Towards a Methodological Framework for Holonic Multi-Agent Systems

Sebastian Rodríguez, Vincent Hilaire and Abderrafiâa Koukam

Université de Technologie de Belfort-Montbéliard
Laboratoire Systèmes et Transport
90010 Belfort CEDEX
FRANCE
email: sebastian.rodriguez@utbm.fr

Abstract. In order to apply the holonic paradigm to design Multi-Agent Systems, there is need of a methodology. This methodology must lead from the analysis of a problem in terms of holons to the design of a Multi-Agent Systems. The aim of this paper is to propose a framework to guide the analysis of such systems. This framework is based upon organizational concepts which have been successfully used in the MAS domain. In order to illustrate this approach we will take as example the adaptive mesh problem.

Keywords: Holonic Multi-Agent Systems, Adaptive Mesh, Organizational Model

1 Introduction

Even if software agents and multi-agent systems (MAS) are recognized as both useful abstractions and effective technologies for modeling and building complex distributed applications, they are still difficult to engineer. When massive number of autonomous components interact it is very difficult to predict the behavior of the system and guarantee that the desired functionalities will be fulfilled. Moreover, it seems improbable that a rigid unscalable organization could handle a real world problem.

The term holon was originally introduced in 1967 by the Hungarian Philosopher Arthur Koestler[13] to refer to natural or artificial structures that are neither wholes nor parts in an absolute sense. According to Koestler, a holon must respect three conditions: (1) being stable, (2) having the capability of autonomy and, (3) being capable of cooperation[2]. The stability means that a holon is capable of reacting when strong perturbations are applied. The autonomy implies that a holon is capable of auto-management in order to achieve its own goals. The capability of cooperation denotes that holons are capable of working in common projects according to shared goals with other holons or other layers of holons.

Holonic organizations have proven to be an effective solution to several problems associated with these hierarchical self organized structures (e.g. [5], [6], [9], [15]).

In many MAS applications, an agent that appears as a single entity to the outside world may in fact be composed of several agents. This hierarchical structure corresponds to the one we find in Holonic Organizations.

Frameworks have been proposed to model specific problem domains, mainly in Flexible Manufacturing Systems (FMS) and Holonic Manufacturing Systems (HMS), such as PROSA[1][17] and MetaMorph[14]. However, the Holonic paradigm has also been applied in other fields such as cooperative work[3].

In order to apply the holonic paradigm to design MAS, there is need of a methodology. This methodology must lead from the analysis of a problem in terms of holons to the design of a MAS. This paper aims to propose a framework to guide the analysis of such systems. Our architecture isn't application domain dependent so it can be easily reused in other domains. This framework is based upon organizational concepts which have been successfully used in the MAS domain [10,12,16,18].

To illustrate this approach we will take as example the adaptive mesh problem. The reader should bear in mind that the problem is intended to illustrate the methodology rather than the complete resolution of this complex problem.

A distinguishing feature of cellular radio mobile networks is the rapid increase of the consumer demand and the ensuing complexity in their design and management. Responding to this demand requires the space to be partitioned between a large amount of service units or cells. The adaptive meshing problem for dimensioning considers traffic statistics as a predefined resource that must be attributed to many adaptive low power Base Transceiver Station (BTS). In order to treat this problem using a HMAS, each resource will be assigned with an agent whose main and unique goal is to find a BTS that could cover the traffic in his resource. This goal must be accomplished respecting certain constraints like geometry and the maximal traffic an antenna can cover.

We base our approach on the Role-Interaction-Organization (RIO) Methodology. RIO uses a specific process and a formal notation that are described in [12].

This work is structured as follows. Section 2 presents the HMAS and holon internal structure. Section 3 gives detail on the basic holon roles and illustrates the different roles on the adaptive mesh problem. Section 4 present the simulator developed and the obtained results . Eventually, Section 5 concludes.

