To My beloved family ...

To my true friends ...

AKNOWLEDGMENT

Firstly, I would like to thank my supervisor Professor Mohammed Mostefai for providing me with supervision, motivation and encouragement throughout the course of my work. Without his care, and friendship, I would not be able to complete this work. In particular I would to express to him my gratitude for his active and constructive participation in this thesis.

I would like to thank Professor Benouattass Nourredine from department of physics for his invaluable and precious help in this work.

I would to thank also all my family members for their help, support and encouragement.

ABSTRACT

This doctoral thesis is part of the research activities of the Laboratory of Automatic of Sétif (L.A.S). It treats specification and verification of production systems in general and mobile enterprises in particular. The major contribution of this thesis appears in the study of the integration of mobility in the enterprises. Indeed, an architecture for mobile enterprises is proposed and comprises a Web-Services layer and a multi-agents layer combining static agents and mobile agents. This contribution is organized in two phases: the first phase concerns the modelling of workflow based enterprises. The model is validated analytically by a performance analysis. The second phase consists to explore and exploit some formal tools for the specification of the mobile enterprises: Two formalisms are retained the REACTNets for the Web-Services layer and MAUDE for the behaviour of the mobile agents.

RÉSUMÉ

Le travail de cette thèse s’inscrit dans le cadre des activités de recherche du Laboratoire d’Automatique de l’Université de Sétif (L.A.S). Il traite de la spécification et de la vérification des systèmes de production "entreprises" en général et des entreprises mobiles en particulier. La contribution majeure de cette thèse apparaît dans l’étude de l’intégration de la mobilité dans les entreprises. En effet, une architecture des entreprises mobiles est proposée et comporte une couche WebServices et une couche multi-agents combinant des agents statiques et des agents mobiles. Cette contribution est organisée en deux phases : la première traite de la modélisation des entreprises moyennant les WorkFlow. Le modèle étant validé analytiquement par une analyse de performances. La seconde consiste à explorer et exploiter quelques outils formels pour la spécification des entreprises mobiles à savoir : les REACTNets pour la spécification et la vérification de la couche Web-Services et MAUDE pour la spécification et la vérification du comportement des agents mobiles.

// ملخـــــــص

تدخل هذه الرسالة في أطار أعمال البحث التي يقوم بها مخبر الأليات بسطيف (L.A.S) و تتمحور حول الوصف النوعي و التحقق من أنظمة الإنتاج بصفة عامة و المؤسسات التي تستغل التكنولوجيات النقالة بصفة خاصة. تظهر المساهمة الأساسية لهذا العمل في دراسة كيفية إدماج التكنولوجيات النقالة في المؤسسات حيث تم اقتراح نموذج لبنية المؤسسات النقالة. يتمثل هذا النموذج في طبقتين الأولى مبنية على تقنية خدمات الويب (Web-Services) و الثانية على تقنية الأنظمة متعددة العملاء (Systèmes multi-agents) و التي تستعمل العملاء الثابتين و المتنقلين. تنقسم هذه المساهمة إلى قسمين:

القسم الأول يخص تمثيل المؤسسات النقالة القائمة على أساس أنظمة تدفق المهام (WorkFlow)، حيث تم إثبات صحة النموذج بطريقة رياضية تحليلية.

القسم الثاني يتمثل في دراسة مجموعة من أدوات الوصف النوعي للمؤسسات النقالة حيث تم اقتراح إطار عام على أساس نموذجين، الأول نوع خاص من شبكات بتري الجبرية (ECATNets) لوصف طبقة خدمات الويب و الثانية لغة MAUDE لوصف حركة العملاء النقالة.

PUBLICATIONS RELATED TO THIS THESIS

Conferences

(1) Faiza Bouchoul -Mohammed Mostefai-Zohour Kaddem

"Towards M-Manufacturing - Case Study : a Fault Tolerant Model for M-Maintenance of Spare Parts with Mobile Agents" IEEE : 4 Th International Conference: Sciences of Electronic, Technologies of Information And Telecommunications) - SETIT 07 - Hamamet marsh 25-29 2007 Tunisia

(2) Faiza Bouchoul - Mohammed Mostefai "MOBIFLEX a generic architecture for healthcare systems" IEEE : First International conference on emedical systems - E-Medisys 07 - FEZ october 24-26 2007 Morocco

(3) Faiza Bouchoul - Mohammed Mostefai "Vers une approche formelle pour la spécification des systèmes distribués réactifs" IEEE : 4th international conference on computer intetegrated manufacturing –CIP 07- November 3-4 Setif Algeria

(4) Faiza Bouchoul – Mohammed Mostefai –Zohour Kaddem "The design of a mobile agent based workflow and its implementation on jade platform" in International Conference of Information Technologies CITIC'09 Sétif 4,5 mai 2009

Publications

Without impact factor

Faiza Bouchoul- Mohammed Mostefai "From E-manufacturing to M-manufacturing" in The international Arab Journal of Information Technology (IAJIT) Vol. 5 N° 2, pp 140-147 April 2008

With impact factor

Faiza Bouchoul - Mohammed Mostefai "Agent-services and mobile agents for an integrated HCIS" International Journal of Computer Integrated Manufacturing, (IJCIM) Taylor and Francis editor Vol. 22 Issue 5 may 2009 pp 458-471

INTRODUCTION

Today mobility has transformed enterprises by incorporating mobile devices, mobile networking and mobile internet technologies to enable new possibilities and perspectives in communication, information access and business transaction when away from the desk. Wireless technologies have matured enough to deliver promising possibilities to resolve many business needs and problem in enterprises. The enterprise activities are generally governed by a business process. According to the Workflow Management Coalition (WFMC) a business process is "a set of one or more linked procedures or activities which collectively realize a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships." A business process can be seen as a flow of related activities that together create a service or a product. Business processes are generally automated partially or totally by workflows. The Workflow management Coalition (WfMC) has defined workflow as "The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules". With the advancement and spreading of various mobile technologies and infrastructures, there is increasing demand for mobile users to connect to Workflow Management Systems. Mobile workflows extend the main business process of the enterprise, since the latter incorporates the functionalities of the first within its own processes out of the traditional boundaries of the organization. Mobile workflow users can be thin clients such as palm-top computers and PDA connected to their enterprise through wireless network; they can be involved in the main workflow system at any time and at any place and perhaps in a disconnected manner. Mobile Workflows provide traditional business processes with possibilities to manage the execution of business activities by large numbers of users distributed over a wide area and using heterogeneous resources and small devices which are connected to a network only occasionally. Mobile devices offer some advantages in comparison to the traditional PC based execution of workflow tasks; because of their portability: they enable the ubiquitous and asynchronous execution of workflow tasks. Users connect to a server in their enterprise to load applications and data in their mobile devices from or to request some task to be done, and then they disconnect from the server and work on those applications and data or wait until the task is remotely performed. After the work has been completed they reconnect with the server to send the results of their work or to receive the reply. Mobile Workflow can be defined as the application of wireless data communications technology to remote workers operating over a wide geographical area.

Examples of applications that could be managed by mobile workflows include construction systems, M-maintenance systems and M-Health.

The main restrictions of mobile devices for the execution of workflow tasks are essentially limited bandwidth, limited resources such as CPU power and memory and limited energy supply. So that and in most cases the performances of mobile applications is greatly restricted due to architectural and design considerations and thus very subject to frequent failures. In such environment, mobile workflows not only inherit the wireless technologies weakness but also are often Internet-based so that mobile users periodically become unavailable due to the lack of network service guarantees. The result is limited or very difficult business information access and activity coordination. The obvious consequence is that traditional workflow management systems have not been designed for dynamic environments requiring adaptive situation induced by mobility. In the other side, considering the growing need of mobility and the more frequent use that organizations make of mobile devices, it is necessary to provide support for the integration of those devices to the work. Unfortunately, mobile workflow users often run into integration problems when attempting to access desktop applications. In other terms, an adequate architecture handling mobile workflow requirements must enable the smooth integration of the mobile workforce within the main business process of the organization and easy mutual synchronization between workers with high fault tolerance mechanisms.

Thus mobile workflow applications need convenient, efficient and robust paradigms suitable for distributed applications, even when partially connected computers are involved. Finally mobile workflow applications can be optimized by implementing minimal tasks on implied mobile devices and centralizing the most resources consuming in the desktop infrastructure of the enterprise.

The aim of this thesis is twofold:

Firstly: We propose the design of MOBIFLEX a generic architecture for mobile workflows. For the solution we propose an architecture integrating mobile agents, static agents with intelligent capabilities and Web-Services, since these new technologies when put together can generate a flexible, reconfigurable, adaptive and integrable framework to fulfil easily requirements of mobile workflows. The architecture is empowered by fault-tolerance mechanism and the choice of mobile agent is validated analytically. Finally we explain how this architecture can be implemented on JADELEAP/JADEX platform and as a case study we show how MOBIFLEX can be a solution for M-Health information systems.

Secondly: We shall try to propose a formal framework for specifying and verifying MOBIFLEX workflows, We have chosen the rewriting logic as the basis of our complex formal framework since this logic is a very powerful unifying paradigm of most of formal models of concurrency. Rewriting logic is a computational logic in which a wide range of logics and models of computation can be represented. For MOBIFLEX we have a large range of possibilities to formalize all the facets of the system, indeed an important number of tools based on rewriting logic such as MAUDE language and the ECATNets exist and can be exploited for our purpose. Finally combining tools with a common semantics enable a homogeneous integration and attenuates the difficulties often encountered during the integration of ad hoc formalisms.

CHAPTER I

MOBILE ENTERPRISE

The Committee on Visionary Manufacturing Challenges established by the National Research Council’s Board on Manufacturing and Engineering Design [National Research Council 97] creates in 1997 a vision of a competitive manufacturing enterprise in 2020, the committee identified six "grand" challenges for manufacturers that represent gaps between classical practices and the vision of manufacturing in 2020.

Grand Challenge 1. Achieve concurrency in all operations.

Grand Challenge 2. Integrate human and technical resources to enhance workforce performance and satisfaction.

Grand Challenge 3. "Instantaneously" transform information gathered from a vast array of diverse sources into useful knowledge for making effective decisions.

Grand Challenge 4. Reduce production waste and product environmental impact to "near zero."

Grand Challenge 5. Reconfigure manufacturing enterprises rapidly in response to changing needs and opportunities.

Grand Challenge 6. Develop innovative manufacturing processes and products with a focus on decreasing dimensional scale.

Network computing seems to be a strategic solution to these challenges, networking was a subject to great changing impacting the evolution of enterprise from traditional practices to modern ones based on high technologies like Web and wireless ones. This introductive chapter gives an overview of evolution of enterprise networking from wired to wireless and the emergence of the mobile enterprise and the mobile computing.

I. Enterprise networking evolution.

I.1 The enterprise multi-net.

A computer network is a system that connects end-user workstations and devices separated in space. Network computing environments in enterprise are generally local area networks (LANs) and are composed of a collection of interactive and cooperative software systems, tools, hardware, enterprise agents and clients.

When first computer networks in the sixties, were expensive mainframes, the current generation of computing, is true distributed computing providing advanced functionality such as concurrency control, transactional support and backup and recovery, in another hand the remote-procedure call (RPC) mechanism provided a programmatic interface through which computing resources at remote hosts can be accessed [Rorive 03].

The rapid expansion of the computing and communications infrastructure is now not only enabling new grand applications, but also influencing the science of computing in terms of paradigms, hardware and software architectures, the specific technologies that realize those architectures and finally the tools and techniques used to construct them. Modern enterprises have now complex information system with many subsystems like ERP (Enterprise Resource Planning), CRM (Customer Relationship Management), KM (Knowledge Management), MES (Manufacturing Execution System), and BPM (Business Process Management) [Nagarajan *et al.* 99].

Fieldbuses are other particular types of LANs dedicated to industrial usage in shop floors. Traditionally, the shop floor was isolated from the rest of the supply chain and within the organization. A traditional fieldbus network consists of several nodes like sensors and actuators physically connected through a wired bus. Fieldbuses are now doted with TCP/IP interfaces with organisation LAN and with Internet [Venzke *et al.* 05], as a consequence of these hardware interconnections between heterogeneous networks in the same organisation, enterprise’s network became a complex infrastructure formed by a number of sub-networks interconnected through gateway interfaces typically Ethernet. Links can be wired or wireless. We call this complex structure (figure 1) a multi-net and define it as follows.

Definition 1: "A multi-net is a complex infrastructure formed by interconnecting a number of heterogeneous LANs like Ethernet LANs, fieldbuses, industrial Ethernet and so on with wired or wireless links" .

In figure 1 a typical multi-net is presented : In this example Ethernet acts as the company's intranet backbone between the traditional enterprise LAN and a fieldbus (industrial LAN in shop floor level) and between the enterprise and the Internet. Ethernet is linked to a controller C and two industrial PCs. The fieldbus, links industrial devices (sensors, actuators, smart devices…), links can be wireless or wired. The networking of a great number of PCs in offices and the proliferation of the Internet around the world has made Ethernet a universal networking standard.

The hardware and related software have evolved to the point where even inexperienced users can build simple networks. In automation, Ethernet is commonly used with other fieldbuses. Industrial Ethernet [Sink 99] is also a new emerging solution.

This infrastructure provides naturally more possibilities to achieve what is called "enterprise integration" (EI) . Enterprise integration can be seen as the re-engineering of business processes and information systems to improve teamwork and co-ordination across organizational boundaries [Nagarajan *et al*. 99]. Virtual enterprises (VEs) [Whitman *et al*. 99] are an other concept that resembles enterprise integration, but goes beyond it in terms of allowing multiple enterprises to merge temporarily to accomplish specific short term goals, so that VE is formed, operated and dissolved after performing its goals.

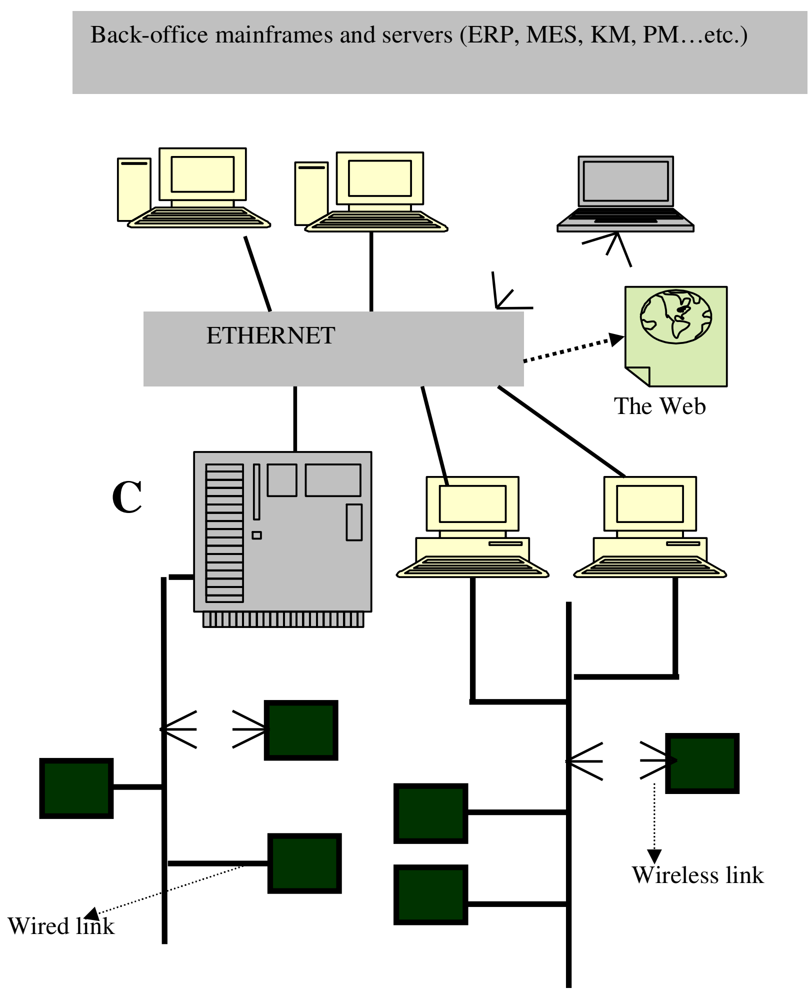


Figure 1 *Example of a multi-net*

I.2 Wireless technologies and new possibilities for enterprises.

Wireless describes the technology that allows exchanging of voice, image, data and other kinds of information through radio or infrared techniques without need of any cable or other physical media, the obvious and most important consequence of wireless technology is mobility; Although the terms "mobile" and "wireless" are similar in theory and often used interchangeably, they are in application very different [Deshpande 04]: Mobile pertains to the ability of an entity to be on the move when wireless pertains to the technology that allows transmission of voice, data and other content through radio waves without being restricted to cables or other physical media.

Mobile devices are portable electronic components that are used by mobile employees to do their work. Mobile devices depend on wireless technology to connect to the enterprise and transfer content to fulfil the users' business needs. Mobile technologies are able to ensure a tangible business benefit by making technical and business information more accessible from anywhere and anytime.

Normally, devices operate within networks that provide metropolitan, state-wide, or nation-wide coverage. Internet-enabled cell phones and Personal Digital Assistants (PDAs) have emerged as the newest products that can connect to the Internet across a digital wireless network. New protocols, such as Wireless Application Protocol (WAP), and new languages, such as WML (Wireless Markup Language) have been developed specifically for these devices to connect to the Internet. Enterprise networking has evolved from traditional LANs to WLANs(Wireless Local Area Networks) (WLAN) are implemented as an extension to wired LANs within a building and can provide the final few meters of connectivity between a wired network and the mobile user [Pierre 03].

II. Mobile enterprise.

Wireless technologies have matured enough to deliver promising possibilities to resolve many business needs and problem in enterprises. Today mobility has transformed enterprises by incorporating mobile devices, mobile networking and mobile internet technologies to enable new possibilities and perspectives in communication, information access and business transaction. Mobile technologies enable business transactions when away from the desk. Information can be accessed via a phone or handheld devices.

Wireless technology improves productivity and allows to easily extend existing business solutions into a mobile environment and develops custom solutions that best serve the organization's specific business needs; they improve the potential of employees, customers and partners by facilitating interaction and immediate reaction to new information. Existing business solutions can be extended easily into M-business solutions, with best flexibility and less cost [Pigneur 02]. For these reasons today many companies are adopting mobile solutions in their different activities. People increasingly have access to their e-mail when they are on the move; mobile messaging is increasingly used for communications when employees are on the road; intelligent alerting is empowering mobile employees; and logistics companies are using mobile technology to dispatch jobs to delivery personnel and provide real-time tracking information for customers. A number of mobile applications are already in place, well-adopted and working well. Mobile e-mail has largely worked, just look at the prolific number of Blackberry devices in use today. Mobile messaging has worked. These possibilities were made easy by a number of opportunities offered by wireless and mobile technologies that can be resumed as follows [Afshar & Radage 03]

- **Availability of a large game of new mobile devices with efficient software at low prices**

Internet-enabled cell phones and Personal Digital Assistants (PDAs) have emerged as the newest products that can connect to the Internet or to the intranet across a digital wireless network. New protocols, such as Wireless Application Protocol (WAP), and new languages, such as WML (Wireless Markup Language) have been developed specifically for these devices to connect to the Internet. new promising technologies such as Bluetooth, provide high possibilities for wireless links between mobile PCs, mobile phones, and other portable handheld devices, and connectivity to the Internet.

- **More possibilities for internet connectivity**

Mobile networks have now been largely enabled with Internet connectivity through both mobile (phone-based) technologies such as GPRS, 1xRTT and 3G, in addition to wireless LAN possibilities.

- **Maturity of mobile software**

An important number of software product exist at the market, These suites provide a standard-based means to develop applications and make them accessible through mobile, messaging, voice channels and exchange of data.

Furthermore, technologies such as Web-Services promise to open up existing enterprise applications and systems and make them accessible from enterprise portals and mobile devices.

II.1 Mobile enterprise: definition and basic concepts.

Enterprises embracing wireless technologies are often said "mobile enterprises", we can define mobile enterprise as follow:

Definition 2

"An enterprise is said " mobile " if its multi-net infrastructures comprise wireless and mobile technologies ."

Mobility in enterprises delivers new concepts related with new forms of business services and work and new methodologies and paradigms in information technologies. Next definitions are borrowed from [Alahuhta *et al*. 05] and [Lilischkis 03].

II.1.1 Mobile business solution.

Mobile business solution "refers to a set of business-oriented applications that are operated using mobile terminals, such as mobile phones, PDAs or wireless laptop".

II.1.2 Mobile application.

Mobile application "stands for a computer program that is executed in a mobile computing platform such as a PDA or a mobile phone. The mobile application may include data storage, data processing or viewing or transmitting it to another application or server".

II.1.3 Mobile service.

Mobile service "is an electronic service that consists of three main components: a mobile application or mobile browsing as a client, wireless networking and server implementation providing the needed functionality or information (Content) to the user".

II.1.4 Mobile work.

Mobile work is a "combination of technology, workplace organization, work facilities and support systems allowing people to work mobile and in multiple locations at different times".

II.2 Mobile work classification.

First of all let us notice that mobile work don't necessary have the use of modern information and communication technologies (ICTs), "ICT work" comprises work practices making use of ICTs to increase efficiency and flexibility in time and place and the sustainability of resource use. ICT work includes the use of hardware tools such as notebooks and mobile phones as well as software applications such as e-mail and the Internet. Next figure shows how mobile work is related to ICT work and Telework.

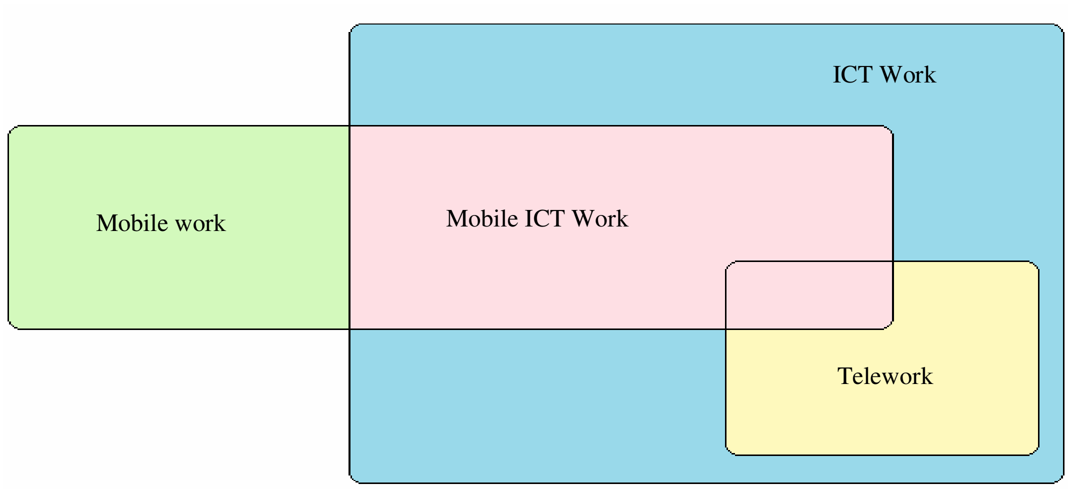


Figure 2. Mobile work, ICT Work and Telework [Lilischkis 03]

One classification of mobile workers has been introduced by [Lilischkis 03] in Star-project [Star-project 04] In this model, the main factors for classifying mobile workers are frequency of changing location and the number of locations where a worker carries out tasks. There are five categories in this classification: On-site movers, Pendulums, Yo-Yos, Nomads and Carriers.

"Yo-yos": The definition of the yo-yo type of mobile work refers to a fixed location as a reference point that is left for a certain time to work elsewhere (see figure 3). This type is generally thought to be the second most prevalent one.

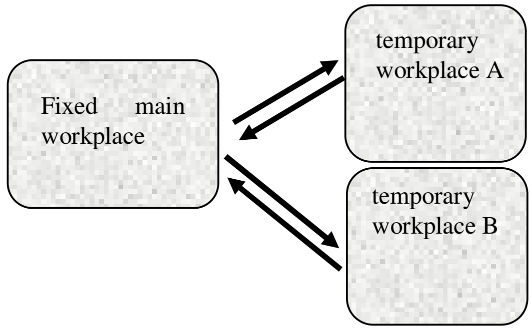


Figure 3. Yo-yo type of mobile worker

Almost every employee may leave the workplace for meetings; customer visits another location from time to time. Thus almost everyone can be considered a mobile worker. Examples of the yo-yo kind of mobile work are work on business trips (e.g., taking part in a meeting in a foreign town), in the field (e.g., face-to-face interviews for scientific research), when travelling (e.g., writing reports while sitting in a train),

"Pendulums": The pendulum type of mobile work includes work with two fixed work locations such as the employer's premises, a home office or a client's premises, between which the workers alternate (see figure 4). The pendulum type includes the classical telework: work being carried out at a distance from where its results are used, implying computer use, and the use of telecommunication for electronically exchanging work results and messages with colleagues.

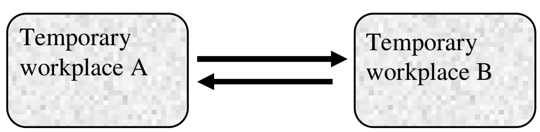


Figure 4. Pendulum type of mobile worker

"Nomads": The nomad type of mobile work refers to people who constantly move from one location of work to another (see figure 5). The number of work locations is more than two, otherwise they should rather be classified as pendulums. They may or may not have headquarters.

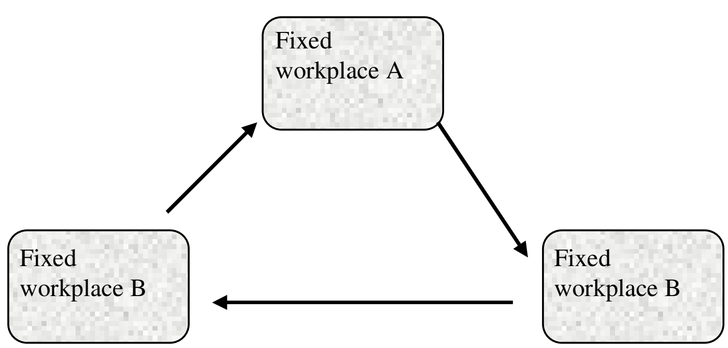


Figure 5. Nomad type of mobile worker

The more frequent the location of work changes, the more useful may be the application of mobile ICTs because a frequent change of locations does hardly allow setting up fixed facilities.

"Carriers": The definition of the carrier type of mobile work refers to personal or commodity transportation involving continuously moving from one place to another (see figure 6). Examples are jobs such as train conductors and ticket collectors.

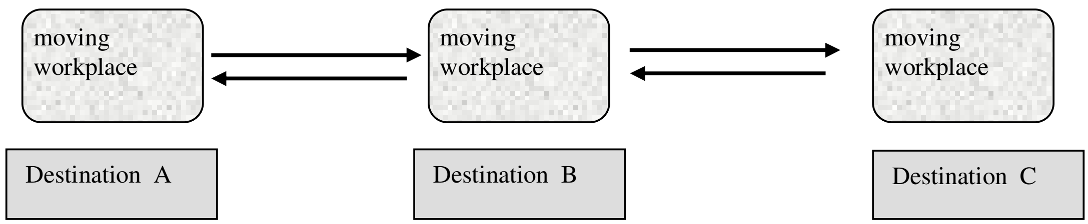


Figure 6. Carrier type of mobile worker

Another classification of mobile workers is given by [Deshpande 04], mobile Users have the usage of ICT tools and are categorized as follows:

• **Desk Warrior:** This is someone who doesn't need to travel and primarily works from a fixed location. The user has a desktop or laptop, and could have a PDA or mobile device (not necessarily wireless) connected to it.

• **Campus Warrior:** This is someone who is quite mobile, but only on the work premises. Examples include IT administrators, facilities staff, service technicians and specific line-of-business workers such as factory floor workers. Visitors and employees from other branch offices also fall into this category. These users normally use a laptop or wireless handheld device, primarily in a WLAN environment. Many enterprises now provision laptops with built-in support for WLANs so that employees can move among office rooms without network connectivity restrictions.

• **Road Warrior: This** is someone who is rarely in the office and does most of his or her daily tasks while on the move. The "road warrior" might use a laptop or handheld device that has multiple network support (WLAN and WWAN).

III. Mobility facets in enterprises.

III.1 Impact of the Web : E-Business and E-Manufacturing.

For the past decade, the impact of Web-based technologies added new promising possibilities to the design, manufacturing, and aftermarket service of a product. An efficient information flow between customers, manufacturing and product development (i.e. plant floor, suppliers and designers) can be done easily with Web-technologies. Recently Web technologies [Papazoglou 06] provide high mechanisms for integrating and interacting heterogeneous Web users. In particular, E-business [Nayak 02] has emerged as a powerful technology to facilitate business transactions around the world. E-commerce is the most important application of E-business. E-business uses the new family of technologies available on the Internet such as multi-agents and Web-Services. These technologies enable people to communicate in new ways, provide new business models, permit businesses to operate more efficiently and take advantage of the new global network economy.

Internet technology is not just E-business, it is also E-production. The Industrial Internet is Internet-based management and collaborative enterprise networking for industry. E-manufacturing [Koç *et al.* 03] has emerged as a new industrial discipline offering to manufacturing processes what E-business do to business ones. E-manufacturing provides an Internet based strategic framework for the factory by satisfying increasing need for communications to and from the factory floor. Manufacturers need systems that can reveal available capacity, status of orders, and quality of a product not just after it comes off the line, but while it is in process. In some fields, using Internet technologies to control production lines or mechanical systems is possible. For example [Leparc *et al.* 04] with E-maintenance it is now possible to make remote diagnostics, to solve and repair problems, to prepare maintenance phases etc…And with E-expertise: it will become possible for experts to operate from their office a machine located somewhere in the world, just using classic Web technologies.

III.2 Impact of mobility : from the E-factory to the M-factory.

Progresses that the wireless networking made provide the enterprises new prospects in choices for technology to be used to carry out solutions not only at the business process level (M-business) but also in the ground (M-manufacturing) [Bouchoul & Mostefai 08].

The wireless networks also presents their own problems of safety and QOS but escapes in an obvious way certain problems involved in the Internet and induced by its public character and its large scope.

III.2.1 M-business.

M-business is an extension of E-business through the usage of wireless technologies. As mobile Internet capabilities improve, it will become the most convenient mode of access to online services, delivering services/products to customers, or mobilizing the enterprise with access to information and applications from anywhere.

There are hundreds of examples that can be given in mobile business processes. For example [Afshar & Radage 03] a salesperson takes an order on a mobile-connected device, that order is entered into the system at headquarters, a supplier is queried to confirm availability, the customer's credit is checked via a partner, the accounts receivable systems is updated, a distributor is notified of expected delivery schedules and destinations and the call center might check status upon a query from the customer. Using the efficiencies of mobility, a customer could call into the call center and request the status of the order immediately after the salesperson submits the order via a mobile device. If the salesperson had to come back to the office to key this information using a PC, then the whole business process would be broken at that point; nothing could happen until the order is keyed in by the salesperson.

A typical example consists in giving maintenance engineers on production lines the ability to locate spare parts for broken-down machinery in real-time using a mobile device instead of having to perform ad-hoc searches through parts stores by hand or have to come back to the office in order to pick information from wired LAN [Bouchoul *et al.* 07].

Another good example of mobile work is given by construction projects [Aziz *et al* 06]: the global nature of many construction projects means that project teams are increasingly geographically dispersed working across time zones and numerous organisational boundaries. These teams are often quickly brought together to deliver a construction project within limited time and resources. At the same time, the construction processes have increased in complexity in the recent years and have become more information-intensive.

Finally healthcare is another domain where mobility can be of great benefit, healthcare information systems (HCIS) are complex, heterogeneous, and spread out over multiple locations making their management and exploitation very onerous and lacking efficiency.

Medical practicioners are inherently mobile so that mobile healthcare solutions can give medical staff instant access to HCIS, resources and services allowing medical tasks such as diagnosis, data exchange or monitoring to be made sooner and with more accuracy. Using wireless and handheld computer technologies provides medical practitioners with instant communication possibilities and mobile access to detailed and latest patient data and medical references [Bouchoul & Mostefai 07] [Bouchoul & Mostefai 09].

III.2.2 M-manufacturing.

New wireless technologies offer possibilities to mobilize access to industrial equipment or to relate wireless heterogeneous industrial networks, so it is possible to do activities such as supervising, controlling and, maintaining industrial process from mobile devices [Koumpis 05], [Rugierro 04]. Wireless fieldbuses are now a reality [Decotignie *et al.* 02].

Mobile computing technology can be used to increase collaboration among field workers by providing on-line access to information and interactive communication facilities.

Over the last few years, radio-frequency technologies have been developing as a relevant and increasingly indispensable communication support. Together with the innumerable applications in our everyday life, the usage of wireless communications in industrial environment seems to start being of great interest to the scientific community. The possibilities of the wireless technologies to add value in monitoring, control and industrial-system configuration are tremendous [Ruggiero 04]. Key benefits are the elimination of cables and connectors from the manufacturing floors, resulting in shorter installation times and reduced machine maintenance costs, and the potential for a flexible topology and mobile applications. Furthermore, wireless technologies will facilitate communications with rotating and moving machine parts and enable systems to be programmed, actuated and automatically report their status back to a central controller or to an operator with a laptop or another wireless device.

