, L., and Winikoff, M. (2003). An Exploration of Bugs and Debugging in Multi-Agent Systems. *Proceedings of the 14th International Symposium on Methodologies for Intelligent Systems (ISMIS)*, Maebashi City, Japan.
- [Pratt et al., 1999] Pratt, D. R., Ragusa L. C., and von der Lippe, S. (1999). Composability as an Architecture Driver. *Proceedings of the International Conference on Interservice/Industry Training, Simulation and Education*, Orlando, Florida.
- [Pree, 1995] Pree, W. (1995). State-of-the-Art Design Pattern Approaches: An Overview. *Proceedings of Technology of Object-Oriented Languages and Systems (TOOLS '95)*.
- [Pressman, 1982] Pressman, R. S. (1982). *Software Engineering: A Practitioner's Approach*. McGraw-Hill.

- [Priyantha et al., 2000] Priyantha, N. B., Chakraborty, A., and Balakrishnan, H. (2000). The Cricket Location-Support System. *Proceedings of MobiCom 2000*.
- [Purvis et al., 2002] Purvis, M., Cranefield, S., Nowostawski, M., Ward, R., Carter, D., and Oliveira, M.A., (2002). Agentcities Interaction Using the Opal Platform. *Proceedings of Agentcities: Research in Large-Scale Open Agents Environments*.
- [Ralph and Shephard, 2001] Ralph, D., and Shephard, C. G. (2001). Services via Mobility Portals. *BT Technology Journal*, 19(1).
- [Rana and Moreau, 2000] Rana, O. F., and Moreau, L. (2000). Issues in Building Agent based Computational Grids. *Third Workshop of the UK Special Interest Group on Multi-Agent Systems (UKMAS'2000)*, Oxford, UK.
- [Rana and Walker, 2000] Rana, O. F., and Walker, D. W. (2000). 'The Agent Grid': Agent-Based Resource Integration in PSEs. *Proceedings of the 16th IMACS World Congress on Scientific Computation, Applied Mathematics and Simulation*, Lausanne, Switzerland.
- [Rao, 1996] Rao, A. S. (1996). AgentSpeak(L): BDI Agents Speak out in a Logical Computable Language. van der Velde, W., and Perram, J. (Eds.) *Proceedings of the 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-96)*, pages 42–55, Springer-Verlag.
- [Rao and Georgeff, 1991] Rao, A. S., and Georgeff, M. P. (1991). Modeling Rational Agents within a BDI-Architecture. Allen, J., Fikes, R., and Sandewall, E. (Eds.) *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pages 473–484, Morgan Kaufmann.
- [Ratsimor et al., 2002] Ratsimor, O., Chakraborty, D., Tolia, S., Kushraj, D., Kunjithapatham, A., Gupta, G., Joshi, A., and Finin, T. (2002). Allia: Alliance-based Service Discovery for Ad-Hoc Environments. *Proceedings of the 2nd ACM Workshop on Mobile Commerce*, Atlanta, USA.
- [Rich and Knight, 1990] Rich, E., and Knight, K. (1990). *Artificial Intelligence*. McGraw-Hill.
- [Ricci et al., 2002] Ricci, A., Omicini, A., and Denti, E. (2002). Virtual Enterprises and Workflow Management as Agent Coordination Issues. *International Journal of Cooperative Information Systems*, 11(3/4), pages 355–379.
- [Ricci et al., 2003] Ricci, A., Omicini, A., and Denti, E. (2003). Activity Theory as a Framework for MAS Coordination. Petta, P., Tolksdorf, R., and

- Zambonelli, F. (Eds.) *Engineering Societies in the Agents World III*, pages 96–110. Springer-Verlag.
- [Ricordel and Demazeau, 2000] Ricordel, P.-M., and Demazeau, Y. (2000). From Analysis to Deployment: A Multi-agent Platform Survey. *Working Notes of the First International Workshop on Engineering Societies in the Agents' World (ESAW-00)*, pages 93–105.
- [Rodríguez et al., 1998] Rodríguez, J. A., Martín, F. J., Noriega, P., García, P., and Sierra, C. (1998). Towards a Test-Bed for Trading Agents in Electronic Auction Markets. *AI Communications*, 11(1), pages 5–19.
- [Roli, 2002] Roli, F., and Zambonelli, A. (2002). Emergent Behaviors in Dissipative Cellular Automata. *Proceedings of the 5th International Conference on Cellular Automata for Research and Industry (ACRI 2002)*, Geneva, Italy.
- [Rosenschein and Kaelbling, 1995] Rosenschein, S. J., and Kaelbling, L. P. (1995). A Situated View of Representation and Control. *Artificial Intelligence*, 73(1–2), pages 149–173.
- [Royal Society of London, 2003] Royal Society of London (2003). Self-Organization: The Quest for the Origin and Evolution of Structure. *Philosophical Transactions of the Royal Society of London*.
- [Russell and Norvig, 1995] Russell, S., and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Prentice-Hall International.
- [Ruth and Hannon, 1997] Ruth, M., and Hannon, B. (1997). *Modelling Dynamic Economic Systems*. Springer-Verlag.
- [Sadri and Toni, 1999] Sadri, F., and Toni, F. (1999). Computational Logic and Multiagent Systems: A Roadmap. Technical report, DFKI.
- [Saeki, 1994] Saeki, M. (1994). Software Specification & Design Methods and Method Engineering. *International Journal of Software Engineering and Knowledge Engineering*.
- [Saha and Mukherjee, 2003] Saha, D., and Mukherjee, A. (2003). Pervasive Computing: A Paradigm for the 21st Century. *IEEE Computer*, 36(3).
- [Sandhu et al., 1996] Sandhu, R. S., Coyne, E. J., Feinstein, H. L., and Youman, C. E. (1996). Role-Based Access Control Models. *IEEE Computer*, 29(2), pages 38–47.
- [Sashima et al., 2002] Sashima, A., Kurumatani, K., and Izumi, N. (2002). Physically-Grounding Agents in Ubiquitous Computing. *Proceedings of Joint Agent Workshop (JAWS'2002)*, Connecticut, USA.

