

USING ORGANIZATIONAL STRUCTURES EMERGENCE FOR MAINTAINING FUNCTIONAL INTEGRITY IN EMBEDDED SYSTEMS NETWORKS

Jean-Paul Jamont

Laboratoire de Conception et d'Intégration des Systèmes

Institut National Polytechnique de Grenoble

F-26000 Valence, France

jean-paul.jamont@esisar.inpg.fr

Michel Occello

Université Pierre-Mendès France

Laboratoire de Conception et d'Intégration des Systèmes

Institut National Polytechnique de Grenoble

F-26000 Valence, France

michel.occello@iut-valence.fr

Abstract This paper deals with the application of multiagent self-organization techniques for giving adaptive features to distributed embedded systems evolving in aggressive environments. Interesting results are presented, showing how functional integrity maintenance of artificial complex systems can take advantage of a collective decentralized approach.

Keywords: Structure emergence, self-organization, multiagent system, wireless sensor network.

Introduction

Multiagent systems are especially adapted for designing complex (open, distributed) systems. We propose an innovative approach for open multiagent systems in the context of wireless networks of intelligent sensors.

Such networks are composed by autonomous hybrid hardware/software entities ensuring measuring tasks and information routing tasks. The sensor wireless nodes have to adapt their behavior according to their independent energy resources and their position in the organization.

We describe how we improve both management of communication and management of energy resources including a strong fault tolerant feature through a totally decentralized approach and inherent multiagent emergence features.

We give finally an insight to the experimental intelligent wireless sensor network architecture used for the EnvSys Project¹ of instrumentation of an underground river system.

1. Emergence and Multiagent systems

Multiagent system. An agent is a software entity embedded in an environment which it can perceive and in which it acts. It is endowed with autonomous behaviors and has objectives. Autonomy is the main concept in the agent issue: it is the ability of agents to control their actions and their internal states. The autonomy of agents implies no centralized control.

A multiagent system is a set of agents situated in a common environment, which interact and attempt to reach a set of goals. Through these interactions a global behavior, more intelligent than the sum of the local intelligence of multiagent system components, can emerge. The emergence process is a way to obtain, through cooperation, dynamic results that cannot be calculated in a deterministic way.

A multiagent system can have several characteristics. It is *open*, as opposed to *closed*, if its structure is capable of dynamical changes: in other words, the system tolerates that the agents enter and leave freely the multiagent system. A multiagent system can be *homogeneous*, as opposed to a *heterogeneous*, if it is constituted of homogeneous agents from the point of view of their theory (representation and properties) and their architecture (particular methodology of agent design).

Emergence. The emergence talks about the not programmed and irreversible sudden appearance of phenomena in a system (maybe multiagent) confirming that “the whole is more than the sum of each part”.

It is difficult to qualify the emergent characteristic of a phenomenon, however Muller and Parunak, 1998 proposes an interesting definition of it. Falling under the prolongation of the work reported in (M.R. Jean, 1997), it affirms that a phenomenon is emergent if:

¹This project is funded by the FITT program (Incitative Fund for Technological Transfert) of the French Rhône-Alpe Regional Council.

- there is a set of agents interacting via an environment whose state and dynamic cannot be expressed in terms of the emerging phenomenon to produce in a vocabulary or a theory D ,
- the dynamic of the interacting agents produces a global phenomenon such as, for example, an execution trace or an invariant,
- The global phenomenon is observable either by the agent (strong sense) or by an external observer (weak sense) in different terms from the subjacent dynamics i.e. another vocabulary or another theory D' .

To give to an agent's system a particular functionality, the traditional method consists in carrying out a functional decomposition of the problem in a set of primitives which will be established in the agents. The alternative suggested by (Steels, 1990) aims to make this functionality emerge from the interactions between the agents. The advantages of the "emergent functionality" approach are first of all a reinforcement of the robustness of the system : it is less sensitive to the changes of the environment. The reason is that, unlike to the case of a programmed functionality (traditional approach), the designer doesn't need to consider all the possibilities for the system reacts according to each situation.

