Engineering Trust in Complex System through Mediating Infrastructures

Alessandro Ricci
DEIS
Università di Bologna – Sede di Cesena
via Venezia 52, 47023 Cesena (FC), Italy
Email: aricci@deis.unibo.it

Andrea Omicini
DEIS
Università di Bologna – Sede di Cesena
via Venezia 52, 47023 Cesena (FC), Italy
Email: aomicini@deis.unibo.it

***Abstract*— Starting from the many research results on trust in state-of-the-art literature, we first point out some open problems related to trust in multiagent systems (MAS), focussing in particular on the issue of the engineering of agent societies, and on the role of agent infrastructures. Then, we discuss two infrastructural abstractions – coordination artifacts, and agent coordination contexts –, and show how they can be exploited for modelling and engineering trust within MAS.**

I. Trust in Complex System Engineering

One of the most relevant problems of our contemporary society is its dependency on information technologies systems which are getting more and more complex and difficult to control. Accordingly, the problem of *trust* between humans and information technology comes out from the inability to provide simple and accessible models to make systems behaviour somehow understandable and predictable for the users themselves. This does not affect only end-users, but also (and, in some sense, mostly) the engineers and developers that are responsible of system design and construction. In particular, the difficulty of conceiving trustworthy models for the engineering of complex and complex systems emphasises the fact that the impetuous technological progress chararcterising our society is a necessary but not sufficient condition for the widespread generation and adoption of innovative processes.

As a simple example, the possibility of checking system behaviour and functioning by inspecting its source code once it is made available (the myth of Open Source wave) is simply not feasible, according to current state-of-the-art models and tools. Turning from the notion of ”program” to the notion of ”system” involves a paradigm shift: the behaviour of a program (as a sequence of instructions of certain (virtual) machines) is, in principle, inspectable, understandable and predictable. Instead, it is typically not possible to formalise nor to have a complete understanding of the behaviour of a software system (as a collection of heterogeneous and independent components interacting in a distributed environment) [23].

According to the current major research lines, the complexity of modern and forthcoming systems can be managed only by introducing models that account for *societies* of heterogeneous actors (objects, components, agents, processes, humans..) which interact and communicate in dynamic and unpredictable environments: at least, this is a suitable model for current web-based systems. So, trust is one of the most important social issues for human as well as for artificial systems: this is evident if we consider scenarios such as e-commerce or e-government, where the edge between human and artificial societies tends to blur: these contexts make it clear that all the social issues involved in human societies, trust in primis, must be faced also in the construction of complex artificial systems.

Accordingly, the applicability (reuse) of models for human societies in the context of artificial systems is a primary issue, exploiting, for instance, the explaination and prediction capabilities of theories both as a scientific and engineering tool to validate engineering constraints of systems [22]. This is especially important if we aim at considering trust beyond conceptually simple applications such as digital signature or e-commerce transactions, facing contexts where trust matters not only for a human actor (users or engineers) w.r.t. the system, but for every human and artificial actor that constitute system society.

Trust is then recognised as a fundamental aspect of engineering systems with MAS: however, trust characterisation and models as found in state-of-the-art literature do not cover some issues which we consider fundamental for the engineering of agent societies. First, a well-defined notion of social trust is missing: few approaches deal with an infrastructure (and then social, objective) support to trust, being mostly focussed on the subjective perception and modelling of trust by individuals. Even the approaches considering forms of social trust (referred as system-level trust in literature) fail to provide a comprehensive model of the trust phenomenon at the social level (including the notion of observation, traceability of actions, etc), limiting their approach to provide some specific mechanisms. Then, trust frameworks (models and mechanisms) are focussed essentially on the behaviour of a individual component (agent), and no account is given for characterising trust at a systemic level, i.e. trust in a group or society of agents in being able to achieve their social tasks. Linked to this point, current models and mechanisms are developed mostly in competitive contexts, where agents are totally self-interested; instead we are interested in modelling trust in systems where agent cooperatively work for a global (system) outcome. In this case we have several points of view concerning trust: trust

M. Baldoni, F. De Paoli, A. Martelli, and A. Omicini (Eds.): *WOA 2004 – Dagli Oggetti agli Agenti, Sistemi Complessi e Agenti Razionali*, Torino, Italy, November 29th – December 1st 2004. Pitagora Editrice Bologna. ISBN 88-371-1533-4. http://woa04.unito.it

of the users relying on a systems of cooperating agents, trust of the engineers in the system he designed, trust of the collectivity of the components (agents) with respect to a specific one, trust of an individual components (agent) of the system with respect to the collectivity.