- [Satyanarayanan, 2001] Satyanarayanan, M. (2001). Pervasive Computing: Vision and Challenges. *IEEE Personal Communications*, 8(4).
- [Sauter et al., 2002] Sauter, J. A., Matthews, R., Parunak, H. V. D., and Brueckner, S. (2002). Evolving Adaptive Pheromone Path Planning Mechanisms. *Proceedings of Autonomous Agents and Multiagent Systems*, pages 434–440.
- [Schmidt and Beigl, 1998] Schmidt, A., and Beigl, M. (1998). New Challenges of Ubiquitous Computing and Augmented Reality. *Proceedings of the 5th CaberNet Radicals Workshop*, Porto, Portugal.
- [Schmidt and Simone, 1996] Schmidt, K., and Simone, C. (1996). Coordination Mechanisms: Towards a Conceptual Foundation of CSCW Systems Design. *Computer Supported Cooperative Work (CSCW)*, 5(2–3), pages 155–200.
- [Schmidt and Simone, 2000] Schmidt, K., and Simone, C. (2000). Mind the Gap! Towards a Unified View of CSCW. Dieng, R., Giboin, A., Karsenty, L., and de Michelis, G. (Eds.) *Designing Cooperative Systems: The Use of Theories and Models*, pages 205–221.
- [Schumacher, 2001] Schumacher, M. (2001). *Objective Coordination in Multi-Agent System Engineering – Design and Implementation*, Springer-Verlag.
- [Schweitzer and Zimmermann, 2001] Schweitzer, F., and Zimmermann, J. (2001). Communication and Self-Organization in Complex Systems: A Basic Approach. Fischer, M. M., and Fröhlich, J. (Eds.) *Knowledge, Complexity and Innovation Systems*, pages 275–296. Springer-Verlag.
- [Scott, 1998] Scott, W. R. (1998). *Organizations: Rational, Natural, and Open Systems*. Prentice-Hall International.
- [Searle, 1969] Searle, J. (1969). *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press.
- [Self and DeLoach, 2003] Self, A., and DeLoach, S. A. (2003). Designing and Specifying Mobility within the Multiagent Systems Engineering Methodology. *Proceedings of the 8th ACM Symposium on Applied Computing*.
- [Sen and Weiss, 1999] Sen, S., and Weiss, G. (1999). Learning in Multiagent Systems. Weiss, G. (Ed.) *Multiagent Systems*, pages 259–299, The MIT Press.

- [Servat and Drogoul, 2002] Servat, D., and Drogoul, A. (2002). Combining Amorphous Computing and Reactive Agent-Based Systems: A Paradigm for Pervasive Intelligence? *Proceedings of AAMAS'02*, ACM Press.
- [Shaikhali et al., 2003] Shaikhali, A., Rana, O. F., Al-Ali, R., and Walker, D. W. (2003). UDDIe: An Extended Registry for Web Services. *Proceedings of the Workshop on Service Oriented Computing: Models, Architectures and Applications*. IEEE Computer Society Press.
- [Shalizi, 2001] Shalizi, C. R. (2001). *Causal Architecture, Complexity and Self-Organization in Time Series and Cellular Automata*. Ph.D. Thesis, University of Wisconsin.
- [Shaw and Garlan, 1996] Shaw, M., and Garlan, D. (1996). *Software Architecture: Perspectives on an Emerging Discipline*. Prentice-Hall International.
- [Shehory and Sturm, 2001] Shehory, O., and Sturm, A. (2001). Evaluation of Modeling Techniques for Agent-Based Systems. *Proceedings of the 5th International Conference on Autonomous Agents*, pages 624–631. ACM Press.
- [Shen and Norrie, 1999] Shen, W., and Norrie, D. (1999). Agent-Based Systems for Intelligent Manufacturing: A State of the Art Survey. *International Journal on Knowledge and Information Systems*, 1(2), pages 129–156.
- [Shen et al., 2002] Shen, W. M., Salemi, M., and Will, P. (2002). Hormone-Inspired Adaptive Communication and Distributed Control for CONRO Self-Reconfigurable Robots. *IEEE Transactions on Robotics and Automation* 18(5), pages 1-12.
- [Shoham, 1991] Shoham, Y. (1991). Agent0: An Agent-Oriented Programming Language and its Interpreter. *Proceedings of AAAI-91*, pages 704–709.
- [Shoham, 1993] Shoham, Y. (1993). Agent-Oriented Programming. *Artificial Intelligence*, 60(1), pages 51–92.
- [Shoham and Tennenholtz, 1995] Shoham, Y., and Tennenholtz, M. (1995). On Social Laws for Artificial Agent Societies: Off-Line Design. *Artificial Intelligence*, 73(1–2), pages 231–252.
- [Sichman, 1998] Sichman, J. S. (1998). Depint: Dependence-Based Coalition Formation in an Open Multi-Agent Scenario. *Journal of Artificial Societies and Social Simulation*, 1(2).