2. Communication management of wireless sensor networks

In this part we expose our practical problem (the Envsys project), the main difficulties of this type of application and the traditional solution to solve this problem.

2.1 Our practical case: the EnvSys project

The purpose of the ENVIRONMENT SYStem project is to monitor an underground river network. Let us present the origin of this project and the problems occurring in such an application (Jamont et al., 2002).

Origin of the project. The ENVSYS project finds its origin in a statement: the measurement of the various parameters in an underground river system is a complex task. In fact, the access to this type of underground galleries is difficult: it requires help from speleologists. Besides, the installation of wire communication networks is difficult, especially because the structure of hydrographic systems is very often chaotic. Finally, in the case of a radio communication network, the underground aspect complicates wave propagation and for the moment the techniques which are used are not totally mastered.

The general idea of the project is to study the feasibility of a sensor network from the existing physical layer. This will allow wireless instrumentation of a subterranean river system. Such a network would present an important interest in many domains: the study of underground flows, the monitoring of deep collecting, flooding risk management, river system detection of pollution risks, etc.

The issue. In a subterranean river system, the interesting parameters to measure are numerous: temperature of air and water, air pressure and if possible water pressure for the flooded galleries, pollution rate by classical pollutants, water flow, draft speed, etc. All this information will be collected at the immediate hydrographic network exit by a workstation like a PC. These data will be processed to activate alarms, study the progress of a certain pollution according to miscellaneous measuring parameters, determine a predictive model of the whole network by relating the subterranean parameters measures of our system with the overground parameter measures more classically on the catchment basin.

We do not wish to carry out this instrumentation with a wire network for obvious reasons of convenience. We shall use electromagnetic waves with low frequencies as a carrier. These waves have an interesting property: they are able to go through rock blocks. Every sensor has a limited transmission range.

Having defined the role of sensors, we can represent the structure of our communication network. It consists of a set of sensors and a listening station as illustrated on the following figure (see fig 1):

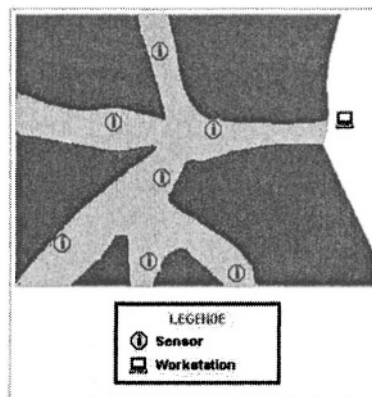


Figure 1. The monitored cave

Here is a non-exhaustive list of problems which one needs to address:

- How to carry out the physical layer and what level of protocol connection to choose above such layer?
- How to route the information in the best way? Each of the sensors cannot physically communicate with the workstation which collects the information. Which sensor should thus make the decision to repeat the information?
- How to monitor such a complex environment?
- What kind of intelligence to give the network?

The main contribution of the work presented in this paper is situated at a logical level, concerning especially the last three points of the problems listed before.

Generally these devices have autonomous energy sources. These constraints must be taken into account in order to optimize the communication management (the energy devoted to communication constitutes an important part of the sensor energetic cost). The energetic parameter is important in the sense that it can create internal faults or that it can influence other parameters like the emission range. Furthermore the environment can be hostile (temperature, pressure, water flood environment...) and can cause internal faults.

The open nature of these networks can be another source of important errors. In fact, insertion and departure of the nodes occur randomly and often unpredictably. Furthermore, in the case of mobile devices the infrastructure of systems are not persistent and the global data monitoring must be organized from local observation.

2.2 Wireless network routing protocols

Networks of wireless autonomous sensors for monitoring physical environments are a recent but very active application and research area. These networks, where the routing process is distributed to all the hosts, are called ad-hoc networks. If the hosts are mobile they can be called MANET networks for Mobile Ad-hoc NETWORK. The associated routing protocols are centered on the flooding techniques which consist in sending messages to all the members of the network to be sure the receiver gets the message: the associated power cost is very high.