2 Holonic Multi-Agent Systems

2.1 The Structure as a whole

The *H*olonic *M*ulti-*A*gent *S*ystem (HMAS) Structure can be seen as a set of hierarchical levels, where the agents can interact only with other agents at the same level or at the level immediately below.

---

[1] The name PROSA stands for Product-Resource-Order-Staff Architecture, which refer to the composing types of holons.

Figure 1 presents a possible disposition of the HMAS Structure. As shown, only agents that are performing a Head role, are present in two different levels. A dashed line is used to show that is the same entity in both levels. The role of this agent is performing is independent of the role he is performing in the lower level.

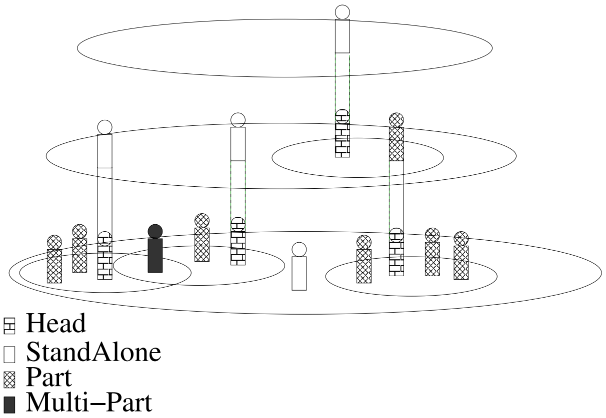


Fig. 1. HMAS Structure

Our approach considers a HMAS as moderated groups, where each head represents the members of his holon with the outside world as introduced by [8,11]

In this organization every agent without any previous engagement is considered to perform what we call a "*Stand-Alone*" role. These agents will interact only with the representatives (heads) of the holons to request a service provided by the Holon.

According to their needs, they could play the "merging" role and request to the heads their admission as a member. If accepted, the agent becomes then a *"part"* of that holon (fusion).

From that moment on, until he leaves the holon, either by self decision or command of the holon's head, he can directly interact only with the members of the holon. Any other request must go through the head, who will then request that service to the outside agents.

If the satisfaction[2] level is not reached, the agent can either leave the holon (fission) or try to join other holons. If no conflict arises, the agent will perform the *"multi-part"* role.

---

[2] See Section 2.2.

2.2 A holon internal structure

An agent that is interacting within a HMAS will change his role according his needs. The whole merging system is inspired by the Immune System [4,7]. According to this approach, it is possible to determine if two agents are suited to work together considering their goals and services.

Each agent has an identifier which gives all relevant information. As in the immune system, an agent can find out the affinity he has with another by comparing his identifier with the identifier of the other agent, then uses this affinity to decide whether or not to merge. Several definitions of "Affinity" can be found depending on the problem nature. In a general sense we can say that two agents have a high affinity with each other if their goals are similar and their services complementary. Once again, several approaches can be taken to define when two goals are similar.

In the meshing example, the holon's identifier should give the position of the agent's resource (X, Y coordinates) and the traffic it contains. Using these values, an agent can determine whether or not to merge.

As explained before, the affinity should give a measure of the compatibility of the agent's goal and services. In this particular case, both agents will have the same goal, to ensure the coverage of their resources. Therefore, the main problem is to ensure that the geometrical constraints are respected. The affinity could be decomposed in two main parts:

- **Distance affinity**: will provide a geometry dependent value used to ensure that the geometrical constraints are respected. As we need square meshes, we will use two parameters to test the distance affinity. First, we will check if the agent trying to merge is inside the acceptance distance (see Figure 2). If not, the agent will be rejected. However, if the agent is inside this distance, the head will calculate the real distance affinity. The affinity equals the number of resources, that are already part of the mesh, and have an unitary distance with the resource the new agent represents.

