Business Continuity Model.
Regeneration System for Manufacturing Components

Diego Marcos-Jorquera, Francisco Maciá-Pérez,
Virgilio Gilart-Iglesias, Jose Vicente Berná-Martínez
Computer Science Department, University of Alicante
{dmarcos, pmacia, vgilart, jvberna}@dtic.ua.es

Abstract

At present, with the expansion of information technologies at the industry, it is vital to implant proactive, self-managed systems that ensure continuous operation and, therefore, business continuity. This article proposes a regeneration system for industrial production elements that transfers the concept of high availability to the manufacturing levels of the organisation, acting automatically under open protocols when there is degradation or failure of any of the components, ensuring that normal operation is resumed within a delimited time. In addition to this proposal, we present a development scenario that allows the validity of the approach to be confirmed, while verifying the drastic reduction in recovery times, the support for the significant heterogeneity existing in these scenarios and their high level of automation, while practically dispensing with the intervention of system administrators.

1. Introduction

The business sector is one of the sectors most affected by the appearance of new technologies; this sector is increasingly basing its business on information technologies (IT). Traditionally, the incorporation of these technologies took place separately in each functional area of the organisations, providing solutions for concrete problems. With time and as the technologies have matured, the process has been carried out in an integrated manner, using systems such as ERP (Enterprise Resource Planning) that give an overview of the organisation, flexible solutions and greater speed of response to the changes inherent in such a dynamic environment. Although business objectives should always determine which technological infrastructures ought to be used help to achieve them, it is unquestionable that the IT have also opened the doors to new business models, as is the case of e-Commerce.

In this scenario, the dependence that organisations have on IT is ever greater, as they make up the technological foundations on which a large part of their business processes are supported. It is vital to establish mechanisms that ensure that said infrastructure operate without interruption.

In industry, the relentless penetration of the IT in business is reaching ever lower levels –plant and manufacturing levels– of companies. In this way, the machinery of industry is less and less governed by mechanics and basic electronics and more and more by the computational systems that are taking over these productive elements, allowing their true integration in the business organisational chart, diluting the traditional barriers imposed by the physical restrictions of these components.

With these new approaches, manufacturing components can be viewed conceptually as any other fragment of the electronic business map, allowing the organisation to evolve from *mass production* to *mass customisation*, offering real integration of the supply channels and the unification of suppliers and consumers with the same IT infrastructure and, above all, with the same business objectives – client-oriented at all times.

Proactive services must be used to ensure that said infrastructures operate continuously and are able to react in a preventive manner, even before incidents occur or, at least, ensuring that they have a minimum impact on business continuity through rapid action and, as far as possible, without intervention by the administrators. In addition, each service, each component and each infrastructure should be self-managing, minimising the need for configuration,

maintenance and recovery, while they are all integrated with the overall business model and architecture.

In this article we propose a system for recovering production elements (industrial machinery, field bus controllers, numerical controllers, etc.) that replicates the concepts of high availability and business continuity at plant levels of the organisation. This system is provided as a service under the SOA model, and is fully integrated with existing IT infrastructures. The study defines the service, its features, how it is organised and its main components. We also provide an example of a use case to validate the proposal, allowing a production plant to be kept operational, with hardly any human intervention. Finally, we describe the main conclusions extracted from the study and the lines of study being followed at present.

2. Related Work

After the mid-1990s, industry began to incorporate more open and flexible control mechanisms, as is the case of industrial computers, in order to enhance the performance of PLCs [1]. The power and connectivity of these new components facilitated their integration at all levels of manufacturing processes [2]. This trend has continued and the current incorporation of embedded systems into industrial devices is leading to new production models where IT are ever more important [3]. The appearance of these new devices, much more sophisticated and complex, has led to a new problem: their administration.

In the field of IT management, the first open standards to approach the devices management in a generic manner were SNMP and CMIP [4], specified by the IETF, and both protocols were mainly oriented towards network supervision and control. For more complex maintenance tasks, such as the installation of software and configuration of component, we see solutions that are normally restricted to specific platforms, as is the case of Microsoft Systems Management Server (SMS) for Microsoft Windows environments.

The significant number of tasks associated with network management, as well as their great diversity and complexity, means that maintenance tasks for these systems involve high costs for organisations, in terms of resources as well as time and personnel. Many companies have thus opted for administration systems including self-management and self-configuration features that facilitate, as far as possible, network management [5, 6]. Examples of these systems are Sun's Solstice Enterprise Manager [7] and NESTOR [8].