- [Sichman and Demazeau, 2001] Sichman, J. S., and Demazeau, Y. (2001). On Social Reasoning in Multi-Agent Systems. *Revista Iberoamericana de Inteligencia Artificial*, 3(13), pages 68–84.
- [Sichman et al., 1994] Sichman, J. S., Conte, R., Demazeau, Y., and Castelfranchi, C. (1994). A Social Reasoning Mechanism based on Dependence Networks. *Proceedings of the European Conference on Cognitive Science*.
- [Sierra et al., 1997] Sierra, C., Faratin, P., and Jennings, N. R. (1997). A Service-Oriented Negotiation Model between Autonomous Agents. *Proceedings of MAAMAW'97*, pages 17–35.
- [Siewiorek, 2002] Siewiorek, D. P. (2002). New Frontiers of Application Design. *Communications of the ACM*, 45(12).
- [Singh, 1997] Singh, M. P. (1997). Formal Methods in DAI: Logic based Representation and Reasoning. *Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence*, pages 331–376.
- [Singh, 1998] Singh, M. P. (1998). Agent Communication Languages: Rethinking the Principles. *IEEE Computer*, 31(12), pages 40–47.
- [Singh, 1999a] Singh, M. P. (1999a). An ontology for Commitments in Multiagent Systems: Toward a Unification of Normative Concepts. *Artificial Intelligence and Law*, 7, pages 97–113.
- [Singh, 1999b] Singh, M. P. (1999b). Write Asynchronous, Run Synchronous. *IEEE Internet Computing*, 3(2), pages 4–5.
- [Singh, 2000] Singh, M. P. (2000). Synthesizing Coordination Requirements for Heterogeneous Autonomous Agents. *International Journal of Autonomous Agents and Multi-Agent Systems*, 3(2), pages 107–132.
- [Singh, 2002] Singh, M. P. (2002). The Pragmatic Web. *IEEE Internet Computing*, 6(3), pages 4–5.
- [Singh, 2003] Singh, M. P. (2003). Distributed Enactment of Multiagent Workflows: Temporal Logic for Service Composition. *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 907–914. ACM Press.
- [Sommerville, 2001] Sommerville, I. (2001). *Software Engineering*. Addison-Wesley.
- [Sousa and Garlan, 2002] Sousa, J. P., and Garlan, D. (2002). Aura: An Architectural Framework for User Mobility in Ubiquitous Computing Environments. *Proceedings of the 3rd Working IEEE/IFIP Conference on Software Architecture (WICSA'2002)*, Montreal, CA.

- [Sparkman et al., 2001] Sparkman, C. H., DeLoach, S. A., and Self, A. L. (2001). Automated Derivation of Complex Agent Architectures from Analysis Specifications. *Proceedings of the 2nd International Workshop on Agent-Oriented Software Engineering*, Montreal, CA.
- [Spivey, 1992] Spivey, J. M. (1992). *The Z Notation*. Prentice-Hall International.
- [Sreenath and Singh, 2003] Sreenath, R. M., and Singh, M. P. (2003). Agent-Based Service Selection. *Journal on Web Semantics*, 1(1).
- [Stafford, 2003] Stafford, T. S. (2003). E-Services. *Communications of the ACM*, 46(6), pages 26–34. ACM Press.
- [Stanford, 2003] Stanford, V. (2003). Pervasive Computing Puts Food on the Table. *IEEE Pervasive Computing Magazine*, 2(1).
- [Stohr and Zhao, 2001] Stohr, E. A., and Zhao, J. L. (2001). Workflow Automation: Overview and Research Issues. *Information Systems Frontiers*, 3(3), pages 281–296.
- [Stone and Veloso, 2000] Stone, P., and Veloso, M. (2000). Multiagent Systems: A Survey from a Machine Learning Perspective. *Autonomous Robots*, 8(3), pages 345–383.
- [Sturm and Shehory, 2003] Sturm, A., and Shehory, O. (2003). A Framework for Evaluating Agent-Oriented Methodologies. Giorgini, P., and Winikoff, M. (Eds.) *Proceedings of the 5th International Bi-Conference Workshop on Agent-Oriented Information Systems*, pages 60–67.
- [Sturm et al., 2003] Sturm, A., Dori, D., and Shehory, O. (2003). Single-Model Method for Specifying Multi-Agent Systems. *Proceeding of 2nd International Joint Conference on Autonomous Agents and Multi Agent Systems*.
- [Suhail, 1998] Suhail, A. (1998). *CORBA Programming Unleashed*. Sams.
- [Sycara and Klusch, 2001] Sycara, K., and Klusch, M. (2001). Brokering and Matchmaking for Coordination of Agent Societies: A Survey. *Coordination of Internet Agents*, Springer-Verlag.
- [Sycara et al., 1999] Sycara, K., Klusch, M., Idof, S., and Lu, J (1999). Dynamic Service Matchmaking among Agents in Open Information Environments. *Journal ACM SIGMOD Record*.
- [Sycara et al., 2002] Sycara, K., Widoff, S., Klusch, M., Lu, J. (2002). LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents