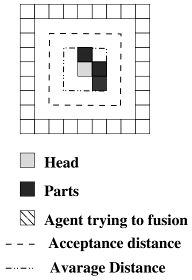
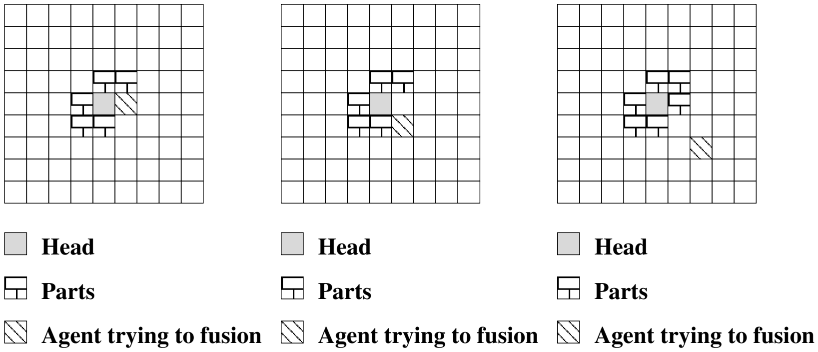


Fig. 2. Acceptance Distance

Lets consider three different situations shown in Figure 3. Agents in (a) and (b) will be accepted into the holon if the maximum traffic that an antenna can handle has not been reached. If the mesh is close to this value, agent (a) will be privileged. However, agent c will be rejected in all circumstances.



(a) Distance Affinity=4 - Accepted Agent

(b) Distance Affinity=2 - Accepted agent

(c) Distance Affinity=0 - Rejected Agent

Fig. 3. Distance Affinity Example.

- **Resource affinity**: Used to ensure that the limits of an antenna are not exceeded.

Another important parameter in the agents behavior is the agent's satisfaction. It enables the agent to move between different roles according to his needs. In the mesh example, we will use the satisfaction to ensure that the agent's resource is covered by a mesh. We present first some definitions and then a state diagram of the agent's roles.

In the next definitions we will use the following notation:

H_i	$Set\, of\, Holons\, the\, agent\, i\, belongs\, to$
$HMAS$	$Set\, of\, all\, agents\, in\, the\, Holonic\, Organization$
R_i	$Role\, of\, the\, agent\, i$
$\#H_i$	$cardinality\, of\, H_i$

Self Satisfaction *(SS_i): Satisfaction for the agent i produced by his own work.*

Collaborative Satisfaction *(CS_i^H): Satisfaction produced for the Agent i by his collaboration with other agents of the Holon(s) he belongs to.*

Collective Satisfaction *(ClS_i^H): Satisfaction Produced for the head of the Holon H by the member agent i*

Accumulative Satisfaction *(AS_i^X): Satisfaction produced for the Agent i*

$$AS_i^X = \sum_{p=0}^{p=X} CS_i^p \; where \, X = \#H_i \tag{1}$$

Leadership Satisfaction *(LS_i^H): Satisfaction produced for the agent i by the fact he is the leader (Head) of the Holon H*

$$LS_i^H = \sum ClS_j^H \; \forall j \neq i, \, j \in H$$

Necessary Satisfaction *(NS^C): Necessary Instant Satisfaction to finish the task the agent has been assigned according to Constraints C. If the constraints are limited to time t and $t = K$, then $NS^C = NS^t = NS^K = constant$.*

Instant Satisfaction *(IS_i): Satisfaction produced by the work done up to the moment (individual and accumulated).*

$$\forall i \in HMAS \; IS_i = \begin{cases} AS_i + SS_i & if \, R_i = Part \\ LS_i + SS_i & if \, R_i = Head \\ SS_i & if \, R_i = Stand - Alone \end{cases} \tag{2}$$

Using the above defined satisfaction, we can see the transitions between roles using only satisfaction, as shown in Figure 4. Transitions are labeled by conditions within square brackets. If the condition is true, the transition is fired. In the next section we present a possible representation of the satisfaction for the mesh problem.