In this paper, then, first we extend the notion of trust to consider also these issues, more related to an engineering point of view on systems and based on infrastructural support to trust. The extension will relate trust to coordination and organisation issues, as fundamental engineering dimensions of systems. Then, we show how some infrastructural abstractions recently introduced for engineering of MAS coordination and organisation – namely coordination artifacts and agent coordination contexts – can play an effective role in defining trust according to our wider vision.

The remainder of the paper is organised as follows: first, in Section II a brief account of state-of-the-art models for trust in MAS is provided; then, Section III remarks some points missing from such models, discussing a wider characterisation of trust including engineering issues. Accordingly, Section IV and Section V discuss how coordination and organisation infrastructural abstraction can play a fundamental role for characterising this enhanced notion of trust. Finally, conclusions are reported in Section VI.

II. Modelling Trust in Agent Societies

Trust has been defined in several ways in distinct domains (see [15] for a comprehensive survey, and [4] for a general description). A definition that is frequently adopted in trust models is:

> "Trust is a belief an agent has that the other party will do what it says it will (being honest and reliable) or reciprocate (being reciprocative for the common good of both), given an opportunity to defect to get higher payoff"
> (adapted from [1])

The various approaches to trust in MAS have been recently classified in two main classes, for some extend in counterposition and complimentary: *individual-level trust* and *system-level trust* [15].

Roughly speaking, in individual-level trust all the burden about trust is in charge of individual agents, and depends on their ability to model and reason about the reciprocative nature, reliability or honesty of their counter-parts. In system-level trust instead the actors in the systems are forced to be trustworthy by the *rules of the encounter* [18] (i.e. protocols, mechanisms) that regulate the systems. So the burden about trust is shifted from agents to some system support, which is realised by designing specific protocols and mechanisms of interaction (i.e. the rules of the encounter). A typical example are auctions.

So the point of view of individual-level trust accounts for an agent situated in an open environment trying to choose the most reliable interaction partner from a pool of potential agents and deliberating which strategy to adopt on it. Following the classification described in [15], trust models for individual-level can be classified in this case either *learning (evolution) based*, *reputation based* or *socio-cognitive based*. In the first, trust is viewed as an emergent property of direct interaction between self-interested agents, who are endowed with strategies that can cope with lying and non-reciprocative agents. Reputation models [19] instead enable agents to gather information in richer forms from their environment and make rational inferences from the information obtained and their counterparts. The models then specify strategies to *gather ratings* that define the trustworthiness of an agent, using relationships existing between member of the community; reasoning methods to gather information from *aggregation of ratings* retrieved from the community (borrowing the concept of social network from sociology); and mechanisms to promote ratings that *truly* describe the trustworthiness of an agent. Finally, the socio-cognitive models adopt a higher level view, modelling the subjective perception of trust in terms of cognitive constructs [3], in contrast to the quantitative view of trust which characterises previous approaches. While the first two models are all based on an assessment of the outcome of interactions between agents, the basic context for socio-cognitive approaches is that of *task delegation* where an agent x wishes to delegate a task to agent y. In doing so agent x needs to evaluate the trust it can place in y by considering the different *beliefs* it has about motivations of agent y.