There are different routing protocols to solve the problem of routing in (mobile) wireless networks. Generally they are a compromise between the control traffic reduction and the latency in finding the route to a destination. These protocols are divided into different families.

The *reactive* families are the on-demand protocols. These protocols never try to find a route to a destination before a message requires

transmission. A reactive protocol attempts to discover routes only on demand. One of the major advantage of this routing technique family is that the bandwidth is mainly used for data transmission. The Dynamic Source Routing protocol (DSR Johnson and Maltz, 1996), that we can see after, is one of the more popular and simplest protocol.

The *proactive* protocols use periodically updated routing tables and for those it is necessary to exchange many types of control messages for creating a “network model”. These messages enable a node to discover its neighborhood for example. The Destination Sequence Distance Vector protocol (DSDV Perkins et al., 1994) for example is one of the first protocols of this family specified by the MANET work group and it takes the RIP functioning principle. The Optimized Link State Routing protocol (OLSR Clausen et al., 2001) and Clusterhead Gateway Switch Routing protocol (CGSR Chiang et al., 1997) agregate the different hosts in clusters.

Hybrid protocols adopt the reactive protocol behavior and, if necessary, use routing tables for increasing efficiency. The majority of these protocols use a proactive or reactive scheme depending on the type of requirements.

3. Our solution based on the emergence

3.1 What should emerge?

Our objective is to decrease the energy expense induced by the inherent floodings techniques. For that we will use a group structure inspired by the clusters of the CGSR protocol. Our organizational basic structures are constituted by (see fig 2):

- one and only one *group representative agent* (r) managing the communication in its group,
- some *connection agents* (c) which know the different representative agents and can belong to several groups,
- some *simple members* (s) which are active in the communication process only for their own tasks (They don't ensure information relay).

With this type of organizational structure, the message path between the source (a) and the receiver (b) is $((a, r), * [(r, c), (c, r)], (r, b))$. If the source is a representative agent the first term doesn't exist. If the receiver is a representative agent the last term doesn't exist.

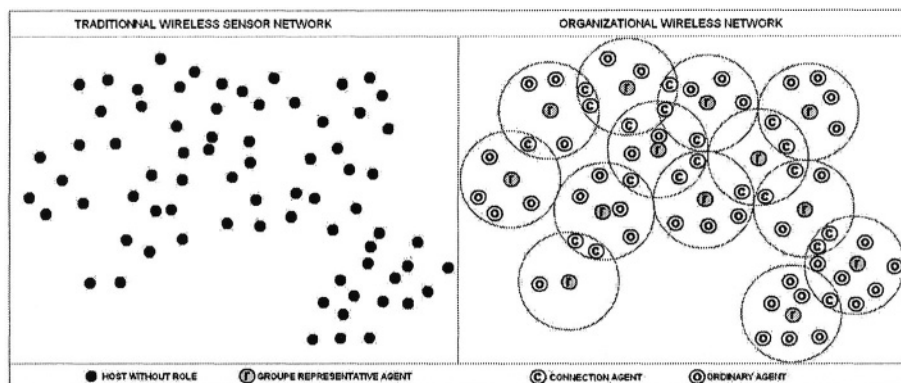


Figure 2. Our organizational structure

The energy saving comes owing to the fact that the flooding is only directed to the representative agent of the groups and to some connection agent. To give an order of idea, a receiver path research with flooding techniques will cost, in the case of a traditional wireless network, a number of emissions equal to the number of stations. In the case of a clustered wireless network, the number of transmitted messages are about twice the numbers of representative agent (all the representative agents are contacted via one connection agent). In our example (fig 2) the cost would be in the first case of 74 messages and in the second of 26 messages is an economy.