The use of multi-agent systems for computer network management offers a number of features that favour automated and unmanned maintenance processes [9, 10]. Projects such as AgentLink III, the premier Co-ordination Action for Agent Based Computing, financed by the European Commission's 6th Framework Program, are clear indicators of the significant interest currently aroused by research into software agents.

Within existing maintenance systems, high availability, recovery from disasters and self-management systems apply techniques aimed at avoiding, or at least minimising, the downtime caused by failures or incidents affecting the services provided [11, 12], allowing full restoration of the information held by the device in delimited times. These mechanisms eliminate the problems derived from the degradation or loss of stored information and also facilitate the start-up of new device, replicating the information obtained from a model with similar features.

There are many regeneration systems, outstanding commercial examples being: Ghost from Symantec and REMBO, as well as open-source projects: Clonezilla, G4L, Linbox and UDP Cast [13]. The main drawback of all these systems is their high dependence on the technology to be recovered. This means that multiple solutions must be applied to cover the technological spread of the organisation.

The Department of Computer Science at University of Alicante has developed Gaia [14, 15], an open code based regeneration system used to carry out automatic and unmanned maintenance of the practical work laboratories at the University, and which has been the object of numerous national and international collaboration projects in recent years. The system is designed to be completely independent of the platform it provides support for, making it the ideal starting point for applying recovery techniques to industrial and embedded systems.

3. Description of the proposal

The main aim of our proposal is to replicate the concept of high availability and business continuity that is currently applied to the IT businesses infrastructures, to the computational elements embedded in industrial machinery and other intermediate components used to automate production

– numerical controllers, industrial computers, programmable industrial robots, embedded systems, etc.

To this end, we propose a *regeneration service* for industrial nodes that, when there is failure, degradation and, in general, a need for the maintenance of one item of industrial component, takes over control and recovers its operational capacity in a brief period. The service can only be successful if the tasks are carried out proactively, without human intervention and independently of the component to be restored.

The independence of the service and the devices must obviously be based on a minimum hardware platform that offers basic functionality allowing the service to take over control of the device. This platform can consist of a connection to the communications network that supports a mechanism for transferring the regeneration agent (e.g. Pre-Boot Execution Environment - PXE), and a network mechanism for switching on the device (e.g. Wake on LAN - WoL). The minimum architecture of the device (See Figure 1) must contain a microprocessor or microcontroller and memory capacity (RAM, EPPROM, Flash RAM, etc.) that allows the recovery agent to be executed. Finally, the software or firmware system of the device must be stored on appropriate media (magnetic discs, flash memory or similar).

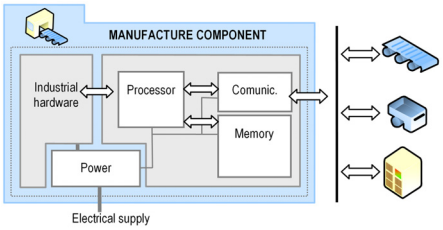

Figure 1. Hardware Architecture of a manufacturing component

However, the lack of any of these elements would not necessarily mean that the regeneration service could not operate, but would only mean that certain tasks would require human intervention (for example, if the device were not compatible with WoL or a similar technology, then someone would have to switch it on).

As can be seen by the definition of the minimum components of the manufacturing device, we have only focused on the physical aspects. This is due to the fact that the regeneration service can recover them from literally zero, meaning that the regeneration agent transferred to the device will have to forecast everything needed to carry out its plan.

In addition, when recovering from disasters in very short periods, the system can provide other services, all related to management of the computational hardware of the manufacturing component: configuration and reconfiguration, updating, installation and maintenance of software and firmware, system management tasks, monitoring, proactive detection and management of faults and preparation of element inventory.

The main features of the proposal can be summarizing as follows: automation of the processes, their independence from manufacturing components, their scalability and their integration with the other IT infrastructures and the global management policies of the organisation.

Automation of the processes is vital for achieving the goals of system proactivity and self-management. In this sense, the regeneration system includes self-configuration, planning and discovery features. The service can also act on demand, behaving as utility computing.

Another of the fundamental aspects of the system is that it is able to offer support to different manufacturing devices, independently of their functionality and hardware or software platforms. The regeneration system makes very few assumptions about the hardware of these components (See Figure 1) and none about their software platform, making it much easier for them to adapt to the enormous variety existing at this level of industry – embedded systems based on microcontrollers, control servers developed on more or less compatible hardware and proprietary systems, mobile devices, numerical controllers, etc., a much richer mix than that found at other levels of the business.