In the overall, trust at the individual level concerns strategies learnt over multiple interactions, the reputation of potential interaction partners, and believed motivations and abilities regarding the interaction. Some problems affecting these approaches have been remarked in the literature: it can be computationally expensive for an agent to reason about all the different factors affecting trust in its opponents; then, agents are limited in gathering information from various sources that populate its (open) environment. Given these limitations, system-level trust approaches shift the focus on the rules of encounter so as to ensure that interaction partners are *forced* to be trustworthy. The mechanisms that dictate these rules of encounter (auctions, voting, contract-nets, market mechanisms, etc) enable agent to trust each other by virtue of the different constraints *imposed by the system*. Always following [15], these system-level mechanisms can be classified in *trustworthy interaction mechanisms*, *reputation mechanisms* and *distributed security mechanisms*. Mechanisms of the first class are adopted to prevent agents from lying or speculating while interacting (auctions are an example, see [20] for an overview); reputation mechanisms [24] make it possible to model the reputation of agents at system level, i.e. it is the system that manage the aggregation and retrieval of ratings (as opposed to reputation models which leave the task to the agents themselves). Finally, the latter class includes security mechanisms and infrastructures which are considered essential for agents to trust each other and each other communication (examples are public key encryption and certificate infrastructures)[14], [5].

According to [15], complex systems require both types of trust approach, individual- and system-level. While the individual-level trust *models* enable agent to reason about its level of trust and of its opponents, the system-level *mechanisms* aim to ensure that opponents' actions can be trusted. It's worth noting that this dichotomy have been remarked also for another dimension focussing on interaction, i.e. coordination, where approaches have been classified as subjective (all the coordination burden on agent and their capabilities) and objective (the coordination burden in charge of abstractions provided by suitable infrastructures)[10].

III. Extending Trust for Engineering Societies

The characterisation of trust in state-of-the-art literature as described in previous section do not give emphasis enough to some issues that we consider as fundamental in the engineering of agent societies. These aspects can be summed up in the following points:

- *Social Trust* – We need to consider in a more general and systematic way the support that infrastructures can provide as a service for societies engineered on their top, beyond specific mechanisms or protocols. This accounts for generalising system-level trust approaches, devising basic abstractions and services on top of which to build trust strategies. Among such basic services, a support for *observation* and *traceability* of both agent action, and interaction among agents and agent-environment. These basic services can be suitably exploited and composed to keep track – for instance – of action history of a specific agent, making it available to some other agents, with the permissions to inspect such information. This infrastructural support is extremely effective when dealing with *open* systems, with heterogeneous agents dynamically participating to the activities of different societies and organisations: infrastructures can provide services to agent organisations to keep track and make information available about agent performance in its interaction life, acting in different contexts, as a kind of "criminal record" publicly available; thus, respecting privacy of the agent, i.e., making available only what is needed to be observed according the type of activities the agent is going to participate.
- *Trust in Societies* – individual-level and system-level approaches share a focus on (trust on) the behaviour of an individual agent. However, in the engineering of complex systems it emerges the need of modelling the notion of trust also related to *groups* or *societies* of agents, delegated of the execution of some social task. More generally we are looking to a systemic acceptation of trust: how much a system (as a structured society of agents) can be trusted in its behaviour, in its ability to achieve the global objectives as outcomes of the cooperative work of its agents? So we are interested in characterising trust also in cooperative scenarios, not only in competitive ones as it typically happens in the literature.
- *Constructive Trust* – As in the system-level (objective) case, we are interested in infrastructural abstractions (services) for creating and managing trust. However, differently from system-level approaches, we characterise these abstractions not only as *barriers*, basically creating trust by enforcing *norms* constraining agent actions and interactions. We are interested also in framing trust from a *constructive* point of view: I can have trust in an system because of the availability of services which provide some (objective) guarantees that not only certain interaction cannot happen, but also that some social tasks can be effectively executed, specifying for instance the workflow or plan useful for achieving the global objective. Considering system-level approaches, it is like modelling trust on the rules of encounters which make it possible to achieve some social goal.
- *Trust and Organisation* – As mentioned in the context of system-level trust, security support has a certain impact on trust in a system [14]. However, when engineering complex systems, some important aspects concerning security – such as access control – cannot be dealt without considering the organisation and coordination model adopted [11]. As an important example, Role-Based Access Control (RBAC) models and architectures – well-known in the research literature concening security in complex information systems, and recently introduced also in MAS [21] – make it possible to model security (access control) policies in the context of role-based organisational models. Accordingly, the presence of such an organisational model can have a significant impact on trust models and services, which can be characterised also considering the notions of roles and related organisational policies.