However, the networks with an organizationnal structure must take care of the maintenance of their routing table. Generally, the adaptive features of these tables come from periodical exchanges between the different nodes. In our approach we do not wish to use this technique to ensure the maintenance of coherence. Indeed, our principle will be “if we do not need to communicate, it is useless to spend energy to ensure the coherence maintenance”. However, we will thus use eavesdropping of surrounding agent communications. We extract from these messages exchange knowledge to update our beliefs about our neighbors. Moreover, our self-organization mechanism will integrate a energy management policy. These structures will thus emerge.

3.2 How to make the solution emerge?

It is necessary for us to wonder now how we will make emerge these structures. The multiagent methods aim at decreasing the complexity of

system design by a decentralized analysis. There are several multiagent system methods (Iglesias et al., 1998). We are thereafter going to be interested in the AEIO decomposition (Demazeau, 1995). We will follow the method of multiagent design discussed in (Occello and Koning, 2000), associated to this MAS decomposition. In fact, we chose to apply this multiagent method for our problem because it privileges an explicit description of the interactions and the environment.

It proposes a decomposition according to four axes collectively accepted today.

Agent axis. The *agent axis* (A) gathers all elements for defining and constructing these entities. It concerns the agent's know-how and knowledge, its model and its architecture. In our problem we use the ASTRO hybrid architecture (Occello and Demazeau, 1998).

Environment axis. The *environment axis* (E) deals with elements necessary for the multiagent system realization such as the perception of this environment and the actions one can do on it.

Interaction axis. The *interaction axis* (I) includes all elements which are in use for structuring the external interactions among the agents (agent communication language, interaction protocols). We defined thirteen different types of small messages.

- *WhoAreMyNeighbors* is used by an agent to know who its neighbors are. This message is transmitted when an agent is created (the first goal of a new agent is to know its neighbors) or when an agent feels that its neighbor table is not coherent with reality.
- *IAmOneOfYourNeighbors* : It makes it possible for an agent to answer the preceding request. With this message, it thus provides its identifier, its role and its membership group.
- *IChangeMyRole* : It is used by any agent to inform its neighbors that it decides to change its role. This message contains the agent identifier, its role and its membership group.
- *AskConnectionAgentGroup* : It is used by representative agents which want to update their knowledge on the close groups. It obliges the connection agent to answer.
- *AnswerConnectionAgentGroup* : It is used by a connection agent to announce to a representative agent the other representative agent that it can contact.

- *VerifyNeighborGroupConsistency* : It is sent by an agent, to its representative, which believes to have detected an inconsistency with a close group. There is an inconsistency between two groups when two agents of different groups see themselves and their representative cannot communicate with a short path (fixed by a time-to-live).
- *ConflictRepresentativeResolution* : It is used by a representative agent, in conflict with one or more other representative agents, for communicating its score. There are conflicts when two neighbors have a representative role.
- *ISuggestYouToBeRepresentative* : It is a suggestion given by a representative to one of the agents of its group. It can give this order for correct an inconsistency problem.
- *FindPacketPath* is used by a representative agent which wants to know the path (list of representative agents) to join another agent.
- *PacketPathResult* : It is the answer of the representative of the recipient of the *FindPacketPath* message.
- *ACKMessage* : It is a configuration message used to confirm to the transmitter that its message has arrived to its destination. These messages play a role in the coherence of the organization.
- *BadWay* : It is a message sent by a representative who has noticed a problem. This message takes the erroneous road and the organization verifies its consistency.
- *EncapsulatedData* : It is a message which encapsulates data.

Organization axis. The *organization axis* (O) allows to order agent groups in organization determined according to their roles. We have identified eleven different self-organization techniques.

The adaptation of our whole multiagent system is obtained through the emergence of organizational structures by self-organization based on role allocation modifications. The organization is built according to an exchange of messages between agents. The decision algorithm is very simple, in case of conflict a mechanism of election is applied according to some criteria (energetic level, number of neighbors...).