- in Cyberspace. *International Journal of Autonomous Agents and Multi-Agent Systems*, 5, pages 173–203.
- [Sycara et al., 2003] Sycara, K., Paolucci, M., van Velsen, M., and Giampapa, J. (2003). The RETSINA MAS Infrastructure. *International Journal on Autonomous Agents and Multi-Agent Systems*, 7(1–2), pages 29–48.
- [Szyperski, 1998] Szyperski, C. (1998). *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley.
- [Tambe et al., 2000] Tambe, M., Pynadath, D. V., Chauvat, N., Das, A., and Kaminka, G. A. (2000). Adaptive Agent Architectures for Heterogeneous Team Members. *Proceedings of the 4th International Conference on Multi-Agent Systems (ICMAS 2000)*, pages 301–308, Boston, USA.
- [Tennenhouse, 2000] Tennenhouse, D. (2000). Embedding the Internet: Proactive Computing. *Communications of the ACM*, 43(5), pages 36–42.
- [Theune et al., 2003] Theune, M., Faas, S., Heylen D., and Nijholt, A. (2003). The Virtual Storyteller: Story Creation by Intelligent Agents. *Proceedings of Technologies for Interactive Digital Storytelling and Entertainment*, pages 204–215.
- [Thomas, 1993] Thomas, S. R. (1993). *PLACA: An Agent Oriented Programming Language*. Ph.D. Thesis, Stanford University.
- [Thomas, 1995] Thomas, S. R. (1995). The PLACA Agent Programming Language. Wooldridge, M. J., and Jennings, N. R. (Eds.) *Intelligent Agents*, pages 355–370. Springer-Verlag.
- [Tolvanen, 1998] Tolvanen, J.-P. (1998). *Incremental Method Engineering with Modeling Tools: Theoretical Principles and Empirical Evidence*. Ph.D. Thesis, University of Jyväskylä.
- [Trastour et al., 2001] Trastour, D., Bartolini, C., and Gonzalez-Castillo, J. (2001). A Semantic Web Approach to Service Description for Matchmaking of Services. *Proceedings of the International Semantic Web Working Symposium (SWWS)*.
- [van de Vijver, 1997] van de Vijver, G. (1997). Emergence et Explication. *Intellectica: Emergence and Explanation*, 2-25.
- [van der Hoek, 2001] van der Hoek, W. (2001). Logical Foundations of Agent-Based Computing. Luck, M., Marík, V., Stepánková, O., and Trappl, R. (Eds.) *Multi-Agent Systems and Applications*, Springer-Verlag.

- [van Harmelen et al., 2002] van Harmelen, F., Horrocks, I., Clark, P., Patel-Schneider, P. F., Uschold, M., Rousset, M.-C., Hendler, J., and Schreiber, G. (2002). Ontologies' KISSES in Standardization. *IEEE Intelligent Systems*, 17, pages 70–79.
- [van Lamsweerde, 2000] van Lamsweerde, A. (2000). Requirements Engineering in the Year 00: A Research Perspective. *Proceedings of the 22nd International Conference on Software Engineering (ICSE'00)*, pages 5–19.
- [van Lamsweerde, 2001] van Lamsweerde, A. (2001). Goal-Oriented Requirements Engineering: A Guided Tour. *Proceedings of the 5th IEEE International Symposium on Requirements Engineering (RE'01)*, pages 249–263.
- [van Lamsweerde and Massonet, 1995] van Lamsweerde, R., Darimont, A., and Massonet, P. (1995). Goal-Directed Elaboration of Requirements for a Meeting Scheduler: Problems and Lessons Learnt. *Proceedings of the 5th IEEE International Symposium on Requirements Engineering (RE'01)*, pages 194–203.
- [van Roy and Haridi, 1999] van Roy, P., and Haridi, S. (1999). Mozart: A Programming System for Agent Applications. *Proceedings of the International Workshop on Distributed and Internet Programming with Logic and Constraint Languages*.
- [Varga et al., 1994] Varga, L. Z., Jennings, N. R., and Cockburn, D. (1994). Integrating Intelligent Systems into a Cooperating Community for Electricity Distribution Management. *International Journal of Expert Systems with Applications*, 7(4), pages 563–579.
- [Venkatraman and Singh, 1999] Venkatraman, M., and Singh, M. P. (1999). Verifying Compliance with Commitment Protocols: Enabling Open Web-Based Multiagent Systems. *International Journal on Autonomous Agents and Multi-Agent Systems*, 2(3), pages 217–236.
- [Viroli and Omicini, 2003] Viroli, M., and Omicini, A. (2003). Coordination as a Service: Ontological and Formal Foundation. Brogi, A., and Jacquet, J.-M. (Eds.) *Foundations of Coordination Languages and Software Architecture*, Elsevier Science.
- [Visser et al., 2000] Visser, W., Park, S., and Penix, J. (2000). Using Predicate Abstraction to Reduce Object-Oriented Programs for Model Checking. *Proceedings of the 3rd Workshop on Formal Methods in Software Practice*, pages 3–182. ACM Press.