Mobile manufacturing practices are considered as E-manufacturing ones and are not differentiated from wired manufacturing practices as it is done for M-business and E-business. On the basis of these remarks we prefer make the difference between E-manufacturing which is based on wired technology and M-manufacturing which is based on wireless remote one [Bouchoul & Mostefai 08].

Definition 3: "The M-manufacturing is the discipline gathering of the activities of control or of supervision in a manufacturing system via a wireless network", we associate the concepts of M-maintenance, M-control, and M-supervision to it

Let us notice that the use of the Internet does not exclude the recourse to mobility, this differentiation allows the company to make a judicious choice for the solution to be adopted between the possibilities offered by E-manufacturing, also M-manufacturing through wireless Internet, or a pure M-manufacturing through a disconnected WLAN. Combining wireless technologies to Internet gathers the advantages of two technologies, but weaknesses of both concerning the real time and the quality of services.

M-manufacturing solutions over WLANs seem in our sense an adequate choice for local applications on the ground. Internet solutions are the best choice for M-business but also for practices of M-manufacturing when the quality of service and the real time are not priorities.

Table 1 compares main characteristic of M-manufacturing and E-manufacturing, when the figure 7 shows the desirable borders for applications of M-manufacturing and of E-manufacturing, the limits in dotted line indicate open borders: The E-manufacturing can be applied through traditional Internet without any geographical restrictions when QOS and real time are not priorities; typically, the M-manufacturing is better when applied in a WLAN.

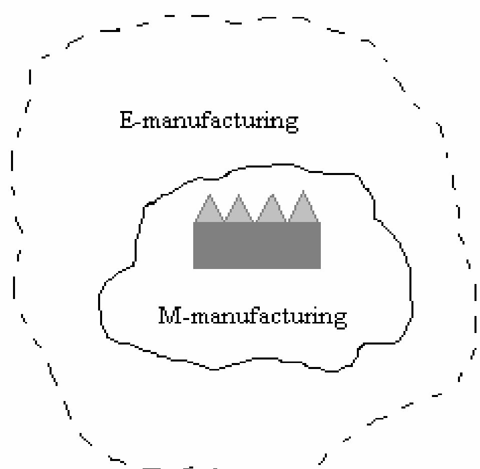


Figure 7. M-manufacturing and E-manufacturing boundaries

We end this section by giving some typical examples of M-manufacturing practices. An example of industrial use of wireless technology on the ground can be the connection of fairly distant points for which the commutated telephone network is not an ideal solution because of the only concern of the possible non-availability of the lines. Other examples: the mobile systems which embark automatisms (sensors and/or actuators to be connected to an automata placed in fixed zone, or even programmable automata to connect to the factory network…). Other examples, the wireless network gives access to zones where the cost of installation of a cable would become onerous for technical reasons or of safety :crossing of a river, crossing of a zone at the risk of explosion, crossing of a zone at very high temperatures (along a blast furnace) [mesures 03].

A typical project of mobile industrial application is the Most project, In The MOST project [Friday *et al*. 99**]** (Mobile Open Systems Technologies for the Utilities Industries), engineers working within the power distribution industry in the field were traditionally coordinated by a single control centre, which approved all switching in the power distribution network, and maintained an overall picture of the current state. The centralised approach ensured that conflicting requirements were resolved safely, but the centre became a bottleneck so that the efficiency was reduced. In order to help field engineers work more efficiently, they were given possibilities to work in a mobile environment; Mobile computers were used to obtain maps indicating the current state of the power distribution system in the area in which they were working. The engineers were also able to communicate with each other and with their control center to coordinate activities and safely resolve switching requirements.

M-manufacturing (via a WLAN)	E-manufacturing (via traditional internet)
Low bandwidth	Large bandwidth
General resource poverty (storage capacity, computing power and battery life)	No problem with resources
Real time Relatively controllable (local network)	No real time
Limited QOS	Limited QOS
Reduced Installation cost (no wiring)	High installation cost
Reduced maintenance	High maintenance (wires)
Local network: best security	Extended network: security causes problems
Greater flexibility and reconfigurability	No flexibility and very difficult reconfigurability
Well adapted in geographically limited area	Well adapted in extended area when QOS and real time are not priorities

Table 1 M-manufacturing versus E-manufacturing

IV Wireless technologies standards: an overview

IV.1 Communication Platforms for WLANs

Wireless networks operate in one of two modes [Pierre 03]: Cellular mode and ad hoc mode; cellular wireless networks were introduced in the early 1980s as a technology for providing access to the wired phone network to mobile users. The network coverage area is partitioned into regions (with diameters ranging from 100s of meters to a few kilometers) called cells, hence the term 'cellular'. In each cell there is a base station (BS), which is connected to the wired network, and through which the mobile devices in the cell communicate over a one hop wireless link.

A wireless ad hoc network also said mobile ad hoc network (MANET) comprises several devices arbitrarily located in a space. Each device is equipped with a radio transceiver, all of which typically share the same radio frequency band. In this situation, the problem is to communicate between the various devices. Nodes need to discover neighbours in order to form a topology, good paths need to be found, and then some form of time scheduling of transmissions needs to be employed in order to send packets between the devices. Packets going from one node to another may need to be forwarded by other nodes. Thus, these are multihop wireless packet radio networks,

IV.2 Wireless technologies

Enterprises have several options regarding the kind of wireless network they deploy, including 802.11b, Bluetooth and wireless WANs [pierre 03].

IV.2.1 802.11b (Wi-Fi).

This specification was defined by the IEEE. 802.11b is used as an extension of Ethernet to wireless communications and is quite flexible about the kinds of network traffic that pass over it. It is primarily used for TCP/IP but also supports AppleTalk and other PC file sharing standards.

Disparate systems such as PCs and Macs can communicate over 802.11b by using PC or PCI cards, as can some of the newer hardware using Universal Serial Bus (USB) and other forms of 802.11b based wireless network cards. Adapters for PDAs such as Palm OS and Pocket PC-based devices are also available.

802.11b is the standard wireless network deployment platform for enterprises and public-area wireless networks such as those found at airports, hotels, conference centers, coffee shops and restaurants.

Other specifications released by the IEEE include: The 802.11a: performance and throughput are significantly increased, the 802.11g works twice as fast as 802.11b. The 802.11i is one of the more anticipated WLAN specifications yet. It has enhanced security and uses Extensible Authentication Protocol (EAP) to provide authentication services. It also comes with enhanced encryption facilities.

IV.2.2 Bluetooth.

This wireless network specification is ideally suited for Personal Area Networks (PANs) that operate in short ranges and need a robust wireless network that allows transmission of bandwidth-intensive information. Bluetooth technology works well for ad hoc networks and situations where device-to-device communication is desired. For example, you can wirelessly connect from your PDA to a printer to print documents, or perhaps synchronize your desktop with your PDA over the air.

IV.2.3 Wireless MANs.

The IEEE has also released a specification for wireless metropolitan area networks (WMANs) called 802.16, or Wi-Max. This supports point-to-multipoint (PMP) architecture in the 10-66 GHz range, with a throughput of up to 120 Mbps. The base station connects to a wired backbone and can transmit wirelessly up to 30 miles to possibly hundreds of subscriber stations.

IV.2.4 Wireless WANs.

While the architectures discussed above are specific to WLAN environments, other technologies provide support for a wireless WAN (WWAN) environment. The latest generation of this technology is called 3G, and although many carriers claim to offer such services, most achieve only 2.5G ratings. There are several WWAN protocols used around the world. Two of the most widely used are:

• Code Division Multiple Access (CDMA): CDMA enables a large number of users to access wireless channels on-demand. Used by many digital mobile phone companies, CDMA delivers performance that is almost eight to 10 times better than traditional analog cell phone systems.

• Global System for Mobile (GSM): This wireless platform provides full voice and data support, with worldwide roaming capabilities. Included in the GSM family is the General Packet Radio Service (GPRS) platform for delivering Internet content on mobile devices, and the Enhanced Data Rates for GSM Evolution (EDGE) and third-generation GSM (3GSM) for delivering mobile multimedia. GPRS is the generally accepted WWAN standard in Europe.

V Mobile computing.

Wireless technologies have largely influenced the emergency and evolution of new concepts in software engineering such a ubiquitous computing, nomadic computing, context-awareness … A good survey dealing with mobility in a software engineering perspective can be found in [Roman *et al*. 99] Mobile computing is presented as "the study of systems in which computational component may change location", where suitable locations are points in space that may be continuous or discrete and can be of two types : physical and logical. "Physical mobility entails the movement of mobile host in a physical space and logical mobility involves mobile units (of code and state) that migrate among hosts that are typically stationary". Context represent the most important aspect of mobile computing, so that mobility in software engineering is named "context-aware computing", the context of a mobile unit is determined by its current location which in return defines the environment where the computation associated with the unit is performed, the context may include current time, resource, services as other component of the system [Roman *et al*. 99].

In [Abowd *et al*. 97] the definition of context computing is extended to consider it as "any attempt to use knowledge of a user's physical and social, and informational and even emotional state as input to adapt the behaviour of one or more computational services".

V.1Mobile applications.

Mobile computing environments are exposed to specific requirements differentiating them from classical environments. Mobile devices and applications must have a high degree of interaction and adaptability to any changes in their setting, mobile applications have to be context-aware.

The principal aspect of mobile environment, is heterogeneity, in the software, hardware and networking levels as the devices that form it are composed of a large number of different applications, middleware systems and hardware and can access different networking infrastructures. The major characteristic of mobile applications can be resumed as follows [Efstratiou *et al*. 01]

- Decoupling: mobile applications must be able to run in a disconnected or weakly connected state and hence computation must be opportunistic
- Context dependency: availability of resources is dependent on nearness to other components, which causes the need for location awareness
- Low bandwidth
- General resource poverty of mobile devices in terms of storage capacity, computing power and battery life

V.2 Calculi and formal specification languages.

V.2.1 The π-family.

The π-calculus [Milner *et al*. 92] is a process calculus introduced as a model of communication in concurrent distributed systems. The π-calculus was extended to allow inputs and outputs between processes to carry more than one object (i.e., object tuples); this is known as the polyadic π-calculus. In the past decade, the π-calculus has become a reference calculus with numerous variants adding enhancements to the basic framework. The asynchronous π-calculus (π_a) [Honda &Tokoro 91], supports asynchronous environments, Dπ [Riely &Hennessy 98], [Sewell 98] extends the polyadic π with channel locations and process mobility. These are achieved, respectively, by adding the ability to transmit tuples of channels and location names, and the addition of a migration operator. Dπ is a simple distributed extension of the π-calculus in which agents are explicitly located, and may use an explicit migration construct to move between locations.

In [Hym 09], "passports" are introduced in Dπ to control mobile migrations. Passports are tied to specific locations, from which migration is permitted. The spi-calculus, extends the π-calculus with cryptographic primitives [Abadi & Gordon 99]. The nomadic π-calculus [Unyapoth 01], which has been introduced to model and study properties of communication infrastructures of mobile processes.

V.2.2 The Join-calculus.

The join-calculus [Fournet & Gonthier 96]: it is an "extended subset" of π_a, with "better locality and better static scoping rules". The Join calculus uses channel locations and is claimed to be implementable in a realistic distributed environment, while retaining the expressive power of the π-calculus. According to the authors, the natural primitives for doing this were message passing, function calling and pattern matching. The Distributed Join-calculus (Djoin) [Fournet *et al.* 96]: adds abstractions to the Join calculus similar to those added to the π_a-calculus by $D\pi$, i.e., process distribution and process mobility. Djoin attempts to "adapt the models and methods developed by concurrency theory to the programming of mobile agent systems distributed over a wide area network.

V.2.3 Mobile Ambients.

Mobile Ambients [Cardelli & Gordon 00] [kwiatkovska *et al.* 09] is another calculus that derives its process primitives from the π-calculus. The Ambient calculus introduces the notion of a bounded environment (the ambient) where processes or mobile agents cooperate. An ambient consists of a set of local agents and possibly other sub-ambients. Ambients are moved as a whole under the control of the enclosed environment.

V.2.4 Mobile UNITY.

Mobile UNITY [McCann & Roman 98] is an extension of UNITY [Chandy & Misra 88] to address the various design issues raised by device mobility and wireless communication. Such issues include: the possibility for applications to be able to run in a disconnected or weakly connected state and provides a transparent management of locations and proximity. Context unity [Roman *et al.* 07] is an extension of Mobile UNITY where mobile units sense aspects of the environment (Context) and use this information to adjust their behaviour in response to changing circumstances.

V.2.5 CMN (Calculus of Mobile Ad Hoc Networks).

CMN [Merro 09] is a process calculus proposed to study the behavioural theory of Mobile Ad Hoc Networks. The operational semantics of this calculus is given both in terms of a Reduction Semantics and in terms of a Labelled Transition Semantics.

V.2.6 Mobile Z.

Z is a formal specification notation based on set theory and first order predicate logic. It is used by industry as part of the software (and hardware) development process in Europe, USA and elsewhere. In combination with natural language, Z can be used to produce structured powerful specifications. Mobile-Z [Bettaz & Maouche 05] extends the Z language with location and mobile operation schemas. The objective is to seek appropriate models for tackling the space and coordination dimensions of mobility in a software engineering perspective. Mobile-Z introduces the notion of location schemas and mobile operation schemas enlarging Z in order to cope with physical and logical mobility in an explicit way. Location schemas involve a distinguished variable whose value specifies the current location of a mobile entity.

V.2.7 P-prisma and F-prisma.

P-PRISMA and F-PRISMA [Bruni & lanese 08] are two parametric calculi that can be instantiated with different interaction policies, defined as synchronization algebras with mobility of names (SAMs). In particular, P-PRISMA is based on name transmission (P-SAM), like π-calculus, and thus exploits directional (input–output) communication only, while F-PRISMA is based on name fusion (F-SAM), like Fusion calculus.

V.3 Programming Languages and extensions.

V.3.1 Pict.

Pict [Pierce and Turner 97] [Bidinger & Compagnoni 09] is a high-level concurrent programming language constructed from the π-calculus. Unlike many other π-based languages that combine π-calculus communication with a functional core language, Pict attempts to take communication (in terms of π-calculus primitives) as the unique mechanism of computation.

V.3.2 The Nomadic Pict.

The Nomadic Pict project [Wojciechowski & Sewell 99] attempts to provide high-level location independent communication facilities for mobile agent-based distributed systems. As a programming language, Nomadic Pict is based on the nomadic π-calculus and extends the Pict language, which is concurrent but not distributed.

V.3.3 Linda.

Linda [Carriero & Gelernter 89] is a process coordination language where multiple processes interact by asynchronously entering and removing tokens from a single, globally shared tuple space. Other extensions of Linda were proposed later. LLinda [Nicola *et al*. 97] is a formalisation of Linda using process calculi techniques. LLinda allows distribution and nesting, but not mobility, of tuple spaces. PLinda (Persistent Linda) [Jeong & Shasha 94] is a fault-tolerant version of Linda. Fault tolerance is achieved by three major extensions to the Linda model: lightweight atomic transactions, continuation committing of critical variables and a checkpoint-protected tuple space. LIME (Linda in a Mobile Environment) [Picco *et al*. 99] [Murphy *et al*. 01] is a middleware that further adapts the communication model of Linda to create a coordination layer that can be exploited successfully for designing applications that exhibit logical and/or physical mobility.

V.3.4 Klaim.

Klaim [Nicola *et al*. 98] and [Fessant 08] is a core language that builds on the Linda model focuses on privacy and integrity of data, as well as mobility. Xklaim [Bettini & Nicola 05] is an extension of KLAIM, X-Klaim naturally supports programming with explicit localities; these are first-class data that can be manipulated like any other data, and coordination primitives that permit controlling interactions among located processes.

V.3.5 Jocaml.

Jocaml [mandel & maranget 08] [Conchon & Fessant 99] is an implementation of the Distributed Join calculus based on the Objective Caml programming language, thus integrating Djoin's primitives for network transparency with the expressiveness of a high-level functional language.

REFERENCES

[Afshar & Radage 03] M. Afshar and K. Radage, "Enterprise mobility what's ahead ?" white paper september 2003 SIIA's UPGRADE magazine available at http://www.siia.net/upgrade/archive/0809_03.asp

[Abadi & Gordon 99] M. Abadi and A. D. Gordon, "A calculus for cryptographic protocols: The spi calculus," Information and Computation, vol. 148[1] pp 1-70 1999.

[Abowd *et al.* 97] G.D. Abowd , C.G. Atkeson , J. Hong , S.Long , R. Kooper and M. inkerton "Cyberguide: A Mobile Context-Aware Tour Guide" Baltzer/ACM Wireless Networks, Vol. 3[5], pp 421-433. 1997

[Alahuhta *et al.* 05] P. Alahuhta, J. Ahola and H. Hakala " Mobilizing Business Applications" Technology Review 167/2005 TEKES Helsinki 2005

[Aziz *et al.* 06] Z. Aziz , C.J Anumba., D.Ruikar, P. Carrillo and D. Bouchlaghem "Intelligent wireless Web-Services for construction—A review of the enabling technologies" Automation in Construction, elsevier Vol. 5[2] pp 113-123. March 2006

[Bettaz & Maouche 05] M. Bettaz, M. Maouche: "Towards Mobile Z Schemas". International journal of computer science and application IJCSA Vol. 2[2], pp 101-117. 2005

[Bettini & Nicola 05] L. Bettini, R. De Nicola. "Mobile Distributed Programming in X-Klaim" Formal Methods for Mobile Computing, Advanced Lectures. LNCS 3465, pp 29-68, M. Bernardo and A. Bogliolo (Eds.), 2005, Springer.

[Bidinger & Compagnoni 09] P. bidinger and A. compagnoni "Pict correctness revisited" Theoretical Computer Science Vol. 410[2,3], pp 114-127. Elsevier Science Publishers February 2009

[Bouchoul & al. 07] F. Bouchoul, M. Mostefai and Z. Kaddem, , "Towards M-Manufacturing - Case Study: a Fault Tolerant Model for M-Maintenance of Spare Parts with Mobile Agents", in Proc. of IEEE: 4Th International conference: Sciences of Electronic, Technologies of Information and Telecommunications Setit'07 marsh 2007 hamamet Tunisia ISBN: 978-9973- 61-474-2

[Bouchoul & Mostefai 08] F. Bouchoul- M. Mostefai "From E-manufacturing to M-manufacturing" in The international Arab Journal of Information Technology (IAJIT) Vol. 5[2], pp 140-147. April 2008

[Bouchoul & Mostefai 09] F. Bouchoul and M. Mostefai "Agent-services and mobile agents for an integrated HCIS" International Journal of Computer Integrated Manufacturing, (IJCIM) Taylor and Francis editor Vol. 22[5], pp 458-471 may 2009

[Bouchoul & Mostefai 07] F. Bouchoul and M. Mostefai, 2007, "MOBIFLEX: A Generic Architecture for Integrated M-Health Applications", in Proc. of IEEE: First international conference on E-medical systems: E-medisys'07 October 2007, FEZ Morocco

[Bruni & Lanese 08] B. Roberto, L. Ivan "Parametric synchronizations in mobile nominal calculi" Theoretical Computer Science, Vol. 402[2,3], pp 102-119. August 2008

[Cardelli & Gordon 00] L. Cardelli and A. D. Gordon, "Mobile ambients," Theoretical Computer Science, Vol. 240[1] pp 117-213 2000.

[Carriero & Gelernter 89] N. Carriero and D. Gelernter, "Linda in context," Communications of the ACM, vol. 32[4] pp 444-458 1989

[Chandy & Misra 88] K. M. Chandy and J. Misra, "Parallel Program Design: a Foundation." Addison-Wesley, 1988. Department of Computer Science, Washington University, March 1998.

[Conchon & Fessant 99] S. Conchon and F. L. Fessant, "Jocaml: mobile agents for objective-caml," in Proc. First International Symposium on Agent Systems and Applications [ASA'99] / Third International Symposium on Mobile Agents [MA'99], [Palm Springs, CA, USA], 1999.

[Decotignie *et al.* 02] J-D Decotignie. "Wireless fieldbusses a survey of issues and solutions". In Proc. 15th IFAC World Congress on Automatic Control [IFAC 2002], Barcelona, Spain, 2002

[Deshpande 04] S. Deshpande "Enabling a Successful Wireless Enterprise" white paper Office of the CTO May 2004 available at http://www.securitytechnet.com/resource/rsc-center/vendor-wp/ca/22996-enabling_a_successful.pdf

[Efstratiou 01] Ch. Efstratiou, K. Cheverst, N. Davies and A. F. Efstratiou, "Architectural Requirements For The Effective Support Of Adaptive Mobile Applications". Proc. of Int Conf on MDM2001 (Mobile Data Management) July 2001.

[Fessant 08] F. L. Fessant, "The jocaml system. Software and documentation available at http://pauillac.inria.fr/jocaml/."

[Fournet & Gonthier 96] C. Fournet and G. Gonthier, "The reflexive CHAM and the join-calculus," in Proc. 23rd Annual ACM Symposium on Principles of Programming Languages, pp 372-385, 1996.

[Fournet *et al.* 96] C. Fournet, G. Gonthier, J.-J. Lévy, L. Maranget, and D. Rémy, "A calculus of mobile agents," in Proc. 7th International Conference on Concurrency Theory [CONCUR '96], pp 406-421, 1996.

[Friday *et al.* 99] A. Friday, N. Davies, G.S. Blair and K.W.J Cheverst . "Developing adaptive applications: The MOST experience". Journal of Integrated Computer-Aided Engineering, vol. 6[2] pp 143–157 1999

[Honda &Tokoro 91] K. Honda and M. Tokoro, "An object calculus for asynchronous communication," in Proc. ECOOP '91 [P. America, ed.], LNCS 512, pp 133-147 Springer-Verlag, 1991.

[Hym 09] S. Hym "Mobility control via passports" Information and Computation, Vol. 207[2] pp 171-193 February 2009

[Jeong & Shasha 94] K. Jeong and D. Shasha, "Plinda 2.0: A transactional/checkpointing approach to fault tolerant linda," in Proc. 13th Symposium on Reliable Distributed Systems, pp 96-105 IEEE 1994.

[Koç *et al.* 03] M. Koc , J. Ni , J. Lee , P. Bandyopadhyay, "Introduction of E-Manufacturing," International Journal of Agile Manufacturing, Special Issue on Distributed E-Manufacturing, Vol. 6, Dec. 2003

[Koumpis 05] K. Koumpis, M. L. H. Andersson and M. Johansson "Wireless Industrial Control and Monitoring beyond Cable Replacement" PROFIBUS International Conference Coombe Abbey, Warwickshire, UK, June 2005

[Kwiatkowska *et al.* 09] M. Kwiatkowska, G. Norman, D. Parker, M. G. Vigliotti "Probabilistic Mobile Ambients" Theoretical Computer Science, Vol. 410[12,13] pp 1272-1303 March 2009

[Leparc *et al.* 04] ph. Leparc , J . Vareille and L. Marcé "E-productique ou contrôle et supervision distante de systèmes mécaniques sur l'internet" RS - JESA. Vol. 38[5] pp 525 - 558 2004

[Lilischkis 03] S. Lilischkis, "More Yo-yos, Pendulums and Nomads: Trends of Mobile and Multi-location Work in the Information Society", in Issue Report N. 36. 2003, Star-project / Empirica

[Mandel & Maranget 08] L. Mandel and L. Maranget. "Programming in JoCaml (Tool Demonstration)", in: 17th European Symposium on Programming (ESOP'08), Budapest, Hungary, pp 108–111 April 2008

[McCann & Roman 98] P. J. McCann and G.-C. Roman, "Compositional programming abstractions for mobile computing" IEEE Transactions on Software Engineering, vol. 24[2] pp 97-100 1998.

[Merro 09] M. Merro "An Observational Theory for Mobile Ad Hoc Networks" Information and Computation, Vol. 207[2] pp 194-208 February 2009

[Mesures 03] "reseaux locaux radio : le sans fil se fait plus" in mesures 759 - november 2003

available at www.mesures.com /archives/759soltransmissionsansfil.pdf

[Milner *et al.* 92] R. Milner, J. Parrow, and D. Walker, "A calculus of mobile processes," Information and Computation, vol. 100 pp 1-40 September 1992.

[Murphy *et al.* 01] A. Murphy, G. Picco, and G.-C. Roman, "Lime: A middleware for physical and logical mobility," in Proc. 21st International Conference on Distributed Computing Systems [ICDCS], pp 524-533 April 2001.

[Nagarajan *et al.* 99] R. Nagarajan, L.Whitman, and. S. H Cheraghi "Enterprise Integration" Proc. of The 4th Annual International Conference on Industrial Engineering Theory, Applications and Practice, San Antonio, TX USA. 1999

[National Research Council 97] National Research Council "Visionary Manufacturing Challenges for 2020" (Free Executive Summary) 1997

[Nayak 02] R. Nayak, "Data Mining for Web-Enabled Electronic Business Applications", Chapter 8 in Architectural Issues of Web-Enabled Electronic Business, editor: Shi Nanshi, Pub: Idea Group Publishers, pp 129-138 2002

[Nicola *et al*. 97] R. D. Nicola, G. L. Ferrari, and R. Pugliese, "Coordinationg mobile agents via blackboards and access rights," in Proc. Coordination '97: Coordination Languages and Models [D. Garlan and D. L. Métayer, eds.], LNCS 1282 pp 220-237, Springer-Verlag, 1997.

[Nicola *et al*. 98] R. D. Nicola, G. Ferrari, and R. Pugliese, "Klaim: a kernel language for agents interaction and mobility," IEEE Transactions on Software Engineering, vol. 24[5] pp 315-330, 1998.

[Papazoglou 06] M.P Papazoglou, " Web-Services Technologies and Standards". ACM Computing Surveys 2006

[Picco *et al*. 99] G. P. Picco, A. L. Murphy, and G.-C. Roman, "Lime: Linda meets mobility," in Proc. 21st International Conference on Software Engineering [ICSE '99], 1999.

[Pierce & Turner 97] B. C. Pierce and D. N. Turner, "Pict: A programming language based on the pi-calculus," Tech. Rep. CSCI-476, Computer Science Department, Indiana University, 1997.

[Pierre 03] S. Pierre "réseaux et systèmes informatiques mobiles : fondements, architectures et applications" presses internationales polytechniques Quebec Canada 2003

[Pigneur 02] Y. Pigneur "An ontology for M-business models", in S. Spaccapietra *et al.* [Eds.] Conceptual Modelling - ER 2002, Tampere, LNCS 2503, pp 3-6 October 2002

[Riely &Hennessy 98] J. Riely and M. Hennessy, "A typed language for distributed mobile processes," in Proc. 25th Annual ACM Symposium on Principles of Programming Lanuages, pp 378-390, 1998.

[Roman *et al*. 07] G. C. Roman, C. Julien, J. Payton "Context Unity" Theoretical Computer Science, Vol. 376[3] pp 185-204 May 2007

[Roman *et al*. 99] G. C., Roman, A. L. Murphy and G. P. Picco "A Software Engineering Perspective on Mobility", WUCS-99-31, Washington University in St. Louis, December 1999.

[Rorive 03] Brigitte RORIVE "L'entreprise réseau : unicité de la formule, diversité des situations" XIVème congrès de l'AGRH Grenoble, 20-22 novembre 2003

[Ruggiero 04] M.o Ruggiero, " Bluetooth in Industrial Environment" , Expert Monitoring Limited marsh 2004

[Scott *et al*. 02] A. Scott, A. O'malley, L. Athie and A. Scott Deloach"Comparing performance of static versus mobile multiagent systems" National Aerospace and Electronics Conference [NAECON] Dayton, OH, October 10-12, 2000.

[Sewell 98] P. Sewell, "Glocal/local subtyping and capability inference for a distributed pi-calculus," in Proc. ICALP '98, LNCS 1443 pp 695-706, Springer-Verlag, 1998.

[Sink 99] P. Sink "Industrial Ethernet: The Death Knell of Fieldbus?" Manufacturing Automation Magazine, Canada, April 1999

[Star-project 04] Star-project, http://www.databank.it/star/index_2.html. 2004

[Unyapoth 01] A. Unyapoth, "Nomadic pi calculi: Expressing and verifying infrastructure for mobile computation," Tech. Rep. TR 514, Computer Laboratory, University of Cambridge, June 2001.

[Venzke *et al*. 05] M. Venzke, Ch. Weyer, and T. Volker "Application specific vs. standard Web-Service interfaces for the vertical integration of fieldbus systems" . In Proc. of the Third Workshop on Intelligent Solutions in Embedded Systems [WISES'05]. Hamburg, Germany, May 20, 2005

[Whitman *et al*. 99] L. Whitman , K. Krishnan, R. K. Agarwal and P. Bhandare "Engineering The Extended Enterprise" Proc. of The 4th Annual International Conference on Industrial Engineering Theory, Applications and Practice November 17-20, 1999, San Antonio, Texas, USA

[Willig *et al*. 05] A. Willig, K. Matheus, and A. Wolisz, "Wireless Technology in Industrial Networks", Proc. of the IEEE, vol. 93[6] pp 1130-1151, June 2005

[Wojciechowski & Sewell 99] P. Wojciechowski and P. Sewell, "Nomadic pict: Language and infrastructure design for mobile agents" In the Proc. of ASA/MA'99 , First International Symposium on Agent Systems and Applications/Third International Symposium on Mobile Agents, October 1999.

MOBILE WORKFLOWS : STATE OF THE ART AND ENABLING TECHNNOLOGIES.

The enterprise activities are generally governed by a business process. According to the Workflow Management Coalition [WFMC 99] a business process is "A set of one or more linked procedures or activities which collectively realize a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships." A business process can be seen as a flow of related activities that together create a service or a product. A workflow is an automation of the business process. When it is assumed to act in mobile environment it is said a mobile workflow, the aim of this chapter is to give an overview of basic definitions and concepts related to business processes and workflows, and a summary state of the art of mobile workflows .

I. Business processes and workflows: an overview.

I.1 Basic concepts [WfMC 99].

I.1.1 Process Definition.

A process definition is a representation of a business process in a computerised form. The representation supports automated manipulation, such as modelling, or enactment by a workflow management system. The process definition consists of a network of activities and their relationships, criteria to indicate the start and termination of the process, and information about the individual activities, such as participants, associated information technology (IT) applications and data, etc. A process instance is the representation of a single enactment of a process including its associated data. It therefore represents an instance of a process definition that includes manual and automated aspects

I.1.2 Activity.

An activity is a description of a piece of work that forms one logical step within a process. An activity may be manual, or a workflow automated activity. A workflow activity requires human and/or machine resource(s) to support process execution: where a human resource is required, an activity is allocated to a workflow participant.

An automated activity is an activity which can be automated using a workflow management system to manage the activity during execution of the business process. A manual activity is an activity within a business process, which can not be automated and hence lies outside the scope of a workflow management system. Such activities may be included within a process definition, for example to support the modelling of the process, but do not form part of a resulting workflow. An activity instance is the representation of an activity within a single enactment of a process. Finally a work item is a representation of the work to be processed by a workflow participant in the context of an activity within a process instance.

I.2 Workflow Definitions.

I.2.1 Definition and example.

WfMC has defined workflow in [WfMC 99] as "The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules". As an example, next Figure 1 shows how a part of an order processing workflow can be automated.

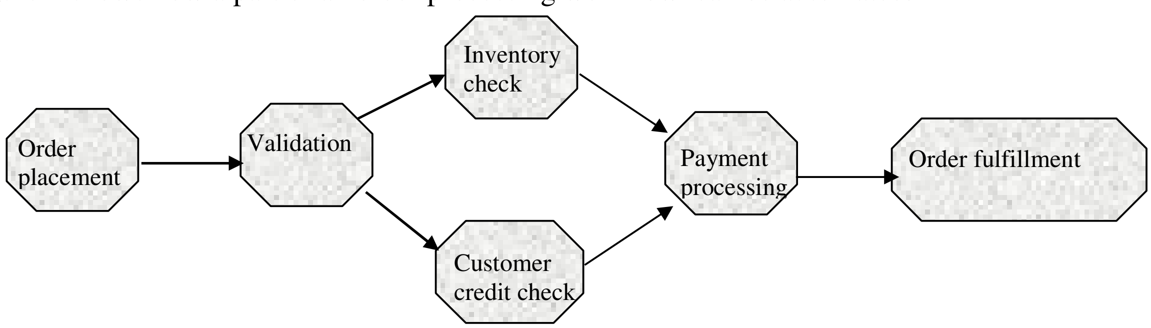


Figure 1. Order processing workflow example

Each node in this workflow represents an activity. Activities can depend on other activities, like "Validation" requires the completion of "Order Placement", before it can start. Some activities can proceed in parallel, like "Inventory Check" and "Customer Credit Check".

I.2.2 Basic terms and terminology.

The basic terms of workflow terminology will be defined. The relationships among them are illustrated in next Figure according to [WfMC 99] a workflow consists of a process that is automated.

A work item or data set is created, processed and changed in stages at a number of processing points to meet business goals. Conditions, that can be expressed mathematically or logically, can be managed by a workflow system.

A workflow process is normally based on several logical steps, each of which is an automatic or a manual activity. An activity can involve manual interaction with a user or workflow participant, or the activity might be executed using machine resources.