The system must ensure that it can grow and adapt to the changing needs of the environment, and that it can do so with the necessary flexibility. The capacity for self-management of the system means that the impact deriving from a considerable increase in the number and variety of devices to manage is minimum. In addition, the modularity with which it has been conceived allows it to grow at the same speed as needs and allows it to be implanted progressively.

The integration of the infrastructures and technologies required to develop the regeneration service is ensured in transparent fashion, thanks to the fact that said infrastructures are the same as those used in the remaining levels of the organisation: TCP/IP and HTTP protocols, service-oriented n-tiered architectures, user directories, Web and Application

servers, etc.

Finally, this integration allows the processes involved in the management of industrial components to be characterised from the point of view of their contribution to the business model of the organisation. This generates a vision allowing them to be incorporated into the global process map and subjected to established management policies.

4. Elements intervening in the Regeneration service

The regeneration service has been conceptually conceived as a multiagent system together with the environment in which they operate. In turn, said environment consists of a number of elements, services and subsystems that provide all the functionality and resources needed by the agents to develop their plans and achieve their objectives. We can organise the elements into three broad levels (See Figure 2): information system (Level 1), system middleware services (Level 2) and service agents (Level 3). We will analyse each of these in more detail in the following sections.

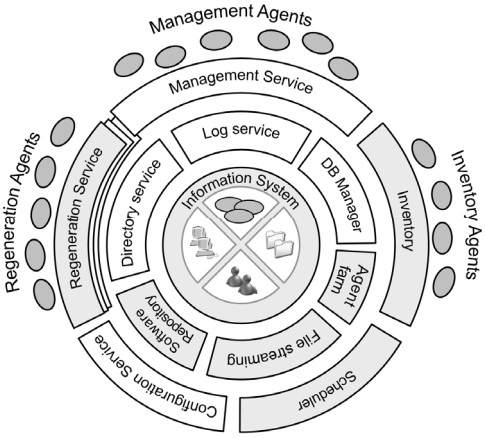

Figure 2. Elements of Management System integrated within TI services

4.1. Information System

The information system is the central core of the system and is one of the basic pillars of the regeneration service in particular and of all the management services in general. It is responsible for gathering all the information related to the services it provides – the software images of the manufacturing elements, the configurations of the different devices, the maintenance plans, the configuration of the services, system agents and activity logs, as well as additional information referring to the different resources of the organisation – users, location of component and network configurations, allowing the automation of all kinds of maintenance processes, as well as the regeneration service itself.

Taken as a whole, the information system of the service must be the same as and be part of the global information system of the organisation, avoiding the duplication of information, the processes for gathering and maintaining it and the resources used to provide it with support.

4.2. Middleware Services of the System

The management services, and in particular the regeneration service, are basically deployed in the clients (the manufacturing components) that are to receive said services. In the next section we will see that the system agents are responsible for executing these processes. However, for these agents to carry out their functions, it is necessary for them to be supported by a group of services that provide them with the necessary resources. These services are those we have identified as the middleware services of the system.

In turn, this level is divided into two sub-levels. The first of these, closest to the information system level, provides the basic services related to the technological platform. These include the relational database manager, the Log manager, responsible for keeping a record of any incidents, the license manager, the software repository and the manager of the bulk transfer of files across the network with support for efficient transmission techniques such as multicast, file caching, real-time compression, etc. This sub-level also includes more specific services, such as the agent farm manager, or standardised services that are currently widespread and complete the regeneration service: a protocol for the remote initiation of device, e.g. PXE, that allows the regeneration agent to be transferred over the network; network self-configuration protocols such as BOOTP or DHCP, that allow network parameters to be configured for the transfer and later use of the agent network; transfer protocols for obtaining the regeneration agent, such as TFTP; a protocol for switching device on, such as WoL, that allows device that is switched off to be switched on by sending datagrams over the network; a time synchronisation protocol, such as NTP, to establish a global clock for the whole system; and a

directory service, such as LDAP, to carry out system access authentication tasks.