- [von Bertalanffy, 1968] von Bertalanffy, L. (1968). *General System Theory*. George Braziller.
- [von Martial, 1992] von Martial, F. (1992). *Co-ordinating Plans of Autonomous Agents*, Springer-Verlag.
- [Vouk and Singh, 1997] Vouk, M. A., and Singh, M. P. (1997). Quality of Service and Scientific Workflows. Boisvert, R. F. (Ed.) *Quality of Numerical Software: Assessment and Enhancements*, pages 77–89.
- [Vygotskij, 1978] Vygotskij, L. S. (1978). *Mind and Society*. Harvard University Press.
- [Walshe et al., 2000] Walshe, D., Kennedy, J., Corley, S., Koudouridis, G., Laenen, F. V., Ouzounis, V., Garijo, F., and Gomez-Sanz, J. (2000). Eurescom P815: An Interoperable Architecture for Agent-Oriented Management. Arabnia, H. R. (Ed.) *Proceedings of IC-AI'2000*, CSREA Press.
- [Wan and Singh, 2003] Wan, F., and Singh, M. P. (2003). Commitments and Causality for Multiagent Design. *Proceedings of the 2nd International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 749–756, ACM Press.
- [Wagner, 2002] Wagner, G. (2002). A UML Profile for External AOR Models. *Proceedings of the 3rd International Workshop on Agent-Oriented Software Engineering*.
- [Wegner, 1997] Wegner, P. (1997). Why Interaction is More Powerful than Computing. *Communications of the ACM*, 40(5), pages 80–91.
- [Weiser, 1991] Weiser, M. (1991). The Computer for the Twenty-First Century. *Scientific American*.
- [Weiser, 1993] Weiser, M. (1993). Some Computer Science Problems in Ubiquitous Computing. *Communications of the ACM*, 36(7).
- [Weiser, 1996] Weiser, M., and Brown, J. S. (1996). Designing Calm Technology. *PowerGrid Journal*, 1-1.
- [Weiss, 1999] Weiss, G. (Ed.) (1999). *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*. The MIT Press.
- [Weiss, 2001] Weiss, R. (2001). *Cellular Computation and Communications Using Engineered Genetic Regulatory Networks*. Ph.D. Thesis, MIT.
- [Weiss, 2003] Weiss, G. (2003). Agent Orientation in Software Engineering. *Knowledge Engineering Review*, 16(4), pages 349–373.

- [Weiss et al., 2003] Weiß, G., Rovatsos, M., and Nickles, M. (2003). Capturing Agent Autonomy in Roles and XML. *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 105–112, ACM Press.
- [Whitaker, 2003] Whitaker, R. (2003). ACM SIGGROUP: Self-Organization, Autopoiesis, and Enterprises. Available at <http://www.acm.org/sigois/auto/Main.html>.
- [Wieland, 2003] Wieland, K. (2003). The Long Road to 3G. *International Telecommunications Magazine*, 37(2).
- [Winikoff et al., 2001] Winikoff, M., Padgham, L., and Harland, J. (2001). Simplifying the Development of Intelligent Agents. *Proceedings of AI2001*, pages 555–568. Springer-Verlag.
- [Wolpert, 1998] Wolpert, D. H. (1998). *Principles of Development*. Oxford University Press, UK.
- [Wolpert and MacReady 1995] Wolpert, D. H., and MacReady, W. G. (1995). No Free Lunch Theorems for Search. Technical Report SFI-TR-95-02-010, Santa Fe Institute.
- [Wood and DeLoach, 2001] Wood, M., and DeLoach, S. A. (2001). An Overview of the Multiagent Systems Engineering Methodology. Ciancarini, P., and Wooldridge, M. J. (Eds.) *Agent-oriented software engineering*, pages 207–222, Springer-Verlag.
- [Wooldridge, 1992] Wooldridge, M. J. (1992). *The Logical Modelling of Computational Multi-Agent Systems*. Ph.D. Thesis, UMIST.
- [Wooldridge, 1997] Wooldridge, M. J. (1997). Agent-Based Software Engineering. *IEE Proceedings Software Engineering*, 144(1), pages 26–37.
- [Wooldridge, 2000] Wooldridge, M. J. (2000). Intelligent Agents. Weiss, G. (Ed) *Multiagent Systems*, pages 27–78, The MIT Press.
- [Wooldridge, 2002] Wooldridge, M. J. (2002). *An Introduction to Multi-Agent Systems*. John Wiley & Sons.
- [Wooldridge and Ciancarini, 2000] Wooldridge, M. J., and Ciancarini, P. (2000). Agent-Oriented Software Engineering: The State of the Art. Ciancarini, P., and Wooldridge, M. J. (Eds.) *Proceedings of the 1st Workshop on Agent-Oriented Software Engineering*, pages 1–28, Springer-Verlag.
- [Wooldridge and Jennings, 1995a] Wooldridge, M. J., and Jennings, N. R. (1995a). Intelligent Agents: Theory and Practice. *Knowledge Engineering Review*, 10(2), pages 115–152.