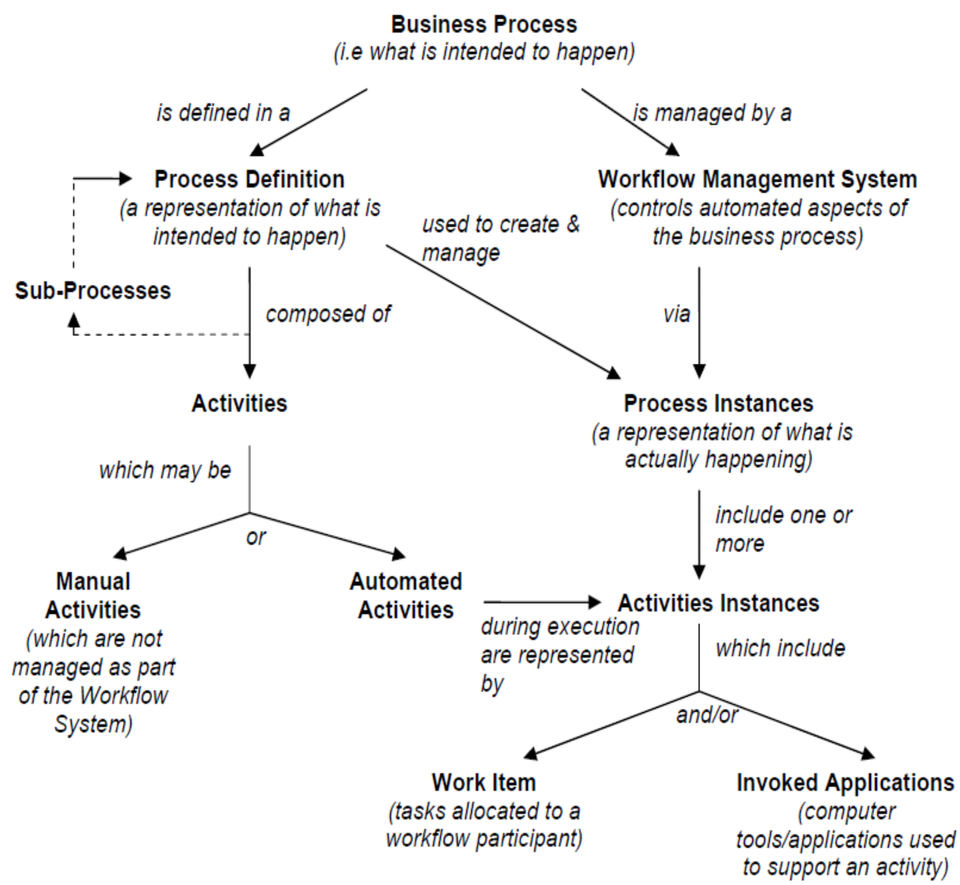


Figure 2. workflow basic terms and relationship

In practice a workflow consists in defined series of tasks (more or less automated) within an organization to produce a final outcome. At each stage in the workflow, one individual or group, or software is responsible for a specific task. Once a task is complete, the workflow software or Workflow Management System ensures that the individuals responsible for the next task are notified and receive the data they need to execute their stage of the process. The system offers the needed interface to set all required information (definition of the individuals or group, of their roles, etc.).

I.2.3 Workflow Management System.

A workflow is created and managed in a workflow management system. WfMC has defined workflow management system in [WfMC 99] as: "A system that defines creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications".

The main goal of a workflow management system is to manage the flow of activities through the workflow system. Users and their roles are managed through the workflow management system. The participation of users are managed by setting deadlines, activity synchronisation and by passing activity data from one participant to another and ensuring that they fulfil their contribution as expected.

I.2.4 Workflow Engine.

A workflow engine is a software service that provides the run-time execution environment for a process instance. A workflow engine may be responsible for the whole run-time process execution, but also for only a part of it.

I.3 Types of workflows.

Several classifications are adopted for workflows, in one classification, workflow systems are segmented into production, administrative, collaborative and ad-hoc, with respect to how they are used and what features they have. Another classification separate workflows into static and adaptive, according to the aptitude of the workflow to react to its environment changes and thus to be redefined at run time.

I.3.1 Ad_hoc, administrative and production workflows.

A first classification of workflows distinguishes between ad hoc, administrative and production workflows according to the degree of repetitiveness and predictability of the workflow tasks and according to how the tasks are performed [Adler 92]. Ad hoc workflows perform office processes, such as product documentation or sales proposals, where there is no set pattern for moving information among people. Ad hoc workflow tasks typically involve human coordination, collaboration, or co-decision thus, the ordering and coordination of tasks in an ad hoc workflow are not automated but are instead controlled by humans.

Administrative workflows involve repetitive, predictable processes with simple task coordination rules, such as routing an expense report or travel request through an authorization process. The ordering and coordination of tasks in administrative workflows can be automated. Administrative WFMS are generally non-mission critical. Finally Production workflows involve repetitive and predictable business processes, such as loan applications or insurance claims. Unlike administrative workflow, production workflows typically encompass a complex information process involving access to multiple information systems. The ordering and coordination of tasks in such workflows can be automated.

I.3.2 Static versus adaptive workflows.

Changes in workflow systems can be static or dynamic (adaptive), [Sadiq et al. 00], Static workflow management systems (static WMS) separate process definition (articulation) from process enactment (activation), and do not handle changes to the definition during enactment. Process definition is the work of process experts, while process participants perform work through workflow clients and invoked applications.

Adaptive WMS [Klein *et al.* 00] [Aalst *et al.* 00] offer important flexibility compared to static systems. Changes to workflow models affect running instances, and exception handling is supported. Adaptive workflow can be considered as intermediate between totally unstructured cooperative work systems such as Computer Supported Cooperative Work (CSCW) and totally structured ones like production-workflows [Aalst 00]. Flexible workflows [Jin *et al.* 08] is an area of research that looks at how conventional systems can be extended and, how static workflow systems can be made adaptive and dynamic.

II. Service oriented business process

Regarding business software development, we observe the emergence of a new and promising paradigm known as service-oriented architectures (SOA) [Demikran *et al.* 09]. The SOA concept promises reuse of components and allows for an easier implementation of communication across heterogeneous platforms and among enterprises based on open and free specifications. A service is a simple or complex task executed within an organization on behalf of a customer [O'Sullivan *et al.* 00]. When new services become available on a technical level they should be transformable into business services and easy integrable into existing business models. Web-Services seem to be a perfect candidate to satisfy these requirements and enable sophisticated SOA applications deployment, Web-Services are modular, distributed applications based on industry standard technologies.

II.1 Web-Services Architecture.

According to [W3C 04] "A Web-Service is a software application identified by an URI, whose interfaces and binding are capable of being defined, described and discovered by XML artefacts and supports direct interactions with other software applications using XML based messages via Internet-based protocols. A Web-Service is a software component that can be connected with other Web-Services in a platform and language independent manner to realize more complex functionality, such as a business processes. Due to the Simple Object Access Protocol (SOAP), they allow communication across heterogeneous platforms, their interfaces are described with the Web-Service Description Language (WSDL) and a repository using Universal Description Discovery and Integration (UDDI) categorizes the Services according to a yellow page approach". Next figure shows in an abstract way the Web-Service architecture.

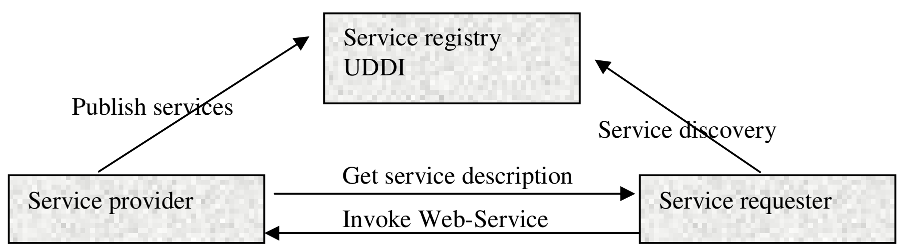


Figure 3. Web-Service architecture

II.1.1 Web-Services interface.

The W3C WSDL group defines WSDL as "an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint." [Christensen *et al.* 01]. WSDL is the interface description for any service that follows the Simple Object Access Protocol (SOAP) [Box *et al.* 01].

II.1.2 Web-Services Composition.

Web-Service composition is an inevitable aspect of Web-Services technology [Karakoc & Senkul 09], which solves complex problems by combining available basic services and ordering them to best suit the problem requirements. Automatic composition gives us flexibility of selecting best candidate services at composition time satisfying quality of service (QoS) requirements; this would require the user to define constraints for selecting and composing candidate Web-Services.

Two major specifications for defining business processes by composing Web-Services emerged in the past : the Business Process Execution Language for Web-Services (BPEL4WS or simply BPEL) which is a language for Web-Services orchestration [OASIS 07] and the Web-Service Choreography Description Language (WS-CDL) [W3C 05]. Orchestration and choreography are two concepts usually confused and in what follows we shall try to present them as clearly as possible.

Orchestration refers to a composed business process that may use both internal and external Web-Services to fulfil its task. The business process is controlled by one of the agents in the system. The process is described at the message level, i.e. in terms of message exchanges and execution order. Choreography addresses the interactions that implement the collaboration between services. Multiple agents are considered where each agent describes its own part in the interaction. Orchestration and choreography address different perspectives. Orchestration is focused on the internal behaviour of a business process. Choreography is focused on the external perspective, looking at process interaction. [Claus & Yaoling 06].

In practice, Web-Services orchestration consists in combining two or more Web-Services to create a new composite one, composition, can be static or dynamic. Static Web-Service compositions are known at design time. Static Web-Service compositions appear currently as the most used Web-Service composition style in both industry and academia [Haas 02]. They are formed by identifying manually (i.e. by human assessment) the applicability of a Web-Service to a particular problem domain. Dynamic Web-Service compositions are one or many compositions in which Web-Services are not known at design time, and which are discovered or their properties resolved based upon a criteria process set at design time. Web-Service orchestration is an executable composition of Web-Services wheareas choreographies are not executable.

II.2 Web-process modelling languages.

A large number of languages have been created to model Web processes.

II.2.1 UML approaches.

Although UML was initially introduced as a modelling language for object oriented software systems, its flexibility and extensibility have attracted business modellers and analysts. In order to describe the behaviour of a business process, UML activity diagrams might be used. In [Kramler *et al.* 05], the authors propose a UML 2-based and platform independent approach for modelling collaborations between Web-Services. In [Gardner 03], a UML profile for BPEL is proposed.

II.2.2 UMM paradigm.

UMM is a UML-based methodology, defined as a UML 1.4.2 profile by the "United Nations Center for Trade Facilitation and Electronic Business" [UN/CEFACT TMG 06] including stereotypes, tagged values and constraints. UMM is a standardized methodology developed by UN/CEFACT and well accepted in the field of B2B modelling. In its early stages, RosettaNet [RosettaNet 02] contributed through its application in the IT, telecommunication, and semiconductor industry to the development of the UMM.

II.2.3 Business Process Modelling Notation (BPMN).

BPMN has been developed by the Business Process Management Initiative (BPMI). Recently, the BPMI contributed their work to the Object Management Group (OMG), as a result of the merger of the business process management activities of those two groups, the BPMN became a final OMG specification [OMG 06]. BPMN provides a small, but clearly defined notation for modelling business processes. The simple notation enhances the understandability of BPMN diagrams among different groups of users [Dijkman *et al.* 08].

II.2.4 Event-Driven Process Chains (EPC).

EPCs are a process-oriented modelling techniques used for the definition of business processes particularly in ERP systems, these methodologies gained a great attention in companies worldwide. ARIS is a tool set that supports besides other modelling approaches the EPC approach and is continuously extended to support recent developments in the IT world [Scheer *et al.* 06].

II.3 Executing Web processes.

II.3.1 BPEL.

In order to implement a business process using Web-Service technology we need to map the flow of a business process model to a set of Web-Service interactions. BPEL has become the de-facto standard for business workflows and is a key element of the Service Oriented Architecture (SOA). [Held & Blocchukger 09] A BPEL process is an orchestration of Web-Services i.e. it describes the Web-Service based process from a specific partner's point of view. We shall discuss details of BPEL structures in Chapter 5.

II.3.2 ebXML.

ebXML is a global electronic business standard that is sponsored by UN/CEFACT and OASIS [OASIS 06]. ebXML thus defines a framework for global electronic business that will allow businesses to find each other and conduct business based on well-defined XML messages within the context of standard business processes which are governed by standard or mutually-negotiated partner agreement. The framework incorporates the concepts of SOA and is the second important approach in this field beside Web-Services.

The ebXML [OASIS 05] framework provides a set of five specifications: Messaging (ebMS), Registry (ebRIM/ebRS), Collaboration Protocol Profiles and Agreements (CPP/A), Core Components (CC) and BPSS. The ebXML messaging [OASIS 06] is defined on top of the SOAP with Attachments (SwA) specification. The SOAP message itself contains technical information for the respective message handlers concerning routing, security, correlation, and reliability, just to name the most important.

The idea of ebXML registries is to provide standardized repositories for managing B2B related content. Such content includes, but is not limited to, business partner profiles and standardized business process descriptions and business documents. In a B2B scenario, registries allow business partners to find each other electronically.

II.3.3 Combining ebxml and Web-Services.

In the Web-Services Programming Model, WSDL is used to describe a Web-Service. In the ebXML specification, on the other hand, CPP is used to describe the same Web-Service. WSDL provides information about a service name, parameters for that service and the endpoint to invoke it. CPP not only produces this information, but also other important parameters, such as the role of an organization in the context of a particular service, error-handling and failure scenarios. In essence, the ebXML business process schema is a more rigorous definition of a Web-Service than simply a WSDL document. It not only identifies business processes but also, for instance, the roles that an organization has to play and messages being exchanged.

UDDI is used in the Web-Services programming model to publish Web-Services to a global UDDI Repository. In ebXML we use a Registry Service Interface to publish an organization's CPP. Once organization 'A' has determined the business processes it can support, it starts building an application to support the understood ebXML standards. This application defines the service interfaces that other organizations can invoke. It also describes the input and output messages that will be given to the service. Organization 'A' already has an Internal Legacy Application, and so all they have to do is create an implementation wrapper around their Legacy Web-Services based Application, to help it understand ebXML messages [Dorn *et al.* 09] .

III. Agent based workflows.

Traditionally, Workflow management systems are used to support static processes, i.e., processes which do not change frequently. This has limited the scope of Workflow management. Moreover, the modern networked economy requires Workflow management systems which are able to deal with dynamically changing workflow processes. Beyond theses orientations, mobility in workflows systems require a lot of flexibility i.e. the capability to react to changes in the workflow during its execution and a great degree of adaptivity. The agent technology seems to be a good candidate to these requirements. Agent systems are by nature distributed, consist of several autonomous entities, each of them possibly developed independently, and with capabilities to communicate in order to achieve a common goal.

III.1 Basic concepts.

An agent is a computer entity situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives. An autonomous agent should be able to act without the direct intervention of humans or other agents, and should have control over its own actions and internal state. [Jennings *et al.* 95]. Depending on the use of the term, there are several attributes assigned to agents. An agent can be characterized by:

- **Autonomy**: Independent acting, he has its own behaviour.
- **Social ability**: The ability to interact with other agents.
- **Reactivity**: The ability to perceive one's environment and react accordingly.
- **Proactivity**: Not only reaction to the environment but the attempt to actively influence it.
- **Mobility**: The ability to move around in a network (across workstations, networks etc.)
- **Adaptivity**: Ability to adapt to new circumstances.
- **Preservation**: Ability to preserve internal state from deactivation to reactivation
- **Rationality**: Agents attempt to find the best way to achieve their goals

In particular BDI-agents (Beliefs-Desires-Intentions) are rational agents with BDI reasoning possibilities as presented in [Bratman 87]. They are equipped with a set of beliefs describing their understanding of his environment (possible states and developments). A subset of all possible states is regarded as desirable (desires). They may be reached either through certain actions taken by the agent or by other circumstances. Sometimes the term "goal" is used to describe those desires. The term intentions finally addresses a set of actions the agent intends to take in order to achieve one or more of his aims. A sequential list of actions is often referred to as a "plan".

III.2 Benefits of agents for workflow systems.

It is now well known that multi-agents systems bring numerous advantages to enterprise information systems at different levels. The multi-agent system has some common characteristics, such as distribution, autonomy, interaction and openness, which are helpful to transform traditional architectures into a distributed and cooperative architecture in an intelligent manufacturing system [Guo & Zhang 09].

Agents are in general a good candidate to SOA applications [Contreras & sheremetov 08] In particular; the advantages of applying agent technology to workflow systems are pointed out in [Jennings *et al.* 00] as follows:

- **Flexibility**: Actions/task-execution can be based on the agent's current situation i.e. local circumstances (e.g. available resources etc.) as opposed to predefining them at design time
- **Agility**: new services/tasks may be added to the workflow without affecting other parts . There is no need to lock the workflow or suspend/cancel its execution. The modifications only have local effects.
- **Adaptability**: As opposed to traditional concepts the agents explicitly allows the workflow to be changed during or after execution. In previous workflow systems these modifications required to change the actual workflow definitions. However in agent-based systems the workflow is dynamically read and transferred to the agents to allow easy modifications while at the same time the agents are enabled to modify the workflow based on feed-back (frequent branches, unreachable states etc.).
- **Component based extensibility** (depending on the architecture): Agents may be placed in different locations, combined, removed or exchanged at will. They do not require continuous access to a central workflow engine. Communication allows multiple agents to be combined in executing a single workflow instance.
- **Event/Exception-Handling**: Many mobile agents systems are explicitly built on event and exception handlers. Whenever an exception occurs during workflow execution or an agent needs to be notified these handlers can be used. To address the deficits in coordination the agent-system may use monitoring-agents with standardized event handlers to call and update the workflow status.
- **Remote Installation**: There is no need to install the workflow system, a present java agent-framework can be deployed throughout the organization and used by the workflow system we well as other agent-based software. The agents can be send to this remote location and commit work there.
- **Support for Mobile Devices**: Mobile devices are characterized by small memory and little cpu-power. Agents generally have a small code-base and carry only small payloads (the workflow) at the same time not requiring complex environments.

III.3 Agent-Based Workflow Systems.

An agent-based workflow is a workflow in which agents perform, coordinate, and support the whole workflow or parts of the workflow. An agent-based WfMS is a set of software agents that manage and coordinate the flow of work defined by a business process [Odgers *et al.* 99]. In an agent-based workflow system, there exist different agent types that manage the workflow. A workflow agent, or sometimes called process agent, is responsible for controlling one workflow instance [Stormer 02]. Using agents to perform WfMS functions is an emerging field where researchers are exploring the ability of agents to improve process integration, interoperability, reusability and adaptability [Huang *et al.* 06]; [Trappey 06], [Trappey, & Lin, 06]. More and more researchers believe that agent technology can provide architectural solutions for integrating multiple heterogeneous workflow system [Kuo 07]. [Huang *et al.* 00] Integrate multi-agent technology to WfMS to facilitate teamwork in collaborative product design, [Xu and Wang 02] uses intelligent multi-agents to monitor the dynamic processes in B2B ecommerce workflow, [Zhuge 03] Define an agent-based mechanism to model, control and manage the cognitive flow process to improve the problem solving ability of a team, [Wang *et al.* 05] Propose a novel workflow monitoring approach based on intelligent agents to perform flexible monitoring tasks in an autonomous and collaborative way in securities trading. [Madhusudan 2005] Embed autonomous agents in a workflow-based distributed systems infrastructure to support industrial design activities. [Huang *et al.* 06] Develop a prototype of agent-based intelligent workflow system for product design collaboration in a distributed network environment [Wang *et al.* 06] Develop an agent-based workflow model to serve inter-enterprise collaboration by applying Web-Service-based technology. The WONDER workflow architecture is also based on agents and deals with the common problems of coordination and security [Filho *et al.* 03].

IV Mobile workflows.

IV.1 Mobile Workflow: definition.

With the advancement and spreading of various mobile technologies and infrastructures, there is increasing demand for mobile users to connect to WFMS. Mobile workflows extend the main business process of the enterprise, since the latter incorporates the functionalities of the first within its own processes out of the traditional boundaries of the organization.

Mobile workflow users can be thin clients such as palm-top computers and PDA connected to their enterprise through wireless network; they can be involved in the main workflow system at any time and at any place and perhaps in a disconnected manner. Mobile Workflows provide traditional business processes with possibilities to manage the execution of business activities by large numbers of users distributed over a wide area and using heterogeneous resources and small devices which are connected to a network only occasionally.

Mobile devices offer some advantages in comparison to the traditional PC based execution of workflow tasks; because of their portability: they enable the ubiquitous and asynchronous execution of workflow tasks. In fact there has been a growing trend towards the adoption of disconnected practices of work. Users connect to a server in their enterprise to load applications and data in their mobile devices from or to request some task to be done, and then they disconnect from the server and work on those applications and data or wait until the task is remotely performed. After the work has been completed they reconnect with the server to send the results of their work or to receive the reply. Mobile Workflow can be defined as the application of wireless data communications technology to remote workers operating over a wide geographical area [Walter *et al.* 04]. Examples of applications that could be managed by mobile workflows include construction systems [Aziz *et al.* 06], M-maintenance systems [Bouchoul *et al.* 07] and E-Health systems [Lu *et al.* 05] [Bouchoul & Mostefai 09].

IV.2 Requirements and related works.

The main restrictions of mobile devices for the execution of workflow tasks are essentially limited bandwidth, limited resources such as CPU power and memory and limited energy supply; so that and in most cases the performances of mobile applications is greatly restricted due to architectural and design considerations and thus very subject to frequent failures.

In such environment, mobile workflows not only inherit the wireless technologies weakness but also are often Internet-based so that mobile users periodically become unavailable due to the lack of network service guarantees. The result is limited or very difficult business information access and activity coordination. The obvious consequence is that traditional workflow management systems have not been designed for dynamic environments requiring adaptive situation induced by mobility.

In the other side, considering the growing need of mobility and the more frequent use that organizations make of mobile devices, it is necessary to provide support for the integration of those devices to the work. Unfortunately, mobile devices run often on many different platforms, which means that their development environments vary widely and mobile workflow users often run into integration problems when attempting to access desktop applications. In other terms, an adequate architecture handling mobile workflow requirements must enable the smooth integration of the mobile workforce within the main business process of the organization and easy mutual synchronization between workers with high fault tolerance mechanisms.

Thus mobile workflow application need convenient, efficient and robust paradigms suitable for distributed applications, even when partially connected computers are involved. Finally mobile workflow applications can be optimized by implementing minimal tasks on implied mobile devices and centralizing the most resources consuming in the desktop infrastructure of the enterprise.

Recently some systems have been developed to address mobility explicitly; a number of systems such as TOXICFARM [Godart *et al.* 04] MOBIWORK [Hackmann *et al.* 06], and DOORS [Preguiça *et al.* 05], are typical examples. DOORS and TOXICFARM are two architectures for cooperative synchronous work in mobile environments. These approaches adapt groupware applications to mobility by supporting workflows in the face of network disconnection. The work environment called workspace in both approaches can be replicated and processed in a disconnected manner. In fact the disconnected mode is not new: Exotica proposed in [Alonso *et al.* 95] is an approach based on a centralized workflow model called FlowMark. FlowMark is centred on an object oriented database and its components can be distributed across heterogeneous systems. FlowMark supports also disconnected clients. The basic idea is to provide the clients with enough information to allow them to proceed without having to consult with the server after every step. WHAM (workflow enhancements for mobility) [Jing *et al.* 00] supports mobile workforce and applications in workflow environment. The prototype is implemented with a client-server architecture. On the client side, a specialized Java worklist handler provides an interface for mobile workers to select and reserve work items from a list of performable tasks in a workflow process, and to invoke the program activities associated with them.

On the server side sits a production workflow server (i.e., IBM's MQSeries Workflow server) with additional functionality to track the location of mobile workers and extra adaptive work activity assignment services that affect mobile workers' worklists.

Magi [Bolcer 00] is an architectural framework that explicitly addresses the coordination of E-business messaging and deployment across a range of computing platforms. The Magi architecture has two key parts: A micro-Apache HTTP server which is a scaled-down, low-memory-footprint version of an Apache HTTP for mobile devices and An extensible generic interface. Magi servers can capture and influence several types of events. To create an event, they can register actions of another Magi server going on- or offline, publishing a document, or even simply downloading a file. Further, users may register a set of workflow actions with an event. Events can trigger onetime actions or can perform recurring workflows to synchronize data or collect and assemble the latest information from a variety of Web destinations.

AWA [Stormer and Knorr 01] is also architecture for flexible workflow systems, which can be distributed and can deal with various levels of adaptability from a process perspective, resource perspective and task perspective. AWA is agent-based and enable dynamic discovering and connecting to Web-Services. The AWA/PDA (Agent-based Workflow Architecture for Personal Digital Assistants) prototype proposes a model where PDAs can be used to execute workflow tasks; it makes use of mobile agents, which travel to PDAs, and allows for an independent and asynchronous execution of workflow tasks. The prototype is JAVA based and the JAVA versions supported is Personal JAVA. Personal JAVA is the old JAVA Virtual Machine for handled devices and it is now obsolete since replaced by J2ME. The system adopts GRASSHOPPER, as mobile-agents technology. But only certain workflow tasks should be completed on PDAs; the main limitations being the bandwidth and technical resources of the PDA such as the small display and storage capacity.

In [Neyem *et al.* 07] an extension to the traditional workflow model is proposed in order to support dynamic settings and Specifically to deal with mobile devices interacting among themselves using a Mobile Ad-hoc The architecture supports mobile workflows by enabling distributed workflow execution and taking advantage of the concept of Active Entity (AE), which is a building block to design abstract models of collaborative BPs. AEs are designed as abstract definitions for each kind of role involved in the process.

IV.3 Some Mobile workflows architectures.

As presented above, there are only few workflows systems dealing with mobility, in this section we shall present architecture of four of them namely Awa, Mobiwork, FlowMark and Active entities based workflow.

IV.3.1 Awa.

AWA [Stormer & Knorr 01] is an agent-based workflow system, there exist different agent types that manage the workflow. A workflow agent is responsible for controlling one process instance. AWA differentiate the following agent types:

Workflow Agent (WA) The WA is created by a subject, which has to provide a complete and correct process definition. The workflow agent represents and manages an instance of a workflow and controls its whole execution.

Process Agent (PrA) For each process definition, a PrA is created. It can be seen as a repository which contains one or more templates of a process definition. The PrA receives a process state, containing a list of already executed tasks from the WA, and returns a list of tasks that have to be executed next. With the introduction of the PrA, different process specification languages can be used within one workflow system.

Task Agent (TA) The TA is responsible for one task in an instance of a process. It is created by the WA and has to search for a subject, deliver the task description and objects to the subject and the results back to the WA.

Worklist Agent (WlA) The WlA stores the organization structure of the workflow environment, i.e. A mapping of all subjects and their assigned roles.

Personal Agent (PA) A personal agent is the interface between the subject and the incoming TAs. It is immobile, controls the incoming task-requests from the TAs, and coordinates the communication between subject and TA. The basic idea in the handling of a workflow task is to create a TA for each task in the underlying process instance.

Mobile Agents AWA/PDA makes use of mobile agents, which travel to PDAs, and allows for an independent and asynchronous execution of workflow tasks.

IV.3.2 Mobiwork.

The MobiWork system [Hackmann *et al.* 01] runs on PDAs. The system is differentiated into four kinds of components based on their roles: (1) planning components, (2) execution components, (3) common components, and (4) external components.

External Components. The external components represent components that are not part of the MobiWork implementation but form an integral part of the overall system.

The Planning Application The planning application is used to inject the plan for the activity to be completed, into the system. User Applications are instantiated by a person using MobiWork in order to assist him or her in completing an assigned task.

Common Components. The common components of MobiWork represent centralized resources that are used by both the planning and execution components of the system and the components which form the bridge between the planning and execution roles of the system.

The Workflow Manager is the central component of the MobiWork system managing both planning and execution activities. During the planning stage, the Workflow Manager accepts the plan from the planning application and passes it on to the planner for allocation. It reports the allocation information to the monitoring applications (if they are instantiated).

The Plan Manager stores and maintains the plan. The plan manager may contain the entire plan or a portion of the plan which reflects the tasks that have been allocated to the particular member(s).

Planning Components. The planning components of the MobiWork architecture are responsible for allocating the tasks in the plan among available members and distributing the allocation information to the relevant members. The Planner is responsible for coordinating these activities. The Workflow Manager initially provides the planner with the plan, which it stores within the Plan Manager.

The Distribution Manager sends the pieces of the plan to the recipients using the Communication Middleware.

Execution Components. The execution components of MobiWork support the actual execution of tasks in the plan. The Executor is the main coordinating entity during the execution of a plan. It is responsible for setting up events.

IV.3.3 FlowMark.

FlowMark [Leymann & Roller 94] is an IBM Workflow project, later in the Exotica Project, FlowMark WFMS and the Lotus Notes replicated file system have been coupled. The Exotica project extended the FlowMark WFMS to support a distributed client-server architecture based on a queue messaging mechanism. FlowMark supports disconnected mode so that mobility is enabled. The main concepts in FlowMark are as follows.

Activities: Activities are the fundamental elements of the model. An activity represents a business action refinement of activities via process models allows for both, modelling business processes bottom-up and top-down.

Containers: The results that are in general produced by the work represented by an activity is put into an output container, which is associated with each activity. Since an activity will in general requires to access output containers of other activities, each activity is associated in addition with an input container too.

Control connectors: Since activities might not be executed arbitrarily they are bound together via control connectors. A control connector might be perceived as a directed edge between two activities; the activity at the connector's end point cannot start before the activity at the start point of the connector has finished (successfully). Control connectors model thus the potential flow of control within a business process model.

Transition conditions: In addition, a Boolean expression called transition condition is associated with each control connector. Parameters from output containers of activities having already produced their results are used as parameters referenced in transition conditions.

Exit conditions: Termination of an activity does not necessarily indicate that the associated task has been finished successfully. In order to allow the measurement of successfulness of the work performed by an activity, a boolean expression called "exit condition" is associated with each activity. Exactly the activities the "exit condition" of which evaluated to 'true' in the actual context are treated as successfully terminated.

Synchronization conditions: Multiple control connectors leaving a certain activity represent parallelism of the activities targeted to by the connectors. The semantics of multiple control connectors pointing to one and the same activity can be defined via a synchronization condition associated with this target activity.

A synchronization condition specifies when the execution can continue: either all of the transition conditions of the incoming control connectors became 'true' or at least one of them became 'true'.

Data connectors: Parallelism of work requires some coordination of the data produced or consumed by activities running in parallel. A directed edge called data connector pointing from a particular parameter type of an activity's output container to a particular parameter type of another activity's input container may be specified. At run time, the corresponding instances are passed along the data connectors.

Tasks: A business process model encompasses the description of the flow of the activities itself between "resources" actually performing the pieces of work represented by each activity. This is done by coupling each activity with a resource resulting in a pair called task. A resource may be specified as a particular person, a role, or an organizational unit. At run time tasks are resolved into requests to particular persons to perform particular activities. Resources are the means to distribute activities to the right people in the sequence prescribed by the control flow aspect of a business process model.

IV.3.4 Active entities based workflow.

Active Entities are the main components of the system [Neyem *et al.* 07], they are service providers which by means of their public service interfaces, are able to accomplish some execution units, by providing and consuming thirdparties (AE) services. Those services are modelled at design time and, then, they are encapsulated in an abstract class that will be instantiated when needed. The architecture consists of a set of components extending traditional server-based workflow engines to be accessible by mobile workers. The architecture has to consider thin clients due to the hardware limitations of mobile devices.

AEs Process Modeller: This component supports the process definition including modelling of activities assigned to active entities, control connectors among AEs, input/output containers and entity contracts.

AEs Process Manager: After the planning stage, this component accepts the plan from the AEs Process Modeller and passes it on to the planner for allocation of AEs. When a process is executed, activity instances are assigned to AE to perform the actions required for the activity. This may be a manual task, or an automatic one.

µWorkflow Manager: This component stores the plan for the active entities assigned to a mobile worker. This Manager uses the µActiveEntity and µTask Manager components to handle the active entities and tasks that have been allocated. This manager needs data about the plan for allocating tasks to determine (1) the active entity from which the inputs for a task are going to come and (2) the active entity to which the results will be returned.

µActiveEntity Manager: This component handles the active entities assigned to a mobile worker. This component needs status information about the tasks allocation for determining the active entity in order to perform the required actions.

µTask Manager: This component handles the task state transition of an active entity. A task being executed in the disconnection mode may change its state according to the current user's situation. For example, a user may terminate or suspend his/her task according to the surrounding business situation. On the other hand, there is a need to handle predictable task state mismatch when reconnected.

SOMU Platform: The Service-Oriented Mobile Unit (SOMU) is a lightweight platform running on PDAs, TabletPC and notebooks. It enables each mobile computing device to produce and consume Web-Services from other peers. Such functionality is implemented on a lightweight Web Server. Thus, the autonomy and part of the interoperability required by mobile workers is supported.

REFERENCES

[Aalst *et al.* 00] W.M.P. van der Aalst, T. Basten, H.M.W. Verbeek, P.A.C. Verkoulen, and M. Voorhoeve. "Adaptive Workflow: On the Interplay between Flexibility and Support". In J. Filipe, editor, Enterprise Information Systems, pp 63-70. Kluwer Academic Publishers, Norwell, 2000.

[Adler 92] P. S. Adler, and T. A. Winograd, Usability – "Turning Technologies into Tools." Oxford University Press, New York, USA, 1992.