In the overall, the social and engineering acceptation of trust that emerges from the above points aims to be wider than the one usually found in the literature, and can be framed in the idea of agent societies used as metaphors to model trust in information technology in the most general way. This includes both trust between humans and systems – i.e. trust between users and systems and trust between designers/engineers and systems – and trust between systems and systems – i.e. trust among system components and trust among components of different systems. The interpretation of systems in terms of societies, promoted by MAS approaches, makes it possible to face these issues within the same conceptual framework, adopting a uniform approach to explore general models and solutions, relevant in computer science as well as in the other related fields.

A possible way to bring to practice such generalised acceptation of trust is to relate them to the coordination and organisation dimensions (and the related models) which characterise the engineering of agent societies. In next sections we follow this line, by presenting two infrastructural abstractions which we have recently introduced in MAS engineering, namely *coordination artifacts* and *agent coordination contexts*, and discussing their role in modelling and engineering such a notion of trust in MAS.

IV. Artifacts for Trust

From the research studies carried on in human (cooperative) activity – mainly with Activity Theory [2], [6] – it clearly emerges the fundamental role of tools or *artifacts* in the development of (social) activities in complex systems framed as societies [7], [16]. According to these studies, every non trivial human activities is *mediated* by some kind of artifacts. An artifact acts as the glue among two or multiple actors, as the tool that enables and mediates their interaction, ruling / governing the resulting global and ”social” behaviour; consequently, an artifact can be considered *the* conceptual place encapsulating all the complexity of the social behaviour that it enables, allowing its factorisation, explicit modeling and engineering, and so freeing the actors of all this *social* burden [16]. Artifacts are widespread in human society: the language can be considered an artifact, as well as the writing, blackboards, maps, post-its, traffic signs such as semaphores, electoral cards or the signature on a document.

Based on this background, recently *coordination artifacts* have been introduced as a conceptual and engineering framework for MAS and agent societies [16], [12]. So the idea here is that coordination artifacts can play a primary role for engineering trust in MAS, providing an answer to the points remarked in Section III.

A. Coordination Artifact Model and Framework

Coordination artifacts have been defined as embodied[1] entities specialised to provide a coordination service in a MAS [12]. As *infrastructure* abstractions, they are meant to improve coordination activities automation; they can be considered then as basic building blocks for creating effective shared collaborative working environments, alleviating the coordination burden for the involved agents.

As remarked for artifacts in general, human society is full of entities like coordination artifacts, engineered by humans in order to support and automate coordination activities: well-known examples are street semaphores, blackboards, queuing tools at the super-markets, maps, synchronisers and so on.

Basically, a coordination artifact *(i)* entails a form of mediation among the agents using it, and *(ii)* embeds and enact effectively some coordination policy. Accordingly, two basic aims can be identified: *(i) constructive*, as an abstraction essential for creating/composing social activities, *(ii) normative*, as an abstraction essential for ruling social activities.

From a constitutive point of view, a coordination artifact is characterised by:

- a *usage interface*, defined in terms of a set of *operations* which agents can execute in order to use the artifacts.
- a set of *operating instructions*, which formally describe how to use the artifact in order to exploit its coordination service.
- a *coordinating behaviour*, which formally describe the coordination enacted by the artifact.

Then, taking the agent viewpoint, to exploit a coordination artifact simply means to follow its operating instructions, on a step-by-step basis.

Among the main properties which exhibit coordination artifacts (and which differentiate them from the agent abstraction) we have:

- *Specialisation* – Coordination artifacts are specialised in automating coordination activities. For this purpose, they typically adopt a computational model suitable for effective and efficient interaction management, whose semantics can be easily expressed with concurrency frameworks such as process algebras, Petri nets, or Event-Condition-Reaction rules.
- *Encapsulation: Abstraction and Reuse* – Coordination artifacts encapsulate a coordination service, allowing user agents to abstract from how the service is implemented. As such, a coordination artifact is perceived as an individual entity, but actually it can be distributed on several nodes of the MAS infrastructure, depending on its specific model and implementation.
- *Malleability* – Coordination artifacts are meant to support coordination in open agent systems, characterised by unpredictable events and dynamism. For this purpsose, their coordination behaviour can be adapted and changed dynamically, either *(i)* by engineers (humans) willing to sustain the MAS behaviour, or *(ii)* by agents responsible of managing the coordination artifact, with the goal of flexibly facing possible coordination breakdowns or evolving/improving the coordination service provided.
- *Inspectability and controllability* – A coordination artifact typically supports different levels of inspectability: *(i)* inspectability of its operating instructions and coordination behaviour specification, in order to let user agents to be aware of how to use it or what coordination service it provides; *(ii)* inspectability of its dynamic state and coordination behaviour, in order to support testing and diagnosing (debugging) stages for the engineers and agents responsible of its management.
- *Predictability and formalisability* – The coordinating behaviour of an artifact strictly follows the specification/service for which it has been forged: given that specification and the agent interaction history, the dynamic behaviour of the artifact can be fully predicted.

