

# A MOBILE AGENT ARCHITECTURE TO ENABLE COLLABORATIVE INFORMATION SYSTEM SURVIVAL

---

H. Mathieu, F. Biennier<sup>1</sup>

(1) INSA de Lyon - Laboratoire PRISMa / IF502 - F69621 Villeurbanne Cedex, FRANCE  
{Herve.Mathieu, Frederique.Biennier}@insa-lyon.fr

*Economical constraints impose to enterprises to regroup themselves into virtual enterprises to be more reactive. This association will create interdependencies between virtual enterprise processes, which can be a problem in case of a process failure, because such an issue can jeopardize the whole activity of the virtual enterprise. Thus, it seems relevant to set « survival » mechanisms enabling to improve processes reliability. First, we propose a method to determine how the information system resources supporting the processes are used. Then, we define the architecture of a system in charge of managing process survival. We designed the architecture so that it is adapted to a large scale, heterogeneous and dynamic context, which characterizes virtual enterprises. We validate our approach with a prototype giving its first results.*

## 1. INTRODUCTION

In the actual economical context, which is more and more complex and changing, it is necessary to form virtual enterprises (VE) to cope quickly and efficiently to the market constraints. Despite of the intrinsic interest of collaboration, setting collaborative business is not obvious (Bremer et al. 1999; Wildeman et al. 1996). A real collaboration is more than just exposing manufacturing processes and services to an outside partner. It must integrate also cross-enterprise collaboration so that partners integrate their resources for mutual gain. However, the cooperation between the different VE entities is also the source of important problems. It is important to take into account that federating the different Information Systems (IS) of the entities composing the VE causes interoperability issues between the IS components, and can deeply impact on the quality of offered services.

IS Infrastructures availability will completely determine the quality of offered services. Thus, infrastructures availability is fundamental because one IS component dysfunction can jeopardize the whole VE's activity, which is not acceptable. To reach a good quality of the global IS, it is first necessary for the infrastructure to be

« available » to support VE's business processes, even if some IS resources are corrupted or not operational. Therefore, it is important to set redundancy mechanisms inside the IS. Thus, in case of resource corruption, tasks performed on the corrupted resource can migrate to a « safe » resource, so that it becomes possible not to interrupt the processes, even in case of failure. Because of the growing IS complexity, of their instability and constant evolution, it is not possible to secure completely an IS to cope with both internal and external threats. Considering this problem, risk management and "survival" capabilities are interesting, because they enable to define and to set mechanisms to assure service continuity, even in case of attack or failure of some IS resources. The problem of these methods is that they are not implemented for now in unstable connectivity, great scale, and heterogeneous environments, which are properties characterizing modern IS. We propose in this paper a method to establish the sequence of tasks implied in business process execution, which enables to establish a chart of the IS resources use. Then, we describe the architecture of a survival system based on the mobile agent paradigm. We present our first results on a virtual enterprise prototype, whose entities share a common research and development process.

## 2. CONTEXT

In a virtual organization, some entities a priori independent form an association to perform a common project. This association impacts deeply on the IS of each entities because the different IS have to merge, at least partly, to support global collaborative processes. This « globalization » imposes constraints in terms of management of the services quality, in terms of interoperability, and in terms of processes safety. In this paper, we focus mainly on this last point. During the collaboration, one of the collaborating entities can experience some problems and dysfunctions. Thus, its information system can be the target of attacks, or some services can fall due to technical issues. However, the virtual enterprise still must support its activities, even if one of the entities does not work properly. This kind of problem is important in a conventional enterprise, but it is much more critical in enterprise networks, because a problem affecting one entity can block the activity of the whole organization. In such a context, it is important to settle mechanisms improving the reliability of the shared processes during their exploitation. This is a crucial issue to solve to improve the reliability of cooperating networks.

The interaction between business processes, tasks and resources can be represented as a model organizing sequentially the tasks being performed by processes as shown on figure 1. These tasks use some resources of the distributed IS. The resources implied by a task performance change in function of the task being performed.