- [Wooldridge and Jennings, 1995b] Wooldridge, M. J., and Jennings, N. R. (1995b). Agent Theories, Architectures, and Languages: A Survey. Wooldridge, M. J., and Jennings, N. R. (Eds.) *Intelligent Agents*, pages 1–39, Springer-Verlag.
- [Wooldridge and Jennings, 1998] Wooldridge, M. J., and Jennings, N. R. (1998). Pitfalls of Agent-Oriented Development. Sycara, K., and Wooldridge, M. J. (Eds.) *Proceedings of the 2nd International Conference on Autonomous Agents*, ACM Press.
- [Wooldridge et al., 1999] Wooldridge, M. J., Jennings, N. R., and Kinny, D. (1999). A Methodology for Agent-Oriented Analysis and Design. Etzioni, O., Müller, J. P., and Bradshaw, J. M. (Eds.) *Proceedings of the 3rd International Conference on Autonomous Agents*, pages 69–76, ACM Press.
- [Wooldridge et al., 2002a] Wooldridge, M. J., Fisher, M., Huget, M.-P., and Parsons, S. (2002a). Model Checking Multi-Agent Systems with MABLE. *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 952–959, ACM Press.
- [Wooldridge et al., 2000b] Wooldridge, M. J., Jennings, N. R., and Kinny, D. (2000b). The Gaia Methodology for Agent-Oriented Analysis and Design. *International Journal of Autonomous Agents and Multi Agent Systems*, 3(3), pages 285–312.
- [Wooldridge et al., 2002c] Wooldridge, M. J., Weis, G., and Ciancarini, P. (Eds.) (2002c). *Agent-Oriented Software Engineering II*, Springer-Verlag.
- [Xing and Singh, 2003] Xing, J., and Singh, M. P. (2003). Engineering Commitment-Based Multiagent Systems: A Temporal Logic Approach. *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 891–898, ACM Press.
- [Xu et al., 2002] Xu, D., Volz, R., Ioerger, T., and Yen, J. (2002). Modeling and Verifying Multi-agent Behaviors using Predicate/Transition Nets. *Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering*, pages 193–200, ACM Press.
- [Yolum and Singh, 2002] Yolum, P., and Singh, M. P. (2002). Flexible Protocol Specification and Execution: Applying Event Calculus Planning using Commitments. *Proceedings of the 1st International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 527–534, ACM Press.
- [Yu, 1995] Yu, E. (1995). *Modelling Strategic Relationships for Process Reengineering*. Ph.D. Thesis, University of Toronto.