The second sub-level of services provides fully functional systems with which the service agents interact and collaborate. These are the planner, responsible for managing the automation of tasks; the inventory service, responsible for ensuring that the information about hardware and software configurations is correctly updated at all times; the configuration service, that allows manufacturing elements to obtain the parameters required to start, obtain the regeneration agent and start the maintenance tasks; the management service, that provides the point of entry for technical staff to administer all these services and infrastructures.

This second sub-level also includes the regeneration service, which is the main element of the service, and where the business logic containing the process for component recovery resides. It acts as the service controller, receiving requests from the different regeneration agents and administrators, initiating and coordinating the actions necessary to carry out the service with the other subsystems. It is also responsible for carrying out the validation, profile management, security and information presentation tasks. The regeneration service is a specific case that can be used as an example of the different services that can be offered by the system, both at manufacturing levels and in the rest of the organisation.

4.3. Service Agents

The service agents are the most important elements of the system. They are the only elements able to interact with the manufacturing component and concentrate most of the proactivity, autonomy and independence of the service.

Our proposal distinguishes three types of agent: the *inventory agents*, the *management agents* and the *regeneration agents*. They are all mobile software agents able to interact with their environment, with the manufacturing elements and with the services and resources provided by the middleware platform.

The *inventory agents* travel through the manufacturing component that is to receive the service, drawing up a detailed report of its configuration and hardware/software composition, and send this information to the inventory service responsible for storing it in the information system. Although this information is vital for the regeneration system, it is not exclusive to it, as most of the management tasks of the organisation's resources, both automatic and manual, use this information.

The *management agents* allow the human administrators to manage the system itself, either by carrying out certain repetitive tasks, gathering information for their work, monitoring the system, generating alarms or simply providing the right interface for each place and time where work is to be carried out. These agents are largely responsible for the self-management and automation capacities of the system.

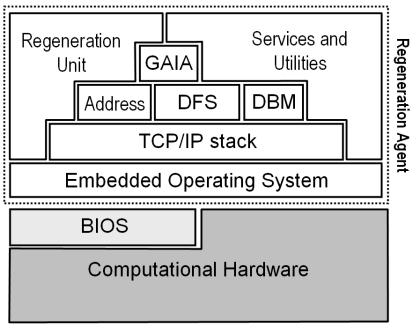

Figure 3. Software architecture of a regeneration agent

The *regeneration agents* are the cornerstone of the regeneration service. They are responsible for travelling to the component they are to manage, taking over complete control of its hardware and carrying out the tasks they have been entrusted: storing in a central repository (within the information system) all or part of the software and data of the component, restoring all or part of the software and data of the component from the software repository, configuring or reconfiguring it, carrying out operating tests and, in general, ensuring that the component is perfectly operational at all times.

Physically, *regeneration agents* should maintain a balance between size and complexity. They must be light enough not to cause a problem when transiting over the communications network, flexible enough to be able to take control of heterogeneous platforms and intelligent enough to be autonomous and obtain the resources needed to carry out their tasks.

Figure 3 shows the software architecture of a regeneration agent. The lowest level corresponds to an embedded operating system suitable for each kind of hardware platform. On top of this layer is the communications layer based on Internet standards. Then we have the elements that allow the agent to access the services provided by the level 2 platform: addressing (such as DHCP) to obtain the configuration

for maintenance, a distributed file system (such as NFS, FTP, file streaming) to transfer the software and the configurations, and a database manager (such as mySQL, XML) to access its planning and diverse information for regeneration. Above the services layer we find the heart of the agent, the embedded core of Gaia [15]. It is basically a library with the management and regeneration functions offered to the higher levels as calls to the system. Finally, the last layer houses the application level, where we can distinguish the utilities and auxiliary services required by the regeneration agent and unit, finally responsible for interacting with the system and managing all the regeneration activity. This module determines the behaviour of the regeneration agent and differentiates it from the other agents.

5. Physical organisation of system elements

The whole regeneration system is based on a technological platform that provides services to the service agents. Each of these services is conceived on the basis of a service-oriented architecture (SOA) that seeks, above all, for them to be independent of each other. This independence allows each service to be assigned the technological resources it requires and that they are not location-dependent. This means that the system can organise itself in very different ways: from a totally centralised model (See Figure 4), to a totally distributed model (See Figure 5), passing through different mixed configurations.