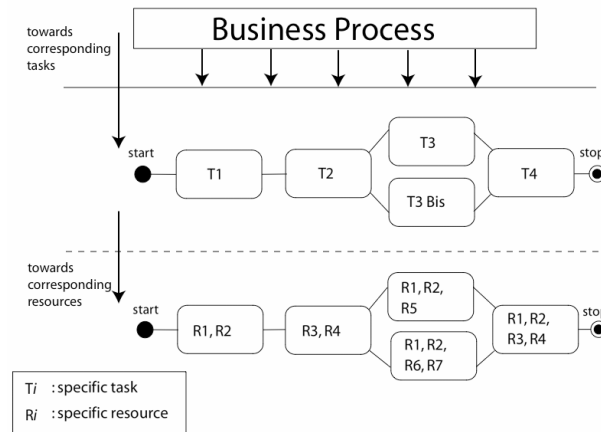


Figure 1 – Business Processes use different resources of the distributed IS system to execute themselves

To improve Business Processes reliability, we can try to settle rescue resources that are able to support services that are no more supported by a deficient resource. This simple idea arises numerous concerns. It is first necessary to be reactive to problems concerning the availability of resources used by the executing Business Process. This point is not obvious in a centralised architecture, but it can become very complex in a highly distributed architecture, such as Information Systems of virtual enterprises. Thus, this kind of architecture can be of large scale, of unstable connectivity, and heterogeneous.

Some workflow and business process management systems based on mobile agents have been proposed (Suh et al., 2001; Budimac et al., 1999, Loke and Zaslavsky, 2001). These models enable to negotiate tasks constraints (precedence, execution conditions...) thanks to mobile agents. This approach is very encouraging in a highly distributed environment. In fact, we can consider the distributed IS as a set of interconnected nodes which will evolve according to the alliances or separations between entities of the VE. In such a case, trying to give survival capabilities to the distributed IS can be considered as giving survival capabilities to a set of interconnected nodes, this set being dynamic, heterogeneous, and presenting unstable connectivity characteristics. In this context, which is highly distributed, it seems convenient to adopt a survival system based on an intrinsic distributed paradigm. Mobile agents enable to work around the limits imposed by centralized environments in terms of infrastructure management (Tomarchio and Puliafito, 2000). The use of mobile agents proved its efficiency to solve key problems bound to distributed infrastructure such as packet routing in dynamic network infrastructures (Kramer et al., 2000; Caro and Dorigo, 1998), or intrusion detection (Fenet, Serge, 2001). Even if the mobile agent use seems adapted to the context, it is necessary to structure the way they are exploited. As underlined in (Schulz et al., 2004), to make the best of the mobile agent paradigm, it is necessary to make them evolve in a well-defined frame. In our context, the specific survival services act

independently from each other. Thus, it seems relevant for us to use several kinds of mobile agents, each one dedicated to the management of one of these specific services. These mobile agents perform all the data collection and modification related to the survival service they are bound. The servers are interconnected and watch themselves. In case of failure of one server, any other server can be elected to replace the corrupted server. Our prototype use the “aglet” platform (SourceForge, 2006) to implement the survivability management system: it is an operational platform, fairly extensive and free, which suits very well to our prototyping constraints.

### **3. ARCHITECTURE OF THE SURVIVAL MANAGEMENT SYSTEM**

#### **3.1 Architecture**

We propose here an approach based on the mobile agent paradigm, to settle survival mechanisms inside the collaborative IS, so that it becomes possible to improve process reliability, even if some IS resources get corrupted. The functionalities of our « survivability » management architecture are implemented by the following mobile agents:

- inventory and planning agents: they are in charge of roaming the network to discover the equipments (nodes), their configuration, services and applications they are hosting. Besides, they determine the nodes which can potentially host applications and services located on other resources, to provide redundancy in case of failure.
- pre-migration agents: these agents are in charge of settle mechanisms to make easier the application or service migration working on a corrupted resource.
- detection agents: they are in charge of roaming the network to control that strategic configuration files have not been modified unexpectedly and to control that the nodes are operating properly, by performing elementary verifications, at the system level as well as at the application level. If a problem is detected, the central server is warned and will launch immediately migration agents enabling the applications and services executing on the corrupted node to migrate toward a safe node.
- migration agents: they are in charge of performing the migration of data and applications from a corrupted node to a safe node.

We developed a prototype implementing this architecture on a small virtual enterprise. To give applications the ability to migrate from one IS node to another is important, but it is also important to know the application transfer duration, to limit the time of service interruption. Knowing this transfer duration will impact significantly the IS services implantation. IS architects will try to implant those services to limit this time of service interruption. To determine this transfer duration, we simulated the corruption of an IS resource, triggering applications and services

migration. The obtained results are interesting and show the potential of mobile agents to solve « survivability » problems related to IS management.

