Dynamic Management of Cooperative Applications for Mobile Systems*

Pierre-Guillaume Raverdy , Phillipe Darche†, Bertil Folliot‡
Laboratoire MASI, Institut Blaise Pascal, Université Pierre et Marie Curie
4 Place Jussieu, 75252 Paris Cedex 05, FRANCE
Tel: +33.1.44.27.34.23 Fax: +33.1.44.27.62.86
E-Mail: {raverdy,darche,folliot}@masi.ibp.fr

Abstract

Management of nomadic computers (either portable laptops or mobile robots) in actual networks brings new challenges for efficient resources access and cooperative work. New systems have to cope with communication breaking of. Systems should also manage resources mobility (due to hardware location change or cell change for wireless communication), that may lead to prohibitive resources use due to communication delay/cost. We propose a new resources management scheme that takes these new challenges into account and apply it in a robotic environment.

Résumé

La gestion d'ordinateurs nomades (robots mobiles ou ordinateurs portables) dans le cadre de réseaux classiques de communication amène de nouveaux problèmes pour la mise à disponibilité des ressources réparties sur le réseau et le travail coopératif de plusieurs entités. Il est d'abord nécessaire de gérer les ruptures de communication, soit volontaires dans le cas d'une deconnection, ou impromptues dans le cas d'un passage dans une zone d'ombre pour les réseaux cellulaires. Il faut de plus tenir compte des déplacements de ressources (déplacement d'un robot mobile, changement de cellule de communication) qui peuvent rendre prohibitif l'accès à une ressource en raison du cout de communication (dans le cas d'applications temps-réel). Nous proposons une gestion de ressources prenant en compte ces différents facteurs et l'appliquons à un environement robotique.

*A version of this paper appeared in *The Information Revolution: Impact on Science and Technology*, J.E. Dubois & N. Gershon editors, Chapter 3: Distributed Data and Information Networks, pp. 141-148, Springer-verlag, 1996

†Affiliation: UFR d'Informatique, Université Paris V

‡Affiliation: UFR d'Informatique, Université Paris VII

Introduction

Introducing mobile computers in classical distributed systems brings new challenges for resources and cooperating applications management. Due to mobility, the location of hosts, applications and resources may change. A mobile host needs to continue its work while disconnected and to connect transparently to the network from various locations. The obligations to cope with batteries limitations and the limited bandwith of wireless communications are also important for cooperatives applications. For these reasons, management of system resources must be dynamic and new management schemes have to be provided for nomadic computing environment.

Due to hosts' mobility, location of resources and processes may change. Thus, the system management must be dynamic.It must support mechanisms to allow disconnection, ensure information consistency during disconnection and allow cooperating programs to resume their communications after host reconnection. Change of location also influences the global system performances by introducing load inbalance due to communication delay. It can be efficient for processes to migrate in order to reduce communication and response time. After a reconnection, processes located on a mobile host needs to locate the remote resources they were acceeding. Resources managed by this host should also be made visible to other hosts (mobile or not).

We propose a cooperative applications management, the territory scheme for a network of mobile robots based on the actor model [1]. Every network node(host) is a hardware actor that executes software actors (processes). Each hardware actor is an autonomous communicating entity. The system is composed of a set of system servers and of users' processes. An application is composed of cooperating processes using different resources and possibly executing on different hosts. Independent applications execute on separated territories, composed of one or more hosts.

The first section presents the nomadic computing model with the wireless communications, robotic computers, and resources mobility. Section 2 describes our resources management schemes based on the territory model. Section 3 presents our Actors Network for nomadic robots: ActNet. We conclude this paper in Section 4.

1 Cooperative Applications in Mobile Systems

In this section, we present our mobile system, with firstly the mobile computing model, and secondly the cooperative application model.

1.1 Mobile Computing Model

In our model, we consider, as mobile computers, computers that are able to move while communicating (mobile robots) and, as nomadic computers, computers that need to be disconnected from the wired network to change of location (notebooks). We first present communication architectures for mobile computers and then the constraints of this new technology.