TuCSoN [13] is an example of agent coordination infrastructure supporting this framework: TuCSoN coordination artifacts are called *tuple centres* [9], spread over the network, collected in the infrastructure nodes. Tuple centres technically are *programmable* tuple spaces, i.e. tuple spaces [9] whose behaviour in reaction to communicating event – the insertion, removal, read of tuples from the spaces – can be suitably programmed so as to realise coordination laws managing interactions (ReSpecT is the language adopted for the purpose). Tuple centres can be framed as general purpose coordination artifacts, whose coordinating behaviour can be dynamically customised and adapted to provide a specific coordination

[1] The term embodied is used here to remark their independent existence from the agents using them.

service.

B. Trust through Coordination Artifacts

The notion of coordination artifacts can be useful to model trust issues as discussed in Section III.

As far as social trust is concerned, coordination artifacts can play the role of the abstractions provided by the infrastructure with suitable expressiveness and effectiveness to construct trust articulated strategies. For instance, coordination artifacts can be used as embodiment of the rules of encounter, being concrete shared tools which are used by the agents to interact according a specified protocol. Operating instructions in this case describe what agents are meant to do in order to participate to the protocols (according to their role), artifact state keeps track of the state of the interactions, and artifact behaviour is concerned with the management of the interaction according to the coordinating behaviour described by the protocol.

More generally, as mediating abstractions, coordination artifacts can be used for supporting the *observation* and *traceability* of agent actions and interactions. They can be designed so as to log / trace all the interactions of interest and related events occuring during its usage, in order to be inspected / observed as interaction history concerning not only a specific agent but also the agent society itself. Actions and interactions history can be useful then to build trust models concerning both the overall society, and the individual participating agents. Such trust models could be created both by humans and agents by inspecting and reasoning about the information reified in the artifact interaction history, made available by suitable infrastructure services. From this point of view then, coordination artifacts can provide a useful support for constructing trust model for individual-level approaches based both on socio-cognitive capabilities and on quantitative formulations: heterogeneous agents could exploit the same information to build different kind of models.

Then, the basic properties characterising coordination artifacts impact on modelling both trust in societies and constructive trust. In this case modelling trust toward a system or a society in charge of a specific social task exploiting a specific coordination artifact accounts for two aspects: *(i)* trusting the effectiveness of the coordination artifact for achieving the objective of the social task; *(ii)* trusting agents in being able to use effectively the coordination artifact. Artifact basic properties – concerning inspectability, predictability, etc. – along with the fact that the correctness of artifact behaviour could be formally verifiable and then “certifiable”, with the availability of operating instructions and of a clear interface – could impact effectively in both previous points. It is worth remarking that this introduces a relatively new ontological framework on which formulating trust, introducing new notions such as *usability* of the artifact, the *complexity* of their operating instructions, and so on. This could change and enrich the cognitive model adopted by socio-cognitive approach to model trust of agents towards the environment.

Finally, from an engineering point of view, inspectability and controllability properties of artifacts could impact significantly on the trust toward a system engineered in terms of coordination artifacts, both for a designer and for a user of the system. In particular, *controllability* – which includes also the possibility of making online tests and diagnosis of artifact behaviour and then of the social core of the system, despite of its openness – is an aspect that heavily contributes to determine trust in the system.

V. Contexts for Trust

The notion of *agent coordintation context* (ACC) has been introduced in [8] as infrastructural abstraction modelling the presence of an agent inside its (organisational) environment. As for coordination artifacts, ACCs have been brought into practice within the TuCSoN infrastructure [17]. Here we show their relevance for modelling and engineering the last aspects of trust mentioned in previous chapter, i.e. trust related to organisation and security.