[Alonso *et al.* 95] G. Alonso, R.Gunthor, M.Kamathy, D.Agrawalz, A.El Abbadiz and C. Mohan, 1995, "Exotica/FMDC Handling Disconnected Clients in a WorkFlow Management System", Third International Conference on Cooperative Information Systems Vienna 1995

[Aziz *et al.* 06] Z. Aziz, C.J. Anumba, D. Ruikar, P. Carrillo and D. Bouchlaghem, , "Intelligent wireless Web-Services for construction A review of the enabling technologies" Automation in Construction, elsevier Vol. 15[2] pp 113-123 2006

[Bolcer 00] G. Bolcer, "Magi: Architecture for Mobile and Disconnected Workflow", IEEE Internet Computing, Vol. 4[3] pp 46-54 2000

[Bouchoul *& al.* 07] F. Bouchoul, M. Mostefai and Z. Kaddem, "Towards M-Manufacturing - Case Study: a Fault Tolerant Model for M-Maintenance of Spare Parts with Mobile Agents", in Proc. of IEEE: 4Th International conference: Sciences of Electronic, Technologies of Information and Telecommunications Setit'07 marsh hamamet Tunisia 2007

[Bouchoul & Mostefai 07] F. Bouchoul and M. Mostefai, "MOBIFLEX: A Generic Architecture for Integrated M-Health Applications" , in Proc. of IEEE: First international conference on E-medical systems : E-medisys'07 October, FEZ Morocco 2007

[Bouchoul & Mostefai 09] F. Bouchoul and M. Mostefai "Agent-services and mobile agents for an integrated HCIS" International Journal of Computer Integrated Manufacturing, (IJCIM) Taylor and Francis editor Vol. 22[5] pp 458-471 may 2009

[Box *et al* 01] D. Box, D. Ehnebuske, G. Kakivaya, and A. Layman. "Simple Object Access Protocol (SOAP) specifi-cation". http ://www.w3.org/2000/xp/Group/, W3C SOAP WORKING GROUP May 2001.

[Bratman 87] M. E. Bratman. "Intentions, Plans, and Practical Reason". Harvard University Press: Cambridge, MA 1987

[Claus & Yaoling 06] P. Claus and Z. Yaoling "A Semantical Framework for the Orchestration and Choreography of Web-Services" Electronic Notes in Theoretical Computer Science Vol. 151[2] pp 3-18 2006

[Christensen *et al.* 01]. E. Christensen, F. Curbera, G. Meredith and S. Weerawarana, "Web-Services Description Language (WSDL) 1.2, Note NOTE-wsdl-20010315", March 2001. available at http://www.w3.org/TR/2001/NOTE-wsdl-20010315

[Contreras & sheremetov 08] M. Contreras and L. Sheremetov "Industrial application integration using the unification approach to agent-enabled semantic SOA" Robotics and Computer-Integrated Manufacturing, Vol. 24[5] pp 680-695 October 2008

[Demikran *et al.* 08] H. Demirkan, R.T.J. Kauffman, J. A. Vayghan, H-G. Fill, D. Karagiannis, and P.P. Maglio "Service-oriented technology and management: Perspectives on research and practice for the coming decade" Electronic Commerce Research and Applications, Vol. 7[4] pp 356-376 2008

[Dijkman *et al.* 08] M. R. Dijkman, M. Dumas and Ch. Ouyang "Semantics and analysis of business process models in BPMN" Information and Software Technology, Vol. 50[12] pp 1281-1294 November 2008

[Dorn *et al.* 09] J. Dorn Ch. Grun H. Werthner and M. Zapletal "From business to software: a B2B survey" in Information System and E-Business Management Vol. 7 pp 123–142 elsevier edition 2009

[Godart *et al. I* 04] C. Godart, P. Molli, G. Oster, O. Perrin, H. Skaf-Molli, P. Ray, and F. Rabhi , "The toxicfarm integrated cooperation framework for virtual team", Distributed and Parallel Databases, Vol 15[1] 67–88 2004

[Filho *et al.* 03] R. S. S Filho, J. Wainer and E. R. M. Madeira, "A Fully Distributed Architecture for Large Scale Workflow Enactment". In International Journal of Cooperative Information Systems, Vol. 12[4] pp 411-440 2003.

[Gardner 03], T. Gardner "UML modelling of automated business processes with a mapping to BPEL4WS" In: 1st European workshop on object orientation and Web-Services at ECOOP 2003 Springer 2003

[Jennings *et al.* 95] N.R. Jennings, J.M. Corera, and I. Laresgoiti (1995). "Developing Industrial Multi-Agent Systems". In Proc. of ICMAS'95, San Francisco, The AAAI press/The MIT press, pp 423-430

[Guo & Zhang 09] Q. Guo and M. Zhang, "A novel approach for Multi-Agent-Based Intelligent Manufacturing System", Information Sciences, doi: 10.1016/j.ins.2009.05.009 published online may 2009

[Haas 02]. H. Haas, Web-Services Activity - W3C Web-Services Activity Group. http://www.w3.org/2002/ws/Activity 2002.

[Hackmann *et al.* 06] G. Hackmann, R. Sen, M. Haitjema, G. C., Roman and C. Gill, , "MobiWork: Mobile Workflow for MANETs", Technical Report WUCSE-06-18, Washington 2006 University, Department of Computer Science and Engineering, St. Louis, Missouri.

[Held & Blochukger 09] H. Markus, B. Wolfgang "Structured collaborative workflow design" Future Generation Computer Systems vol. 25 pp 638-653 2009

[Huang *et al.* 00] C. J. Huang, , A. J. C. Trappey and Y. H. Yao, "Design and development of a WfMC-based modularized network collaborative design system". Journal of Chinese Institute of Industrial Engineers, Vol. 20[4] pp 422–434 2003

[Huang *et al.* 06] C. J. Huang, A. J. C. Trappey and Y. H. Yao, "Developing an agent-based workflow management system for collaborative product design". Industrial Management and Data Systems, Vol. 106[5] pp 680–699 2006

[Jennings *et al.* 00] N. R. Jennings, T. J. Norman, P. Faratin, P. O'Brien, B. Odger: "Autonomous Agents for Business Process Management". In International Journal of Applied Artificial Intelligence Vol. 14[2] pp 145-189 2000.

[Jing *et al.* 00] J. Jing, K. Huff, B. Hurwitz, H. Sinha, B. Robinson and M. Feblowitz, "WHAM: Supporting Mobile Workforce and Applications in Workflow Environment", 10th Int. Work. On Res. Issues in Data Eng., San Diego, pp 31-38 February 2000

[Karakoc & Senkul 09] E. Karakoc, P. Senkul "Composing semantic Web-Services under constraints" Expert Systems with Applications Vol. 36 pp 11021–11029 Elsevier 2009

[Klein *et al.* 00] M. Klein, C. Dellarocas and A. Bernstein, "Special Issue on Adaptive Workflow Systems", Computer Supported Cooperative Work, vol. 9[3,4] 2000.

[Kramler *et al.* 05] G. Kramler, E. Kapsammer, W. Retschitzegger and G. Kappel: "Towards using UML 2 for modelling Web-Service collaboration protocols" . In: Proc. of the Con-ference on Interoperability of Enterprise Software and Applications (INTEROP-ESA), Springer London pp 227-238 2005

[Kuo 07]. M. H. Kuo,. "An intelligent agent-based collaborative information security framework". Expert Systems with Applications, Vol. 32[2], pp 585–598. 2007

[Leymann & Roller 94] F. Leymann and D. Roller, "Business Process Management with FlowMark", Digest of papers, Cat. No. 94CH3414-0, Spring COMPCON 94, pp 230-234 1994

[Lu *et al.* 05] Y-C Lu, Y. Xiao, A. Sears and J. A. Jacko, "A review and a framework of handheld computer adoption in healthcare", International Journal of Medical Informatics, Vol. 74[5] pp 409-422 2005

[Jin *et al.* 08] L. Jin Z. Xiao-Fang W. Xin-Cai and Z. Cheng "Research on a Flexible Workflow Management System Based on Web-Services" in: Information Processing (ISIP), 2008 International Symposiums on Publication 23-25 May 2008 China Univ. of Geo Sci., Wuhan;

[Madhusudan 05] T. Madhusudan "An agent-based approach for coordinating product design workflows". Computers in Industry, Vol. 56 pp 235–259 Nwana, H. S. 1996

[Neyem *et al.* 07] A. Neyem, D. Franco, S. F. Ochoa, J.A. Pino "Supporting Mobile Workflow with Active Entities". Proc. of CSCWD 2007. 11th International Conference on Computer Supported Cooperative Work in Design, 2007 26-28 pp 795 – 800 April 2007

[OASIS 05] OASIS (2005) ebXML registry information model. Version 3.0

[OASIS 06] OASIS (2006) OASIS ebXML messaging services. Version 3.0, Public Review Draft 01

[OASIS 07] OASIS (2007) Web-Services business process execution language. Version 2.0

[Odgers *et al.* 99], B. R. Odgers, J. W. Shepherdson and S. G. Thompson "Distributed workflow co-ordination by proactive software agents". In Proc. of intelligent workflow and process management, IJCAI-99 Workshop. 1999

[OMG 06]. Object Management Group (OMG) (2006) Business process modelling notation specification. Version 1.0

[O'Sullivan *et al* 00] J. O'Sullivan, D. Edmond and A.H.M. Ter Hofstede Service description: a survey of the general nature of services. report FIT-TR-2003-02. 2002. available at http://www.service-description.com/papers/tr-dapd-osullivan.pdf 2000

[Preguiça *et al* 05] N. M. Preguiça , J. L. Martins, H. J. L. Domingos, and S. Duarte, Integrating synchronous and asynchronous interactions in groupware applications., LNCS 3706 pp 89–104 2005

[RosettaNet 02]RosettaNet (2002) RosettaNet implementation framework: core specification. V02.00.01

[Sadiq *et al* 00] S.W. Sadiq, O. Marjanovic and M.E. Orlowska, "Managing change and time in dynamic workflow processes", International Journal of Cooperative Information Systems, 9(1-2):93-116, World Scientific Publishing Company, 2000-09-13

[Stormer & Knorr 01] H. Stormer and K. Knorr: "PDA- and Agent-based Execution of Workflow Tasks". In: Kurt Bauknecht, Wilfried Brauer and Thomas A. Mück (eds), Proc. of the GI Jahrestagung 2001, ISBN 3-85403-157-2, Vienna, Austria, pp 968-972. September 2001

[Stormer 00] H. Stormer "Task Scheduling in Agent-Based Workflow" in Proc. of International ICSC Congress on Intelligent Systems & Applications (MAMA'2000: Multi-Agents and Mobile Agents in Virtual Organizations and E-Commerce). December 11-13, Wollongong, Australia 2000

[Trappey 06] A. J. C. Trappey, C. V.Trappey, J. L. Hou and B. J. G. Chen "Mobile agent technology and application for online global logistic services". Industrial Management and Data System, Vol. 104[2] 169–183 2004

[Trappey, & Lin 06]. A. J. C. Trappey, C. V. Trappey and F. T. L. Lin . "Automated silicon intellectual property trade using mobile agent technology". Robotics and CIM, Vol 22 pp 189–202 2006

[UN/CEFACT TMG 06] UN/CEFACT TMG (2006) UN/CEFACT's Modelling methodology (UMM), UMM Meta Model—foundation module. Technical specification, Version 1.0

[W3C 05]. W3C (2005b) Web-Services choreography description language. Candidate recommendation, Version 1.0

[W3C 04] W3C, Web-Services Architecture, W3C Working Group Note 11 February 2004 http://www.w3.org/TR/ws-arch/wsa.pdf [Accessed September 10 2007]

[Walter *et al* 04] T. Walter, L. Bussard, Y. Roudier, J. Haller, R. Kilian-Kehr, J. Posegga and P. Robinson, "Secure mobile business applications – framework, architecture and implementation", In information Security Technical Report, Vol. 9[4], PP 6-21 2004

[Wang *et al.* 05] M. Wang, , H. Wang, and D. Xu "The design of intelligent workflow monitoring with agent technology. Knowledge-Based Systems", Vol. 18 pp 257–266 2005

[Wang *et al.* 06] S. Wang, W. Shen, and Q. Hao . "An agent-based Web-Service workflow model for inter-enterprise collaboration". Expert Systems with Applications, Vol. 31[4] 787–799 2006

[WFMC 99] "Workflow Management Coalition Terminology & Glossary" Document Number WFMC-TC-1011 February 99 available at http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf

[Xu and Wang 02] D. Xu, and H. Wang "Multi-agent collaboration for B2B workflow monitoring". Knowledge-Based Systems, Vol. 15 pp 485–491 2002

[Zhuge 03] H. Zhuge, "Workflow and agent-based cognitive flow management for distributed team cooperation". Information and Management, Vol. 40 pp 419–429 2003

CHAPTER III

TOWARDS AN ARCHITECTURE FOR MOBILE WORKFLOWS.

The purpose of this chapter is to propose the design of MOBIFLEX a generic architecture for mobile workflows. For the solution we propose an architecture integrating mobile agents, static agents with intelligent capabilities and Web-Services, since these new technologies when put together can generate a flexible, reconfigurable, adaptive and integrable framework to fulfil easily requirements of mobile workflows. The architecture is empowered by a fault-tolerance mechanism and the choice of mobile agent is validated analytically. Finally we explain how this architecture can be implemented on JADELEAP/JADEX platform and as a case study we show how MOBIFLEX can be a solution for M-Health information systems.

I MOBIFLEX architecture.

Comparing to related works, MOBIFLEX architecture brings together some features that seems very promising, for example the disconnected mode adopted in many works and which can be a good solution for cooperative work in mobile environment such as mobile groupware applications is in our sense not very realistic for workflows, since a workflow is assumed to be a well established enactment of interdependent tasks that have to be done in a fixed order. In MOBIFLEX the workflow is enacted in the proper order by a mobile agent. In the other hand MOBIFLEX is powered by a fault tolerance mechanism which is missing in the works cited in the previous chapter.

In MOBIFLEX we try to resolve limitations of mobile devices by a judicious combination of PCs and mobile devices since hard tasks are not executed on PDAs but on fixed nodes in the network and the PDA is used to launch the execution of the workflow and receive the results. MOBIFLEX combines in a convenient manner PCs and mobile devices so that the mobile device is used with respect to its limited resources.

Finally compared to other approaches MOBIFLEX combines together mobile agents, intelligent agents and Web-Services in the same workflow architecture. Agents systems and Web-Services when used together are suitable candidates to face most of mobile workflows issues. the association agent/Web-Services enhances MOBIFLEX architecture with more flexibility and adaptability.

At implementation level MOBIFLEX needs a development framework enabling not only the deployment on both desktop and mobile devices but also the usage of mobile agents, static agents and intelligent agents. The choice of JADE [JADE 07] and JADEX [Braubach *et al.* 04] as development environment for MOBIFLEX seems in our sense a good choice since this association is perhaps the unique homogeneous solution to all these requirements together.

I.1 MOBIFLEX key elements.

The key elements of our architecture can be enumerated as follows:

● It combines static agents and mobile agents so that usage of agents is optimised. The multi-agent system approach (MAS) is an excellent way to develop large distributed systems with easy integration and maintenance. Generally as explained in the previous chapter, agents exhibit at last three important general characteristics autonomy, adaptation, and cooperation. Applying agent technology to mobile workflow systems can easily fulfil dynamicity, flexibility and adaptability requirements. Dynamicity is achieved by the fact that agents allow easy modifications of the workflow during or after execution, agents are naturally adaptive and can easily have a flexible and intelligent behaviour. In MOBIFLEX Intelligent agents are used to handle complex functions offering flexible and reconfigurable possibilities.

Agent mobility adds better performances to mobile workflows. A mobile agent can autonomously migrate from one agent platform to another to interact with other agents and to do specific tasks; it can for example, perform local processing, or retrieve information and bring back the results. Mobile agents are advantageous in particular in mobile environments where there is intermittent connectivity, low bandwidth and limited local storage; and for information retrieval in heterogeneous networks with local real-time interaction requirements. Furthermore, according to [Lange & Oshima 99] there are seven good reasons to use mobile agents: They reduce the network load; they overcome network latency; they encapsulate protocols; they execute asynchronously and autonomously; they adapt dynamically; they are naturally heterogeneous, and finally they are robust and fault tolerant.

The workflow can be enacted in a centralized manner by a sequence of remote procedure calls (RPC) from an agent residing in the mobile device. Agents communicate via Acts Communication Language (ACL) such as KQML and Fipa-ACL. A static agent can invoke remotely a Web-Service via XML (via SOAP or XML_RPC for example).

XML files can be generated by a marshalling of ACL messages into SOAP messages, but unfortunately, in a mobile environment it seems that XML protocols are too "heavy" for mobile devices. The overhead involved in XML processing presents a huge problem for mobile Web-Services applications. In [Adaçal & Bener 06] mobile-agents are introduced to resolve this problem.

Furthermore, many works proved that agent technology in general outperforms client server technology [Patel & Garg 05] and that mobile-agents systems can in most conditions outperform static agent systems [O'Malley *et al.* 00]. For the special case where an interaction is initiated from a mobile device to a sequence of fixed sites it was shown analytically in [Bouchoul *et al.* 07] that we have better results if the itinerary is executed by a mobile agent rather than by a sequence of centralized RPC from a static agent residing on the mobile device. In the centralized approach a bottleneck is caused and the number of low-bandwidth links is larger than in mobile-agent based approach.

● It is fault tolerant since migration of agents can induce real problems when altered. A first level of fault-tolerance in MOBIFLEX architecture is obtained from the usage of mobile agents themselves; for example in the case of disconnections, the operation can resume without problems after its establishing. But this is not enough, if a mobile agent reaches site n and site n fails or the mobile agent itself crashes, all execution results that occurred between sites 1 and n are lost with the mobile agent and the execution must be performed again which can be catastrophic. In MOBIFLEX We intend to use Checkpointing mechanisms proposed in [Osman *et al.*04].

● It is service oriented to achieve high interoperability for disparate systems: Recently Web-Services technologies had added new possibilities to workflow systems. Web-Services can work across organizations and be composed to create new complex Web-Services. Consequently, automated Workflow activities can be implemented as services, and distributed services can be replaced or modified to improve performance or quality without the need to change the business process. Furthermore the prevailing opinion today is that Web-Services paradigm will become the dominant form of distributed computing within this decade and beyond, business processes are becoming more and more service-oriented and workflows can benefit greatly from Web standards like BPEL and SOAP.

In MOBIFLEX Web-Services can be used as the main integrating tool not only at intra-enterprise level but at inter-enterprise level too. It becomes possible to connect different information subsystems and even legacy ones in a homogeneous manner. In another side, although the mobile workflow is simply an extension of the enterprise business process, in some cases its scope goes beyond the enterprise borders, for example if implied in a virtual enterprise or an extended one, but since the mobile workflow is enacted by a mobile agent and this one can migrate only among sites of its platform, the mobile workflow has to be interfaced to the external partners by some brokering mechanism. By this way the integrity of the enterprise can be preserved through secured interactions and the interoperability with heterogeneous environment becomes feasible. A solution can be achieved by creating specific local Web-Services interfacing the information system of the enterprise to external ones, these gateway Web-Services can be complex composition of local and external Web-Services or can provide only one functionality which is the invocation of some relevant external information resource that mobile agents can not reach without technical issues or security problems.

I.2 MOBIFLEX multi-agents system.

The architecture takes into account the characteristics of mobile devices so that a minimal configuration is deployed on such devices when the core components of the system are on the static platform. The architecture proposed for managing workflows in MOBIFLEX is composed of three types of agents as depicted in (Figure 1).

● A workflow manager (WF-Manager): the WF-Manager is the core component of the mobile workflow engine on the mobile device. The WF-Manager in this scenario takes full responsibilities of a workflow management system, which means that it has possibilities to oversee its workflow.

● An agent planner (PL-Agent): It is an intelligent agent which resides on the main system in a fixed server. IA mechanisms and performances metrics can be used by the PL-Agent to choose the most appropriate and most coherent sequences of Web-Services composing the workflow to be enacted. When the WF-Manager is initiated by the user by a specific request, it sends the request to the PL-Agent that returns details of the workflow susceptible to achieve the goal.

● A mobile worker agent (M-Worker): the M-Worker is a mobile agent that enacts effectively the workflow. To enact the workflow the WF-Manager sends an M-Worker to execute it. The M-Worker carries with him an itinerary which offloads the execution sequence for tasks. The WF-Manager can launch multiple mobile agents for a given workflow which are not dependent upon each other and can, therefore, be executed in parallel.

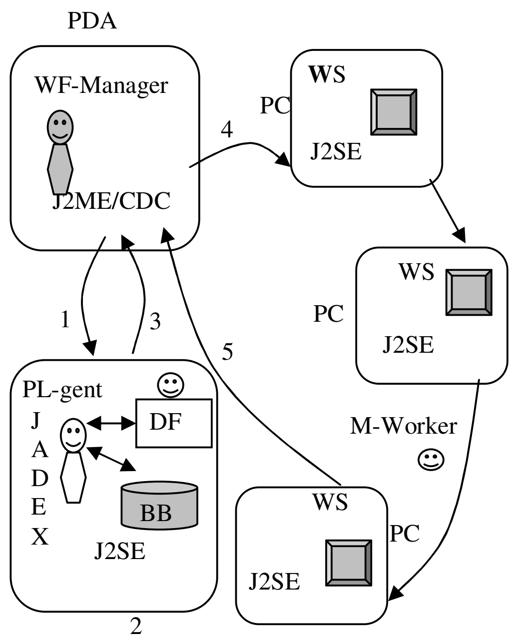


Figure 1. MOBIFLEX Abstract architecture

I.3 Fault tolerance model.

In MOBIFLEX, prior to each M-Worker migration, an archival copy of the agent is stored at the site from which it is migrating. One way the WF-Manager can recognize the failure is based on expiration of the duration for the M-Worker to complete the sequence or to send a notification when needed (i.e., a timeout occurs). The late can be caused either by a crash of the M-Worker, or by some unavailability of the destination. When this occurs, the WF-Manager can recover the last known copy of the M-Worker through a search across the sequence path as presented in [Bouchoul *et al.* 07] and then do suitable actions to resume the workflow by reactivating the copy of the M-Worker, and eventually changing its itinerary. (See the algorithm in listing 1).

```
begin
 Send an M-Worker at time T0
while workflow not completed do
   -Compute the approximate duration D to
    Complete the itinerary
    /* L: Admissible timeout percentage
    -at time T > T0+D(1+L)
    if the M-Worker is back with result
       -Do nothing.
    else /*failure
       -Perform a binary search to locate the site
       of the last correct M-Worker's copy.
       -Identify the cause of late.
       if the M-Worker
          - Reactivate the M-Worker copy.
       else
          If insolvable next step
             -Update the M-Worker itinerary.
             -The M-Worker resumes its itinerary
          else /* The user
             While possible
                -Notify the user
             end while
             if not ok/* (no reply from user)
                -abort
             endif
          endif
       endif        .
    endif
end while
end
```

Listing 1. Fault tolerance algorithm

II Validating MOBIFLEX : A performance analysis model.

We think about a scenario where workflow management in a mobile environment is under control of a classical centralized Workflow management system (WFMS) engine (figure 2) . Communication are generally done through RPC protocols which are not well suited for mobile devices. Mobile agent's technology as adopted in MOBIFLEX seems to be a good candidate to resolve the problem. In this section we propose a performance model for MOBIFLEX workflows. The objective is to compare analytically performances of MOBIFLEX workflows and classical centralized WFMS, in the other hand we provide a mathematical model for the evaluation of the system under various considerations by fixing certain parameters as constants and making others variable. This analysis is based on some works namely [osman *et al*. 04] [patel & Garg 05] [Straber & schwehm 97].

In [Cardoso *et al.* 04] mathematical methods have been used to analyze and estimate the overall quality of service (QoS) of Web-Services based processes. Our formulation takes into account the migration of the mobile agent and the invocation of Web-Services in a single mathematical model.

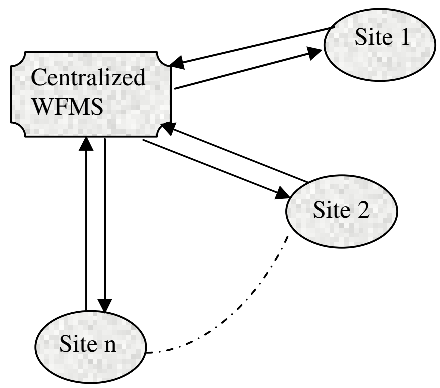


Figure 2 Centralized WFMS

II.1 Comparing the network load.

At each stage i a request of size β_r is sent to the concerned node i and a reply of size β_p is received The network load (in bytes) for a simple remote procedure call from the mobile device to a node S_i consists of the size of the request B_r and the size of the reply B_p received and thus the total load for the n interactions can be calculated as the sum of all the requests sent and the replies received.

$L_{static} = n(\beta_{r+}\beta_p)$ (1)

In the case of MOBIFLEX interactions , The network load for the migration of an agent *A* from a site S_1 to a different site S_2 is calculated as follow.

$$L_{mobile}(S_1, S_2, A_S) = Code_S + 2State_S + 2data_S + (1-\sigma)\beta_p \qquad (2)$$

Where $code_s$ is the size in kilo-bytes of code, $state_s$ the size of execution state of the agent, $data_s$ the size of data transported by the agent and β_r the size of the request. The size of reply is represented by β_p . β_p is reduced (filtering and compressing) to $(1\text{-}\sigma)Rep_s$ by the agent, with $(0\leq\sigma\leq1)$ where σ models the selectivity of the agent. At last stage when the M-Worker comes back, only $state_s$ and the replies are brought to S_0

Total network load for the mobile agent approach equals:

$$L_{mobile}(S_1, S_2, A_S) = Code_S + 2State_S + 2data + n(1-\sigma)\beta_p \qquad (3)$$

II.2 Comparing the execution time.

II.2.1 Case of the centralized model.

In this context when the workflow is launched, interactions are traditionally done through RPC [Straber & Schwehm 97], a (classical) RPC includes binding to the server (destination site), marshalling, transfer, unmarshalling of (the request parameters), execution of the request, and marshalling, transfer and unmarshalling of the reply (marshalling and unmarshalling are transformation of data in a transport format and back).

The execution time taken by a single Web-Service invocation has three components: Service Time (S), Message Delay Time (M) and Waiting Time (W). Service Time is the time that the Web-Service takes to perform its task. Message Delay Time is the time taken by the SOAP messages, in being sent/received by the invocation call. Waiting Time is the Web-Service invocation delay caused by the load on the system where the Web-Service is deployed. Thus, the Total Invocation Time (T) for a Web-Service ω is given by the following formula:

$$T(\omega) = M(\omega) + W(\omega) + S(\omega) \quad (4)$$

Since the bandwidth (throughput) and size of information transferred are very significant in our comparison, we prefer to give a more detailed formula by expanding it.

In practice before been sent the XML message is processed as any other message over the network, the processing time include specific tasks such as checking bit errors, determining output link, and of course marshalling of the message, for our study we consider only the marshalling time with factor μ since other tasks are very negligible. If the Web-Service is local, marshalling and unmarshalling are not necessary. In the other hand, the delay time *Dt* (latency) includes the queuing time *Qt*, the transmit time *Tt* and the propagation time *Pt*. The sum δ of the propagation time Pt and the queuing time *Qt* is said basic latency so that $Dt = \delta + Tt$: if the XML message has a size $\beta(\omega)$ in Kilobytes then $T = \beta(\omega)/\tau$ where τ is the throughput at the node where the Web-Service resides. In our study, we are interested to know how long it takes to invoke the Web-Service and receive the reply back, so we consider RTT (round-trip time). Since the mobile workflow is assumed to be enacted on the Intranet of the enterprise and for the sake of simplicity we consider that the difference between queuing times of the two ends of a link is negligible so that the basic latency is similar at these two points.

In the other hand we consider that the request (the Soap message sent) and the reply (the Soap message received) have identical sizes. The M-Worker has also to perform an action with time ρ (for example to generate the soap request message) With these assumptions the RTT for a simple invocation of a Web-Service ω on site S1 with a request Soap message of size $\beta_r(\omega)$ and a response Soap message of size $\beta_p(\omega)$ by a WF-Manager residing in site S0 (mobile device) can be expressed as follow:

$$RTT(w) = 2\delta(S_0, S_1) + \left(\frac{1}{\tau(S_0, S_1)} + 2\mu\right)\left(\beta_r(\omega) + \beta_p(\omega)\right) + W(\omega) + S(\omega) + \rho(\omega) \quad (7)$$

The total execution time *Tc* for the centralized workflow in whole can be calculated as follow.

$$T_c = \sum_{i=1}^{n}\left(2\delta(S_0, S_1) + \left(\frac{1}{\tau(S_0, S_1)} + 2\mu\right)\left(\beta_r(\omega) + \beta_p(\omega)\right) + W(\omega) + S(\omega) + \rho(\omega)\right) \quad (8)$$

II.2.2 Case of MOBIFLEX workflow.

A (classical) migration includes serialization, transmit and deserialization of code, data and execution state of the agent to the server (serialization is the marshalling process specific to mobile agents). The M-Worker has obviously also to perform a local invocation of the Web-Service; in this case the message delay time is negligible. With these assumptions, the execution time for the M-Worker's itinerary, without checkpointing can be calculated by adding together the total serialization/deserialization time, the total transmit time, the total latencies time and the total Web-Services waiting times and service times (4).

$$\mu(n+1)(code + 2state + 2data) + 2\delta_m + (n-1)\delta_f \; +$$
$$\left(\frac{2}{\tau_m} + \frac{n}{\tau_f}\right)(code + state + data) + \sum\left(W(\omega_i) + S(\omega_i) + Ds(\omega_i)\right) \quad (9)$$

Where δ_m and δ_f indicate average values of latencies of mobile links and fixed links respectively when τ_m and τ_f, are average values for wireless and fixed throughputs.

In the fault tolerant model, the total network execution time for the M-Worker's itinerary *REKs* must be calculated taking into account that a copy of the agent is stored at each visited site, if a certain timeout occurs the sender performs a search among all sites of the itinerary, to recover the last available and correct copy of the M-Worker.

If this copy is found at site *k*, it sends a message of size to site *k* so that a new M-Worker is restored from the copy and reactivated to resume the itinerary. We suppose that a binary search is performed and require at each step an informational message of size I_s to be sent by the sender and a response of size R_s to be received.

Furthermore To model the average checkpointing-recovery-rollback costs, we need to evaluate additional load in the network, execution time of the checkpoint, time to detect failure and time to recover state after a failure, so we need these additional variables:

_ *Di* : Speed of Local Disk Write in site Si measured in kilo- bytes per second

_ *F*: probability of Sequence Path Failure,

_ *L*: factor of acceptable timeout as percentage of expected execution time.

Since a binary search among n nodes implies at average log2(n) steps, we conclude that additional load necessary to perform the recovery-rollback procedure is given by (5):

$A_L = (log_2(n)*(I_s+R_s)*(I_s+R_s))+REK_s$ *(10)*

Additional load is comprised of all informational messages and responses respectively sent and received plus the message *REKs*. As assumed above a checkpoint consists of a local disk write of a copy of the M-Worker, so that total checkpointing time can be calculated as follows:

$$T_{check} = \sum_{i=1}^{n} \frac{code + state + data}{Di} \quad (11)$$

While recovery-rollback time is calculated like this

$$RR = \left(\log_2(n) * \left(\frac{I_s + R_s}{\tau_m} \right) \right) + \frac{REK_s}{\tau_m} \quad (12)$$

Where τ_m is the average throughput for wireless links in the network. Formulation for expected additional execution time for this fault-tolerance model can be given by this equation:

*AT =Tcheck +F(RR +L * TT mobile)* (13)

II.3 Numerical validation.

This section tries to compare the centralized approach and the mobile-agent approach for a typical scenario.

For this scenario we suppose that in some construction project, an engineer who is obviously a mobile worker needs to receive data out in the field - then processes that information and takes appropriate action. This data can be obtained by a workflow from complex constructions diagrams and plans, or some maintenance records and manuals or processed by other static workers. He has the usage of a mobile device such as PDA or laptop (location S0). This device has obviously only wireless low bandwidth access to the internet. The sites providing Web-Services (S1 ... Sn) are fixed and situated on the enterprise Intranet and thus have probably similar technical properties so for the sake of simplicity and without loss of significance, we assume disk write speeds, identical for all the fixed sites (S1 ...Sn); and we consider W and S the average waiting time and service time of Web-Services on the enterprise network. Table (1.) presents the presumed characteristics of this typical scenario, these values are proposed from some experimental studies given in works cited in section such as [osman *et al.* 04] [patel & Garg 05] [Straber & schwehm 97]. Particularly in [Chandrasekaran 02]. δ_m , τ_m , δ_f , τ_f are average values for wireless and fixed throughputs and latencies (The purpose is essentially to compare the two approaches in the same conditions.