1.1.1 Wireless Communications

The channel used in mobile communication is the atmosphere. This channel is an extremely noisy one and is subject to various interferences (scrambling, etc..). Information is carried by a wave characterized by its frequency. Three frequence domains are mainly concerned: the visible domain with a coherente laser transmission, the infrared domain and the radio frequency domain. Laser links are rare and they are mostly used for static point-to-point communications. Due to high attenuations, infra-red communications are reserved for short distance links, a room or an office. Radio-frequency links concern short to long distances. Thus, only these two last communication domains are used in mobile systems.

The used topologies mainly of three types: bus, star, and ring. The topology is static. Wireless Local Area Networks (WLAN) use in most cases the same channel and the access type is, in general, Pure-Aloha type or Slotted-Aloha type [2]. Star topology is mostly reserved to long distance communications (e.g. satellite), and frequency channels are divided between several users. The communication zone is divided in cells, in order to limit transmission power and to avoid interferences. A data concentrator by cell allocates one or several frequencies to one mobile, used for user data and management data. This allocation does not change during the communication if the mobile stays in the same cell. Cells are linked between them by specialized wired links (see Figure 1).

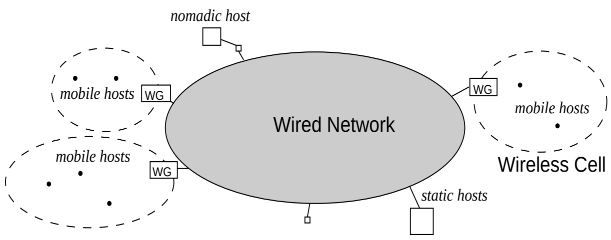

Figure 1: Mobile Networks

1.1.2 Mobile Computing Constraints

Mobile communication is characterized by the following criteria:

- energetic autonomy,
- transmission reliability,
- communication speed,
- distances bringing into play,
- and cost.

Embedded devices require to be equipped with electronic module using minimum current, because energy is limited (battery). We call that the energetic autonomy criteria. It is primordial because it has a direct effect on the operation duration and the transmission power (the maximal distance of communication). The transmission power is generally low (some watts) and in spite of the bringing into play of efficient amplification types (C, D in particular), the maximal distance of communication is lower than some kilometers, essentialy due to the channel attenuation. The energetic autonomy which depends of the electrical consumption is of the order of some hours.

Reliability criteria is more difficult to solve because the type of used channel, in our case the atmosphere, is accessible to all people and therefore could be disturbed (intentionnaly or not). Wired links are more reliable by definition in spite of electromagnetic interference like crosstalk or attenuation. It also be subject to intentionnal access (listening, disruption or intrusion of false data).

The communication speed is higher for a wired link (100 MBits/s) than a wireless one (10 MBits/s), in spite of compression and efficient modulation mechanisms. Moreover, the set of frequency channels is limited in number and in bandwidth.

A system for mobile networks has to take into account the numerous variable pertubations of the communication medium and its limited performances.

1.2 Mobile Environment Model

The base of our system is the actor entity. Actor are used by cooperative applications to execute their processes and to manage both mobility and resources access.

1.2.1 Actor Model

Our system is based on the actor model [1]. An actor can be described as an autonomous active entity which uses messages to communicate. An actor can only be acceded through its message queue. A main characteritic of an actor is its infinite lifetime.