### **3.2 Impact of application transfer times on the IS implantation**

We modelled the following virtual enterprise (figure 2). It is composed of 3 interconnected entities, (one in the USA, and 2 in France) which are collaborating on a common research and development project. The goal of this project is to develop an open source enterprise service bus (ObjectWeb, 2006), and some connectors specific to open source or proprietary ERP. This bus should enable clusters of enterprise to place common orders for buying, selling, or transporting goods. One of the strategic resource of the research and development project is the database containing all the data related to the development of applications (source code, documentation, compilation procedures). The difficulty is that this development project is an open source one, meaning that there are constantly actors coming and going out of the project. The environment is naturally dynamic and unstable. One key problem in such a context is to assure the conception project survival by assuring the survival of the resources supporting this process.

An important problem is to decide where to implant the DBMS (on which site, and on which node of that site), knowing that in case of problem appearing on this resource, or on the site hosting the resource, data must migrate easily towards a resource capable of relaying the defective resource, this rescue resource being located ideally on an other site. We implemented for this problem inventory agents and pre-migration mobile agents. Inventory agents are launched on each site by the « process survival server », and they detect the nodes which are able to host the database server, in terms of physical capacity (memory, CPU, disk space, bandwidth...). On each of those sites, a node is detected. The three potential candidates to host the DBMS are alpha, beta and gamma. They are said to be eligible to host the DBMS server. The next question is to locate where to implement the DBMS server. For this, pre-migration agents will install a « fake » database containing around 3Go of data on the eligible nodes. The corruption of the node hosting the DBMS is then simulated. The database (configuration + data) will migrate towards the closer eligible node. If the migration time is short enough, this means that in case of attack or failure the database is able to migrate fast enough, then the tested node is retained to host the database permanently.

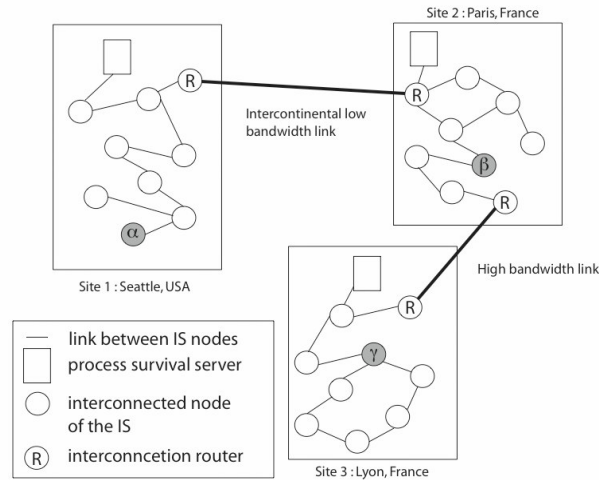


Figure 2 – Virtual enterprise organization for prototype implantation

In a first place alpha is chosen to host the database, and a problem is simulated on alpha. The problem simulation on alpha involves data migration towards beta, which is the closest eligible node. Because of the low throughput of the link binding the US site to the Paris site, the database migration lasts around 9000 seconds (2 hours and 30 minutes, figure 3). This important migration duration is not acceptable, because during this time, the database can be seriously damaged or completely destroyed.

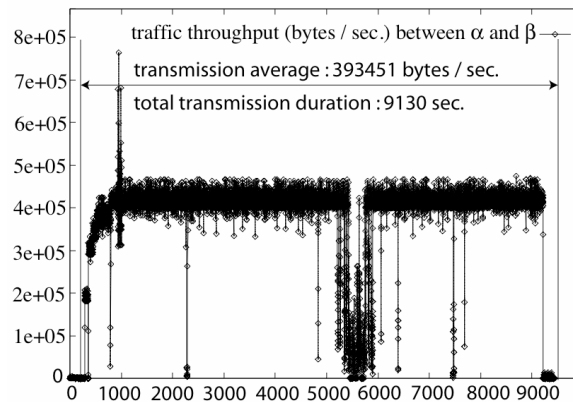


Figure 3 – Traffic generated by the database migration from the alpha node to the beta node

Consequently, the alpha node is not chosen to host permanently the database server. The same scenario is followed for the beta node. A problem simulation performed by the simulation agents causes the database migration towards the gamma node. The bandwidth being more important between Paris and Lyon than in Seattle and Paris, the duration of the database migration is 8 minutes and 30 seconds

(figure 4), which is much more acceptable. Consequently, the beta node is retained to host the main database.