Relations between agents are going to emerge from the evolution of the agents' states and from their interactions. We are only going to fix the organization parameters, i.e. agents' tasks, agents' roles.

The ideal representative agent is the one having the most important number of neighbors and the most important level of energy. The level of energy is an important parameter in the sense that the representative agent is the most solicited agent in the group from a communication point of view. We use role allocation based self-organization mechanisms involving the representative agent election. Our election function integrates some data on neighbors and energy levels. This function estimates the adequation between its desire to be the boss and its capacity to access to this position. The organization is modified only when a problem occurs. We do not try to maintain it if we have no communication. In addition to the configuration messages, all agents use eavesdropping. In fact, when some communicating entities (humans, robots etc.) share a common environment they might intercept some messages (broadcasted or not). From this eavesdropping message they can extract some authorised information like the receiver, the sender, the type of message and the packet's path.

Our algorithm, presented below, can be adjusted by other agents' suggestions such as an organization inconsistency. Moreover, an agent can give up its role because its power level quickly fall or fall under a limit that the agent thinks dangerous for its integrity. So it can become a simple member.

```

IF neighborNumber $\neq$ 0 THEN
  * One has neighbors
  IF neighborRepresentativeNumber=0 THEN
    * None of our neighbors is representative: one decides to become to it. This
    * case intervenes when one has just created the agent or when he is isolate.
    * One does not proceed to a vote because one make the system unstable (the
    * sensor goes surely to carry on its path)
    myRole = REPRESENTATIVE;
  ELSE IF myRole = REPRESENTATIVE THEN
    * I am a representative agent too: I enter in conflict with the other applicants
    * to this role an election will take place and the agent with the best score will
    * remain in place.
    RepresentativeElectionProcedure()
  ELSE IF neighborRepresentativeNumber=1 THEN
    * One of our neighbors is representative: one subjects oneself to its authority
    * and his even if the organization is less effective than otherwise. One privi-
    * leges, for the moment, stability to performance in the organization. One will
    * await a failure or its wish to leave its mandate.
    myRole = SIMPLEMEMBER;
  ELSE
    * There are, in our vicinity, several representatives: one becomes connection
    * agent for these representatives
    myRole = CONNECTION
  ENDIF
ELSE
  * One does not have a neighbor: one has any more no role
  myRole = NOTHING
ENDIF

```

4. Results of our approach

It is significant when one is interested in self-organizing systems to think to the results evaluation problems. Indeed, as we noted previously, the self-organized systems are essentially fault tolerant and thus are particularly suited when one cannot plan all the situations which will occur. It is thus difficult to evaluate a self-organization method whose power is precisely to adapt to unforeseen situations.

In this part, we are thus going to give elements which will be used as a basis for the evaluation of self-organized system. We will initially be interested in the observability of these systems then, in the analysis of these observations.

4.1 Observation

We can distinguish three observation levels for agents organization: An external level of the multiagent system (case 1, picture 5) considering the system as a black box (we can observe only input/output), an internal level of the multiagent system (case 2, picture 5) if we focus on interactions between agent society, an internal level agent (case 3, picture 5) considering the agent and its architecture.

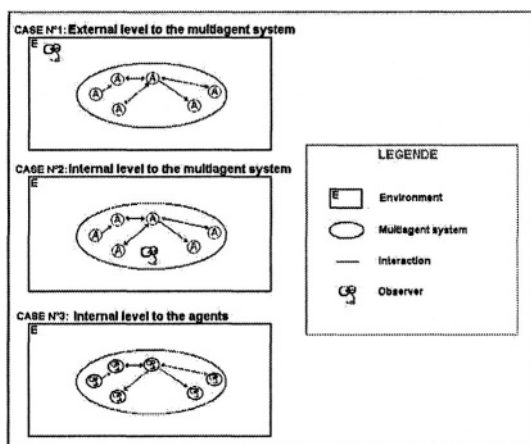


Figure 3. Observation of agent organization

In self-organized multiagent systems, the stable states are an emergence of structure: we want to identify, to observe, these structures. We want to focus on global energy expence generated by messages exchange: the adapted level of observation is so the internal level of the multiagent system (case 2).