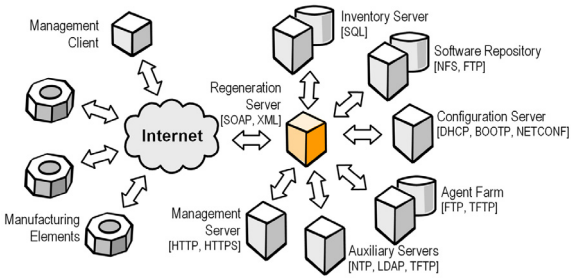

Figure 4. Centralised model

With a *totally centralised organisation* (Figure 4), the manufacturing elements are regarded as external to the system, interconnected by means of a general network such as Internet or the corporate Intranet. These elements can only access the system through the regeneration server. The regeneration server is responsible for organising the rest of the system according to the requests it receives. The *backend* elements are organised around a more or less complex local network specialised in housing the servers.

With this organisational model, the points of access to system administration are also regarded as external (e.g. from a Web Browser with connection to Internet or the Intranet), as their task is simply to send the management requests for the system to handle.

The main advantage of this model is that it offers a configuration that allows easy management of the system and the physical elements making it up but, above all, facilitates secure access to its resources. On the other hand, this model significantly limits the capacity to adapt of the service and the possibility of taking advantage of IT resources in highly distributed environments.

At the other extreme we find the *totally distributed configuration* (Figure 5). In this case, each service can be deployed independently on its technological resources and both manufacturing component and management and system elements are at the same level. However, the security aspects must be transferred to each element, thus making management more difficult. On the other hand, the exploitation of resources and the flexibility to adapt to needs is considerably increased, providing a much finer grained level of scalability than in the previous case – if system capacity is insufficient, the elements affected can be easily identified for replacement or replication as required, without the other elements being affected or, at the worst, with a minimum impact on their operation

.

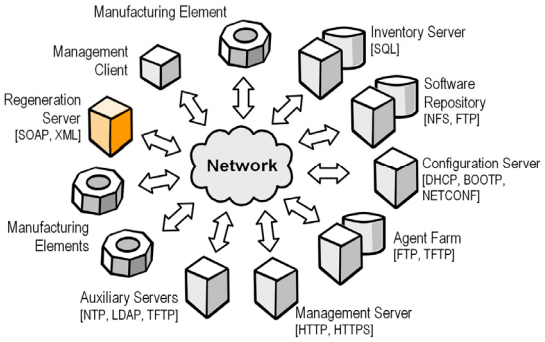

Figure 5. Distributed model

Perhaps one of the most interesting aspects of the proposal is that, as we have mentioned above, a mixed model can be applied to allow centralised and distributed elements to coexist at the same time. In addition, thanks to the use of standard protocols and open technologies, each of the elements making up the system can take advantage of infrastructures and

services that already exist in the organisation (such as DHCP, LDAP, MySQL, inventory systems, etc.).

However, only detailed analysis of each scenario can identify the best configuration. Aspects such as the location of the IT resources of the organisation and the manufacturing elements, the size of the organisation, its global IT policy, the budget, the bandwidth of the network or the qualifications of the technical staff give us a general idea of the different parameters and requirements affecting said study. The final goal is to achieve a balance between costs, complexity and the efficiency and effectiveness of the system. If we also take into account the fact that this balance can vary over time, the proposed approach becomes even more interesting.

6. Regeneration Model

The regeneration service is based on a model of services managed by a mobile software agent. The service can be provided *on demand*, as requested by the manufacturing element needing the maintenance tasks, or *by planning*, meaning that the service can carry out certain maintenance tasks on certain manufacturing components.

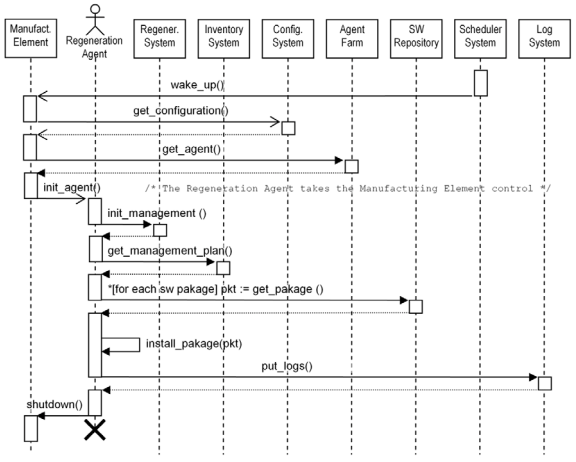

Figure 6. Sequence diagram of a regeneration plan

Independently of how the process is triggered, the protocol is basically the same (See Figure 6). In the first case, the manufacturing element, once its *boot* process has initiated, analyses its status and, if it detects the need for maintenance, makes the request to the regeneration system. With planning, the system itself starts the *boot* process of the manufacturing element. From then on, in both cases, the regeneration system sends the correct *regeneration agent* for the manufacturing element.