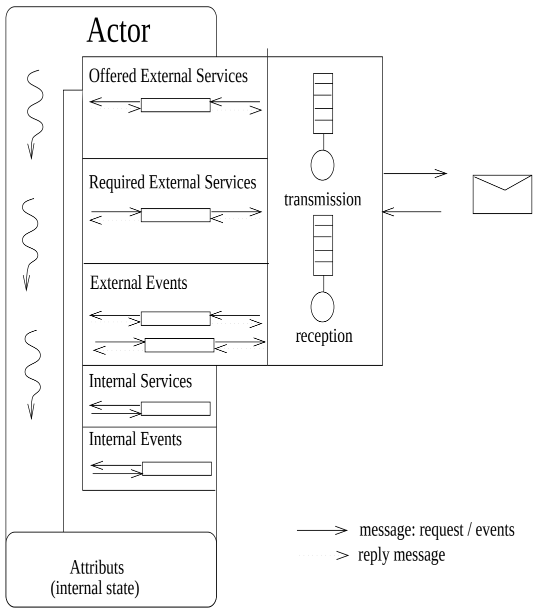

Figure 2: Actor

Others actors can send messages to the actor's message queue. These message will be processed, and according to the externel services offered, the actor can send messages, creates new actors or execute threads in its space (see Figure 2).

Actors offer a uniform communication scheme between all the entities of our mobile system, hardware or software, static or mobile. This allows to implement transparent services needed by mobile hosts such as disconnection, automatic localization and coherency management while disconnected.

1.2.2 Cooperative Applications Model

In our model, we consider independant applications composed of communicating processes. These processes use resources distributed over the whole network. These resources can be sofware (file) or hardware (printer) or more specific one (motor) and can be located on static or mobile hosts. With this model, message exchange occurs mostly between processes of the same application or between a process and the manager of an acceded resource. So, we have independant applications.

1.2.3 Resources in Mobile Systems

Cooperative applications use a set of resources distributed over the global network. After a reconnection, actors located on a nomadic host needs to locate the remote resources they were acceeding. Resources managed by this host should also be made visible to other hosts.

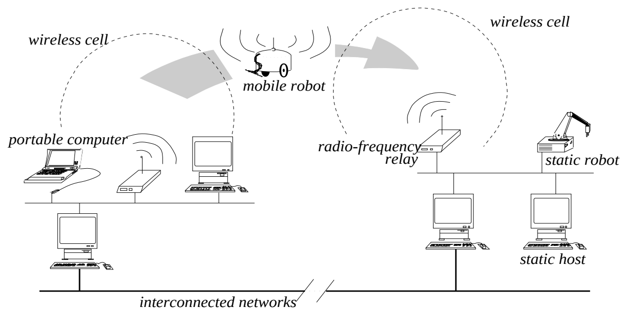

Figure 3: Mobility between two wireless cells

To define the location method of resources after a reconnection, we classified resources in three types:

- **Static resources** that have a specified location on the network. An actor using these resources still uses the same ones after it migrates or its host moves (i.e printer device).
- **Duplicated resources** that have fixed location but are duplicated on the network. After the actor migration (due to mobile hosts or load balancing), the actor will use the closest version of the resource accessible to improve performances (i.e system files).
- **Mobile resources** that can migrate with an actor to better answer the application need and improve their performances (users' files).

After reconnection, depending on the kind of resource an actor wants to accede (static, duplicated or mobile), it calls the resource location, a manager on the network to find a copy, or asks the resource to migrate.

2 Territory Scheme for Resources Management

In this section, we present our dynamic management, the execution territory scheme, which allow an efficient use of the resources distributed over the network and cooperative work between static and mobile units.

Cooperatives application management in a local network of static hosts already set many problems like load sharing, fault tolerance and efficient resources access. Numerous studies have been completed on this problems: Utopia [3], Charlotte [4] and Gatos [5] for load balancing, Coda [6] for disconnected file operations, Amoeba [7], Star[8] for fault tolerance. and object management [9].

This management is even more complex in the case of mobile computers [10] due to:

- frequent disconnection of mobile computers,
- resources duplication to allow disconnect work,
- reconnection after a location's change,

2.1 Territory Management Scheme

In our system, an execution territory is associated to each cooperative application.

We introduce the **Execution Territory** (ET) to allow an efficient control of cooperative application processes. An ET, associated to an application, is composed of a set of hosts on which application's processes are executing. When an application starts, a territory is affected to it and evolves according to hosts load. On this territory will be performed processes migration to solve load inbalance introduced by processes execution.