- [Yu, 1997a] Yu, E. (1997a). Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. *Proceedings of 3rd International Symposium on Requirements Engineering*, pages 226–235.
- [Yu, 1997b] Yu, E. (1997b). Why Agent-Oriented Requirements Engineering? *Proceedings of 3rd International Workshop on Requirements Engineering*.
- [Yu, 1999] Yu, E. (1999). Strategic Modelling for Enterprise Integration. *Proceedings 14th World Congress of the International Federation of Automatic Control*.
- [Yu, 2001] Yu, E. (2001). Agent-Oriented Modeling: Software versus the World. *Agent-Oriented Software Engineering II*, Springer-Verlag.
- [Yu and Cysneiros, 2002] Yu, E., and Cysneiros, L. M. (2002). Agent-Oriented Methodologies – Towards a Challenge Exemplar. *Proceedings of the 4th International Bi-Conference Workshop on Agent-Oriented Information Systems*.
- [Yu and Liu, 2002] Yu, E., and Liu, L. (2002). Designing Web-Based Systems in Social Context: A Goal and Scenario based Approach. *Advanced Information Systems Engineering*, pages 37–51, Springer-Verlag.
- [Yu and Liu, 2003] Yu, E., and Liu, L. (2003). Organization Modelling Environment. Available at <http://www.cs.toronto.edu/km/ome>.
- [Yu and Mylopoulos, 1998] Yu, E., and Mylopoulos, J. (1998). Why Goal-Oriented Requirements Engineering? *Proceedings of the 4th International Workshop on Requirements Engineering*, pages 15–22.
- [Yu and Singh, 2002] Yu, B., and Singh, M. P. (2002). An Evidential Model of Distributed Reputation Management. *Proceedings of the 1st International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 294–301. ACM Press.
- [Yu et al., 2001] Yu, E., Liu, L., and Li, Y. (2001). Modelling Strategic Actor Relationships to Support Intellectual Property Management. *Conceptual Modeling*, pages 164–178, Springer-Verlag.
- [Yunos et al., 2003] Yunos, H. M., Gao, J. Z., and Shim, S. (2003). Wireless Advertising’s Challenges and Opportunities. *IEEE Computer*, 26(5).
- [Zambonelli and Parunak, 2002] Zambonelli, F., and Parunak, H. V. D. (2002). Sign of a Revolution in Computer Science and Software Engineering. *Proceedings of the 3rd International Workshop on Engineering Societies in the Agents’ World*, Springer-Verlag.

- [Zambonelli and Parunak, 2003] Zambonelli, F., and Parunak, H. V. D. (2003). Toward a Change of Paradigm in Computer Science and Software Engineering: A Synthesis. *The Knowledge Engineering Review*, 18, 2003.
- [Zambonelli et al., 2000] Zambonelli, F., Jennings, N. R., and Wooldridge, M. J. (2000). Organisational Abstractions for the Analysis and Design of Multi-Agent Systems. Ciancarini, P., and Wooldridge, M. J. (Eds.) *Agent-Oriented Software Engineering*, pages 127–141, Springer-Verlag.
- [Zambonelli et al., 2001a] Zambonelli, F., Jennings, N. R., Omicini, A., and Wooldridge, M. J. (2001a). Agent-Oriented Software Engineering for Internet Applications. *Coordination of Internet Agents: Models, Technologies, and Applications*, pages 326–346, Springer-Verlag.
- [Zambonelli et al., 2001b] Zambonelli, F., Jennings, N. R., and Wooldridge, M. J. (2001). Organizational Rules as an Abstraction for the Analysis and Design of Multiagent Systems. *International Journal of Software Engineering and Knowledge Engineering*, 11(4), pages 303–328.
- [Zambonelli et al., 2003] Zambonelli, F., Jennings, N. R., and Wooldridge, M. J. (2003). Developing Multiagent systems: The Gaia methodology. *ACM Transactions on Software Engineering and Methodology*, 12(3), pages 417–470.
- [Zave, 1997] Zave, P. (1997). Classification of Research Efforts in Requirements Engineering. *ACM Computing Surveys*, 29(4), pages 315–321.

Index

3APL, 57

AAII, 174

ABLE framework, 54

ACL (Agent Communication Language), 22, 47, 260, 417

Activity Theory, 283, 284

ADELFE, 40, 53, 154, 158, 232, 338

agent, 9, 20, 21, 36, 48, 90, 130, 179, 191, 248, 260, 305, 396, 398, 407, 416, 433, 434