The regeneration agent immediately takes control of the computational hardware of the manufacturing element, carried out certain very basic verification and configuration tasks and requests the central regeneration system for the *plan of work* it will immediately put into practice on the component.

A regeneration plan can include such simple tasks as: updating certain applications or even individual files; gathering information on the configuration and status of the element for the central inventory, configuring hardware and software elements, or carrying out a full regeneration of the manufacturing element, consisting of requesting configurations and software from the software repository, either directly (with the distributed organisation model) or through the regeneration server (with the centralised model).

Figure 6 shows a flow chart synthesising the basic elements that make up the service, together with the main tasks of a regeneration plan.

7. Test Scenario

To validate the proposed model, it was implemented on the basis of the J2EE platform. Figure 7 shows the test scenario with the different elements making it up, while Table 1 shows the software configuration of the different server used.

Table 1. Utilized software to deploy the service

System Elements	Server	Applications / Services
Management Service	Web Server	Apache (v. 2.0.55); Tomcat (v. 5.5.15)
Regeneration Service Scheduler	Application Server	JBoss (v. 4.0.4)
Inventory; Log Service DB Manager	Inventory Server	MySQL (v. 5.0)
Information System File Streaming Software Repository Agent Farm	Storage Server	NFS (v. 4); OpenSSH (SFTP) (v. 4.2) SAMBA (v. 3.0.20)
Configuration Service Directory Service	Auxiliar Server	OpenLDAP (v. 2.3.20); NTP (v. 4.2.0); TFTPD (v. 0.1.7); DHCPD (v. 3.0.2)

When choosing the platforms and services, we have selected mature free software projects backed up by a broad scientific-technical community. Debian 3.1 (Sarge), a GNU/Linux distribution, was chosen as the operating system for the servers.

An embedded Linux core was also used to develop the regeneration agents, as it offers many of the features required by the system: support for multiple

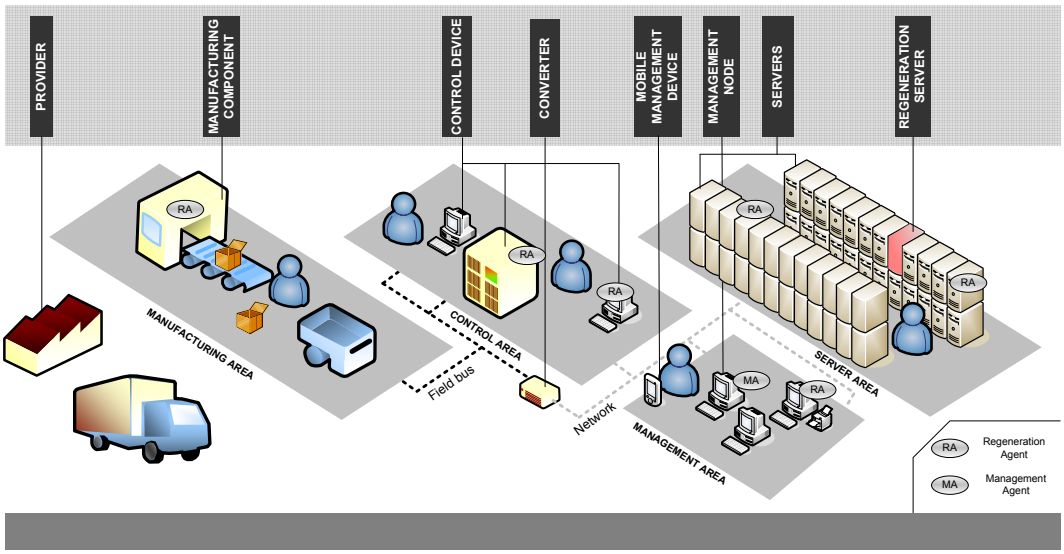

Figure7. Test scenario

hardware architectures, file systems, types of network, scalability, security, *diskless* operation and a high degree of integration. In addition, the fact that it is an *open source* platform means that the code can be modified to suit the specific needs of the agent.