Each ET is managed by a **Territory Manager** (TM) in charge of data gathering from the hosts belonging to its territory (e.g. inter-process communications, host loads, resources utilization). The TM, is also in charge of keeping application informations during disconnection.

With these informations, the TM maintains an execution graph of the application and is able to detect any load inbalance due to an overloaded host or inefficient resource access. Statistics are only gathered at the territory level by the TM, so that each manager has a local view of the system state.

All the TM are linked with a virtual ring, allowing them to exchange management information. The processes are acceding the resources thru resources managers (RM) dispatched over the network (see Figures 4 and 5).

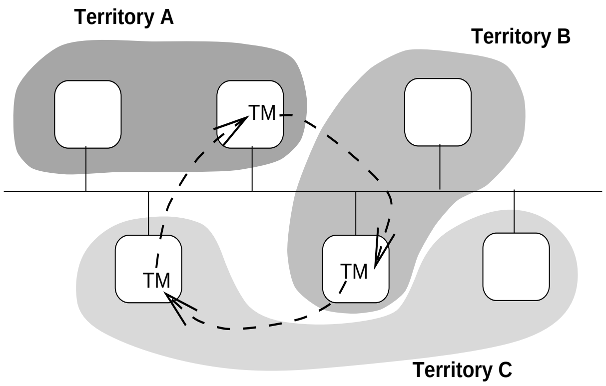

Figure 4: Territory Scheme

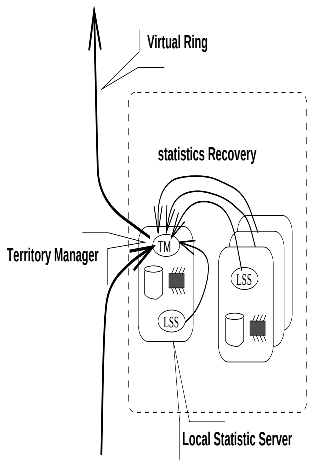

Figure 5: Territory management

2.2 Processes and Resources Migration

An application can be described by a dynamic execution graph that represents resources utilization by the different processes of the application. When a mobile host moves, it can create a load inbalance due to the executing processes or system resources on the mobile host (e.g. a process on the mobile host was formely using a close resource located on a static host. After location change and due to network latenty, the request on this resource can be drastically costly). Load inbalance can also be the result of application evolution (e.g. start of new processes)

For load balancing, the application territory can proceed to processes migration, and in case of resources, can ask for its migration or for the use of another instance (if duplicated) by sending a hint to the resources manager (RM) of the resource. Resources or processes migrations are decided according to a dynamic execution graph by using attraction/repulsion mechanisms [11]. This graph represents the resources utilization for each process of the application and the communication between these processes.

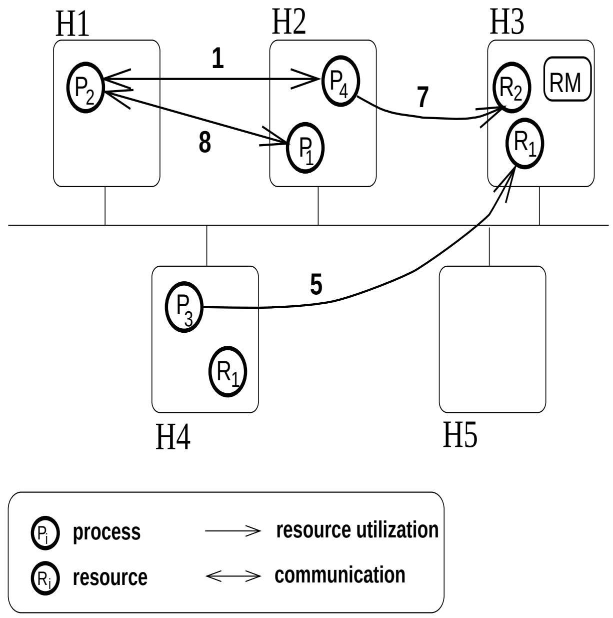

Figure 6: Execution graph (1)

In the execution graph, there are two kinds of multi-valued node, one representing the different hosts (nomadic or static), the other representing the actors. These valuations correspond to host characteristics and utilization (resp. mobiles and duplicated resources utilization). The execution graph nodes are also connected by two kinds of multi-valued link representing the use of static resources located on a host or the communication between two application's actors.