This study shows that execution time in the centralized approach model is better than execution time in MOBIFLEX based model only for one station (figure. 3). But when the number of stations increases, it is the MOBIFLEX based–model which outperforms the centralized model in time consuming. The raison is the number of low-bandwidth interactions against the number of high bandwidth interactions in the two models. At a first step the M-Worker migrates from the PDA (S0) to the first site of its itinerary (S1) through a mobile connection, then performs its itinerary from S1 to Sn through high bandwidth connections and finally returns back from the last site (Sn) to its original location through a mobile connection again. For a number of stations different from one, the number of low bandwidth links increases in the centralized model so that it is not possible any more to produce good performance in time (figure 3). Figure 4 shows that centralized WFMS produces less load than MOBIFLEX model; for a number of stations less than six, beyond that, number of low bandwidth interactions increases in the centralized model so that it is not possible any more to produce good performance in load also. Furthermore, figures 5 and 6 show that additional time and load induced by fault-tolerance mechanism are negligible.

parameter	meaning	value
$Code_s$	Size of mobile agent code	25 Kb
$State_s$+$data_s$	Size of mobile agent state+data	8 Kb
ρ	Execution time of a local task	100 ms
βrs	Size of the soap request	4Kb
βp	Size of the soap reply	6 Kb
Is	Size of informational message	2 Kb
Rs	Size of reply for Is	3 kb
REKs	Size of reactivation message	3 Kb
Di i= 1…n	Disk write Speed	2000 Kb/s
μ	Marshalling/unmarshalling factor	5 ms /Kb
σ	Compression factor of the agent	0.40
F	Probability of failure	0.1
L	Percentage of acceptable timeout	0.01
τ(S0,S1)	Throughput of wireless link	50 kb/s
δ(S0,S1)	Latency of wireless link	200 ms
τ(Si,Si+1) i = 1 … n	Throughput of wired links	500 Kb/s
$\delta\tau$(Si,Si+1) I = 1 … n	Latency of wired links	50 ms

Table 1. Parameters retained for the case study

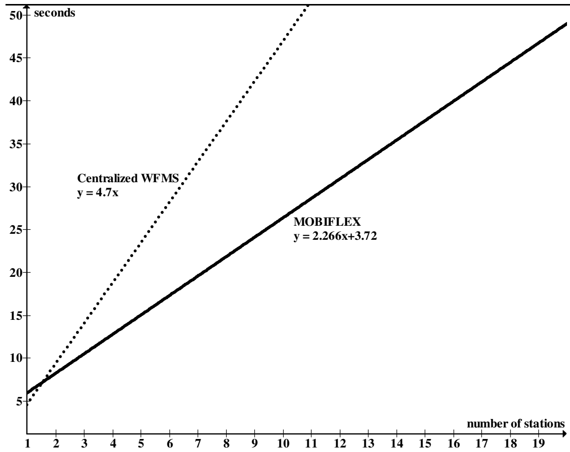


Figure 3. Execution time of centralized model and MOBIFLEX based model in mobile environment

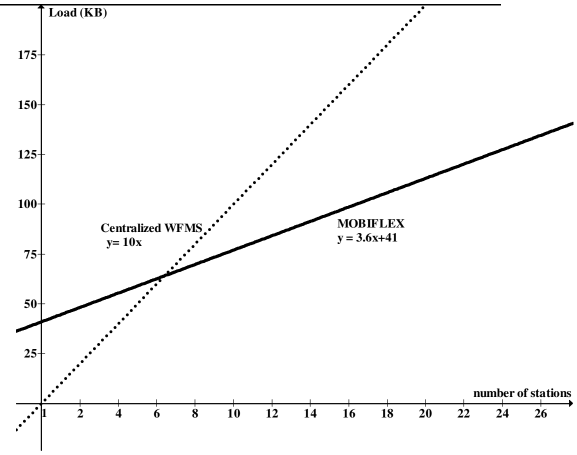


Figure 4. Comparing the network load in centralized and MOBIFLEX model

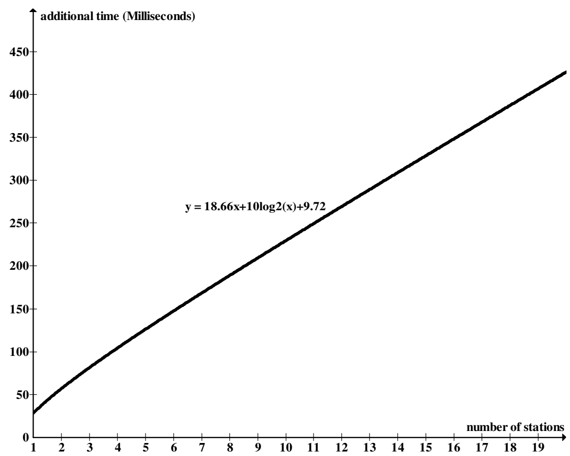


Figure 5. Additional time induced by fault-tolerance mechanism

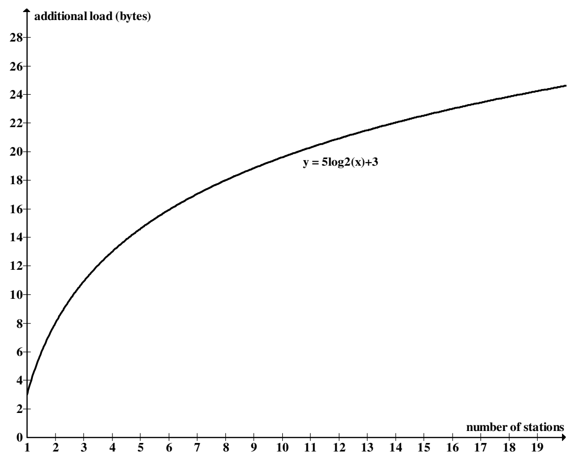


Figure 6. Additional load induced by fault-tolerance mechanism

III Case study: MOBIFLEX for M-Health.

Healthcare information systems (HCIS) are complex, heterogeneous, and spread out over multiple locations making their management and exploitation very onerous and lacking efficiency. Integration of these sub-systems seems necessary and needs a judicious choice of technologies and an adapted architecture. Significant benefits in terms of better economic costs and higher quality of care can be obtained by adopting good integration strategies and suitable technologies. This section studies the importance of HCIS integration and proposes a MOBIFLEX based solution to the problem.

III.1 Healthcare Information systems overview.

Modern healthcare environments are built upon distributed, complex and heterogeneous resources spread out over multiple places. Since healthcare is information intensive, health systems have been developed over many years using various generations of information and communication technologies (ICT). Electronic tools of many types have been introduced, from Hospital Information Systems (HIS) and related subsystems such as Laboratory Information System (LIS), Radiology Information System (RIS) or Pharmacy Information System (PIS) to newer tools such as Electronic Records [Scott 07]. In (Figure 7) shows in an abstract manner a typical architecture of HCIS.

A HIS, also called Clinical Information System (CIS) is the main integrated information system designed to manage the administrative, financial and even clinical information in a hospital. This encompasses paper-based information processing as well as data processing machines. The HIS is often directly interfaced to other clinical systems, such as the LIS, the RIS generally coupled with the Picture Archiving and Communication System (PACS) and the PIS. A LIS is a software for managing information generated by medical laboratory processes. A RIS is used by radiology departments to manage patient radiological data and imagery. PACS is a system for the digital processing of radiological images when PIS is a system for pharmaceutical acts management such as potential medical interactions, patient allergies and dosage errors.

Other important components of health systems are electronic records. A medical record is a folder containing from one to many patient health documents and reports. Today medical records are massively digitalized. An electronic record may be created for each service a patient receives from a medical action, such as radiology, laboratory, or pharmacy, or even administrative one. Numerous types of electronic records are used in healthcare environments. Some examples are the Automated Medical Record (AMR), the Computerized Medical Record (CMR), the computer-based patient record (CPR), the Electronic Medical Record (EMR), and the Electronic Patient Record (EPR). Although there are differences between these concepts, all these terms describe systems that provide a "structured, digitized and fully accessible patient record." the reader can consult [Scott 07] for more details.

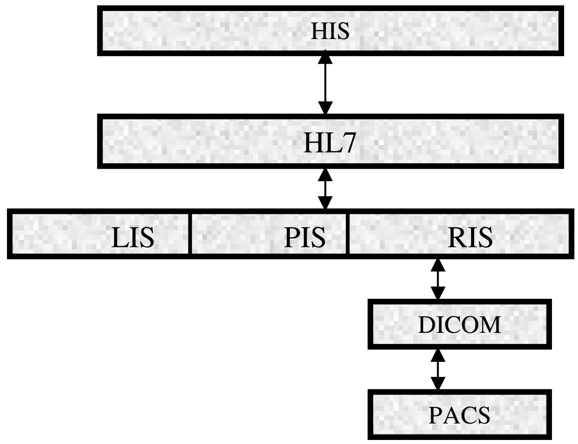


Figure 7. Typical HCIS architecture

Today the term EHR has turned out to be the favoured nomenclature for a sophisticated, generic term covering all concepts encompassed in the terms given above. The Health Information Management Systems Society's (HIMSS) defines Electronic Health Record (EHR) as "a longitudinal electronic record of patient health information generated by one or more encounters in any care delivery setting. Included in this information are patient demographics, progress notes, problems, medications, vital signs, past medical history, immunizations, laboratory data, and radiology reports." [HIMSS 07] Unfortunately, in a typical HCIS, devices and systems interoperate only with the protocols from the same vendor, and can't interoperate with other devices or systems running on different communication protocols standards [Lenz *et al.* 07]. In particular, different electronic records are often captured and remain in disparate and not integrated systems, so that a typical HCIS consists of many independently developed silos. A patient may have multiple medical records at each location with frequent redundancy. Each record may contain partial information, and the process of retrieving data and updating records is very hard. Clinical information of a patient is spread out over a number of medical centers which makes it difficult to get its exact state.

In order to improve the quality of care and to reduce costs, cooperation and information sharing among different health information systems are strongly required. Integration of different healthcare sub-systems and devices is one of the most active research areas and a suitable solution. MOBIFLEX tries to give a software solution to the problem which seems more appropriate than standards based one, since MOBIFLEX architecture is a combination of mobile agents, static agents and Web-Services.

The integration and exploitation of healthcare sub-systems becomes easy and more efficient. Since mobility is inherent to healthcare environments, MOBIFLEX mobile workflows give a natural solution to HCIS management.

III.2 ICT for healthcare

The impact of (ICT) in the healthcare domain has increased considerably in the last few years. Medical informatics also called Health informatics has emerged as a new discipline in the intersection of information science, medicine and healthcare. It deals with the resources, devices and methods required to optimize the acquisition, storage, retrieval and use of information in health and biomedicine [Telemedicine Alliance 07].

New terms like E-Health, telehealth, telemedicine and M-Health have appeared; In (Figure 8) adapted from [Telemedicine Alliance 07] relationships between these new concepts is presented. E-Health [Eysenbach 07] is defined as "an emerging field in the intersection of medical informatics, public health and business, referring to health services and information delivered or enhanced through the Internet and related technologies". Telehealth has become a more generic term to describe a wider definition of telemedicine. Telemedicine [Field & Grigsby 02] is the use of electronic information and communications technologies to provide healthcare when distance separates the participants.

The terms Telemedicine and E-Health are sometimes confused or used interchangeably. Telemedicine normally refers to the provision of medical services from a distance, while e-Health is a more generic term referring to the administration of health data electronically. For the purpose of this work M-Health is also considered (added in Figure 8 to original one). M-Health (mobile-health) is the application of E-Health in the mobile world; a mobile solution should fulfil what is called in [CASCOM 06] the "5 ANYs".

Any network (combining both mobile and wired) (e.g. GSM, GPRS, UMTS, Satellite, Wireless LAN, ADSL, etc.).

- Any channel (e.g. Web, WAP, MIDLETS, etc…) or device (Mobile phone, PDA, Medical devices, PC, IDTV, etc…).
- Any user (any age, any culture, any expertise, etc.).
- Any place (local, regional, national, European; in a city, countryside, road, etc.).
- Any service (i.e. a platform that can be tailored to any specific vertical application).

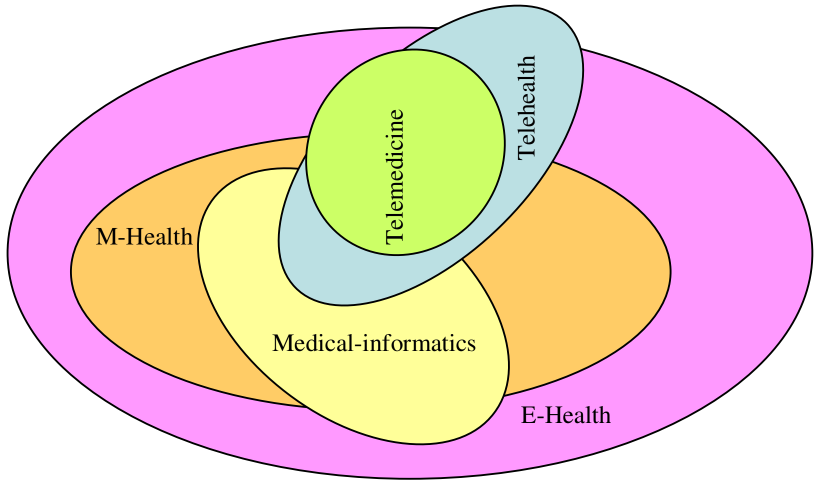


Figure 8. Terminology in ICT based health systems

III.3 New technologies for healthcare: An economic viewpoint.

Usually the success and failure of new technological innovations are explained by the functional and economic advantages that new technologies provide over traditional ways of doing things. Significant economic effect can be obtained if ICT are introduced massively and used optimally in healthcare activities. Information technology can be seen as a strategic tool for change in being competitive, innovating new ways to do healthcare tasks, and creating novel opportunities in this domain. In [CASCOM 06] the benefits of E-Health are summarized in three categories.

- Improving the quality of care: ICT provide the clinician with the patient's entire health history or, even in some instances, the patient's latest clinical information.
- Extending the reach of effective healthcare: The implementation of E-Health technologies such as telemedicine can help rural and low-income communities continue to have facilities attracting and retaining healthcare professionals and service facilities.
- Reducing healthcare costs: ICT give healthcare providers the opportunity to reduce overall healthcare expenses by lowering the costs of administrative and clinical services.

In many countries, great emphasis is currently placed on the implementation of E-Health projects in order to improve the efficiency and effectiveness of the healthcare system. This includes efforts to support evidence-based healthcare, to reduce errors and to make health care more accessible.

For this purpose The European Commission and the CEN E-Health Standardization Focus Group state the necessity of coordinated and interoperable electronic health services and recommend a Europe-wide E-Health platform [Bergmann *et al.* 07]. Today in many European countries, integrated healthcare systems are already in place or are being developed. In Canada Since the late 1980s decisive steps had been taken to embrace ICT in its healthcare system. This evolution was fostered by a number of high profile commission reports examining the state and possible future of the Canadian healthcare system [Moehr *et al.* 06].

In Algeria, HCIS are of particular interest since the costs have been constantly soaring, and thus rationalisation of health care practices became necessary. Currently new ICTs are used in a very limited way and existing healthcare information systems are too fragmented to work in most efficient manner. Maintenance of several different information systems is done with very high costs and the quality of care for patients is not done efficiently. We think that one of the main obstacles in HCIS development in Algeria is the rigid organisation structures and cultures in the healthcare field. In our opinion, Algerian healthcare organisations could benefit from new information technologies to achieve an integration of disparate health sub-systems and thus to create savings for exiguous budgets and promote good and efficient healthcare practices.

It is certain that significant benefits in terms of better economic costs and higher quality of care can be obtained by adopting good integration strategies. In particular, mobile healthcare solutions can give medical staff instant access to HCIS, resources and services allowing medical tasks such as diagnosis, data exchange or monitoring to be made sooner and with more accuracy. Using wireless and handheld computer technologies provides medical practitioners with instant communication possibilities and mobile access to detailed and latest patient data and medical references. The need to find and manage hard copies of patient records from LIS, HIS or RIS systems for example is reduced considerably, and less time is spent in trying to exchange patient data and recommendations. In fact the benefits of the wireless technology for HCIS have already been illustrated in a number of different applications (CASCOM 2006).

Clinicians can have access to patient history, laboratory results, pharmaceutical data, insurance information, and medical resources from anywhere. They can prescribe medication, consult with colleagues and change treatment regardless of their location.

Inventory tracking can become instantaneous with a mobile staff finding what they need on the fly i.e. by means of their mobile device without any need to move to look for useful information. A patient's vitals and location can be monitored with a simple handheld device, providing better control with improved flexibility over traditional methods.

In [Lu *et al.* 05] a practical study on the impact of wireless technologies on healthcare systems was done. The authors and in addition to Internet search engines used Medline, the National Library of Medicine's searchable database of peer-reviewed publications [Medline 07] and the published Proc. of one primary conference (Proc. of Healthcare Information and Management Systems Society [HIMSS 07]. The study had attested that the benefits of handheld computers in healthcare systems are essentially the following.

- Cost saving: The cost reduction associated with electronic documentation handheld computers adoption was very significant.
- Education: Handheld devices were used successfully to assess educational effectiveness on learning evidence-based medicine and a real improvement in participants' educational experience was reported.
- Time saving: Important amount of time is saved during information retrieval, by the medical staff. In particular works, PDA technology was used for billing and reimbursement: reimbursement time was reduced considerably.

III.4. Towards Integration of Healthcare Information Systems.

III.4.1 HCIS Integration: Standards and requirements.

For the purpose of integrating HCIS, first efforts were deployed in standardization; many solutions were proposed to enable homogeneous interoperability between health systems and medical devices. Currently, for the storage and exchange of medical data, three standards have been established: DICOM (Digital Imaging and Communications in Medicine) [DICOM 07], HL7 (Health Level Seven) [HL7 07] and CEN/TC251 [ECS 07].

In practice the integration at the standards level is not enough, every day large Volumes of data are generated from primary care surgeries, radiology departments, laboratories and different other services of the hospital. Since information data is the key to digital HCIS and rather than standardizing at the level of the vendor only, a new tendency is to standardize at the level of the data. New standards in information technologies are adopted for this purpose; however the solution must take into account following requirements.

- Distribution: HCIS is distributed over disparate nodes and built upon disparate technologies.
- Interoperability: interoperability between different medical systems must be possible through heterogeneous platforms among heterogeneous subsystems and medical devices and via different mediums or different links (PDA, mobile phones, wired or wireless links…).
- Mobility: healthcare practitioners are inherently mobile; thus the architecture must be implementable on desktop devices as well as mobile ones.
- Integrability: it must be possible to integrate sub-systems in the same hospital or with other healthcare systems to realize a regional EHR for example or to integrate legacy systems such as old databases or old medical devices.
- Flexibility and adaptability: the architecture must be flexible and adaptable enough to deal with frequent changes and new circumstances inherent to health environment.
- Heterogeneity: the architecture has to bring together very disparate and heterogeneous components such as LIS/RIS systems, medical monitoring devices, databases and so on.
- Complex coordination: there are several kinds of interactions to coordinate: human resource, resource-resource and human-human.
- Intelligency: healthcare actions need intelligency to achieve flexibility and adaptability to frequent changing and unpredictable circumstances of patients and the environment.

III.4.2 MOBIFLEX: a suitable Architecture for HCIS applications.

The two technologies handled in MOBIFLEX seem to be particularly convenient candidates for health domain: multi-agents systems and Web-Services. Healthcare is a promising research area where the agent paradigm is able to be applied. In [Moreno 03], the author gave a list of fields where researchers have already applied successfully the agent paradigm in health systems.

- Information management: The increasing health information available online (Internet and other electronic sources) has led to the development of information agents to collect, filter and organize this information.

- Community care: Agent-based systems have been applied in the coordination of all the activities that have to be performed for providing an efficient healthcare to citizens of a community especially older or disabled citizens.
- Decision support systems: A distributed decision support system based on the agent paradigm can monitor the status of a hospitalized patient and help to support cooperative medical decision-making.
- Education and simulation: Agents can help to improve medical training and education in distance-learning tutoring.
- Planning and resource allocation: For example, an agent-based coordination across a hospital could provide significant improvements in the time required to pull together the resources required for tissue and organ transplant operation.

Furthermore Web-Services are becoming the technology of next generation of devices of different types and usage. The Devices Profile for Web-Services (DPWS) [Schlimmer 04] proposed by Microsoft in August 2004 defines a subset (profile) of Web-Service protocols for its usage by devices. Today Internet technology can connect not only high level devices like computers and PLCs (Programmable Logic Controlers) but also devices of all kinds even at the lowest level of the device hierarchy, i.e. sensors and actuators. Intelligent networked devices become able to collaborate with each other to achieve their own individual goals. In medical domain, Microsoft has already submitted three Web-Services specifications (WS-Addressing, WS-Security and WS-Reliable Messaging) as an update of the HL7 standard [Shodjai 06]

MOBIFLEX architecture combines agents and Web-Services for the design of a platform enabling integration of HCIS and their management with mobile workflows in a natural way. MOBIFLEX Multi-agent systems with its static ordinary agents, mobile agents and intelligent agents offers a broad range of possibilities for HCIS. In particular, intelligent agents are used to handle complex functions offering flexible and reconfigurable possibilities to HCIS. It is service oriented to achieve high interoperability and integrability in healthcare systems. If agents are the best solution to fulfil most of the requirements of the distribute applications, they are not sufficient to face interoperability issues, because of their coupled dependency to specific multi-agents platforms. In contrast Web-Services, are mainly designed to solve such interoperability problems. According to [Dogac *et al.* 06] Web-Services offer to HCIS many advantages.

- It becomes possible to provide the interoperability of medical information systems through standardizing the access to data through WSDL and SOAP rather than standardizing documentation of electronic health records.
- Web-Services will extend the healthcare enterprises by making their own services available to others.
- Web-Services will extend the life of the existing software by exposing previously proprietary functions as Web-Services.
- Web-Services can be used to achieve interoperability not only between heterogeneous platforms and standards but also between legacy databases and infrastructures such as old RIS/LIS medical devices. .

Finally MOBIFLEX is fault–tolerant and takes into account the poor resources of mobile devices so that a minimal configuration is deployed on such devices when the core components of the system are on the fixed platform. A MOBIFLEX based HCIS architecture can be achieved by multi-agents and Web-Services layers upon the traditional HCIS Layers and then we obtain an integrated and intelligent architecture as shown in figure 9.

III.4.3 MOBIFLEX integrating possibilities for HCIS.

MOBIFLEX is service oriented since all functionalities and operation of the system are implemented as Web-Services. Web-Services are used to enable high degree of interoperability between heterogeneous systems and devices in the same HCIS. But this is not sufficient; although the HCIS is included in the main hospital business process, in some cases its scope goes beyond the hospital borders, for example if implied in a virtual E-Health system such as regional or national EHR or if extended to an external structure such as a homecare system managed from the hospital. In this case the HCIS have to be interfaced to the external partners by some brokering mechanism to ensure a secured interaction, this way the integrity of the HCIS is preserved.

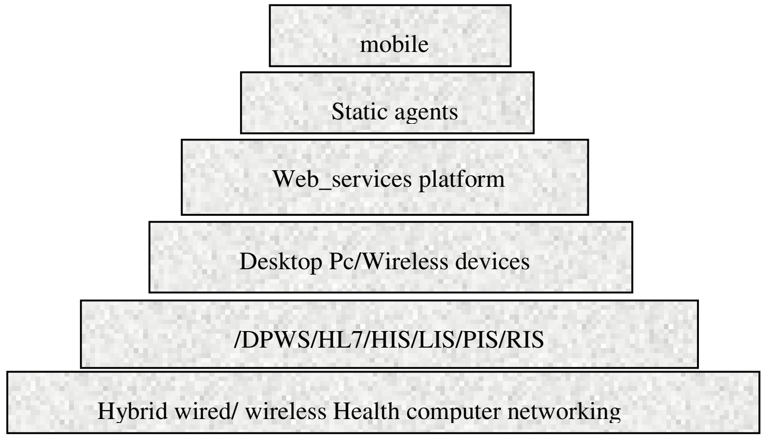


Figure 9.Buiding blocks for MOBIFLEX based HCIS

In MOBIFLEX Web-Services are used as main tools for the integration. MOBIFLEX differentiates between three classes of Web-Services to be used to integrate different sub-systems of the HCIS and the HCIS with external partners (figure 10).

- Native Web-Services(NT-ws): native Web-Services mean those included originally in devices or systems such as DPWS or HL7 based ones.
- Broker Web-Services(BR-ws): they are Web-Services created to interface the HCIS with legacy not service oriented devices and systems.
- Database Web-Services(DB-ws): these Web-Services enable requesting and retrieving data from heterogeneous data-bases systems.

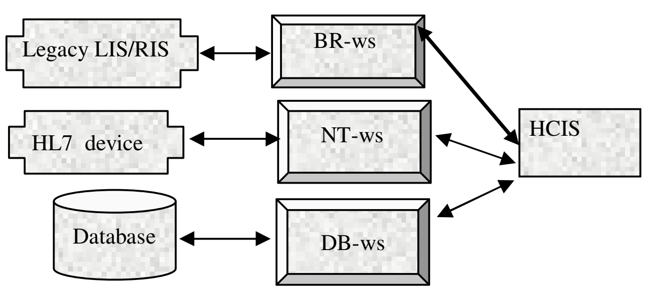


Figure 10. MOBIFLEX integration mechanism

To improve the integration possibilities in HCIS we can enhance the initial MOBIFLEX architecture by an agentification of Web-Services when necessary. Classic interoperability between agents and Web-Services is not obvious even though feasible because agents communicate in ACL and Web-Services interact with soap messages.

In the other hand if the agent is mobile and wants to move to invoke the Web-Service locally, it can not migrate only among sites of its platform. An agentification of Web-Services seems appropriate to make the architecture more homogeneous and facilitate the interactions. Fortunately, the standards exist yet. Agents-services are proposed by the FIPA organization as an agentification standard of Web-Services. In this architecture services are capabilities exposed by agents, these capabilities are not registered in UDDI but in special yellow pages called Directory Facilitator (DF) that are managed by specific servers on agent platforms (Figure 11). This new vision presents the advantages to bring the gap between agents and Web-Services and to simplify their usage since they interact uniformly by FIPA–ACL messages. For this purpose these two cases can be considered in MOBIFLEX to improve possibilities for HCIS (figure 12).

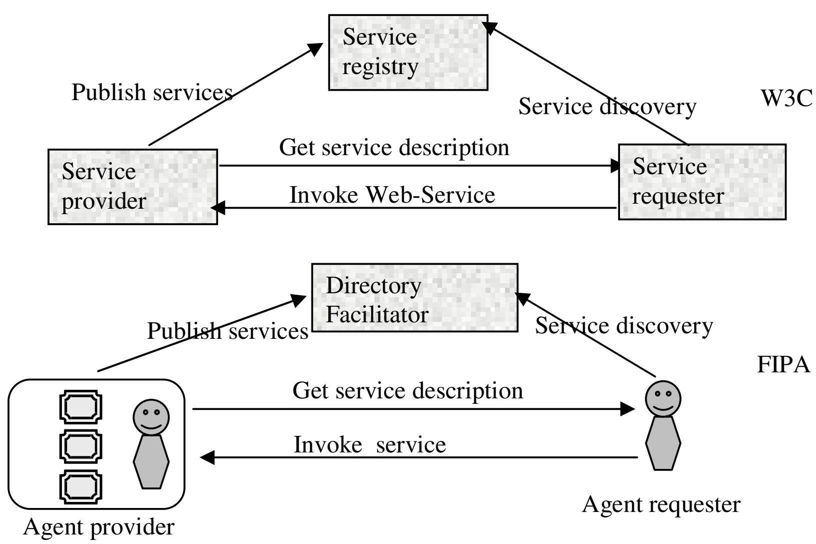


Figure 11. W3C versus FIPA based service oriented architecture

If the Web-Service implied in the HCIS is local to the hospital or in a system directly linked to it (e.g. homecare): an agent-service is used else if the Web-Service implied in the HCIS is external and provided by some external enterprise: a broker agent-services is created which capabilities can vary from a simple invocation of an external Web-Service to a complex composite one combining local and external ones. For the case where the external system does not expose its functionalities as Web-Services, local agents-services can be created to interface the HCIS to external partner’s functionalities through broker Web-Services.

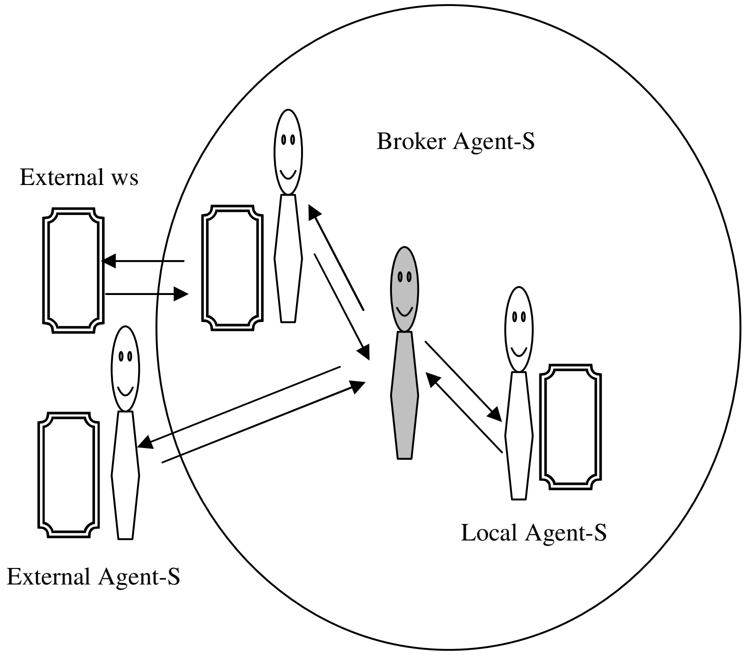


Figure 12. Enhancing MOBIFLEX with

III.5 Typical applications of MOBIFLEX in HCIS.

III.5.1 Medical data retrieval: a virtual EHR.

Patient reports are usually spread out over different and heterogeneous information systems as explained before. Physicians frequently need to have an updated medical report for a given patient that combine information from his EPR, some specific imaging from RIS and some laboratory results from LIS. The objective is to have the complete medical information of a patient available in one consistent application rather than over several information systems. A MOBIFLEX based workflow can be used as an intelligent agent-based information management system to access and retrieve information homogenously from heterogeneous sources [Bouchoul & Mostefai 09]. A virtual EHR can thus be built on the fly with the newest and updated data without the need to create a true combined EHR with very expensive maintenance tasks.

III.5.2 Guidelines based careflows.

The standard definition of Clinical Practice Guidelines (CPGs) is that of Field and Lohr [Field & Lohr 92] who indicated that CPGs are "Systematically developed statements to assist practitioners and patient decisions about appropriate healthcare for specific circumstances". The treatment for a specific patient is a path through a subset of the guideline steps, where each step is a test, treatment or decision task that might need to be undertaken.

Computerised guidelines offer obvious benefits above those offered by paper-based guidelines since recommendations about what medical procedures to perform for an individual patient can be automatically generated. Medical guidelines are generally computerized as workflows. These specific workflows are called careflows [Quaglini *et al.* 00]. In this case the reasoning mechanism of the PL-Agent may encompass in some specific notation the guidelines to be done for specific cases. The M-Worker can act as a guideline manager and executes the workflow, starting from its entry point, by selecting each time the next activity to be performed. A task can be medical data retrieval, a communication of some medical data to a health practitioner, a presentation of instructions to a specific practitioner in the workflow (eventually the patient) or an alert to a specific user to do some medical task and so on. In a particular step the M-Worker may wait in the related site until the task is done, for example until the patient notifies it that the task was done.

III.5.3 Careflows for homecare.

Decentralized healthcare services and home assistance are key tools to achieve two objectives: enabling patients to spend their time in a familiar environment and reducing the hospital expenses. A homecare careflow may involve different scheduled activities related to chronic or elderly patients for example such as retrieving data from remote monitoring devices, automated storage of data about the patient's health state in clinical records, alerting such users when it is time to perform an action and assisting them in the action execution, and alerting the hospital when an emergency is needed. The reasoning mechanism of the PL-Agent can be programmed to plan homecare careflow which are executed by an M-Worker under control of a physician at the hospital.

III.6 Implementation of a prototype for medical data retrieval: a virtual EHR.

Currently a prototype illustrating our approach is under development. The objective is to build a virtual EHR on the fly as explained above and thus to have the complete medical information of a patient available in one consistent application rather than over several information systems. Since a mobile workflow is enacted from a device with low resources capacities typically a PDA, the architecture must be built so that only the minimal configuration and capabilities are held by the mobile device.

In this section the implementation of such architecture is proposed . The choice of the appropriate technology is motivated by next requirements.

- It can be deployed on both wired and wireless devices.
- Both static and mobile agents have to be enabled.
- Provide possibilities to use intelligent agents.
- Is portable on different platforms.
- Provide service oriented management facilities.

The platform JADE [Jade 07] is perhaps the unique one that fulfils all these requirements together. JADE is a FIPA compliant middleware implemented in Java for the development and execution of peer-to-peer multi-agents systems. Peer-to-peer systems are distributed ones where all nodes are peers in the sense that they are both clients and servers in the same time. A peer could be a computer, a personal mobile terminal or some other device. A platform JADE is composed of one main container and many sub-containers. Only one container can be launched on each JVM and then on each device. A JADE Main Container is used to host the Agent Management System (AMS) and Directory Facilitator (DF) agent in conformance to FIPA standards. DF is an agent that offers Yellow pages of the services that can be offered by other JADE agents. (AMS) is an agent that offers a White pages service to control the access and use of the agents' platform. Message Transport Service (MTS) is used to communicate agents which are in different JADE platforms. Agents communicate with each other directly via messages through a FIPA-ACL–Communication protocol. Basic communication protocols like FIPA-Query, FIPA-Request and FIPA-Contract Net are enabled. Agents can move from one machine to another one, as and when required.