A. The Agent Coordination Context Abstraction

The ACC abstraction can be framed as the conceptual place where to set the boundary between the agent and the environment, so as to encapsulate the *interface* that enables agent actions and perceptions inside the environment. A useful metaphor for understanding ACCs is the *control room* [8]. According to this metaphor, an agent entering a new environment is assigned its own control room, which is the only way in which it can perceive the environment, as well as the only way in which it can interact. The control room offers the agent a set of admissible inputs (lights, screens,...), admissible outputs (buttons, cameras,...). How many input and output devices are available to agents, of what sort, and for how much time is what defines the control room *configuration*, that is the specific ACC. So, the ACC abstraction can be fruitfully exploited to model the *presence* or position of an agent within an organisation, in terms of its admissible actions with respect to organisation resources and its admissible communications toward the other agents belonging to the organisation.

ACCs are meant to be inspectable: it must be possible for an agent to know what kind of ACC it can obtain from an organisation – and so what roles and related actions it is allowed to do.

Two basic stages characterise the ACC dynamics: *ACC negotiation* and *ACC use*. An ACC is meant to be negotiated by the agents with the MAS infrastructure, in order to start a *working session* inside an organisation. The agent requests an ACC specifying which roles to activate inside the organisation. The request can fail for instance if an agent requests to play a role for which he is not allowed, or which is not compatible with the other roles currently actively played by the agent inside the organisation. If the agent request is compatible with (current) organisation rules, a new ACC is created, configured according to the characteristics of the specified roles, and then released to the agent for active playing. The agent then can use the ACC to interact with the organisation environment, by exploiting the actions/perceptions enabled by the ACC.

The ACC framework has been used to model and implement Role-Based Access Control architecture on top of TuCSoN infrastructure [17].

B. Trust through Agent Coordination Contexts

ACCs – supported by suitable infrastructures – guarantee the enforcement of organisational rules and related security policy inside a social environment: they can act as a barriers for agents, filtering only the patterns of actions and perceptions allowed according to their roles. This clearly impacts on the trust that we can have on the systems, providing a generalisation of the security mechanism mentioned for system-level trust. In particular ACC abstraction makes it possible to link trust with the organisational model adopted: agents can participate to activities only by playing some roles through dynamically requested ACC enabling and ruling their actions. In the overall, we can frame an ACC as the embodiment of a *contract* established between a specific agent and the system (organisation) where he is actively playing.

Each organisation can define (and change dynamically) the set of available roles and rules, and then the set of ACCs which can be released to agents. This information can be then made available – by means of suitable infrastructure services – for creating trust in agents and users aiming at participating at or using the systems.

VI. Conclusion

The notion of trust has a deep impact on the future of artificial systems. How trust is modelled, how it is engineered – that is, how it is actually built into artificial systems – are then crucial issues that are already discussed in literature, and in particular in MAS literature. In this paper, we first shortly summarised the many different acceptations of the trust notion, then we pointed out some fundamental open issues that seem to be of particular relevance to the modelling and engineering of trust in the context of complex artificial systems, in general, and of MAS, in particular.

As the main contribution of this seminal paper, we adopted the viewpoint of MAS infrastructures (as the most natural *loci* where to embed trust in MAS) and showed how two different infrastructural abstractions recently introduced (coordination artifacts and agent coordination contexts) can be exploited for modelling and engineering trust within MAS.

References

[1] P. Dasgupta. Trust as a commodity. In D. Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*, pages 49–72. Blackwell, 1998.

[2] Y. Engeström, R. Miettinen, and R.-L. Punamaki, editors. *Perspectives on Activity Theory*. Cambridge University Press, 1999.

[3] R. Falcone and C. Castelfranchi. Social trust: a cognitive approach. In R. Falcone, M. P. Singh, and Y. Tan, editors, *Trust in Cyber-Societies, Integrating the Human and Artificial Perspectives*, volume 2246 of *LNCS*. Springer-Verlag, 2001.