4.2 Results analysis of self-organization

First, we want to evaluate the self-organization without taking our application into account. Generally, a process of self-organization must be stable, sensitive and convergent (Groupe MARCIA, 1996) at the same time to be interesting and exploitable:

- **Stable:** the stability of a system refers to its constancy in time. The system must thus be stable in order to highlight persistent structures which would change only under the influence of internal or external disturbances.
- **Sensitive:** the sensitivity refers to its capacity to make a structure evolve into a partial or total recognition state according to whether all the selected criteria were satisfied or not. It is thus necessary that the process of self-organization be sensitive, so that it is possible for the emerging structures to be questioned.
- **Converges:** the convergence of a system reflects its capacity to evolve to known structures. The system must thus be convergent so that one leads to new structures.

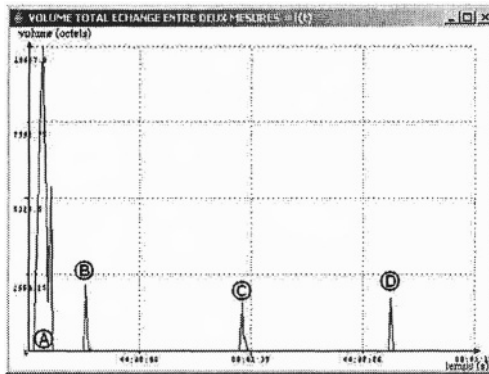


Figure 4. Message volume exchanged for self-organization

Figure 5 shows the stability, the sensitivity and the the convergence of our self-organization process. The stable state are between the 4 evenements (A: Self-organization of 300 sensors, B: Re-organization to correct inconsistency, C: Adding 10 sensors, D: re-organisation to correct an error detected by eavesdropping in a receiver search).

In a second time we take into account our application to quantify the correlation between our aim and the result of the self-organization process.

As we saw previously, this part of the analysis consists in quantifying the adequacy between the process of car-organization conceived and the results which were expected. It will thus be necessary for us to determine if the system is *valide* and/or *pertinent*. These two properties will be correlated with *interest* and the *simplicity*. Let us clarify these terms:

- the system is known as *valid* if the result produced by the self-organized process is in conformity with the expected one.
- the system is known as *pertinent* if the structure can be observed as easily as its implementation and as the comprehension that an observer would have of it.

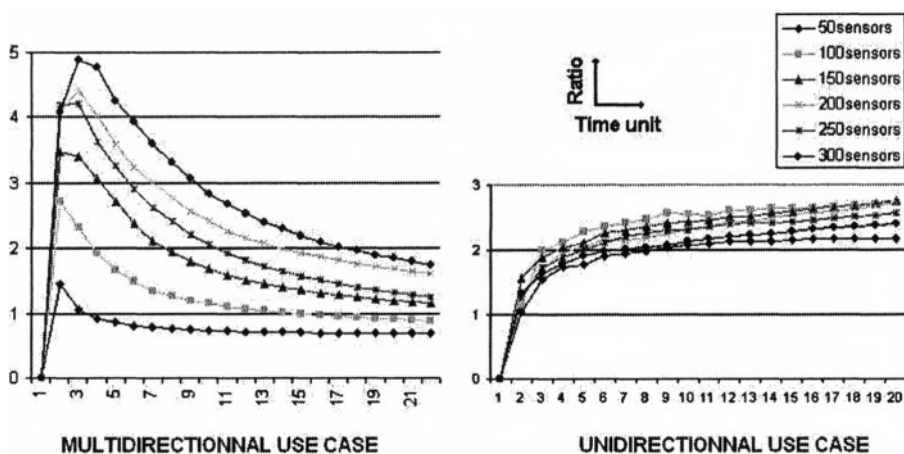


Figure 5. Ratio ASTRO efficiency divided by DSR-ROUTAGE efficiency

Figure 5 shows that in most of case our approach is efficient as expected, the validity is so proved.