The Lightweight Extensible Agent Platform (LEAP) [Caire 03] is an extension for JADE, which enables agents to use mobile devices as agent platforms. With LEAP it is possible to create a platform which is not only distributed over different servers but can even be extended to devices which are connected by a wireless connection like PDAs or mobile phones.

JADEX is another JADE extension which makes it possible to use the BDI agents (Beliefs, Desires and Intentions). Beliefs represent the information an agent has about the world it inhabits. Desires represent the agent's wishes and drive the course of its actions.

Plans are the means by which agents achieve their goals and react to occurring events. JADEX supports four types of goals [Braubach *et al*. 04]

- A perform goal specifies some activities to be done, therefore the outcome of the goal depends only on the fact if activities were performed or not.
- An achieve goal represents a goal in the classical sense by specifying a target state that shall be achieved, the agent has to perform specific activities for achieving the target state.
- A maintain goal has the purpose to observe some desired world state and the agent actively tries to re-establish this state when it is violated.
- Query goals allow for an easy information retrieval from the beliefbase and when the result is not available the BDI mechanism will invoke plans for retrieving the needed information.

(Figure 13) shows the structure of a JADEX agent. For the reasoning mechanism, JADEX does not require any special kind of knowledge representation, but allows arbitrary Java objects to be stored as facts in the beliefbase (BB in Figure 1). Implementation also incorporates concepts from the relational database world. A set oriented declarative query language allows retrieving subsets of beliefs, or evaluating expressions over the belief base state. Each retrieved belief can generate an internal event that initiates a new query goal for the next step in planning. Each step can be generated according to QoS metrics or some applicability rules (resource availability for example).

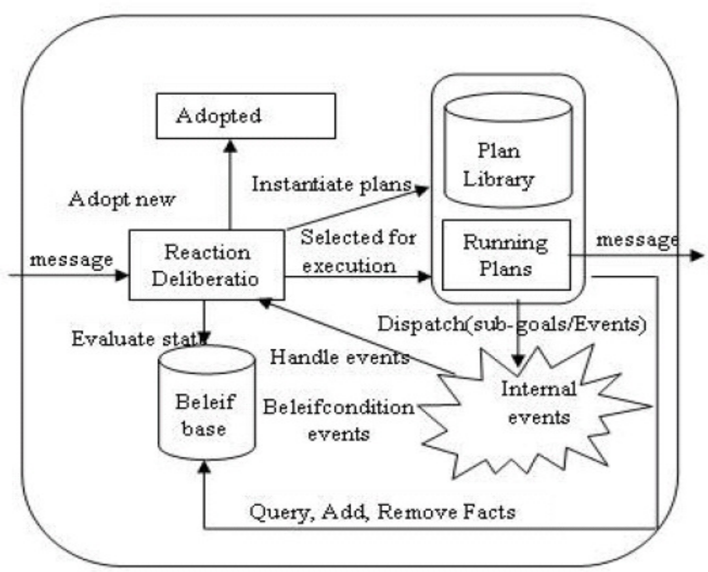


Figure 13. JADEX agent Abstract architecture

In MOBIFLEX architecture, WF-Managers are JADE-LEAP agents residing in J2ME containers on mobile devices such as PDA, Palm-Tops or smart phones running the CDC version of J2ME or devices with powerless resources running the CLDC-MIDP version of J2ME. These two SUN technologies offer very powerful possibilities for handheld devices and are fully supported by JadeLeap. Unfortunately, currently there is no free JVM for J2ME; the most used commercial ones are CrEmE [CREME 09] and WebSphere IBM J9 [IBMJ9 09]. The wireless connectivity is possible under JadeLeap with multiple alternatives such as GPRS, Wi-Fi and BlueTooth. The PL-Agent is a Jadex agent residing in the main-container of the JADE platform (J2SE), notice that until now JADEX can run only on J2SE containers and there is no version for J2ME. As depicted in figure 1, when the PL-Agent receives the FIPA-query message from the WF-Manager (Step 1) the planning process is initiated (Step 2): a query goal is generated, and an information retrieval from the beleifbase is initiated, when the result is not available the BDI mechanism will invoke plans for retrieving the needed information.

Flexibility in BDI plans is achieved by the dynamic selection of suitable plans for a certain goal which is performed by a process called "meta-level reasoning"; this process decides with respect to the actual situation which plan will get a chance to satisfy the goal. If a plan is not successful, the meta-level reasoning can be done again allowing a recovery from plan failures. The goal is to find a suitable workflow enactment to be done by the M-Worker including the sites to be visited and the tasks to be done (Step 3). The PL-Agent can consult the DF for the best agent-servers candidates. Finally a plan is initiated to return the result in a convenient form to the M-Worker as a FIPA-inform message. This message contains the itinerary to be performed to achieve the goal.

The WF-manager initiates the creation of an M-Worker (Step 4) to enact the mobile workflow; the M-Worker then performs its itinerary, and finally comes back to its original location with the result (Step 5). From [Ferreira *et al.* 03] an elegant JADE solution to implement the M-Worker itinerary is adapted: The WF-manager initially adds the behaviour ItineraryBehaviour to the M-Worker, that sets the itinerary he must follow, and then, this latter starts the migration. The M-Worker execution is controlled by the methods beforeMove and afterMove that controls the migration and allows the execution of the job, respectively. The agent job is implemented in a behaviour called JobBehaviour.

For the checkpointing mechanism JADE offers sufficient methods to clone the agent (doClone()) deactivate it (deactivate()) and reactivate it (doActivate()) for more details on these techniques see [Jade 06].

In the experimental platform installed, the WebSphere IBM J9 is used as JAVA virtual machine for mobile devices. The choice is motivated by the fact that this JVM has been successfully tested and used in PDAs running JadeLeap, when some problems have been noticed with CrEme.

In (Figure 15) the experimental platform MOBIFLEX on JadeLeap/Jadex is launched. From the top to the bottom one can see the three standards agents of JadeLeap, the PL-agent residing in the main container, the WF-manager is in container-2 on the mobile device together with an M-Worker (called here MW-01) ready to start its itinerary. At each step, the M-Worker has to request a local Agent-s able to retrieve information from electronic records stored in healthcare databases.

In the JADE based architecture of MOBIFLEX (Figure. 1), WF-Managers are JADE-LEAP agents residing in J2ME containers on mobile devices such as PDA, Palm-Tops or smart phones running the CDC version of J2ME or devices with powerless resources running the CLDC-MIDP version of J2ME. These two SUN technologies offer very powerful possibilities for handled devices and are fully supported by JADELEAP. Unfortunately currently there is no free virtual machine for J2ME; the most used commercial ones are CrEme [CrEme 09] and WebSphere IBM J9 [IBMJ9 09].

The wireless connectivity is possible under JADELEAP with multiple alternatives such as GPRS, Wi-Fi and BlueTooth. Figure. 14 shows in an abstract manner the MOBIFLEX experimental platform.

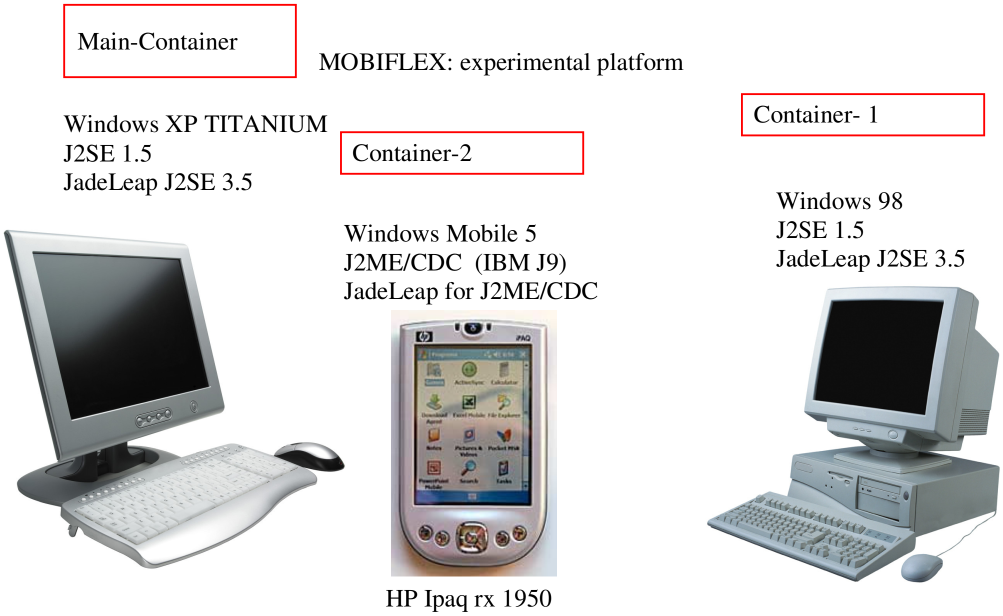


Figure 14. MOBIFLEX experimental platform

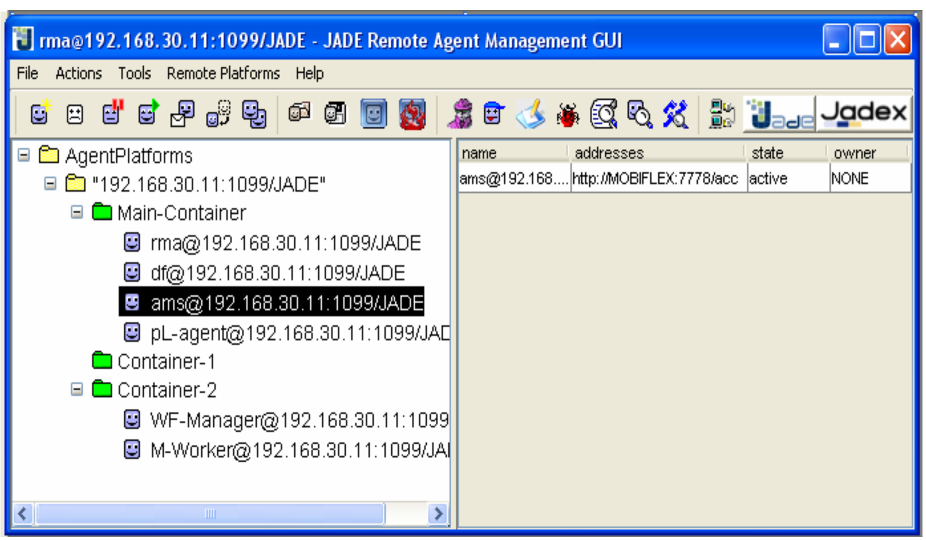


Figure 15. MOBIFLEX platform launched on Jade/Jadex

REFERENCES

[Adaçal & Bener 06] M. Adaçal and A B. Bener " Mobile Web-Services: A New Agent-Based Framework" IEEE Internet Computing, Vol. 10[3] pp 58 – 65 May-June 2006

[Bergman *et al*. 07] J. Bergmann, O. J. Bott, D. P. Pretschner, and R. Haux. "An e-consent-based shared EHR system architecture for integrated healthcare networks", International Journal of Medical Informatics Vol [76] pp 130–136 2007

[Bouchoul & Mostefai 09] F. Bouchoul; M. Mostefai "Agent-services and mobile agents for an integrated HCIS" International Journal of Computer Integrated Manufacturing, (IJCIM) Taylor and Francis editor Vol 22[5] pp 458-471 may 2009

[Bouchoul *et al*. 07] F. Bouchoul M. Mostefai and Z. Kaddem, 2007. "Towards M-Manufacturing - Case Study: a Fault Tolerant Model for M-Maintenance of Spare Parts with Mobile Agents", in IEEE:Proc. of Fourth international conference: Sciences of Electronic, Technologies of Information and Telecommunications SETIT 2007 mars 25-29 Hamamet, Tunisia

[Bouchoul *et al*. 09] F. Bouchoul , M. Mostefai and Z. Kaddem "The design of a mobile agent based workflow and its implementation on jade platform" in International conference of Information technologies CITIC'09 Sétif 4,5 mai 2009

[Braubach *et al*. 04] L. Braubach, A. Pokahr and W. Lamersdorf, "Jadex: A short overiew", in: Main Conference Net.ObjectDays , AgentExpo. 2004.

[Caire 03] Caire G., "LEAP 3.0 User Guide" available at http://JADE.tilab.com/doc/LEAP UserGuide.pdf 2003

[Cardoso *et al*. 04] J. Cardoso, A. Sheth, J. Miller, J. Arnold and K. Kochut, "Quality of service for workflows and Web-Service processes", In *Web Semantics:* Science, Services and Agents on the World Wide Web, Vol. 1[3] pp 281-308 April 2004

[CASCOM 06] : "Context-Aware Business Application Service Co-ordination in Mobile Computing Environments, Deliverable D3.2: Conceptual Architecture Design", available at http://www.ist-cascom.org/dmdocuments/IST-2003-511632-CASCOM-WP3-D3.2-v2.0.pdf [Accessed June 13 2007]

[Chandrasekaran 02] Chandrasekaran S., 2002, "Composition, performance analysis and simulation of Web-Service", A thesis submitted to the graduate faculty of the university of Georgia in partial fulfilment of the requirements for the degree master of science athens, georgia , Available at http://lsdis.cs.uga.edu/lib/download/thesis/ Senthilanand_Chandrasekaran_Thesis_200212.pdf

[CrEme 09] CrEme java virtual machine, : www.nsicom.com

[DICOM 07], Digital Imaging and Communications in Medicine (DICOM), http://medical.nema.org/. [Accessed june 13 2007]

[Dogac *et al*. 06] A. Dogac , G. Laleci , S. Kirbas , Y. Kabak , S. Sinir , A. Yildiz and Y. Gurcan "Artemis: Deploying Semantically Enriched Web-Services in the Healthcare Domain", Information Systems Journal (Elsevier), Vol. 31[4,5] pp 321-339 2006

[ECS 07] European Committee for Standardisation, Technical Committee for Health Informatics. http://www.centc251.org/. [Accessed April 28 2007]

[Eysenbach 01] G. Eysenbach "What is E-Health?", Journal of Medical Internet Research 2001;3(2):e20 <URL: http://www.jmir.org/2001/2/e20/> 2001 [Accessed june 13 2007]

[Ferreira *et al*. 03] E. Ferreira de Araújo Lima, P. Duarte de Lima Machado, J. C. Abrantes de Figueiredo and F. Ronison Sampaio, "Implementing Mobile Agent Design Patterns in the JADE framework", In: in Special Issue on JADE of the TILAB Journal EXP 2003

[Field & Grisby 02] M. J. Field Grigsby J. 2002. "Telemedicine and Remote Patient Monitoring" , In JAMA.2002; 288: 423-425.available at http://jama.ama-assn.org/cgi/content/full/288/4/423#ACK#ACK [Accessed june 13 2007]

[Field & Lohr 92] M.J. Field and K.N. Lohr "Guidelines for clinical practice: from development to use". Institute of Medicine, Washington, D.C: National Academy Press 1992

[HIMSS 07] http://www.himss.org/ASP/topics_ehr.asp. [Accessed April 25 2007]

[HL7 07], Health Level-7 Standards homepage, http://www.hl7.org [Accessed june 13 2007]

[IBMJ9 09] IBM J9 virtual machine http://www

[Jade 06] Jade Programmer's Guide, 2006. available at http://JADE.tilab.com/doc/programmersguide.pdf [Accessed May 02 2007]

[JADE 07] http://JADE.tilab.com/. [Accessed May 02 2007]

[Lange & Oshima 99] D. B. Lange, and M. Oshima, "Seven Good Reasons For Mobile Agents" communication of the ACM, Vol 42[3] march 1999

[Lenz *et al*. 07] R. Lenz, M. Beyer and K. A. Kuhn, "Semantic integration in healthcare networks", International Journal of Medical Informatic., Vol. 76[2,3] pp 201-207 2007

[Lu *et al*. 05] L. Y-Co, Y. Xiao, A. Sears and J. A. Jacko "A review and a framework of handheld computer adoption in healthcare" International Journal of Medical Informatics, Vol. 74 pp 409–422 2005

[Medline 07]. http://medline.cos.com/ [Accessed October 8 2007]

[Moehr *et al*. 06] J.R. Moehr, J. Schaafsma, C. Anglin, S.V. Pantazi, N.A. Grimm and S. Anglin "Success factors for telehealth—A case study", International Journal of Medical Informatics Vol. 75 pp 755–763 2006

[Moreno 03] A. Moreno, "Medical Applications of Multi-Agent Systems", in Ninth European conference on artificial intelligence in medicine AIME'03 cyprus 2003

[O'Malley *et al*. 00] A. O'Malley, L. S. Athie and S.A. Deloach "Comparing performance of static versus mobile multiagent systems", in National Aerospace and Electronics Conference (NAECON) Dayton, OH, October 10-12. 2000

[Osman *et al*. 04] T. Osman, W. Wagealla and A. Bargiela "An approach to rollback recovery of collaborating mobile agents", IEEE Trans. Syst., Man, Cybern., vol. 34[1] pp 48–57 2004

[Patel & Garg 05] R. B. Patel and K. Garg "A Comparative Study of Mobile Agent and Client-Server technologies in a Real Application", in 11th International Conference on Management of Data (COMAD 2005), Goa, India, January 6-8 2005

[Pico 01] G. P. Picco "Mobile agents: an introduction" Microprocessors and Microsystems, Vol. 25[2] pp 65-74 20 April 2001

[Schlimmer 04] J. A. Schlimmer "Technical Introduction to the Devices Profile for Web-Services", Microsoft Corporation May 2004 http://msdn2.microsoft.com/en-us/library/ms996400.aspx 2004 [Accessed April 25 2007]

[Scott 07] R. E. Scott "e-Records in health—Preserving our future", International Journal of Medical Informatics, Vol. 76[5,6], pp 427-431. 2007

[Shodjai 06] P. Shodjai, "Web-Services and the Microsoft Platform", Microsoft Corporation, available at http://msdn2.microsoft.com/en-us/library/aa480728.aspx 2006 [Accessed June 13 2007]

[Stormer & Knorr 01] H. Stormer and K. Knorr, "Pda- and agent-based execution of workflow tasks". In Proc. of the Informatik 2001, pp 968–973 2001

[Straber & Schwehm 97] M. Straber and M. Schwehm. "A performance model for mobile agent systems" Proc. of Int. Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA' 97) CSREA – VoL . 2 pp 1132-1140 1997

[Telemedicine Alliance 04]. "Telemedicine 2010: Visions for a Personal Medical Network", White paper, The Telemedecine alliance, Available at http://esamultimedia.esa.int/docs/BR-229Web.pdf [Accessed June 13 2007]

CHAPTER IV

A FORMAL FRAMEWORK FOR MOBIFLEX. PART I : THE REACTNets

The usage of formal tools for verification, simulation and prototyping designed to facilitate the modelling of complex systems is of great interest. In particular, improved methods are needed to insure reliability, security and robustness of business processes systems. In this chapter we shall try to propose a formal framework for specifying and verifying MOBIFLEX workflows, an enactment of such system can been seen as a complex combination of mobile agents moving over a set of nodes where reside Web-Services, these Web-Services can be simple ones or composite ones i.e. BPEL processes. Figure 1 shows the two layers of MOBIFLEX architecture, a static service oriented layer (Which we call the SO_Layer) is the Web-Services platform of the enterprise, and a dynamic layer obtained by a reconfigurable and very flexible workflow enactment patterns to be performed by mobile agents over the static layer (we call it the MA_Layer). To overcome the complexity of the system we have to distinguish between these two layers and then we propose to formalize each one by itself. This will be done by a judicious combination of formal tools to deal with all facets of the system.

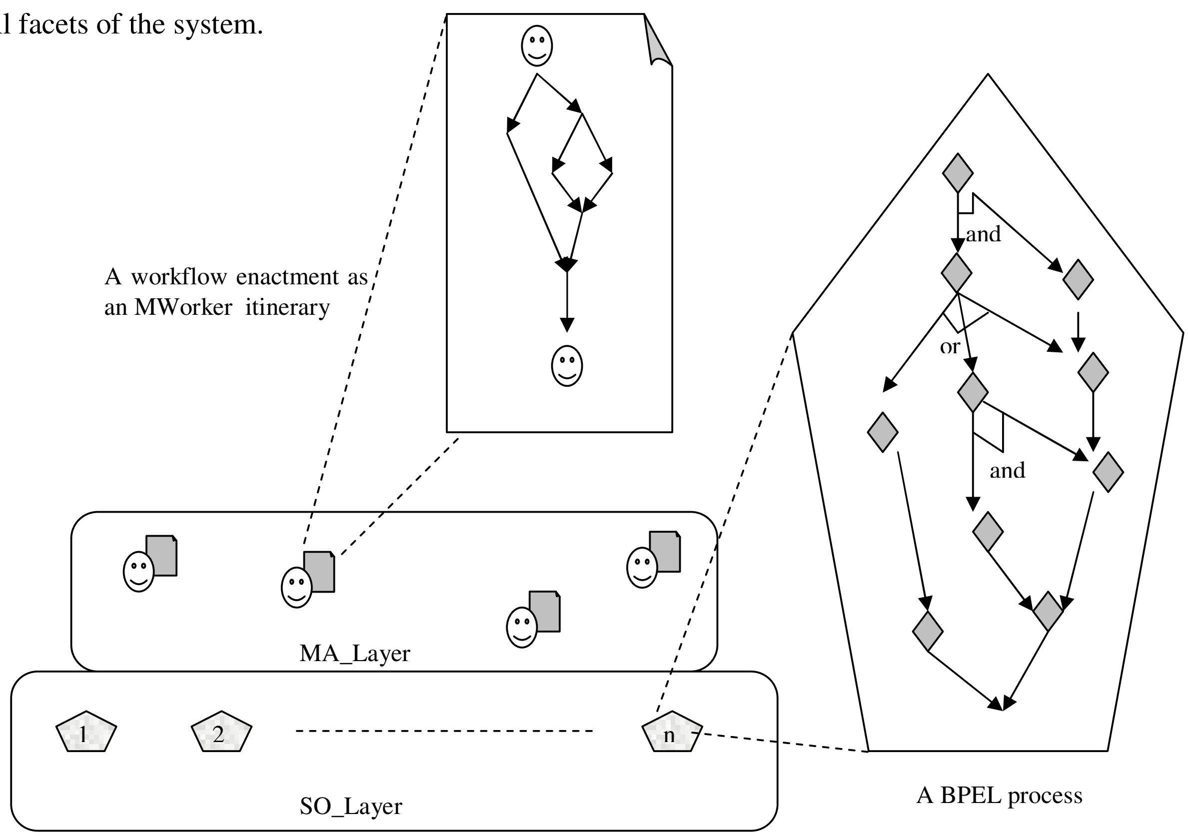


Figure 1. MOBIFLEX abstract architecture

We have chosen the rewriting logic [Meseguer 92] as the basis of our complex formal framework since this logic is a very a powerful unifying paradigm of most of formal models of concurrency. Rewriting logic is a computational logic that can be efficiently implemented and that has good properties as a general and flexible logical and semantic framework, in which a wide range of logics and models of computation can be represented. In particular, for programming language semantics [Şerbănuţă *et al*. 09].

For MOBIFLEX we have a large range of possibilities to formalize all the facets of the system, indeed, an important number of tools based on rewriting logic such as MAUDE language [Meseguer 92bis] [Duran & Meseguer 07] and the ECATNets [Bettaz 92] [Belala *et al*. 00] exist and can be exploited for our purpose. Finally combining tools with a common semantic enable a homogeneous integration and attenuates the difficulties often encountered during the integration of ad hoc formalisms. The first section is a background about the rewriting logic theory and the two formalisms chosen for our framework: the MAUDE language and the ECATnets. In The second section we propose a new formalism which extends the ECATNets to deal with reactivity and distribution inherent to the structure of the SO_Layer of MOBIFLEX architecture. Our solution is mainly based on the proposition of two sets of design patterns; design patterns are reusable design artefacts. The concept of reusable design is one of great prevalence in the field of software engineering, among other fields. The concept of a design pattern "describes a problem ubiquitous to a given environment as well as a solution to that problem in a way that allows for its reuse under varying circumstances" [cicirello *et al*. 05]. Our formal framework is composed of two sets of design patterns, the first set relative to the SO_Layer is an algebraic Petri nets based specification of BPEL patterns, the second relative to the MA_Layer is a formalization of the most relevant mobile agent movement patterns. In this chapter we present the rewriting logic and propose an extension of the ECATnets a kind of high level algebraic Petri nets to adapt them to our needs, in next chapter we present our formal framework.

I Background: The rewriting logic.

I.1 Basic definitions.

Definition 1

A signature Σ is a pair (*S,F*) where *S* is a set of sorts (types) and *F* a set of function symbols such that *F* is equiped with a mapping type $F \rightarrow S^* \, x \, S$ which expresses the type or functionality of each function symbol.

We write $f : s_1 \times s_2 \times \ldots \times s_n \rightarrow s$ to mean that $f \in F$ with type(f) =($s_1, s_2, \ldots, s_n$)

Example

Let Σ_{nat} be a signature for natural numbers

Σ_{nat} = ({nat}, {zero,succ})

Where zero : $\rightarrow$ nat

Succ : nat $\rightarrow$ nat

Definition 2

Let *X* be an S-sorted set of variables. For every sort s $\in$ *S* we define the set $T_{\Sigma\_}(X)s$, of terms of sort s with variables in X, as the least set containing:

1. Every variable of sort s, i.e., $X_s \subset T_\Sigma(X)_s$
2. Every nullary function symbol (constant) c $\in$ F with c : $\rightarrow$ s
3. Every term f ($t_1, \ldots\ldots, t_n$) where f: $s_1 x \ldots x \, s_n \rightarrow s \in F$ and each t_i is a term in $T_\Sigma(X)_{si}$, i = 1; n.

These sets form the family of sets $\{T_\Sigma(X)_s\}_{s \in S}$, which we shall refer to as $T_\Sigma(X)$.

Note that the terms in the family $\{T_\Sigma(X)_s\}_{s \in S}$ are elements (words, strings) of the set.

Definition 3

Given a signature Σ = (S,F) , a Σ-algebra A consists of an S-sorted family of nonempty carrier sets $\{A_s\}_{s \in S}$ and a total function f^A: $A_{s1} \, x \ldots x \, A_{sn} \rightarrow A_s$ for each function symbol

f: $s_1 x \ldots x \, s_n \rightarrow s \in F$.

I.2 Rewriting logic [Meseguer 92] [Meseguer & Rosu 07]

Although the rules of rewriting logic resemble those of equational logic, their meaning is very different; rewriting logic is a logic to reason about changes in a concurrent system, not about equalities. Each rewrite rule is a general pattern for a basic action that can occur concurrently with other actions in a concurrent system.

Rewriting logic then allows us to reason about what other complex changes are possible in a system, given that changes corresponding to the basic actions axiomatized by the rules are possible. In this way, we can reason about concurrent programs in a logic intrinsic to their computations.

The models of rewriting logic are precisely concurrent systems in the intuitive sense of the word, i.e., they are machine-like entities whose state is distributed and can change by actions taking place simultaneously. Such models are formalized as categories with algebraic structure and this yields a general triangular correspondence between logic, concurrency and category theory by which ideas and methods can be transferred between these fields.

I.2.1 Rewriting logic semantics.

The rewriting logic is nothing but a generalization of equational logic in order to adapt it to changes (Meseguer 1992). The rules are similar to those of equational logic but have a completely different significance. A rule $T \rightarrow T'$ do not mean any more T equal T' but T becomes T'. The rule is a basic action allowing the transition of the system from one state to another. The rewriting logic describes the changes of the system so that the state is represented by an algebraic term, the transition becomes a rewriting rule and the distributed structure, an algebraic structure modulo a set of axioms E. Syntax in logic of rewriting is given by a signature (Σ,E) where Σ is a set of functions and E a set of axioms. A rewriting theory $T=(\Sigma, E, L, R)$ in rewriting logic is composed of a signature (Σ,E) and by a set of labelled rules R with labels in L. These rules describe the behaviour of the system and the rewritings are performed on the classes of equivalences of the terms modulo the axioms E .

In practice a theory of rewriting $T=(\Sigma, E, L, R)$ can be used as an executable specification allowing a rapid prototyping of the modelled system and its checking. This notion of rewrite theory is very general and expressive. In the first place, it allows rewriting modulo "structural axioms" *E,* thus increasing the expressive power.

In addition, it allows conditional rules of a very general form, where the conditions need not require equalities to hold but only the existence of rewritings among pairs of terms in the condition, which further increases the expressive power. Finally, it allows labelling of the rewrite rules; this is quite natural for many applications, and customary for automata-viewed as labelled transition systems-and for Petri nets, which are both particular instances of this definition.

I.2.2 Concurrent rewriting in rewriting logic.

The states of the system are specified as algebraic data types, the basic changes which may occur in the system and in parallel are specified by rewriting rules, the set of axioms *E* capture the structural properties of the system, a possible state of the system is represented by an equivalence class $[\ t\ ]$ of a term t modulo the structural axioms s E. A rule axiomatising an elementary change has the form : $[\ t\ ] \rightarrow [\ t'\ ]$ where $[\ t\ ]$ and $[\ t'\ ]$ are partial states of the system (two sub-states). This rule mean that if $[\ t\ ]$ is a part of the whole state of the system the it is replaced by $[\ t'\ ]$.

t et t' may contain variables from a set $X = \{x_1, \ldots, x_n\}$ eventually infinite. The rule becomes $[t(x_1, \ldots, x_n)] \rightarrow [t'(x_1, \ldots, x_n)]$

Rules can be

Conditional : $[t(x_1, \ldots, x_n)] \rightarrow [t'(x_1, \ldots, x_n)]\ if\ c$

labelled : $rl : [t(x_1, \ldots, x_n)] \rightarrow [t'(x_1, \ldots, x_n)]$

Given a rewrite theory *R,* we say that *R* entails a sequent $[\ t\ ] \rightarrow [\ t'\ ]$ and write

$R \vdash [\ t\ ] \rightarrow [\ t'\ ]$ if an only if $[\ t\ ] \rightarrow [\ t'\ ]$ can be obtained by finite application of the following rules of deduction :

- Reflexivity.

$$\forall\ [t] \in T_{\Sigma,E}(X) \qquad \overline{[t] \rightarrow [t]}$$

- Congruence.

For each f (function symbol) $\in \sum_n$ n$\in$ IN

$$\frac{[t_1] \rightarrow [t'_1] \ldots [t_n] \rightarrow [t'_n]}{[f(t_1,\ldots,t_n)] \rightarrow [f(t'_1,\ldots,t'_n)]}$$

- Replacement.

For each rewrite rule $r\colon\ [t(\overline{x})] \rightarrow [t'(\overline{x})]$ we have

$$\frac{[w_1] \rightarrow [w'_1] \ \ldots\ [w_n] \rightarrow [w'_n]}{[t(\overline{w}/\overline{x})] \rightarrow [t'(\overline{w'}/\overline{x})]}$$

- Transitivity.

$$\frac{[t_1] \rightarrow [t_2]\ [t_2] \rightarrow [t_3]}{[t_1] \rightarrow [t_3]}$$

Equational logic is obtained by adding the next symmetry rule

$$\frac{[t] \rightarrow [t']}{[t'] \rightarrow [t]}$$

This rule has no meaning in rewriting logic because changes in time are irreversible.

I.2.3 Rewriting logic as a unifying paradigm.

This section discusses a variety of models of concurrency that can be obtained as special cases of concurrent rewriting [Meseguer 92]. A natural way of studying specializations of this kind is to impose restrictions on the rewrite theories being used. The most obvious restriction is fixing the set E of structural axioms in term of associativity, commutativity and identity element (A, C, I). Three cases are considered:

- **Syntactic rewriting, i.e., $E = \varnothing$**

This is the particularly simple case of unconditional rewrite theories obtained by imposing the additional requirements that $\Sigma = \Sigma_o$, i.e., Σ only involves constants, and that all the rules in R only involve ground terms, i.e., they are of the form $r : a \rightarrow b$ for *a, b* constants.

This case includes labelled transition systems, functional programming, (Parallel) Functional programming, The lambda calculus and combinatory logic.

- **String rewriting, i.e. $E = AI$, associativity and identity**

The acronym *AI* stands for associativity and identity, i.e., we assume a binary operator which is associative (therefore, we can use empty syntax for the operator and denote the result of applying the operator to arguments x and y by xy) as well as a constant λ which is an identity element for that binary operation. This is of course a world of strings where the algebraic and therefore distributed structure of a state is a linear structure, in which the linear order of the elements is fundamental and cannot be forgotten. This case includes Turing machine and Chomsky grammars.

- **Multiset rewriting, i.e., $E = ACI$, associativity, commutativity and identity**

The acronym *ACI* stands for associativity, commutativity and identity, i.e., we add to *AI* a commutativity law $xy = yx$. We keep the same juxtaposition notation, but due to commutativity we can represent *ACI* equivalence classes as a multiset of elements. The ACI axioms lead to the system a state structure that is distributed as a commutative word, multiset, or bag, which is at the same time its concurrent structure. This case includes Petri nets, the chemical abstract machine, CCS, and the general logical theory of concurrent objects of MAUDE language and the UNITY model of computation as special cases.

II.The MAUDE Language.

MAUDE [Duran & Meseguer 07] is a fully reflective programming language and development environment that utilizes rewriting logic and its equational logic sublanguage to specify formal executable environments. The reflective nature of rewriting logic allows MAUDE programs to create the actual algebra that defines the programmer's specific application. Programs are able to define their own syntax, operations, and data types to model the behaviour of concurrent systems.