8. Conclusions

In this article we have presented and analysed a regeneration service for manufacturing elements that contain embedded computational systems. This service offers high availability at plant level by applying a technology that has been widely validated at other levels of the organisation. The service is able to absorb the enormous variety existing in this environment and integrates seamlessly with existing IT elements.

The service is based on a node regeneration system developed at Alicante University and validated in multiple scenarios through national and international collaboration projects. The analyses carried out allow us to state that using the regeneration service drastically reduces recovery times after unforeseen events and faults with the platforms and applications installed, providing high availability and cold recovery features. There is also a significant reduction in risk before any disaster, acting within time values ranging from asynchronous replication to backup copies.

The proposal is scalable and can be rolled out progressively as the organisation requires, it integrates with the global management policies of the company and is based on standard technologies and protocols that avoid dependence on proprietary solutions. It also minimises the impact on the environment it is implanted in as it does not require the installation of additional software in the manufacturing component and operates over heterogeneous environments.

The fact of using a system based on a multiagent system consisting of mobile software agents, gives the proposal interesting features that resolve different aspects, such as adaptation of the agent to each platform and its updating. This approach also offers the autonomy and proactivity required by the service.

We are currently concentrating on the development of a collaborative model that improves the massive information diffusion features through wide area networks to ensure its operation over Internet allows sufficiently delimited times to be established in this environment.

9. References

[1] R.P. Moreno, “Ingeniería de la automatización industrial”, Ed. Ra-Ma, Madrid, 2004.

[2] H. Chang, “A Model of Computerization of Manufacturing Systems: an International Study”, Information and Management 39 7:605-624, 2002.

[3] F. Jammes, H. Smit, “Service-Oriented Paradigms in Industrial Automation. IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS 1 1: 62-70, 2005.

[4] U. Black, “Network Management Standards: SNMP, CMIP, TMN, MIBs and Object Libraries, 2nd edition”, Ed. McGraw-Hill, 1994.

[5] M. Kim, M. Choi, J. W. Hong, “A load cluster management system using SNMP and web”, International Journal of Network Management 12 6: 367-378, 2002.

[6] H. Saïdi, B. Dutertre, J. Lew, A. Valdes, ”Self-regenerative software components”, In Proceedings of the 2003 ACM workshop on Survivable and self-regenerative systems, pp 115-120, 2003.

[7] J. E. López de Vergara, V.A. Villagrá, J. Berrocal, “Semantic Management: advantages of using an ontology-based management information meta-model”, in Proceedings of the HP Openview University Association Ninth Plenary Workshop (HP-OVUA'2002), 2002.

[8] V. Konstantinou , D. Yemini, Florissi , “Towards Self-Configuring Networks”, in DARPA Active Networks Conference and Exposition, pp 143-145, 2002.

[9] T. C. Du, E. Y. Li, A. P. Chang, “Mobile Agents in Distributed Network Management”, in Communications of the ACM 46 7: 127-132, 2003.

[10] J. Guo, Y. Liao, B. Parviz, “An Agent-based Network Management System”, in Hamza HM (eds). Internet and Multimedia Systems and Applications (IMSA 2005), Honolulu, pp 20-88, 2005.

[11] A. Brown, D.A. Patterson, “Embracing Failure: A Case for Recovery-Oriented Computing (ROC)”, in High Performance Transaction Processing Symposium. Asilomar, CA, 2001.

[12] K. Ivinskis “High availability of commercial applications”, in Carey M, Schneider D (eds).Proceedings of the 1995 ACM SIGMOD international conference on Management of data, ACM Press, New York, pp 433-434, 1995.

[13] J. A. Cuff, G.M.P. Coates, T.J.R. Cutts, M. Rae, “The Ensembl Computing Architecture. In Genome Research”, Cold Spring Harbor Laboratory Press 14: 971-975, 2004.

[14] F. Maciá-Pérez, “Modelos de Administración de Redes Heterogéneas de Computadores. Sistema de Regeneración de Nodos de Red”, Doctoral thesis. Departamento de Tecnología Informática y Computación. Universidad de Alicante, 2001.

[15] F. Maciá-Pérez, J.M. García-Chamizo, F.J. Mora-Gimeno, D. Marcos-Jorquera , V. Gilart-Iglesias, J.A. Gil-Martínez-Abarca, J.C. Monllor-Pérez (eds) “Desarrollo de Grandes Aplicaciones Distribuidas sobre Internet”. Publicaciones U. Alicante, 2005.