Multiagent System Engineering: The Coordination Viewpoint

Paolo Ciancarini[1], Andrea Omicini[2], and Franco Zambonelli[3]

[1] DSI - Università di Bologna, Mura Anteo Zamboni 7
Bologna, Italy
ciancarini@cs.unibo.it
[2] LIA - DEIS - Università di Bologna, Viale Risorgimento 2
Bologna, Italy
aomicini@deis.unibo.it
[3] DSI - Università di Modena, Via Campi 213b
Modena, Italy
franco.zambonelli@unimo.it

Abstract. The paper focuses on the design of multiagent systems and argues that traditional approaches fall short when dealing with complex multiagent systems. On this basis, this paper shows how an approach based on coordination models can help in the design of multiagent systems. A simple example in the area of conference management is assumed as a case study to clarify the concepts expressed.

1 Interaction in the Agent's Space

Multiagent systems are more and more becoming an ubiquitous paradigm for the design and implementation of complex software applications. Even though somehow blurred throughout the vast literature on multiagent systems, the notion of agent can be characterised by few fundamental key-points: *(i)* autonomy, *(ii)* interaction, and *(iii)* task. In other words, an agent may be thought as an *autonomous* software component which *interacts* with its environment in order to achieve its *tasks*.

Autonomy guarantees that an agent will pursue its goals proactively, so as to accomplish its tasks. Accordingly, the process of designing a multiagent system is typically conceived as picking out the multiplicity of tasks which have to be accomplished in order to achieve the intended multiagent system's goal(s), and delegating them to the agent's responsibility. In other words, the focus is mainly on intra-agent aspects, such as agent's internal structure, its view of the world, its beliefs, desires and intentions. This typically includes also delegating to agents the responsibility of coordinating itself with the other application agents accordingly to specific interaction protocols embedded into the agents [15]. Some recent proposals elect interaction to a key issue in the design of multiagent systems [25], to be tackled with from the preliminary analysis phase. Still, even in this case, agent interactions are simply considered in terms of *communication* with other agents and interaction protocols, without considering higher-level design issues related to the fact that agents will eventually live dipped into possibly complex multiagent systems.

N.R. Jennings and Y. Lespérance (Eds.): Intelligent Agents VI, LNAI 1757, pp. 250–259, 2000.

Both the above approaches typically lead to a two-phase engineering methodology for multiagent system: first, the agents composing the systems are identified and designed according to the tasks they have been assigned; then, they are simply made interacting so as to actually build the multiagent ensemble. However, this is usually done without any explicit modelling of the global behaviour of the resulting multiagent systems. According to this (compositional) perception, a multiagent systems is viewed as the mere sum of its parts (the agents). What is relevant, then, is the role of autonomous agents acting so as to achieve their tasks: agent interaction adds nothing but the capability to compose agents into an ensemble, enabling them to work together. This corresponds to considering multiagent systems as a simple multitude of individuals, and disregarding the social aspects such as collective behaviour, social rules, and so on.

Moreover, this endorses a view of the space of agent's interaction as merely the *space of communication*, where agent-generated knowledge is made available and consumed, and where interaction histories simply results from the chaotic interleaving of the observable behaviours of single agents [23]. Correspondingly, many research efforts in the multiagent field actually deal with the problem of *enabling communication* among agents: ACLs [21] (such as KQML [9] or FIPA [10]), allowing knowledge to be transferred from one agent to another, middleware components (mediators [24], information brokers) and infrastructures (such as ORBs [13]), providing communication with some degree of transparency concerning ontology, name systems, services, and so on. However, simply focussing on enabling communication and defining agent interaction protocols does not give multiagent system designers a way to effectively deal with the complexity of the agent communication space: it makes it exist, but does not really help in governing it.

Instead, an holistic (non-compositional) view of multiagent systems is currently being promoted by several recent research efforts, introducing notions such as *social agency* [21], *socialware* [14], and *social laws* [20]. These approaches assume that the full understanding of multiagent systems calls for a comprehensive theoretical setting for *agent societies*, defining what is the world where agents live and socialise, which kinds of individuals populate the world, and how the world is ruled. Moreover, they recognise that social rules, far from being a mere limitation of the behaviour of individuals, constitutes a further source of intelligence in multiagent systems (we could call it *social intelligence*), which goes beyond the one provided by single agents, and represents a natural place where global system properties can be embodied.

This suggests that, in the context of multiagent systems, engineering the social aspects is at least as much relevant as engineering single agents. Accordingly, the social issue of a multiagent system should be accounted for since the earliest design phases, and social rules and collective behaviour should be the subjects of a separate design phase, independent from the design of individual agents.

2 Coordination as a Design Dimension

Shifting focus from the agent internal structure to the agent interaction space changes the way in which a multiagent system is designed. The design process of a multiagent system can still be conceived as a two-phases process, but of a quite different sort. In the