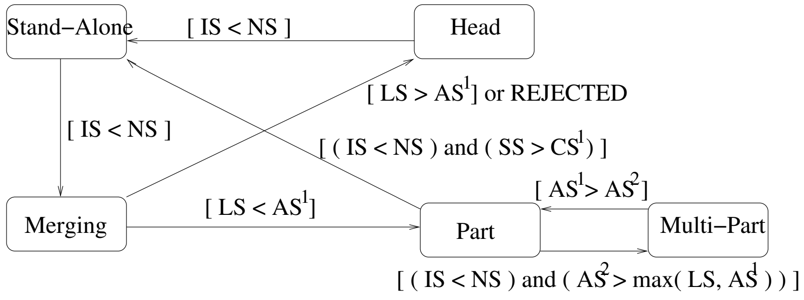


Fig. 4. State Transitions Diagram according to the agent's satisfaction parameter

As said before, in the example the only goal of the agent is to ensure the coverage of his resource and, at the same, time respect the constraints imposed by the problem (Geometrical and Maximum coverage).

3 Holon Roles

An agent that wants to be part of a holonic organization should be able to perform different behaviors according to his status in the HMAS. Several models were proposed in the literature to manage the interaction and the organization between the agents [8]. In our approach each Holon has a representative (Head) that will interact with other agents and Holons. A holon has four roles (StandAlone, Head, Part and Multi-Part) and one transitory state (Merging) .

3.1 StandAlone

As an agent joins a HMAS Organization, he has no special bindings with other agents, and does not collaborate with any other agent. This situation represents a Stand Alone Behavior. In this state, the agent's decisions are not attach to any restriction but his own goals and objectives. The agent will remain in this state as long as he is satisfied.

In the adaptive mesh problem, the goal of the agent is to ensure the coverage of his resource, then he will try to join a mesh immediately. The only situation where he remains in a stand-alone role is when his resource can get an antenna for him alone, which is a highly improbable situation. We can then say that NS is the maximum traffic an antenna can cover and SS the agent's resource. If we look at equation 2 we see that $IS = SS$. Then if the agent's resource is not big enough to receive an antenna of its own, he will try to merge.

3.2 Head

There are several ways how an agent can raise to the Head state. The first and more straight-forward is when the agent becomes head by his own decision. In this case, the negotiations with other Holons failed in the merging State (see 3.5). Another way is to become a Holon head by Command. In this case the actual Holon head is going to change his state and needs to give up his role to other agent. This agent becomes the representative of all the others with the exterior world. He is then able to take engagements with other agents that will benefit the Holon as a whole.

Several condition could lead an agent to leave his role as a the head of a Holon: (1) The agent finished his task and commands a coworker to take his place as the Head of the Holon. The new head is now free to take any decision according the common goals. (2) The agent performing the head role has been chosen to be the head of a new holon. In this case, his former Holon will be added as a part of the new Holon he is representing. The new designed Head is conditioned in his actions, as he must become a part of the new formed Holon in the higher level as described. (3) The agent has no more common goals with the rest of the agents and needs to leave the holon to search new co-workers.

In the adaptive mesh problem, an agent that performs the head role will be responsible to respect the constraints of a mesh. He will be representing a possible mesh in the system, and will accept or refuse other agent's requests to fusion according the constraints.

Although all heads represent possible meshes in the system, a Holon Head can decide to leave his role if, after trying to improve the Holon's satisfaction, the satisfaction is insufficient to remain as a Holon. In order to improve the Holon's satisfaction, will accept new agents to increase the Holon covered resource, or will command member agents to leave the Holon if they don't respect the geometrical constraints or if the covered resource has exceeded the maximum.