agent

agent architecture, 45, 445

agent class, 117

agent lifeline, 240

agent model, 35

agent-based model, 197

cooperative agent, 159, 164, 167

types of agents, 408

agent class diagram, 248, 254

agent platform, 53

agent technology, 431, 432, 434

agent-based software development, 13

Agent0, 56

AgentBuilder, 42

AgentSpeak (L), 56

agentTool, 38, 43, 60, 122, 172

AIP (Agent Interaction Protocol), 166

AIP (Agent Interaction Protocol), 85

Albert (language), 37

AMAS (Adaptive MultiAgent Systems), 157, 158, 327

amorphous computing, 300, 304, 361

analysis, 34, 35, 41, 70, 71, 76, 108, 163, 181

AOSE (Agent Oriented Software Engineering), 19, 394

AOSE (Agent-Oriented Software Engineering), 3

April language, 55

ASL, 262

AUML, 40, 84, 166, 172, 173, 178, 237

autocatalysis, 358, 360

Autonomic Computing, 7

autonomy, 10, 14, 49, 129, 267, 398

Bee-gent, 262

CLIPS language, 55

co-construction, 285

commitment, 13, 15

communication, 12, 15, 22, 47, 260, 417

complex system, 322, 433

component, 21

Concurrent-METATEM, 39, 55

ConGOLOG, 39, 56

conversation, 117, 119

cooperation, 284, 285, 328

coordination, 48, 273, 274, 286

coordination artifact, 284, 291

Data Coupling Diagram, 222

debugging, 58, 60, 229, 270, 439

decentralization, 351

delegation of responsibility, 23

dependency, 91, 274

deployment, 446

design, 34, 41, 71, 74, 77, 108, 117, 121, 165, 441

design

architectural design, 90, 93, 218, 222

architecture design, 423

detailed design, 90, 94, 218, 226

high-level design, 181

low-level design, 182

DESIRE, 44, 173

development environment, 42

Dia, 255

dialogic framework, 204

DISCERN, 386

distribution, 46, 351, 433, 441

dynamic, 11, 14, 352, 380

Electronic Institution Model, 197

emergence, 196, 308, 326, 347

emergence

emergent behaviour, 195

emergent functionality, 158

emergent phenomenon, 325

- environment, 12, 24, 75, 76, 87, 161, 162, 220, 279, 357, 382, 433
- Equation-Based Models, 196
- evaluation (of a methodology), 67, 129, 134, 138, 143
- evolutionary computation, 196, 210

- FIPA (Foudation for Intelligent Physical Agents), 260
- FIPA (Foundation for Intelligent Physical Agents), 20, 54, 239, 437
- FIPA-OS, 263
- Formal Tropos, 94
- functional adequacy, 164, 328, 334

- Gaia, 39, 52, 66, 69, 124, 134, 173, 178, 220, 231
- goal, 58, 76, 108, 110, 179, 220
- goal analysis, 102
- Grasshopper, 263
- grid, 394, 413, 414, 447
- Growing Point Language, 310, 314

- heterogeneity, 8, 11, 14, 441

- i*, 36, 58, 90
- implementation, 34, 55
- implementation tool, 444
- infrastructure, 273, 278
- INGENIAS, 41, 43, 172, 194
- interaction, 24, 179, 180, 187, 275, 281, 382
- interaction
 - interaction diagram, 225
 - interaction model, 74, 77
 - thread of interaction, 241

- JACK, 229
- JADE, 44, 48, 54, 61, 264

- KAOS, 37

- language, 55
- lifecycle, 66, 127, 132, 196
- Lisp, 56

- MABLE, 60
- MADKIT platform, 41, 52, 54, 61
- MAML, 57
- MAS-CommonKADS, 38, 178
- MaSE, 37, 66, 107, 125, 143, 172, 232
- MASIF, 54
- MASSIVE, 173
- mediation, 382
- MESSAGE, 41, 52, 154, 173, 177, 232
- message, 130, 243, 267
- meta-model, 41, 81
- methodology, 65, 127, 153, 435, 436

- Microsoft Visio, 255
- mobile computing, 399
- modelling language, 66, 127
- morphogen gradients, 306
- Mozart language, 56
- multiagent system, 107, 197, 378, 397, 435, 441

- NCS (Non Cooperative Situations), 159, 167, 170, 328, 330, 335
- norm, 130, 197, 204, 205
- notation, 85, 87, 131, 224, 238

- observation, 382
- online engineering, 378, 383
- ontology, 47, 198
- Opal, 263
- open architecture, 5, 6
- open system, 8, 301, 379, 380, 382, 441
- open system
 - a model of open computational system, 381
- OpenTool, 172, 173, 255
- organization, 52, 69, 130, 179, 181, 192, 252, 292, 344
- Organization Theory, 98
- organization view, 180
- organizational rules, 76, 77
- organizational structures, 76, 78, 87

- PASSI, 40, 43, 172, 173
- performative structure, 204
- pervasive computing, 315, 399
- pheromone, 310
- Prolog, 56
- Prometheus, 155, 173, 217
- Prometheus Design Tool, 228
- protocol, 13, 15, 80, 130, 190, 197, 226, 245

- requirement, 35, 181, 220
- requirement
 - early requirement, 90, 91
 - final requirement, 161
 - late requirement, 90, 92
 - preliminary requirement, 161
- RETSINA framework, 54
- ROADMAP, 79
- role, 36, 71, 76, 108, 109, 113, 115, 130, 179, 198, 240, 249
- RUP (Rational Unified Process), 161, 178, 181

- SADDE, 53, 154, 196, 232
- scene, 197, 198, 204
- self-assembly, 311
- self-organization, 52, 158, 300, 325, 331, 344–346, 348
- semantic composability, 27
- semantic extensibility, 28
- semantic interoperability, 25

- sequence diagrams, 112, 239, 253
- service, 74, 421, 422
- service discovery, 419, 420
- service-oriented approach, 415
- SOLACE, 385
- specification, 218, 220
- stereotype, 167
- swarming, 300, 341–343, 353

- task, 108, 114, 115, 130, 179, 182, 291
- testing, 34, 57, 229, 440
- TOTA, 316
- Tropos, 58, 66, 90, 125, 138, 173, 220, 232
- TuCSon, 293

- ubiquitous computing, 394, 395, 400, 401
- UDDI (Universal Description Discovery and Integration), 14, 404, 415, 420, 422
- UML, 40, 84, 173, 177, 178, 237
- use cases, 112, 162, 221

- validation, 57, 383, 384, 440
- verification, 57, 58, 94, 440
- VIRTUE, 385

- Web services, 6, 420

- Z, 38
- Zeus, 37, 42, 60, 263