If the load inbalance is due to application processes, the execution territory can gain or loose some hosts. In the case of inefficient resource access, and depending of the resource (mobile, duplicated or static), the territory will also evolve.

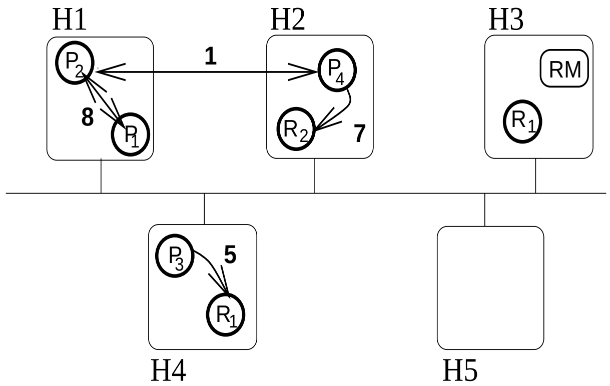

Figure 7: Execution graph (2)

Figure 6, an execution territory composed of three hosts, H1, H2 and H4. After statistics recovery, the TM detects that host H2 is overloaded and that execution of process P4 (resp. P3) is slowed by access to resource R2 (resp. R1). Because processes P2 and P1 have a high communication level, the TM decides to transfert process P1 onto host H1. And send request to the RM of R2 (resp. R1) with a hints about resource access. With these hints, and according to others access on these resources, the RM decides that resource R2 should be transfered on H2 and that process P3 should accede another instance of P1 (see Figure 7).

2.3 Disconnection Management

The territory scheme allows a mobile host to easily reconnect by contacting the last location of its TM. If, during disconnection time, the TM has migrated, the contacted host will ask its own TM to search the new location of the mobile host TM.

The TM is aalways located on a static host on the network. It s role is to keep information about resources located on mobiles hosts. So, in case of failure, it will be able to unlock resources. It s second role, but also a major one is to provide all information needed by a mobile host at reconnection time. La reconnection d'un site mobile implique:

- reprendre les communications entre les processussur le site mobile et ceux sur le reseau (etablissement connexion, traitement messages en attente, continuation comm.)

- mise a jour des localisation des ressources (mise a jour au niveau des serveurs responsables).

- calcul du graphe d'execution, equilibrage de charge, acces aux ressources les plus efficaces.

pour cela, la gestion par territoire d'execution semble la plus adequatte. En effet, le site mobile connait la position du GT lors de la deconnexion. Lorsqu'il se reconnecte, il contacte alors son GT aui peut proceder aux pprecedentes etapes (gestion/controle centralise des applications). Si jamais, a la suite de l'evolution du systeme, le gt n'est plus localise au meme endroit, il peut etre retrouve car tout les GT commuiquent sur un anneau virtuel. Soit l'ancien site est un GT, soit il est un esclave connaissant son GT. Donc il y a un premier forward du message de reconnection vers le GT du nouveau territoire, puis un parcours sur l'anneau virtuel pour retrouver le bon GT. apres cette etape, la reconnection peut alors commencer.

3 ActNet: a Robotic Network

As the control structure in an actors model is message passing, the actors communicate through the medium of a local network named "ActNet" for "Actors Network".

ActNet is a robotic testbed for new hardware actors models. This system allows to design and test actors based applications [12].