3.3 Part

Once an Agent was accepted in a Holon, his autonomy is reduced because of his obligations with the Holon. In this state, the agent can receive requests from other members of the holon, and Commands from the Holon's head. Commands and Requests are treated differently. In both cases, a member(sender) of the Holon is demanding a service provided by other agent (receiver)[3]. However, the head can either make a request or a command[4] to a member of his holon. Commands should be used only for time-critical demands and to execute engagements of the Holon[5] with external agents. If a holon-part needs a service provided by another member to execute a time-critical task, he must request this service to the Head, who will then send a command to the service provider.

As mentioned before, heads are the only that can send *Commands* to other agents and, as any other member of the Holon, can also send requests.

The *Requests* are simple demands to an agent for his services. The agent can then analyze the demand and accept or refuse it. The criteria of this decision is problem-dependent and there are no general rules. *Commands*, on the other hand, are actual orders to the agent, that he can not refuse and should be treated before any request.

As the head acts as the representative of all the agents in the Holon, he is in title to engage the Holon in new activities[6]. In order to provide the Head of a group of services he can count on (see service allocation, section 3.6), as an agent joins the Holon, the head will allocate a group of services.

These allocated services can be used by the head to complete a task. Although this procedure can be achieved using a sequence of request/accept-refuse with the members of the Holon, this method speeds the execution up and ensures that time-critical task can be done. It also avoids possible dead-locks where the only holder of an important service blocks the whole holon.

---

[3] The receiver could be at the same time a Holon Head and he might need to request the service to another agent.

[4] The services to which a head can issue a command are restricted to the services allocated in the merging process (Section 3.5).

[5] These engagements are requests of external agents that the Holon Head accepted.

[6] This could be from rendering one single service to an external agent to compromised the whole holon to become part of a higher level Holon.

In the adaptive mesh problem, the agent gets to this role if negotiations with a Holon succeeded. He will remain in the holon if his satisfaction level is raising. In order to calculate this value, we will consider AS^H as the sum of the resources of the members of the holon H. If the agent remains in the holon, his goal will be accomplished and will stay in the HMAS until the antenna is finally assigned. However, is is also possible that the agent receives a command to leave the holon, in that case, he must return to Stand-Alone and restart the merging process.

3.4 Multi-Part

The Multi-Part Role is a special case of the Part Role. This state is reached when an agent belongs to more than one Holon. There are several situations to consider(e.g: requests conflicts, forwarding requests, etc) however these issues are out of the scope of this paper. In the presented example, a resource must be covered only by one antenna, then no agent can assume this role. However, other approaches could be used to enable this behavior, like to overlap different meshes.

3.5 Merging

The agent will get to this state if an unsatisfactory condition has arrived in his earlier state. Meaning that is the agent's understanding that he will not achieve his goals respecting the imposed constraints if he remains in the previous state. These constraints could be externally imposed (time, defined by the problem itself, etc) or self imposed due to the need of services. There are two possible previous states:

StandAlone: The agent is requesting to become part of a holon with his whole services available. This means that the agent has no previous engagements with other holons. If these negotiations fail (he was not accepted by a Holon), he may then become a holon head himself and try to find coworkers. If he is accepted by a Holon, he becomes a part.

Part or **Multi-Part**: The agent is already part of other(s) holon(s). This is of special interest to the head of the holon where the agent is requesting his entry. The agent must send his status when he requests the fusion into the holon. He also sends his *disponibility*, a measure of the probability that the agent can provide a determined service. The disponibility should be defined according the problem's definition and constraints.

3.6 Service allocation

Conflict solving is an important part of the Part and Multi-Part Roles. As explained in 3.3 an agent can receive two different kinds of demands, *Requests* and *Commands*. Requests can be refused, so to solve a conflict an agent could use his right to refuse a demand. However, Commands can not be refused, so the head's right to send commands is limited to a set of services.

In the merging process, the Head must allocate a set of services. After an agreement was reached, the Holon's head is entitled to send command concerning only this set of services, while must send requests to use other services provided by this agent.

Normally, the Head will allocate those services that are not present in other holon-members. As the agent joins a new Holon, only a limited part of his services could be allocated.