Each agent local algorithm managing the role selection presents a very low complexity due to the decentralization. The global structure constituted by the agent's localization can be easily observed and understood. The approach can be qualified of pertinent according to the definition given previously.

Conclusion

We presented in this paper a hybrid software/hardware application for the management of an environmental sensor network. We proposed a multiagent analysis of this system and detailed how we use collective features to make the system adaptive. The innovative aspect of this

work stands in the use of self-organization techniques based on emergence of structure. The concept of emergence is usually quite difficult to defend in an applicative real world context. We have made an effort to show through quantitative results the validity and the pertinence of the approach. This paper wants to contribute, by this way, to show that artificial intelligence mechanisms as self-organization can lead to interesting results and can improve classical techniques.

References

- Chiang, C., Wu, H.-K., Liu, W., and Gerla, M. (1997). Routing in clustered multihop, mobile wireless networks. In *Proceedings of IEEE Singapore International Conference on Networks*, pages 197–211, Singapore.
- Clausen, T., Jacquet, P., Laouti, A., Muhlethaler, P., Qayyum, A., and L.Viennot (2001). Optimized link state routing protocol. In *Proceedings of IEEE International Multitopic Conference - INMIC'2001*, Pakistan.
- Demazeau, Y. (1995). From interactions to collective behavior in agent-based systems. In *European Conference on Cognitive Science*, Saint-Malo France.
- Groupe MARCIA (1996). Auto-organisation := évolution de structure(s). In *PRC-GDR Intelligence artificielle*, pages 139–152. Hermes.
- Iglesias, C., Garrijo, M., and Gonzales, J. (1998). A survey of agent oriented methodologies. In *Proceedings of Workshop on Agent Theories, Architectures, and Languages*, volume LNAI 1555, pages 163–176, Paris, France. Springer-Verlag.
- Jamont, J.-P., Occello, M., and Lagrèze, A. (2002). A multiagent system for the instrumentation of an underground hydrographic system. In *Proceedings of IEEE International Symposium on Virtual and Intelligent Measurement Systems - VIMS'2002*, Mt Ayleska Resort, AK, USA.
- Johnson, D.-B. and Maltz, D.-A. (1996). Dynamic source routing in ad hoc wireless networks. In Imielinski, T. and Korth, H., editors, *Mobile Computing*, pages 153–181. Kluwer Academic Publishers.
- M.R. Jean (1997). Emergence et sma. In *JFIADSMA'1997 - Journée Francophones IAD et SMA*, Nice. Hermes.
- Muller, J.-P. and Parunak, V. D. (1998). Vers une méthodologie de conception de systèmes multi-agents de résolution de problème par émergence. In *JFIADSMA - Journée Francophones IAD et SMA*, Pont--Mousson, France. Hermes.
- Occello, M. and Demazeau, Y. (1998). Modelling decision making systems using agents for cooperation in a real time constraints. In *Proceedings of 3rd IFAC Symposium on Intelligent Autonomous Vehicles*, volume 1, pages 51–56, Madrid, Spain.
- Occello, M. and Koning, J.-L. (2000). Multi-agent based software engineering: an approach based on model and software reuse. In *From Agent Theory to Agent Implementation II - EMCSR 2000 Symposium*, pages 645–657, Vienna.
- Perkins, C.-E., Royer, E.-M., and Das, S. (1994). Highly dynamic destination-sequenced distance-vector (dsv) routing for mobile computers. In *Proceedings of Special Interest Group on Data Communication conference- ACM SIGCOMM'94*.
- Steels, L. (1990). Cooperation between distributed agents through self-organisation. In Demazeau, Y. and Muller, J.-P., editors, *Decentralized AI*. Elsevier Science Publishers B.V. (North Holland).