From a hardware point of view, ActNet is a heterogeneous actors network, which can be splitted between hardware actors and software actor embedded by hardware actors

In our model, we have characterized two type of hardware actors:

- Static Robotic Actor (SRA),

- Mobile Robotic Actor (MRA).

Executive sub-networks of moving actors (M.A.S.N.) and sedentary actors (S.A.S.N.) are associated to each type of hardware actor, and Gateways Actors manage communications between sub-networks. Moreover, a users' subnetwork (U.S.N.) is dedicated to the development and control of users' applications (see Figure 8 and 9).

Thus, the network is heterogeneous by its topology (programmable: M.A.S.N, ring or bus), but also by its kind of processors equipped every node. The USN is a local commercial network with a variable topology defined by computers supporting the developments of the users. In this study, it is reduced to a Personal Computer equipped with an interface board for SASN The SASN is a double ring, with opposite directions,

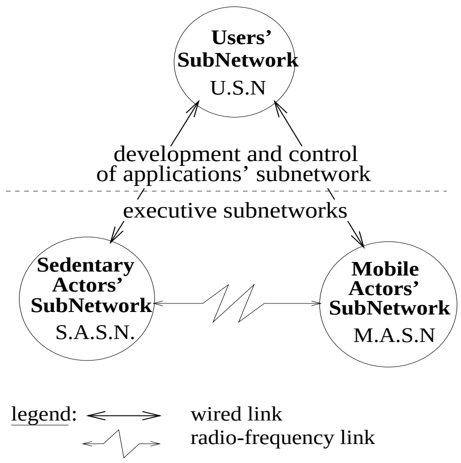

Figure 8: ActNet Topology

to increase reliability and insure a high and warranted speed for the flow of data.

The MASN is a radio frequency network (V.H.F.) to guarantee the autonomy of mobiles with communications. We want an asynchronous full duplex communication. the modulation speed is the one of a serial RS232C type link equipped the control board, and this one is setting from 1200 to 19200 bauds. A channel coding is needed to adapt the information. To achieve a duplex communication, it needs to spot the informations coming from several interlocutors. An analogue modulation will be used. The frequencies' space will be cut in identical bands called radio-frequency channels and two channels will be allocated to each actor of the MASN, which will be therefore characterized by two frequency addresses, one for the emission and one for the reception. For spectrum occupation and reliability, a frequency type (F.M.) discrete angular modulation, the binary minimum frequency shift keying modulation (B.M.F.S.K. modulation) has been chosen, because it is optimizing the channel bandwidth necessary for the transmission of a digital signal, without an elaborated source coding. The use of digital technologies (Phase Lock Loop particularly) allows us to compute sending and receiving frequencies of each mobile. The use of an antenna's duplexer ensures full duplex communications, but because of duplex spacing between the emission and reception frequencies, it must be done to define two distinct frequency bands B1 and B2.

The wanted maximum modulation speed and the employed type of modulation are defining the channel width (50 Khz). So these technical choices enable us to obtain the best fitted topology of computable network for the application of the user (bus, ring, star, etc..). A particular working of MASN is deduced from faults

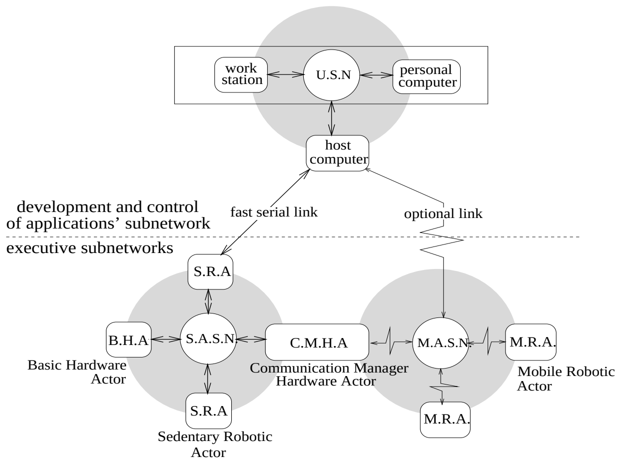

Figure 9: ActNet Topology: Details

tolerance which has guided our study. The communications by wire with mobile robots is only considered in the following cases:

- when a failure occurs in the embedded energy supplying system; an external energy supply is necessary for the electronic devices and for recharging the batteries; the power line of the network then fulfills this role,
- in the implementation phase for fast teleloading of a software prototype or for debugging applications,
- when failure of the radio frequency system occurs.