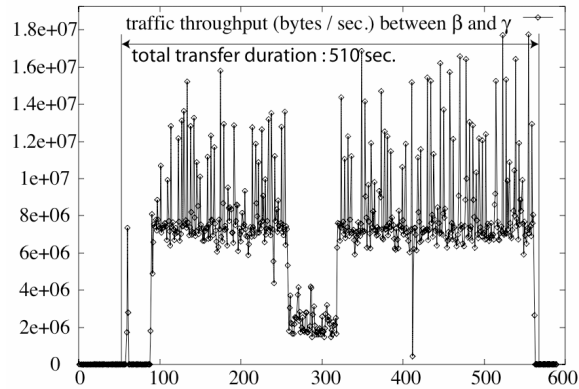


Figure 4 – Traffic generated by the database migration from the beta node to the gamma node

Thus, these first results show that it is possible to use mobile agents to implement survival policies to make stronger the shared processes in a virtual organization.

#### 4. CONCLUSION

We showed in this paper how to improve to reliability of collaborative processes during their exploitation. We defined for this a method to sequence tasks performed by collaborative processes, and also indirectly the resources that are used by these tasks. A mobile-agent based architecture for improving process survival is then described. We prototyped this architecture and we tested it on a simple collaborative process of information sharing to validate our approach. An interesting project about this platform would be to generate automatically the mobile agents depending on the needs and on the context, to come closer to a reactive and adaptive management of the Information System. In this paper we try to improve process survivability but only during their exploitation phase. An interesting perspective would be to try to anticipate the needs of survivability. For that purpose, it would be interesting to integrate risk management concerns in the mobile agent generation process. The risks are different during the steps of execution of the BP, and it would be appropriate to generate specific agents (tagging, monitoring...) according to the risks. This would enable to refine the agent generation and to better define the survival management tasks that have to be done. We could start from the risk management methods existing for the Information System such as EBIOS (DCSSI, 2004) or OCTAVE (Reedy, 2003). We could use these methods from a process point of view, which could enable us to identify the critical phases of a process. These critical phases would be particularly watched by the survival management system.

## 5. REFERENCES

1. Bremer CF, Mundim APF, Michilini FVS, Siqueira JEM, Ortega LM. A Brazilian case of VE coordination. Infrastructures for Virtual Enterprises, Boston, Kluwer Academic Publishers 1999; pp. 377-386.
2. Budimac Z, Ivanovic M and Popovic A, Workflow management systems using mobile agents, In : Advances in Databases and Information Systems : Third East European Conference, ADBIS'99, 13-16 sept. 1999, Maribor, Slovenia, Springer, 1999, pp. 168-178.
3. Caro GD and Dorigo M, AntNet: Distributed Stimergetic Control for Communications Network, Journal of Artificial Intelligence Research, 1998;9:317-365.
4. DCSSI, Expression des Besoins et Identification des Objectifs de Sécurité: EBIOS, Direction centrale de la sécurité des systèmes d'information (french governmental organization), Technical Report, 2004, (<http://www.ssi.gouv.fr/fr/confiance/ebios.html>)
5. Fenet Serge, Toward a paradigm for programming distributed applications based on social insects behavior: application to network security, PhD Thesis (Claude Bernard University, Lyon), 2001.
6. Kramer KJ, Minar N, Maes P, Mobile Software Agents for Dynamic Routing, Mobile computing and Communications Review, 3(2):12-16, 2000.
7. Loke SW and Zaslavsky A, Towards distributed workflow enactment with itineraries and mobile agent management, Lecture Notes in Computer Science : E-Commerce Agents, 2001, vol. 2033.
8. ObjectWeb, Open Source Middleware, (<http://www.objectweb.org>), 2006
10. Reedy R., Healthcare Information Risks Assessment : Empowering Executives to Manage Information Risk, Master Thesis, School of Business Administration, Kennedy-Western University, 2003, 192p.
11. Tomarchio O and Puliafito A, Using mobile agents to implement flexible network management strategies, Computer Communications, 23(8):708-719, 2000.
12. Schulz S, Schulz M, Tanner A., Frame of Interest Approach on Quality of Prediction for Agent-based Network Monitoring, Proceedings of the International Conference on Architecture of Computing Systems 2004 (ARCS'04), Augsburg, Germany: Springer; p. 246-259, 2004.
13. SourceForge, Aglets framework, (<http://aglets.sourceforge.net/>), 2006
14. Suh Y, Namgoong H, Yoo J and Lee D, Design of a mobile agent-based workflow management system, Mobile Agents for Telecommunication Applications : Third International Workshop, MATA'01, 14-16 August, 2001, Montreal, Canada, Springer, 2001.
15. Wildeman L. and Stoffelen R. Alliances and networks of the next generation. KPMG Alliances Networks & Virtual Organisations. K. report. Amsterdam, The Netherlands, 1996.