II.1 Functional Modules.

MAUDE is a declarative language designed as a metalanguage to define formal systems. Every program is built from logic theories that are expressed as programming modules. Computation is equivalent to logical deduction based on the rules defined by the logic in the programs. Functional modules form the foundation of that logic. They create the data types and operations used in the equational theories. Data types are specified in terms of sorts and subsorts. The keyword sort is used to define any type in the system and subsorts are used to define more specific types within a sort. The following example creates Positive and Negative sorts as specific subsorts of the Integer sort.

subsorts Positive Negative < Integer .

Operations define the syntax used to create sorts. MAUDE allows both prefix and mixfix operators to be defined using the *op* or *ops* commands. An operation consists of the *op* keyword followed by the operator symbols, a colon, then a list of sorts for the arguments, a right arrow, the sorts for the results, and then any operator attributes. Underscores are used to specify where mixfix arguments are placed.

```
op OpName : Sort0 … Sortk -> Sort [OperatorAttributes] .
--- Examples
op + : Integer Integer -> Integer .
op _+_ : Integer Integer -> Integer .
```

The possible operator attributes include associative, commutative, identity, precedence levels, and constructor. Constructors are operations that take no arguments but have syntax and produce a sort. Identity operators do not affect the sort if included (like adding 0 or a null set). To give these operational definitions meaning, equations are used. Equations define the rules for determining equivalence and serve to simplify the operation. When operators are used in expressions, MAUDE will evaluate the equations defined for each operator to determine what expressions evaluate to. Equations must be confluent and deterministic (Church-Rosser) so they can always be reduced to a single sort. Variables are simple placeholders for sort types used to help define equations. The syntax and examples for both are given in the module below [Clavel & Duran 02].

```
**** functional module example ****
fmod CARD-DECK is
sorts Number Suit Card .
ops A 2 3 4 5 6 7 8 9 10 J Q K : -> Number [ctor] .
ops Clubs Diamonds Hearts Spades : -> Suit [ctor] .
op _of_ : Number Suit -> Card [ctor] .
op CardNum : Card -> Number .
op CardSuit : Card -> Suit .
var N : Number . var S : Suit .
eq CardNum( N of S ) = N .
eq CardSuit( N of S ) = S .
endfm
```

II.2 System Modules.

The system modules build on this equational logic by adding rewrite rules which transition a system from one state to another. This forms a full 4-tuple rewrite theory $\mathcal{R} = (\Sigma, E \cup A, \varnothing, R)$ where Σ is the signature of the type definitions, E is the set of equations, A is the set of attributes, R are the rewrite rules, and $\varnothing$ is the identity elements set of arguments to Σ. While equations specified simplifications in the system, rewrite theories in general define one-way transitions between states. Function modules can be thought of as the basic data types of a system, defining structures and their operations, and rewrite rules are all the possible methods to apply to those structures.

Each rewrite rule consists of the rewrite operator *rl*, a label enclosed in square braces followed by a colon, a required term followed by an =>, and the resulting term with any statement attributes following in square braces. The required term is like a set of arguments that must be present, but in Full MAUDE, those arguments can include conditions on attributes as well.

rl [Label] : Term-1 => Term-2 [StatementAttributes] .

Both terms must be of the same sort. Just as with equations, rewrite laws may have conditions that can be any MAUDE expression that evaluates to a Bool sort, membership axiom, or pattern to search. The conditional equations can even contain additional rewrite laws.

II.3 Full MAUDE.

Full MAUDE is an object-oriented extension of the Core MAUDE modules written in MAUDE using an interactive loop mode. Full MAUDE defines a generic syntax for objects, classes, messages, and configurations to allow the programming of event based systems. A class defines the structure of an object, and objects are specific instances of a class. A class can also be thought of as a high level sort, defining the possible object signatures. A class consists of a class identifier (Cid), which is a sort, and a list of attributes that are sorts, including class or object identifiers. Classes also support multiple inheritance with subclasses that inherit all the attributes of parent classes.

class C | attribute$_1$*: Sort*$_1$*, … , attribute*$_n$*: Sort*$_n$ *.*
< Oid_Name : C | attribute$_1$*: variable*$_1$*, … , attribute*$_n$*: variable*$_n$ *>*
msg syntax : Oid Sort$_1$ *Sort*$_n$ *-> Msg .*

Each object has an object identifier (Oid), a class identifier, and a list of the instances of its attributes. Objects serve as variables in rewrite rules but actual instances are also returned as MAUDE programs are run. Messages are assumed to be "sent" to objects to convey information, but are mostly left to the design of the programmer. They are analogous to operators but at the object level. Each message type has a name and a list of arguments that starts with a destination Oid and results in an Msg. The Msg and Configuration sorts are used as placeholders to for groups of msg and Oid sorts. Configurations are used to express the sort needed for the results of rewrite laws and equations that have many objects and messages that are not directly part of the rule.

The configuration is the distributed state of the concurrent object-oriented system and is represented as a multiset of objects and messages according to the following syntax:

subsorts Object Message < Configuration .
op -- : Configuration Configuration => Configuration [assoc comm id: ∅]

Where the operator -- is associative and commutative with identity ∅. The system evolves by concurrent rewriting (modulo ACI) of the configuration by means of rewrite rules specific to each particular system, whose left-hand and right-hand sides may in general involve patterns for several objects and messages.

The following is an example object module that specifies a bank accounts system.

```
********* object module example ************

omod ACCOUNT is
protecting INT . class Accnt .
att bal : Accnt -> Nat .
msgs credit,debit : OId Nat -> Msg .
msg transfer-from-to-: Nat OId OId -> Msg vars A B : OId .
vars MNN' :Nat.
rl credit(A,M) < A : Accnt / bal: N > => < A : Accnt / bal: N + M >
rl debit(A,M) < A : Accnt I bal: N > => < A : Accnt / bal: N - M >  if N >= M .
rl transfer M from A to B < A : Accnt / bal: N >
< B : Accnt / bal: N' > =>
<A: Accnt / bal: N - M > < B : Accnt /  bal: N' + M > if  N >= M
endom
```

Rewrite rules in Full MAUDE transition the system from a configuration of objects and messages to a new configuration of objects and messages. As messages are sent from object to object, new messages are fired and objects may be created, destroyed, or updated. As was mentioned before, rewrite rules define all the possible transitions for a concurrent system.

In MAUDE, the general form required of rewrite rules used to specify the behaviour of an object-oriented system is as follows:

$$\mathbf{m_1,\ldots,m_n} < O_1: C_1 / atts_1 > \ \ldots\ < O_m: C_m / atts_m > \Rightarrow$$
$$< O_{i1}: C_{i1} / atts_{i1} > \ \ldots\ < O_{ik}: C_{ik} / atts_{ik} >$$
$$< Q_1: D_1 / \mathbf{atts_1''} > \ \ldots\ < O_p: D_p / \mathbf{atts_p''} > \mathbf{m_1'},\ldots,\mathbf{m_q'} \quad \textit{if}\ \ C$$

where the $m_1,\ldots,m_n$ are message expressions, $O_1,\ldots,O_n$; $O_{i1}, \ldots ,O_{ik}$ and $Q_1,\ldots, Q_n$ are objects such as $\{ O_{i1}, \ldots ,O_{ik} \} \subset \{ O_1,\ldots,O_n \}$

C is the rule’s condition. A rule of this kind expresses a communication event in which *n* messages and *m* distinct objects participate. The outcome of such an event is as follows:

- The messages $\mathbf{m_1,\ldots, m_n}$ disappear;
- The state and possibly even the class of the objects $\mathbf{O_{i1}, \ldots ,O_{ik}}$ may change;
- All other objects vanish;
- New objects $\mathbf{Q_1,\ldots, Q_n}$ are created ;
- New messages $\mathbf{m_1'},\ldots,\mathbf{m_q'}$ are sent.

IV.The ECATnets.

Petri nets are well known as a powerful graphical and mathematical tool applicable to many systems. They are a promising tool for describing and studying complex systems that are characterized as being concurrent, asynchronous, distributed, parallel, nondeterministic and/or stochastic. ECATNets are a kind of High-Level Algebraic Nets [Bettaz 92][Belala *et al*. 00]. They are proposed as a way for specifying and modelling various aspects of distributed and parallel systems. They are built around a combination of three formalisms. The first two formalisms constitute a net/data model, and are used for defining the syntax of the system, in other terms to capture its structure. The net model, which is a kind of advanced Petri net, is used to describe the process architecture of the system; the data model, which is an algebraic formalism, is used for specifying the data structures of the system. The third formalism, which is a rewriting logic, is used for defining the semantics of the system, or in other words to describe its behaviour. According to this logic, the system behaviour may be explained by formal reasoning. In Figure 2 a generic ECATnet is represented.

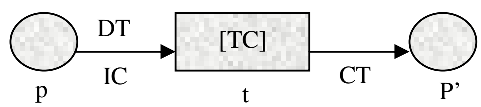


Figure. 2. A generic ECATNet.

IC(p,t) (Input Condition), DT(p,t) (Destroyed Tokens) and CT(p',t) (Created Tokens) are multisets of tokens, with $\oplus$,$\cap$, $\subset$, \ being respectively the multiset union, intersection, inclusion and difference. TC (Transition Condition) is a boolean expression which may contain variables occurring in IC, DT and CT.

The tokens are equivalence classes of terms defined on a user declared algebraic specification of an abstract data type. We let $[x]_E$ or just [X] denote the equivalence dass of x, w.r.t. the axioms (equations) given by the user in his (her) specification. Let $\varnothing_M$ denote the identity element of $\oplus$, and $[X]_\oplus$ the equivalence class of x, w.r.t. the ACI (associative, commutative and with Identity element) axioms for $\oplus$. [TC(t)] is a boolean term which may contain variables occurring in IC(p,t), DT(p,t) and CT(p',t). To each place is associated a capacity C(p) defined as a multiset of closed (equivalence classes of) terms. The marking M(p) of a place p of the net, which is itself a multiset of closed terms, is defined w.r.t. the capacity (which may be infinite). The pair (P,M(P)) define the current state of the place P.

The set of all the pairs (place,marking) is a multi_set with ⊗ multiset union and $\varnothing_B$ the identity element. The state of the E**CATNet** is the union (⊗) of the states of its places.

A transition t is fireable when various conditions are simultaneously true.

- Every IC(p,t) for each input place p is enabled.
- TC(t) is true.
- The addition of CT(p,t) to each output place p must not result in p exceeding its capacity when this capacity is finite.

When t is fired,

- DT(p,t) is removed from the input place p
- CT(q,t) is added to the output place q.

Transition firing and its conditions are expressed by rewrite rules which are strongly depending on the form of the syntactic notation used for representing IC. Those rewrite rules together with a set of deduction rules define a rewriting logic [Bettaz 92] which gives the semantics of the net.

IV The REACTnets.

The SO_layer of MOBIFLEX architecture is a distributed system whose entities are BPEL based Web-processes; these entities exhibit a dynamic and reactive behaviour. The formal specification of reactive distributed systems must exhibit the structures of control and has to imply explicitly the relevant aspects of the distribution, such as the concurrency, the reactivity and the interaction between the entities. This section proposes the fundamentals of a formal approach for the specification of reactive distributed systems based on object-oriented paradigm. Object's behaviour is modelled as REACTNets.

The REACTNets enhance the ECATNets with explicit distribution and reactivity [Bouchoul & Mostefai 07]. We associate to the classic ECATNets MAUDE rules to handle interactions between objects. The two formalisms have a common semantics in term of rewriting logic so that interesting prospects are opened for their integration.

A distributed system can be seen as a number of heterogeneous and autonomous entities which can interact by the means of suitable interfaces. The complexity of these systems increases with the number of entities which compose them.

Various works relating to formal modelling of such systems are continuously proposed for the purpose of verification or rapid prototyping; each one with different objectives, concepts, tools and possibilities. In particular, the expression of concurrency and reactivity constitutes a crucial aspect during the development of the model. Concurrency can arise between the system entities (inter-entities concurrency) and also inside the same entity (intra-entities concurrency). Reactivity deals with the possibility for the system to react dynamically to its environment.

Thanks to their logical autonomy and to their modularity, objects are naturally predisposed for the role of concurrent units. They not only make it possible to describe structural properties of the system but also to handle naturally the distribution [Dolvan *et al.* 08]. However, the object oriented approach presents an evident weakness to suitably express the dynamic aspects of distributed systems. For this reason, the objects are often enhanced with formalism for the description of the dynamic aspects of their behaviour. In particular, the approaches associating Petri nets and objects are more and more gaining the interest of several groups of researchers. We shall propose the fundamentals of a formal approach for the specification of reactive distributed systems with true concurrency semantics at inter- and intra-entities level. The idea is to associate the ECATNets(Extended Concurrent Algebraïc Term Nets) [Bettaz *et al.* 93] and the theory of concurrent objects proposed by Meseguer in MAUDE [Meseguer 02bis]. The ECATNets are a kind of high level algebraic Petri nets with rewriting logic semantics. First we propose the REACTNets that enhance traditional ECATNets with reactivity[Bouchoul & Mostefai 07]. REACTNets should be used to describe individual objects' behaviours and to express not only the actions which the object carries out but also its interactions with its environment in term of messages emitting/receiving.

IV.1. Object/Petri nets formalisms.

IV.1.1 The object/Petri net complementarity.

Object/Petri nets association is based mainly on the interesting complementarity of the two formalisms for the specification of distributed systems. Petri nets deal with the most crucial aspects of concurrency; objects offer necessary tools to express various aspects of distribution.

Furthermore, distributed systems are often reactive and the behaviour of a reactive system is usually modelled by event-condition-actions rules called commonly production rules or simply ECA rules (Event-Condition-Action); The significance of an ECA rule is that if the event in the environment occurs, and the condition is true, the reactive system performs the action[Eshuis *et al.* 03]. The problem is that the token-game semantics of Petri nets does not model behaviour of reactive systems, the non-reactivity of the token-game semantics can be seen immediately from the definition of the firing rule.

A transition in a Petri net is enabled once the conditions of firing are true, however the environment of the Petri net does not influence in any way its firing. In contrast, in a reactive system a relevant transition needs some additional input event to become enabled. So, the token-game semantics models closed systems, whereas a reactive system is open, otherwise it cannot interact with its environment. Furthermore, in a reactive system an enabled transition must fire immediately. In the token-game semantics, an enabled transition may fire, but does not necessarily have to. A Reactive Petri net can simply be built by changing for internal transitions the rule "the transition may fire" by the rule "the transition must fire"; [Eshuis *et al.* 03]; while for external transitions expressing the interactions with the environment the traditional rule can be preserved to ensure the network stability. Thus a reactive Petri net has two possible states: stable and unstable. The system must continue to fire the internal enabled transitions as a long time as it does not reach a stable state, in other words until no internal transition is enabled; before being able to fire external transitions from the environment. But, the Petri net must explicitly comprise sufficient constructions to model the interaction with the environment by external transitions handling the events that influence its internal behaviour and expressing the reactivity. For this purpose object paradigm offers to Petri nets an elegant solution. And we can conclude that the complementarity of the two approaches is twofold, on one hand objects need Petri nets to express their dynamic behaviour and on the other hand Petri nets need objects to have modularity and reactivity through object interaction mechanisms.

IV.1.2 Object / Petri net Approaches: state of the art.

The object/Petri net association is not new, and among the multitude of works integrating objects and Petri nets, two tendencies are distinguishable, designated successively by "Objects in Petri nets" and "Petri nets in objects" [Bastide 95].

The principle of the "objects in Petri nets" approach is to model a system by a single Petri net, whose tokens are objects. This single network can be structured by using a hierarchical decomposition, typically in the form of super-transitions or super-places. The type of tokens is described in an external formalism to Petri nets, for instance an algebraic notation or a programming language.

The formalism POP/POT [Engelfriet 90] belongs to this type of approaches. POT (Parallel Object–based Transition) system is another example: A POT is a simple Petri net where objects are tokens with associated structures of memories; the state of an object is explicitly modelled by places. Another example is given by LOOPN [Lakos & Keen 91] which is a language for simulation and specification of distributed systems with timed coloured Petri nets. It includes object properties such as the sub-typing, inheritance and polymorphism which allow an adequate modularization of complex specifications.

The "Petri nets in objects" approach consists in using Petri nets to describe the internal behaviour of the objects. This approach proposes to model the system by several independent Petri nets (objects) which can interact. The network marking models the internal state of the object and the transitions model the execution of its methods. The fundamental interest of this type of approach is to allow the use of the concepts resulting from the object paradigm (classification, encapsulation) to describe the structure of the system, instead of using a purely hierarchical structuring.

The COOPN (Competitor Object Oriented Petri Net) [Buchs & Guelfi 00] and PROTOB [Baldassari *et al*. 91] belong to this type of formalisms. In particular, PROTOB is a C.A.S.E (Computer Aided Software Engineering) for the specification, simulation and prototyping of the concurrent systems. A PROTOB Object is defined by its attributes, actions and communication ports. The behaviour is described by a PROT which is a high level Petri net which integrates Petri nets and DFDs (DataFlow-Diagramms). In [Wang é Wu 98] another similar formalism is presented: the CTOPN (Colored Timed Object-Oriented Petri-Nets) are proposed for the modelling of the automated manufacturing systems.

Objective-Linda [Holvoet & Kielmann 98] is another formalism for the formal specification of active objects' behaviour, using high level Petri nets (HLPN). The EP-Nets [Guan & Lim 02] associating objects and Petri nets are proposed for the modelling of the interactive multi-media orchestrations. In [Baresi & Pezzè 01] the dynamic model of UML is enhanced by high level timed Petri nets to cover the language gaps. Another example is given by HOONets (Hierarchical Object- Oriented Petri Net).

HOONEts deal with several oriented object aspects such as abstraction, encapsulation, modularity, interaction by messages, inheritance and polymorphism [Hong & Bae 00]. However the work closest to our proposed approach is probably the CO-Nets [Aoumeur & Saake 02]; the CO-Nets constitute a multi-paradigm integrating algebraic Petri nets and the object-oriented paradigm, the model is semantically interpreted by a rewriting logic theory largely inspired from that of ECATNets.

IV.2 The ECAObjects.

The REACTNets results from the integration of the ECATNets and MAUDE; in addition to the advantages of an object/Petri nets association as explained above, the two formalisms have the same semantics based on rewriting logic; so that this common semantic enable an homogeneous integration, on the other hand this association makes it possible to specify not determinist distributed systems with a true concurrency semantics at inter-object level (thanks to MAUDE rules) and intra-object level (thanks to ECATNets); finally the object paradigm adds the distribution and reactivity which are missing in traditional Petri nets to ECATNets. The object that we call ECAObject (Figure 3) is described by its structural aspects and its behavioural aspects. The structure of an ECAObject consists of its static description in term of its name (unique identifier), its attributes, its communication ports and the events describing its behaviour.

The attributes model the ECAObject's static properties such as:

- Parameters of ECAObject (name, first name, age…).
- States of ECAObject (busy, idle …).
- References to other ECAObjects

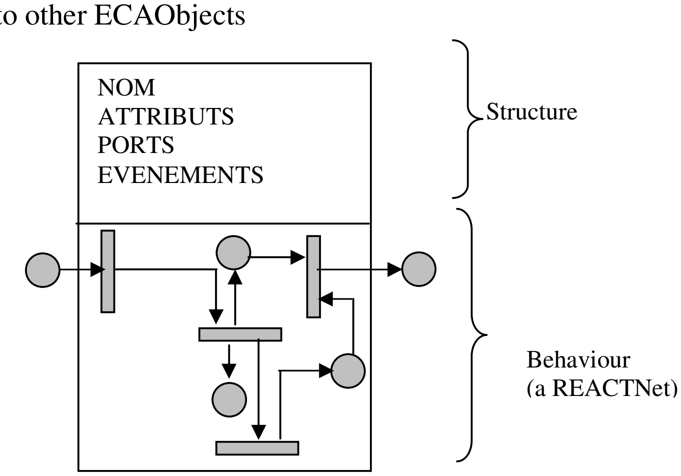


Figure. 3. Abstract Architecture of an ECAObject

The ports are the ECAObject's access points used for messages' emission and reception. The current state of an ECAObject is given by the set of its attributes. The event is the elementary activity of the ECAObject dependent on its state and modifying it. It is the granule of its concurrent behaviour. An ECAObject can carry out several events in parallel. The identification of the events depends on the level of abstraction agreed to describe this behaviour.

The events can be either internal (local operations in the ECAObject) or visible (emission or interception of messages). The visible events constitute the interface of ECAObject and model the services needed or offered by him. The behaviour of the ECAObject consists of its dynamic evolution and can be described by the set of its acceptable life cycles. A life cycle represents a possible succession of events implying this ECAObject during its evolution and can comprise concurrency, mutual exclusion, and sequencing.

A place may be:

- An attribute of the ECAObject
- A port for an external interaction
- An intermediate place added for the needs of specification

The behaviour of an ECAObject is described by a REACTNet exhibiting not only its internal events but also its external events expressing its interaction with the environment through emission or reception of message in specific ports (Figure 4). The places P-out (emission) and P-in (reception) are communication ports for the ECAObject's visible events. This case of figure could be brought back to a composition by transition (also called by rendez-vous) of the two Petri nets (Figure 5). It is a particular case of the composition by a sequential process and it was proven that properties of aliveness and boundness are preserved in the composite network [Souissi & Memmi 90]. In addition we agree that REACTNets are considered with respect to the stability rules of classical reactive Petri nets theory as presented in [Eshuis *et al*. 03]. The communication ports allow specifying the simultaneous emission and interception of several different messages in parallel whereas the input/output places of classical Object/Petri nets approaches are generally managed in FIFO in accordance with the traditional vision of communication ports of concurrent objects. The transition T models an internal event which is an action undertaken by the ECAObject. Let us note here that any change which can affect the state of an ECAObject (its attributes) constitutes a stage of one of its possible life cycles and have to be expressed in the REACTNet.

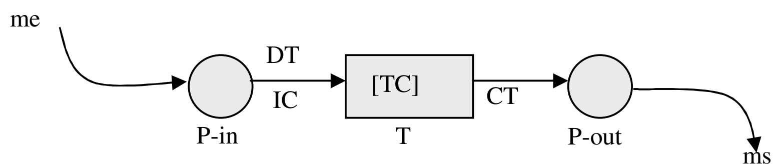


Figure 4. A generic REACTNet.

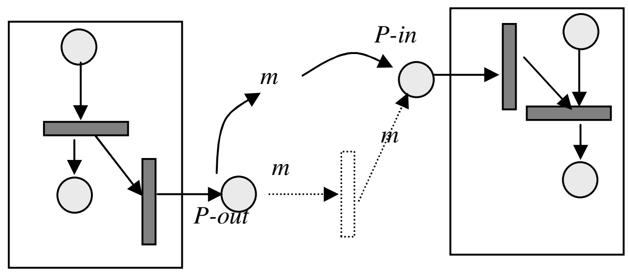


Figure. 5. Emission/reception of messages as a composition by "rendez-vous"

IV.3 The REACTNets' semantics.

IV.3.1 The state of the REACTNet as an object configuration.

The state of the system called configuration is specified as a multi-set of ECAObjects and messages, provided with an operator ACI, with the identity element $\varnothing$. The pair (P, M (P)) defines the current state of the place P. the set of these pairs (place, marking) has a structure of a multi-set with $\otimes$ union on this multi-set and $\varnothing_B$ the identity element. The state of the ECAOBJect is the union ($\otimes$) of the states of all its places and is expressed with the term $< O: C/ P_1: m_1 ... P_n: m_n >$ Where,

- O: name of the ECAOBject.
- C: classe of the ECAOBject P_i: i^{th} place of the associated REACTNets.
- $m_{i:}$ marquing of the i^{th} place of the associated REACTNets.

The tokens are algebraic terms. IC(input condition), DT(Destroyed Tokens), CT(Created Tokens) are multi-sets of terms (tokens), where $\oplus$, $\cap$, $\subset$, Θ stand for respectively union, intersection, inclusion and difference on the multi-sets and $\varnothing_M$ the element identity. [TC] is a Boolean algebraic expression eventually containing variables appearing in IC, DT and CT.

To each place P are associated a sort S(P) and a capacity C(P) defined as a multi-set of closed terms (constants). The marking M (P) of a place is defined in respect of its capacity (which can be infinite).

The transition T materializes an internal event and is enabled if the following conditions are true:

- IC (P1, T) is enabled: IC indicates the multi-set of tokens that have to be present in P1
- TC (T) is true.

the addition of CT to the place P2 must not result in exceeding its capacity

When T is fired

- The multi-set M (P1) $\cap$ DT) is removed from the input place P1.
- The multi-set CT is added to P2.

IV.3.2 The rewriting theory of the system.

A REACTNets-based specification has a rewriting logic semantics combining the semantics of ECATNets and that of MAUDE and therefore it is a particular case of a conditional rewriting theory. The rewriting system obtained inherits the four groups of ECATNets' rules to which we add two other groups derived from MAUDE, the first one expresses the reactivity by the means of the interaction with environment and the second models the creation/destruction of objects.

IV.3.2.1 Equational logic rules: These rules are derived from the algebraic equations describing the types of tokens (by ADTs). Usually, the ECATNets use OBJ3 [Goguen at al. 87] as a functional sub-language. The evaluation of the tokens can be done using a concurrent equational rewriting.

IV.3.2.2 Transitions rules: The form of the rules derived from the transitions depends on the form of IC. The form of the rule is derived from the ECATNets rules as well as MAUDE events of communication in the sense that we explicitly express the object nature of the REACTNet. If we suppose that the generic REACTNet presented at the preceding paragraph is associated to an object O of class C which we represent in accordance with MAUDE notation by the expression < O:C >, we will have the following cases:

- **Case 1: IC is of the form [m]$_{\oplus}$**

IC = DT

We agree to express the rule as follows:

$T: < O{:}C/P1{:}\ IC > \ \Rightarrow < O{:}C/P2{:}\ CT >$

Where expressions P1: IC and P2: CT are in conformity with the ECATNet notation i.e. they respectively express the suppression of IC of P1 and the addition of CT to P2.

IC $\cap$ DT = $\emptyset_M$

The multi-set IC must be included in M(P) but does not have to be removed after firing, to express it the idea is to transform IC into itself:

$T: < O{:}C/P1{:}\ IC;\ P1{:}DT \cap M\ (P1) > \ \Rightarrow < O{:}C/P1{:}IC;\ P2{:}CT >$

IC $\cap$ DT $\neq \emptyset$M

For this case, it was shown [Bettaz & Maouche 92] that it is possible to split the transition T in two transitions T1 and T2 of the simple type (two preceding cases) whose simultaneous firing is equivalent to that of T so we derive two rules.

$T1 : < O{:}C/P1{:}\ IC_1 > \ \Rightarrow < O{:}C/P2{:}\ CT_1 >$

$T2: <O{:}C/P1, IC2> \otimes <O{:}C/P, DT2> \ \Rightarrow \ <O{:}C/P, IC2> \otimes \ <O\ {:}C/P2, CT2>$ With :

$IC= IC1 \cup IC2,\ DT= DT1 \cup DT2$

$IC1 = DT1,\ IC2 \cap DT2 = \emptyset_M$

- **Case 2 : IC is of the form ~ [m]$_{\oplus}$**

The form of the rule is given by:

$T :< O{:}C/P1{:}\ DT \cap M\ (p) > \ \Rightarrow < O{:}C/P2{:}\ CT >$ if $(IC \backslash\ (IC \cap M(p) = \emptyset_M\ ) \Rightarrow$ false

- **Case 3: IC= $\emptyset_M$**

The form of the rule is given by:

$T: <O{:}C/P1, DT \cap M(p)) \Rightarrow (O{:}C/P2,CT)$ if $(M(p) = \emptyset_M) \Rightarrow$ true

When the place capacity C(p) is finite, the conditional part of the rewrite rule will include the following component:

$(CT \oplus (M(p) \cap C(p)) \Rightarrow CT \otimes M(p)$ (**Cap**)

In the case where there is a transition condition TC, the conditional part of our rewrite rule must contain the following component: $TC \Rightarrow$ true

IV.3.2.3 Identity rules.

$\varnothing_M \oplus X \Rightarrow X$

$\varnothing_B \otimes Z \Rightarrow Z$

IV.3.2.4 Inferences rules.

The two following rules allow by splitting and recombination of the set of tokens, to carry out the rewriting rules with a maximum of concurrency at the level of the ECAOBJect itself, in fact this splitting/recombination of the state of the ECAOBject exhibits explicitly intra-object concurrency which is missing in MAUDE.

- **Splitting:** $< O{:}C/\ P{:}X \oplus Y > \Rightarrow < O{:}C/\ P{:}X > \otimes < O{:}C/\ P{:}Y >$
- **Recombination:** $< O{:}C/\ P{:}X > \otimes < O{:}C/\ P{:}Y > \Rightarrow < O{:}C/\ P{:}X \oplus Y >$

IV.3.2.5 Visible events rules : They are asynchronous events related to the ports of the ECAObject. The explicit separation between the communication interface and the other activities for the same object makes it possible to have an additional level of intra–object concurrency.

The communications can be done in a completely independent manner of the internal activities.

- **Intercepting a message.**

This rule can be expressed according to the adopted syntax as follows:

$m < O{:}C > \Rightarrow < O{:}C/(P\text{-}in, m) >$

- **Emitting a message.**

The agreed rule is as follows:

$< O{:}C/\ (P\text{-}out, m) > \Rightarrow m < O{:}C >$

IV.3.2.6 Object creation/destruction rules : The object creation/destruction model considered is borrowed from that of MAUDE and inherits in particular, its declarative nature.

- **Object creation.**

The creation of an object requires a rule which makes it possible to specify explicitly that a message m_C is a creation message, while revealing the object created on the right of the rule in accordance with MAUDE syntax .

Example: $m_C \Rightarrow < O{:}C/S >$

This rule specifies that m_C is a message of creation; the effect is the generation of an object O of class C; S is the initial state of the associated REACTNet , i.e. the pairs set (place: marking) which starts the life cycle of the ECAObject created.

The identity of the ECAObject O and its initial state S can be the message parameters. Creation can be made, as presented in [Meseguer 92bis] in two stages, initially the sending of a message to a particular object (Meta–object) associated to the class then the emission by this last of the effective message of creation. The objective is to manage the unicity of the identity and the validity of the creation.

- **Object destruction.**

The destruction can be specified by the interaction of a destroying message and the object to be destroyed, which will have to disappear from the right of the rule.

Example: $m_D < O{:}C > \Rightarrow \varnothing$

Just as for creation, the destruction of an object can be processed by a particular object (a priori the same charged by creation) in order to check that the object to be destroyed really exists and to eliminate it in the affirmative from the list of the objects of the current configuration, by transmitting the destructive message.

IV.4 Case study : The router system.

The usage of multiple switches to connect test points or devices to instruments for the purpose of testing, measuring or monitoring some systems such as industrial ones through router systems is very common. Although this example is not relevant to our work, the choice is motivated by the high degree of parallelism implied in such systems so that we can show the possibilities of the REACTnets.

IV.4.1 Abstract specification.

The system is composed of several senders and several receivers communicating via the router. A sender emits from a queue of packets. Each emitted packet must be acknowledged. The sender does not send a new packet to a given receiver if its predecessor is not acknowledged yet. The receiver receives the packets in a queue. For each received packet, an acknowledgement is sent to the sender. The router has at a given moment a set of packets and acknowledgements to treat. It can intercept many packets and acknowledgements in parallel and rout them in the same time to the receivers.

IV.4.2 A formal model for the router system.

The system is composed of three ECAObjects classes: Sender, Receiver and Router. The messages' exchange between these three ECAObjects can be done according to the protocol presented in figure 6 where S, RT and R are respectively, the ECAObjects of the Sender class, the Router class and the Receiver class: The sender S sends a Pck(S, D, R) message to the router RT who transmits it to the concerned receiver R in the form of the routed message (S, D, R). D (Data) is the contents of the message.

After the message reception, the receiver R returns an acknowledgement Ack(R, S) which is routed to S in the form (R,S). The distinction between packets and acknowledgements before and after routing is necessary since each message type is associated to a distinct visible event. Indeed, the Pck(S message, D, R) have to be intercepted by the router RT whereas the message (S, D, R) have to be intercepted by the receiver S.

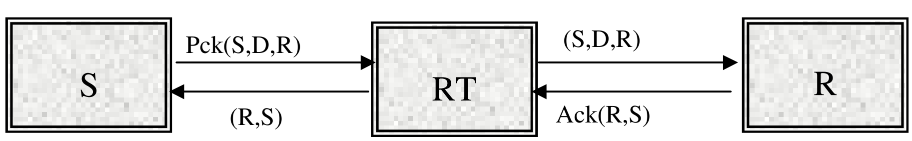


Fig 6. Interaction protocol between the ECAObjects

IV.4.2.1 The ECAObject Receiver.

- Attributes:

  Recq : queue of received packets.

  Ports:

  Ack_out, Pck_in

- Internal events:

  Treating–Pck: processing of a received packet (queuing in Recq and emission of an acknowledgement)

- Visible events:

  Output messages: {Ack(R,S) }

  Input messages: { (S,D,R) }

IV.4.2.2 The ECAObject Router.

- Attributes:

  Acknowledgement: a set of packages and acknowledgements to be treated at a given moment.