4 Conclusion

The use of nomadic computers implies more secure and flexible operating systems because both applications and resources are mobile. To control the application needs and then decide which process to migrate or which resource to use, we introduce the application territory. This allows to dynamically benefit from all available resources according to the applications needs.

Cooperative applications management is done at two level. Inside an execution territory for an independant application where a centralized scheme is used, and on the global network with a distributed scheme to equilibrate resources utilization between applications. One major advantage of this model is its scalability (territory size is independant of the global network size). Having a centralized algorithm allows to take better decisions for application managemnt and help mobile computers to locate formely acceded resources or communicating processes when reconnecting.

We currently implements our system for the robotic actors network **ActNet** [12]. This network is representative because it includes mobile entities (mobile robotic actors) and static ones (static robotic actors, workstations).

References

[1] Carl Hewitt. Viewing control structures as patterns of passing messages. *Artificial Intelligence*, 8(3):323–364, June 1977.

[2] N. Abramson. Development of the ALOHANET. *IEEE Transactions on Information Theory*, 2(32):119–123, March 1985.

[3] Songnian Zhou, Xiaohu Zheng, Jingwen Wang, and Pierre Delisle. Utopia: A Load Sharing Facility for Large, Heterogeneous Distributed Computer System. Technical Report 257, Computer Systems Research Institute, Universite de Toronto, Canada, April 1992.

[4] Yeshayahu Artsy and Raphael Finkel. Designing a Process Migration Facility: The Charlotte Experience. *IEEE Computer*, 22:47–56, September 1989.

[5] Bertil Folliot. *Méthodes et outils de partage de charge pour la conception et la mise en oeuvre d'applications dans les systèmes répartis hétérogène*. PhD thesis, Université Paris VI, December 1992.

[6] Mahadev Satyanarayanan, James J. Kistler, Puneet Kumar, Maria E. Okasaki, Ellen H. Siegel, and David C. Steere. Coda: A Highly Available File System for a Distributed Workstation Environment. *IEEE Transactions on Computers*, 39(4):447–459, April 1990.

[7] Andrew S. Tanenbaum, Robbert van Renesse, Hans van Staveren, Gregory J. Sharp, Sape J. Mullender, Jack Jansen, and Guido van Rossum. Experiences with the amoeba distributed operating system. *Communication of the ACM*, 33:46–63, December 1990.

[8] Pierre Sens. STAR: a Fault-Tolerant System for Distributed Applications. In *Proceedings of the fifth IEEE symposium on Parallel and Distributed Processing*, pages 656–660, December 1993.

[9] Peter William Dickman. *Distributed Object Management in a Non-Small Graph of Autonomous Networks with Few Failures*. PhD thesis, University of Cambridge, september 1991.

[10] B. R. Badrinath, Arup Acharya, and Tomasz Imielinski. Impact of mobility on distributed computations. *ACM Operating Systems Review*, 27(2):15–20, April 1993.

[11] Mario Tokoro. Computational Field Model: Toward a New Computing Model/Technology for Open Distributed Environment. In *2nd Workshop on Future Trends in Distributed Computing Systems*, Cairo, September 1990.

[12] Phillipe Darche and Gerard Nowak. *ActNet: A Heterogeneous Network of Actors for Learning of Parallelism, Communication and Synchronization*, volume 116 of *Control Technology in Elementary Education, NATO ASI Series F*, pages 289–307. Springer Verlag Berlin, 1993.