It is impossible then that an agent receives two commands for the same service. It should also be considered, that some services can not be allocated independently, e.g. the case of a create-destroy mechanism.

4 Simulation Results

One of the mayor problems in radio mobile networks is to obtain balanced and optimal BTS usage. A simulation application was developed to tackle this problem and test the principles exposed in this paper. As in our approach one agents is assigned to each resource, the number of agents increases rapidly when using more detailed maps, as shown in Table 1. This problem can be easily solved if we consider the hierarchical nature of the HMAS. The map could be divided in independent zones, where the holon responsible of the zone will report the final result to the simulation holon.

Map size	Number of Agents
40x40	670
78x78	2694
152x152	10793
300x300	43223

Table 1. Number of Agents

The system was implemented with the Madkit platform[1], and an extension library is being developed to support roles and protocols needed by a HMAS. Because of the high number of agents in each simulation zone, several servers were required. In order to support these highly detailed resources maps a server/client structure was implemented. Each server will execute the simulation of a reduced area. The servers are provided as well with a simulation responsible called Server Network Holon. The former is in charge of the communications with the Client Network Holon in order to report the progress of the simulation in real-time.

For the moment, the simulation uses a fixed maximum number of meshes, which is translated in a fixed number of agent that are allowed to move to the Head Role. These heads consider them self as the center of the mesh and accept or refuse possible member using their coordinates to respect the geometrical constraints.

Future work will enable the head to reassign his role to a better suited member. This is aimed to obtain dynamically positioning of the BTS.

Figure 5 shows the resource of the meshes found in a simulation with a 40x40 resource map. For this simulation, the maximum BTS coverage was fixed to 200. It involved 670 agents and resulted in 21 meshes. As shown, each BTS presents a balanced and close to its maximum usage (Y axis).

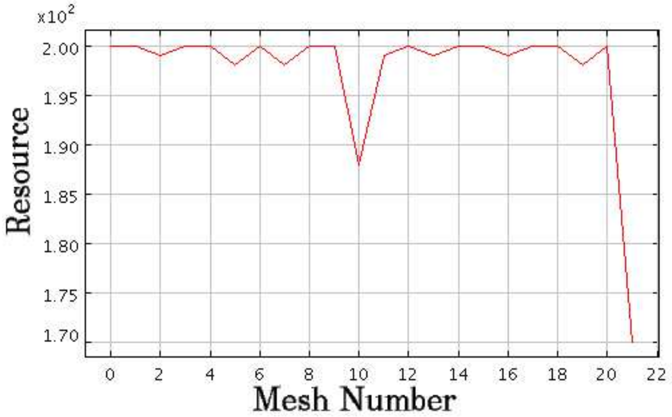


Fig. 5. Mesh resource vs Mesh Number

5 Conclusion

In this paper we have presented an approach to design HMAS. This approach is based on the definition of typical behaviors, called roles, a holon can play over time. Using these generic roles the agent's behavior can be detailed to solve a wide range of problems. We give rules which define the possible transitions between these roles. Moreover, we give generic state variables for holons and a merging mechanism inspired by the immune system. Using the adaptive mesh problem we have illustrated how to apply these concepts.

Other frameworks and methodologies have been proposed [3,14,17] and, although they have shown to be effective inside specific domains, a more generic framework is needed. Indeed, it is difficult to design a HMAS without clear and specific definitions that can lead from the analysis in terms of holon to the design of the system.

The aim of the paper is to contribute to the definition of a framework for the analysis and design of HMAS. We have given some elements of this framework but it needs more work to constitute a methodology for HMAS. We plan to give formal definition of the concepts presented using the formal notation included in RIO which enables prototyping and formal verification (model checking or semi-automatic deduction of properties). We also plan to apply this approach to other problems.