[4] R. Falcone, M. P. Singh, and Y. Tan, editors. *Trust in Cyber-Societies, Integrating the Human and Artificial Perspectives*, volume 2246 of *LNCS*. Springer-Verlag, 2001.

[5] Y. Mass and O. Shehory. Distributed trust in open multi-agent systems. In R. Falcone, M. P. Singh, and Y. Tan, editors, *Trust in Cyber-Societies, Integrating the Human and Artificial Perspectives*, volume 2246 of *LNCS*. Springer-Verlag, 2001.

[6] B. Nardi, editor. *Context and Consciousness: Activity Theory and Human-Computer Interaction*. MIT Press, 1996.

[7] B. Nardi. Studying contexts: A comparison of activity theory, situated action models and distributed cognition. In B. Nardi, editor, *Context and Consciousness: Activity Theory and Human-Computer Interaction*. MIT Press, 1996.

[8] A. Omicini. Towards a notion of agent coordination context. In D. Marinescu and C. Lee, editors, *Process Coordination and Ubiquitous Computing*, pages 187–200. CRC Press, 2002.

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

[10] A. Omicini and S. Ossowski. Objective versus subjective coordination in the engineering of agent systems. In M. Klusch, S. Bergamaschi, P. Edwards, and P. Petta, editors, *Intelligent Information Agents: An AgentLink Perspective*, volume 2586 of *LNAI: State-of-the-Art Survey*, pages 179–202. Springer-Verlag, Mar. 2003.

[11] A. Omicini, A. Ricci, and M. Viroli. Formal specification and enactment of security policies through Agent Coordination Contexts. *Electronic Notes in Theoretical Computer Science*, 85(3), Aug. 2003.

[12] A. Omicini, A. Ricci, M. Viroli, and C. Castelfranchi. Coordination artifacts: Environment-based coordination for intelligent agents. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, New York, USA, 2004. ACM Press.

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

[14] S. Poslad, M. Calisti, and P. Charlton. Specifying standard security mechanisms in multi-agent systems. In *Workshop on Deception, Fraud and Trust in Agent Societies*, pages 122–127, Bologna, Italy, 2002. AAMAS 2002, Proceedings.

[15] S. D. Ramchurn, D. Hunyh, and N. R. Jennings. Trust in multi-agent systems. *Knowledge Engineering Review*, 2004. to appear.

[16] A. Ricci, A. Omicini, and E. Denti. Activity Theory as a framework for MAS coordination. In P. Petta, R. Tolksdorf, and F. Zambonelli, editors, *Engineering Societies in the Agents World III*, volume 2577 of *LNCS*, pages 96–110. Springer-Verlag, Apr. 2003. 3rd International Workshop (ESAW 2002), Madrid, Spain, 16–17 Sept. 2002. Revised Papers.

[17] A. Ricci, M. Viroli, and A. Omicini. Agent coordination contexts: From theory to practice. In R. Trappl, editor, *Cybernetics and Systems 2004*, Vienna, Austria, 2004. Austrian Society for Cybernetic Studies. 17th European Meeting on Cybernetics and System Research (EMCSR 2004), Vienna, Austria, 2004. Proceedings.

[18] J. Rosenschein and G. Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. MIT Press, 1994.

[19] J. Sabater and C. Sierra. REGRET: A reputational model for gregarious societies. In *1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2002)*, pages 475–482, Bologna, Italy, 2002. ACM Press. Proceedings.

[20] T. Sandholm. Distributed rational decision making. In G. Weiss and S. Sen, editors, *Multi-Agent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 299–330. AAAI/MIT Press, 1999.

[21] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, 1996.

[22] M. Viroli and A. Omicini. Coordination as a service: Ontological and formal foundation. *Electronic Notes in Theoretical Computer Science*, 68(3), Mar. 2003. 1st International Workshop “Foundations of Coordination Languages and Software Architecture” (FOCLASA 2002), Brno, Czech Republic, 24 Aug. 2002. Proceedings.

[23] P. Wegner. Why interaction is more powerful than algorithms. *Communication of ACM*, 40(5):80–91, May 1997.

[24] G. Zacharia and P. Maes. Trust through reputation mechanisms. *Applied Artificial Intelligence*, (14):881–907, 2000.