References

1. Madkit - multi-agent development kit. http://www.madkit.org.
2. Emmanuel Adam, René Mandiau, and Christopher Kolki. *Une Méthode de modélilsation et de conception d'organizations multi-agents holoniques*, chapter 2, pages 41–75.
3. Emmauel Adam. *Modele d'organization multi-agent pour l'aide au travail cooperative dans les processus d'entreprisee: application aux systemes administratif complexes.* PhD thesis, Universite de valenciennes et du hainaut-cambresis, 2000.
4. M. Bakhouya, J. Gaber, and A. Koukam. A middleware for large scale networks inspired by the immune system. In *Workshop on Biologically Inspired Solutions to Parallel Processing Problems BIOSP02 (IPDPS 2002 Workshop)*, Fort Lauderdale, Florida, April 2002.
5. G.A Bell, A. Cooper, M, M Kennedy, and J. Warwick. Using the Holon Framework: from Enquiry to Metrication - A Higher Education Case Study. In *In Proceedings of the 19th International System Dynamics Conference, Atlanta, Georgia.*, 2001.
6. Hans-Jürgen Bürckert, Klaus Fischer, and Gero Vierke. Teletruck: A holonic fleet management system.
7. Dipankar Dasgupta. *Artificial Immune System and Their Applications.* Springer, 1998.
8. Christian Gerber, Jörg Siekmann, and Gero Vierke. Holonic Multi-Agent Systems. Technical report, Deutsches Forschungszentrum für Künztliche Inteligenz - GmbH, Postfach 20 80, 67608 Kaiserslautern, FRG, May 1999. Main.
9. Ling Gou, Tetso Hasegawa, Peter B. Luh, Shinsuke Tamure, and John M. Oblak. Holonic Planning and Scheduling for Robotic Assembly Testbed. *Proceedings of the Fourth International Conference on Computer Integrated Manufacturing and Automation Technology*, pages 142–149, October 1994.
10. P. Gruer, V. Hilaire, Abder Koukam, and Krzysztof Cetnarowicz. A formal framework for multi-agent systems analysis and design. *Expert Systems with Applications*, 23, December 2002.
11. G.Vierke H.-J. Bürckert, K. Fischer. Transportation scheduling with holonic mas - the teletruck approach. In *Proceedings of the Third International Conference on Practical Applications of Intelligent Agents and Multiagents*, pages 577–590, 1998.
12. Vincent Hilaire, Abder Koukam, Pablo Gruer, and Jean-Pierre Müller. Formal specification and prototyping of multi-agent systems. In Andrea Omicini, Robert Tolksdorf, and Franco Zambonelli, editors, *Engineering Societies in the Agents' World*, number 1972 in Lecture Notes in Artificial Intelligence. Springer Verlag, 2000.
13. Arthur Koestler. *The Ghost in the Machine.* Hutchinson, 1967.
14. Francisco Maturana. *MetaMorph: an adaptive multi-agent architecture for advanced manufacturing systems.* PhD thesis, The University of Calgary, 1997.
15. Christian Russ and Gero Vierke. Agent-based configuration of virtual enterprises.
16. Michael Wooldridge, Nicholas R. Jennings, and David Kinny. A methodology for agent-oriented analysis and design. In Oren Etzioni, Jörg P. Müller, and Jeffrey M. Bradshaw, editors, *Proceedings of the Third Annual Conference on Autonomous Agents (AGENTS-99)*, pages 69–76, New York, May 1-5 1999. ACM Press.

17. J. Wyns. *Reference architecture for Holonic Manufacturing Systems - the key to support evolution and reconfiguration.* PhD thesis, Katholieke Universiteit Leuven, 1999.
18. F. Zambonelli, N. R. Jennings, A. Omicini, and M. Wooldridge. Agent-oriented software engineering for internet applications. In A. Omicini, F. Zambonelli, M. Klusch, and R. Tolksdorf, editors, *Coordination of Internet Agents: Models, Technologies, and Applications*, pages 326–346. Springer-Verlag: Heidelberg, Germany, 2000.