- Ports:

  Ack_in, Pck_in , Ack_out, Pck_out

- Internal events:

  Routing–Pck : Routing of a packet.

  Routing–Ack : Routing of an acknowledgement.

- Visible events:

  Output Messages: { (S,D,R), (S, R) }

  Input Messages: {Ack(R,S), Pck(S,D,R)}

IV.4.2.3 The ECAObject Sender.

- Attributes:

  Sendq: queue of the packets to emit.

  Receiver: identifier of the receiver from which an acknowledgement is expected.

- Ports:

  Ack_in, Pck_out

- Internal events:

  Emitting–Pck : this action consists in emitting a packet when the conditions are true (file not empty and no acknowledgement waited from the receiver)

  Treating–Ack: processing of a received acknowledgement.

- Visible events:

  Output Messages: { Pck(S,D,R) }.

  Input Messages: {(R, S)}.

IV.5 The REACTNets of the router model.

The type Queue[elt] is supposed to be predefined with the usual operations Remove, Empty, Add. We consider the functions Send, Rec and Data which give respectively for a packet or an acknowledgement the sender (S), the receiver (R) and the data (D):

IV.5.1 The REACTNet "Receiver".

The packets (S, D, R) are received in the input port Pck–In (Figure 7). For each received message, an acknowledgement is emitted via the output port Ack–Out. The data D is added to the file Q in the Recq place. The parameter id used is supposed referring the identity of the object associated to the REACTNet.

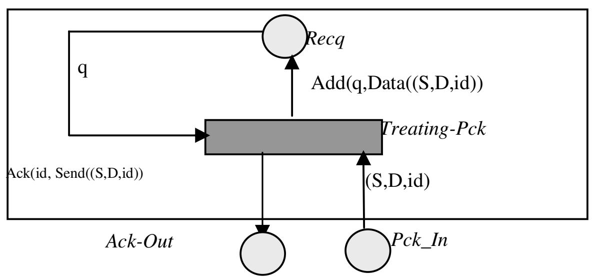


Fig. 7 The REACTNet Receiver

IV.5.2 The REACTNet "Sender".

Sendq contains the file Q of the packets (Pck(S, D, R)) to emit. The packets are emitted via the output port Pck–Out (Figure 8). For any emission a reference of the receiver R whose a acknowledgement is awaited is stored in the Receiver place. A packet Pck(S, D, R) is emitted only if no acknowledgement is awaited from the receiver R. The expression ~Rec(Head (Q)) expresses that the identity of the receiver of the packet at the head of file should not be in the Receiver place and $\varnothing_m$ indicates that no token is destroyed. The acknowledgements are received in the port Ack–In.

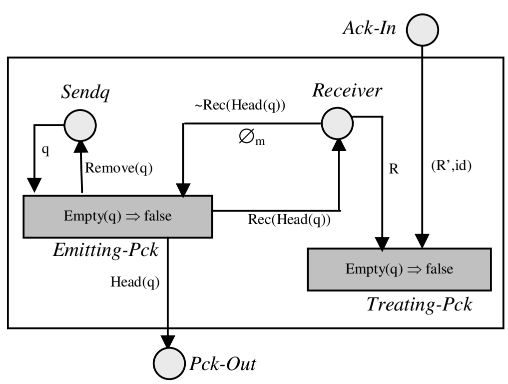


Figure 8. The REACTNet Sender

IV.5.3 The REACTNet "Router".

The Router ECAObject (Figure 9) has two input ports Pck–In and Ack–In, respectively for the packets and the acknowledgements and two output ports Pck–Out and Ack–Out. messages and are collected in the place "Messages". The transitions Receiving–Pck and Receiving–Ack are used to pass the received messages of the input ports to the place "Messages"

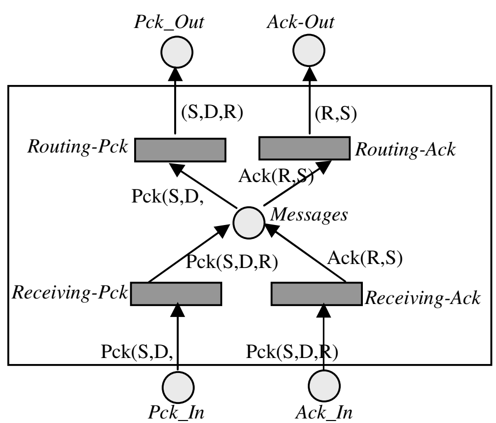


Figure 9. The REACTNet Router

IV.6 Specification of the system : The Object module "ROUTER.

The module object ROUTER (Listing 1) is the specification in the adopted MAUDE notation of the example of the router introduced in precedent paragraphs. The following types are supposed to be predefined: Mset[elt] (multi-set of elements), Queue [elt] (file of elements) and Bits (sequence of bits). We show in what follows how the specification above can be used for a rapid prototyping of the system, we start from a given configuration and rewrite the prototype.

Initial Configuration:

<RT:Router/(Pck–Out,$\varnothing_m$)⊗(Ack–Out,$\varnothing_m$) ⊗(Messages,$\varnothing_m$)⊗(Pck–In,$\varnothing_m$) ⊗(Ack–In,$\varnothing_m$)>

<S1:Sender/(Sendq, Pck(S1,D1,R1).Pck(S1,D2,R2))⊗(Pck–Out,$\varnothing_m$) ⊗ (Ack–In,$\varnothing_m$) ⊗(Receiver,$\varnothing_m$)>

<S2:Sender/(Sendq, Pck(S2,D3,R2)) ⊗(Pck–Out,$\varnothing_m$) ⊗(Ack–In,$\varnothing_m$)⊗(Receiver,$\varnothing_m$)>

<R1:Receiver/ (Recq,$\varnothing_m$)⊗ (Ack–Out,$\varnothing_m$)⊗(Pck–In,$\varnothing_m$)> <R2:Receiver/ (Recq,$\varnothing_m$)⊗ (Ack–Out,$\varnothing_m$)⊗(Pck–In,$\varnothing_m$)>

Step (1)

Fireable rules:

Object S1: Emitting–Pck

Object S2: Emitting–Pck

Step (2)

Fireable rules:

Object S1: R1

Emitting–Pck

Object S2: R1

Step (3)

Fireable rules:

Object S1: R1

Object RT: R100 (twice)

Step (4)

Fireable rules:

Object RT : R100

Receiving–Pck (twice)

Step (5) and final

Fireable rules:

Object RT: Receiving–Pck

Final Configuration

<RT:Router/(Pck–Out, (S1,D1,R1) $\oplus$ (S2,D3,R2)) $\otimes$ (Ack–Out, $\varnothing_m$) $\otimes$ (Messages, Pck(S1,D2,R2)) $\otimes$ (Pck–In $\varnothing_m$,) $\otimes$ (Ack–In, $\varnothing_m$)> <S1:Sender/(Sendq, $\varnothing_m$) $\otimes$ (Pck–Out, $\varnothing_m$) $\otimes$ (Ack–In, $\varnothing_m$) $\otimes$ (Receiver, R1 $\oplus$ R2)> <S2:Sender/(Sendq, $\varnothing_m$)) $\otimes$ (Pck–Out, $\varnothing_m$) $\otimes$ (Ack–In, $\varnothing_m$) $\otimes$ (Receiver, R2)> <R1:Receiver/ (Recq, $\varnothing_m$) $\otimes$ (Ack–Out, $\varnothing_m$) $\otimes$ (Pck–In, $\varnothing_m$)> <R2:Receiver/ (Recq, $\varnothing_m$) $\otimes$ (Ack–Out, $\varnothing_m$) $\otimes$ (Pck–In, $\varnothing_m$)>

```
OMOD  ROUTER
protecting configuration / specification of the  configuration withsortes msg, objects
and Oid(object identifier) and communication events
  protecting Queue[elt]
protecting Mset[elt]
 protecting Bits
make  Msg–queue  is Queue[msg]  endmk
make  Msg–mset  is Mset[msg]  endmk
 msg   Pck(–,–,–) : Oid Bits Oid  →  msg
 msg   (–,–,–) : Oid Bits Oid  →  msg
 msg  Ack(–,–) : Oid Oid  →  msg
 msg   (–,–) : Oid Oid  →  msg
var q:msg–Queue
var S,R,R',RT : Oid
var D : Bits
Class Sender /  Atts: Sendq:Msg–queue, Receiver:Oid; Ports :Ack–In,Pck–
Out:Msg–mset
Emitting–Pck : <S:Sender|(Sendq,q) ⊗ Receiver,∅m)> ⇒
<Sender|(Sendq,Remove(q)) ⊗ (Pck–Out,Head(q)) ⊗ (Receiver, Rec(Head(q))>  if
((Empty(q)⇒false) and (M(Receiver)Θ(M(Receiver) ∩ Rec(Head(q)))= ∅m)⇒false)
Treating–Ack : <S:Sender|(Receiver,R) ⊗ (Ack–In, (R',S))> ⇒ ∅B  if ((R=Rec
((R',S))) ⇒ true)
R1 : <S:Sender|(Pck–Out,Pck(S,D,R))> ⇒ <S:Sender> Pck(S,D,R) **Pck–Out rule
R2 : (R',S) <S:Sender> ⇒ <S:Sender|(Ack–In, (R',S))> | Ack–In rule
Class Receiver |Atts: Recq:Msg–queue; Ports :Ack–Out,Pck–In:Msg–mset.
Treating–Pck :  <R:Receiver|(Pck–In,(S,D,R)) ⊗ (Recq,q) > ⇒
  <R:Receiver|(Recq, Add(q,Data((S,D,R)))) ⊗ (Ack–Out,Ack(R,Send (S,D,R))>
R10 : <R:Receiver,(Ack–Out,Ack(R,S))> ⇒   <R:Receiver> Ack(R,S) ** règle
associée à la  place Ack–Out.
R20 : (S,D,R) <R:Receiver> ⇒ <R:Receiver|(Pck–In,(S,D,R))>|  Pck–In rule.
Class Router | Atts :Messages; Ports: Pck–Out,Ack–out,Ack–In,Pck–In : Msg–mset
Receiving–Pck : <RT:Router|(Pck–In,Pck(S,D,R)> ⇒
<RT:Router|(Messages,Pck(S,D,R))>
Receiving–Ack : < RT:Router |(Ack–in,Ack(R,S)> ⇒
<RT:Router|(Messages,Ack(R,S))>
Routing–Pck : <RT:Router|(Messages,Pck(S,D,R)> ⇒<RT:Router|(Pck–Out,
(S,D,R))>
Routing–Ack : < RT:Router |(Messages,Ack(R,S)> ⇒ <RT:Router|(Ack–Out,
(R,S))>
R100 : Pck(S,D,R)<RT:Router> ⇒ <RT:Router|(Pck–In,Pck(S,D,R))> ** Pck–In
rule.
R200 : Ack(R,S)<RT:Router> ⇒ <RT:Router|(Ack–In, Ack(R,S))> **  Ack–In rule.
R300 : <RT:Router|(Pck–Out, (S,D,R))> ⇒ <RT:Router> (S,D,R) ** Pck–Out rule.
R400 : <RT:Router|(Ack–Out, (R,S))> ⇒<RT:Router > (R,S) **Ack–Out rule.
ENDOMOD
```

Listing 1. The object module router

REFERENCES

[Aoumeur & Saake 02] N. Aoumeur and G. Saake "A component-based Petri net model for specifying and validating cooperative information systems". In Data & Knowledge Engineering, Vol. 42[2], pp 143-187. 2002,

[Baldassari *et al*. 91] M. Baldassari, G. Bruno and A. Castella "PROTOB: an object oriented CASE tool for modelling and prototyping distributed systems" in Software practice and experience Vol. 21[8], pp 823-844. 1991

[Baresi & Pezzè 01] L. Baresi and M. Pezzè, "Improving UML with Petri nets". In Electronic Notes in Theoretical Computer Science, Vol. 44[4], pp 107-119. 2001

[Bastide 95] R. Bastide "Approaches in unifying Petri nets and the object-orientation approach". In 1st Workshop on Object-oriented Programming and Models of Concurrency, Torino, Italy, 1995

[Belala *et al*. 00] F. Belala, M. Bettaz, L. Petruci. "Concurrent Systems Analysis Using ECATNets". Logic Journal of the IGPL, Vol. 8[2] pp 149-164, 2000.

[Bettaz *et al*. 93] M. Bettaz, M. Maouche, M. Soualmi and M. Boukebeche. "Protocol specification using ECATNets". In Networking and distributed computing, Hermes, Paris, Vol. 3[1] pp 7-35 1993

[Bettaz & Maouche 92] M. Bettaz and M. Maouche "How to specify non determinism and true concurrency with algebraic term Nets". In LNCS 655, pp 164 – 180. 1992

[Bettaz & Maouche 94] M. Bettaz and M. Maouche "On the specification of protocol objects". 10th ADT + Compass general meeting'94, Genova, Italy 1994

[Bouchoul & Mostefai 07] F. Bouchoul and M. Mostefai "Vers une approche formelle pour la spécification des systèmes distribués réactifs". Iternatioanl Conference on computer integrated manufacturing CIP'07 SETIF 3-4 novembre 2007

[Boudiaf *et al*. 06] N. Boudiaf, K. Barkaoui and A. Chaoui "Implémentation des règles de réduction Des ECATNets dans MAUDE" . 6e Conférence Francophone de MOdélisation et SIMulation -MOSIM'06 - du 3 au 5 avril 2006 – Rabat – Maroc pp 505-514

[Bounoua & Bettaz 92] Bounoua O., and Bettaz M.. "A graphical editor-simulator for Algebraic term Nets". Proceeding of the second maghrebian conference on SE and AI, Tunis 1992 pp 2-16

[Bruni & Meseguer 06] R. Bruni, J. Meseguer "Semantic foundations for generalized rewrite theories" Theoretical Computer Science, Vol. 360[1,3], pp 386-414. August 2006

[Buchs & Guelfi 00] D. Buchs, and N. Guelfi "A Formal Specification Framework for Object-Oriented Distributed Systems" IEEE Transactions on Software Engineering Vol. 26[7], pp 635 - 652 July 2000

[Clavel & Durán 02] M. Clavel, F. Durán "MAUDE: specification and programming in rewriting logic" Theoretical Computer Science, Vol. 285[2], pp 187-243. August 2002,

[Clavel *et al*. 03] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer and C. Talcott, "The MAUDE 2.0 System". In Proc. Of Rewriting Techniques and Applications, , Springer-Verlag LNCS 2706 , pp 76-87. 2003

[Djemame *et al*. 98] K. Djemame, D.C. Gilles, L.M. Mackenzie and M. Bettaz, "Performance comparison of high-level algebraic nets distributed simulation protocols". Journal of Systems Architecture, Vol. 44[6,7], pp 457-472. March 1998,

[Dovland *et al*. 08] J. Dovland, E- B Johnsen, and O.Owe "Observable Beheviour of Dynamic Systems: Component Reasoning for Concurrent Objects" Electronic Notes in Theoretical Computer Science 203: pp 19-34, 2008. Elsevier

[Duran & Meseguer 07] F. Durán and J. Meseguer "MAUDE's module algebra" Science of Computer Programming, Vol. 66[2], pp 125-153. April 2007

[Engelfriet 90] J. Engelfriet, G. Leth, and G. Rozenberg, "Net based description of parallel object based systems or POTs and POPs". Technical report, Noordwij Kerhool fool workshop 1990.

[Eshuis *et al*. 03] R. Eshuis and J. Dehnert "Reactive Petri nets for workflow modelling". In In W.M.P. van der Aalst and E. Best editors, Application and Theory of Petri Nets, LNCS 2679, pp 295-314. 2003

[Goguen *et al*. 87] C. Goguen, H. Kirshner, J. Meseguer, A. Megrelis and T.Winkler. "An introduction to OBJ3". In J.-P. Jouannaud and S. Kaplan, eds., Proc. of the 1st International Workshop on Conditional Term Rewriting Systems, France, LNCS 308, pp 258-263. SpringerVerlag, 1987

[Guan & Lim 02] S-U Guan and S-S. Lim "Modelling with enhanced prioritized Petri nets: EP-nets". Computer Communications, Vol. 25[8], pp 812-824. 2002

[Hicheur *et al*. 06] A. Hicheur, K. Barkaoui, and N. Boudiaf. "Modelling Workflows with Recursive ECATNets". In 8th IEEE Int'l. Sympos. on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC'06), pp 389-398, 2006.

[Holvoet & Kielmann 98] T. Holvoet and T. Kielmann. "Behaviour specification of parallel active objects". In Parallel Computing, Vol. 24, Issue 7, July 1998, pp 1107-1135

[Hong & Bae 00] J.E. Hong and. D.H. Bae. "Software modelling and analysis using a hierarchical object-oriented Petri net". In Information Sciences, Vol. 130[1-4], pp 133-164. 2000

[Lakos & Keen 91] C.A. Lakos and C.D. Keen "Modelling Layered Protocols in LOOPN", Proc. of the Fourth International Workshop on Petri Nets and Performance Models, Melbourne, 1991

[Meseguer & Rosu 07] J. Meseguer and G. Rosu "The Rewriting Logic Semantics Project" Theoretical Computer Science Vol. 373[3], pp 213-237. 2007

[Meseguer 92] J. Meseguer, Conditional rewriting logic as a unified model of concurrency, Theoretical Computer Science Vol. 96[1] pp 73–155. 1992

[Meseguer 92bis] J. Meseguer "A logical theory of concurrent objects and its realization in the MAUDE language" . Technical report SRI CSL 1992, SRI international

[Şerbănuţă *et al.* 09] T.F. Şerbănuţă, G. Roşu and J. Meseguer "A rewriting logic approach to operational semantics" Information and Computation, Vol. 207[2], pp 305-340. 2009

[Souissi & Memmi 90] Y. Souissi, and G. Memmi, "composition of nets via a communication medium". LNCS 483 advances in Petri Nets 1990 pp 457–470

[Wang & Wu 98] L. C. Wang and S. Y. Wu "Modelling with colored timed object-oriented Petri nets for automated manufacturing systems". In Computers & Industrial Engineering, Vol. 34[2] pp 463-480. 1998

[Wilbur 87] M.C Wilbur Ham 1987. "Numerical Petri nets, a guide version 2" , Telecom Australia, Research Laboratories, 1987

CHAPTER V

A FORMAL FRAMEWORK FOR MOBIFLEX. PART II : REACTNets and MAUDE.

I. Formal design patterns for the SO_Layer.

The SO_layer of MOBIFLEX architecture is Web-Service based, each node can be a simple web-service or a complex Web-process obtained by a composition of two or more other Web-Services. At their early stages linking Web-Services together into a business process or a composition was possible with two languages the WSFL from IBM and XLANG from Microsoft. The Business Process Execution Language for Web-Services (BPEL) represents the merging of WSFL and XLANG, and became the basis of a standard for Web-Service composition. We shall in what follows propose a set of REACTNet based design patterns to formalize the SO_Layer.

I.1 Overview of the structure of BPEL4WS [BPEL 03].

For this technical section we limit ourselves to the presentation of BPEL in the official document [BPEL 03] since the objective is to give as clearly as possible a sufficient technical description of BPEL enabling us to formalize its main constructs. As an executable process implementation language, the role of BPEL is to define a new Web-Service by composing a set of existing services by orchestration. The interface of the composite service is described as a collection of WSDL portTypes, just like any other Web-Service. The composition (called the process) indicates how the service interface fits into the overall execution of the composition. Figure 1 illustrates this outer view of a BPEL process. The entire type of the service (that is, the set of portTypes of the service) is implemented by one single BPEL process. The portTypes are specific "entry-points" corresponding to external users invoking the operations of the interface are indicated within the BPEL description. These entry points either consume WSDL operations' incoming messages from input-only or input-output operations. In the latter case, the process must also indicate where the output message is generated. BPEL only uses and supports input-only and input-output (request-response) operations of WSDL; output-only (notification) and output-input (solicit-response) operations are not required nor supported.

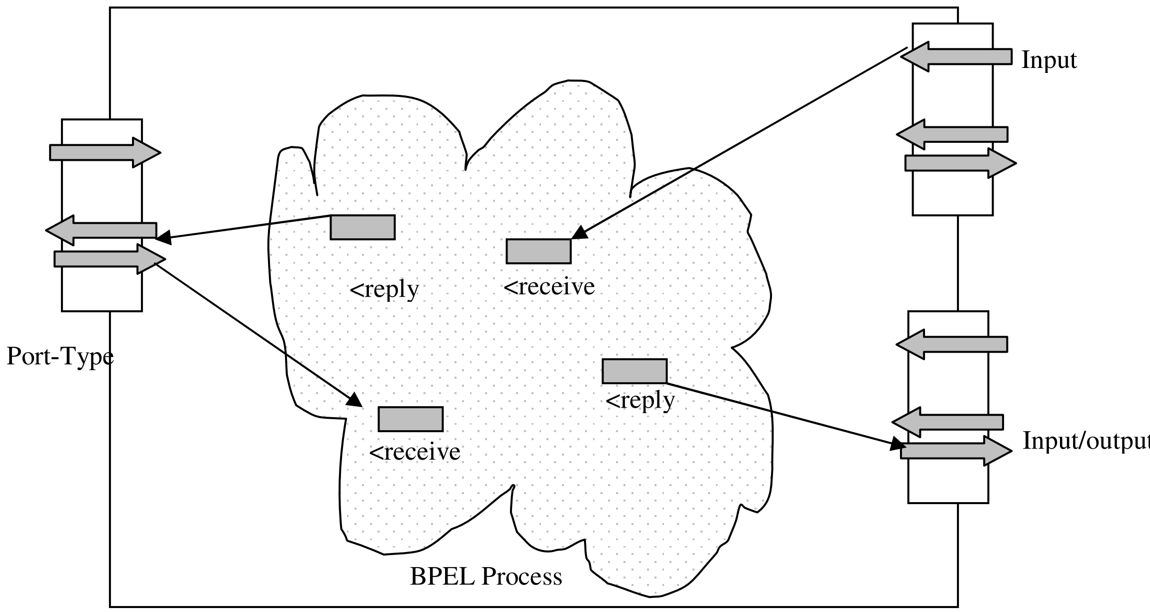


Figure 1. The outer view of a WEB Process

I.1.1 BPEL activities.

The BPEL process itself is basically a flow of synchronized steps. Each step in the process is called an activity. Basic activities are invoking an operation on some Web-Service (<invoke>), waiting for a message to operation of the service's interface to be invoked by someone externally (<receive>), generating the response of an input/output operation (<reply>), waiting for some time (<wait>), copying data from one place to another (<assign>), indicating that something went wrong (<throw>), or terminating the entire service instance (<terminate>). These primitive activities can combined into more complex algorithms using structured activities. These are the ability to define an ordered sequence of steps (<sequence>), the ability to have branching using the now common "case-statement" approach (<switch>), the ability to define a loop (<while>), the ability to execute one of several alternative paths (<pick>), and finally the ability to indicate that a collection of steps should be executed in parallel (<flow>).

Within activities executing in parallel, one can indicate execution order constraints by using the "links". BPEL allows us to recursively combine the structured activities to express arbitrarily complex algorithms that represent the implementation of the service.

I.1.2 Interactions: Partners and correlations.

The composition together of a set of services into a new service in BPEL processes mainly consists of making invocations to other services and/or receiving invocations from clients. The prior is done using the <invoke> activity and the latter using the <receive> and <reply> activities. BPEL calls these other services that interact with a process partner. Thus, a partner is either a service the process invokes "invoked partners" as an integral part of its algorithm, or those that invoke the process "client partners"

So, partners are one of the following:

- Services that the process invokes only.
- Services that invoke the process only.
- Services that the process invokes and invoke the process (where either may occur first).

The first two are "invoked partners" and "client partners", respectively. To model the third kind we need "service link types". Instead of defining the relationship between the service and the process from the point of view of one of these participants, a service link type defines a collection of roles, where each role indicates a list of portTypes. Basically, a partner is defined by giving it a name and then indicating the name of a service link type and identifying the role that the process will play from that service link type and the role that the partner will play. In the pure invoked partner and pure client partner cases, the service link type will have just one role in it and, hence, only one is indicated at partner definition time. The partner name is then used in <receive>, <reply> and <invoke> activities to indicate the desired partner. But Since we can have different instances of a business process at the same time, messages need to be delivered not only to the correct port, but also to the correct instance of the business process. The mechanism used in BPEL to route the messages to the correct process instances is to carry a set of tokens in all transactions between the partners. This set is called Correlation Set. Once a correlation set is initiated the correlation tokens get some values that must be same for all the messages in that correlation group.

I.1.3 WSDL specification of a BPEL Process.

BPEL process model is built on top of WSDL service model. A BPEL process and its partners are defined as abstract WSDL services, and they use abstract messages defined by WSDL model for interaction. Figure 2 gives an overall view to the structure of a business process in BPEL. A process is defined by specifying its partners (Web-Services that this process interacts with), a set of variables that keep the state of the process and an activity that defines the logic behind the interaction between this process and its partners. This structure for defining business process is just a template for creating business process instances. Whenever a message arrives for a Start Activity, a new instance of the business process is created. Start Activities are then defined in the business process template according to the desired business logic. Therefore, the process creation is always implicit in BPEL. Start Activity is a receive (or pick) activity that is annotated with createInstance=true; i.e. whenever the corresponding message is received by this activity a new instance of the process must be created (there are some conditions on this).

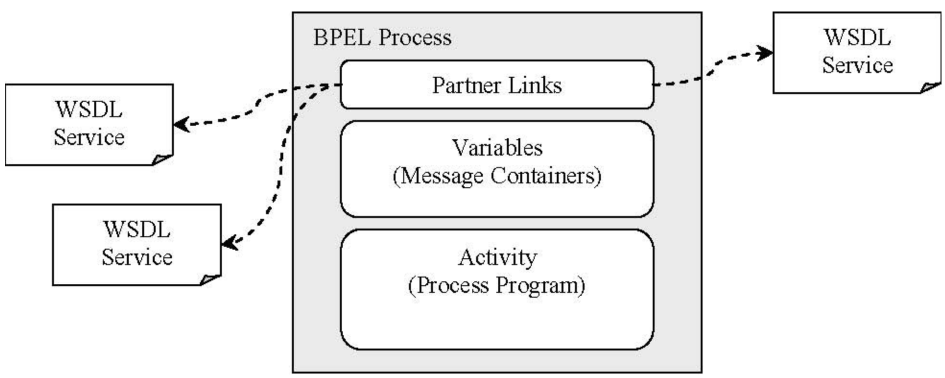


Figure 2. Outer view of a WEB Process

To illustrate the use of Web-Service in practice we present an example borrowed from [Khalaf 04]. In this example (see Figure 3), a customer sends a request for a loan; the request gets processed, and the customer finds out whether the loan was approved. Initially, the middle step will involve sending the application to a Web-Service enabled financial institution and telling the customer what it decided. From the customer's point of view, the process will consume his application and then send him an answer. The numbers on the arrows indicate the order in which the steps occur.

The Gray envelope is the message containing the loan request. The white envelope is the message containing the answer to that request. Once a client sends a message to a process manager with the appropriate triplet, a process instance is created and starts running. In the given example, the process would start up the sequence, which would in turn start the receive activity. The message has arrived so it will be put into the "request" container. The invoke will then occur. After the message that resulted from the invocation is placed in the "approvalInfo" container, the reply will take it and send it to the customer at which point that instance of the process ends. Multiple instances of the same process may be running simultaneously.

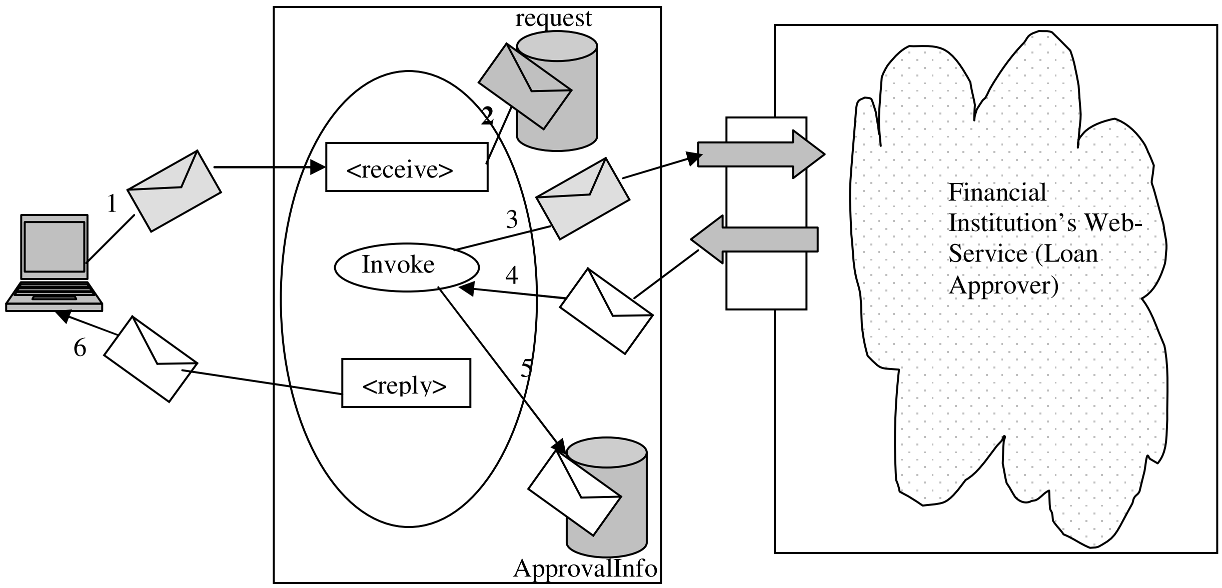


Figure 3. WEB process example

Setting up the process.

The behaviour above consists of getting a message, then invoking the financial institution's Web-Service, and finally replying to the customer. These three actions are defined in BPEL using the <receive>, <invoke>, and <reply> activities as mentioned above. However, the process needs to define the relation of such simple activities to each other in order to know how and when to run them. Such relations are defined in BPEL by using structured activities that define restrictions on how to run the activities they enclose. In this example, the three activities have to occur one after the other. This ordering may be achieved in BPEL using a <sequence> activity, that would contain first the <receive> to consume the message, followed by an <invoke> to talk to the financial institution, and ending with a <reply> to send the answer to the customer.

Creating the service descriptions: Using WSDL.

BPEL compositions rely heavily on WSDL descriptions of the involved services in order to refer to the messages being exchanged, the operations being invoked, and the portTypes these operations belong to. In the example, we will need the description of the financial institution and the process itself. Consider that the financial world uses a unified set of messages for describing loan information, and has those defined in the loan definitions in Listing 1.

```
<definitions targetNamespace="http://tempuri.org/services/loandefinitions"
          xmlns:tns="http://tempuri.org/services/loandefinitions"
          xmlns:xsd="http://www.w3.org/2001/XMLSchema"
          xmlns="http://schemas.xmlsoap.org/wsdl/">
 <message name="creditInformationMessage">
   <part name="firstName" type="xsd:string"/>
   <part name="name" type="xsd:string"/>
   <part name="amount" type="xsd:integer"/>
 </message>
 <message name="loanRequestErrorMessage">
   <part name="errorCode" type="xsd:integer"/>
  </message>
</definitions>
```

Listing 1: Loan Definitions WSDL (loandefinitions.wsdl)

Listing 2 describes a financial institution that provides a loan approval service. It contains one a single operation, "approve", which it uses to decide the status of a loan request. The operation takes information about the customer as input, and outputs an approval message containing the answer. The definition for the input message is defined in the loan defintion WSDL above. The process itself simply forwards the input and output messages to and from this service. Therefore, it will present the same description to the user by referencing the above portType. One more required thing is to define serviceLinkTypes for the services used. The serviceLinkType defines up to two roles that refer to the portTypes that are provided and required by any two services it links together. In the case of this example, this serviceLinkType will be used to link the customer to the process, as well as the process to the loan approver (Listing 3).

```
<definitions targetNamespace="http://tempuri.org/services/loanapprover"
                   xmlns:tns="http://tempuri.org/services/loanapprover"
                   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                   xmlns:loandef="http://tempuri.org/services/loandefinitions"
                   xmlns="http://schemas.xmlsoap.org/wsdl/">
  <import namespace="http://tempuri.org/services/loandefinitions"
     location="http://localhost:8080/bpws-samples/loanapproval/loandefinitions.wsdl"/>
  <message name="approvalMessage">
   <part name="accept" type="xsd:string"/>
  </message>
  <portType name="loanApprovalPT">
   <operation name="approve">
     <input message="loandef:creditInformationMessage"/>
     <output message="tns:approvalMessage"/>
     <fault name="loanProcessFault"
          message="loandef:loanRequestErrorMessage"/>
   </operation>
  </portType>
   <binding ...> ... </binding>
 <service name="LoanApprover">....</service>
 </definitions>
```

Listing 2: Loan Approver definition

```
<definitions
     targetNamespace="http://loans.org/wsdl/loan-approval"
     xmlns="http://schemas.xmlsoap.org/wsdl/"
     xmlns:slnk="http://schemas.xmlsoap.org/ws/2002/06/service-link/"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
     xmlns:lns="http://loans.org/wsdl/loan-approval"
     xmlns:apns="http://tempuri.org/services/loanapprover">
  <import namespace="http://tempuri.org/services/loanapprover"
        location="http://localhost:8080/bpws-samples/loanapproval/loanapprover.wsdl"/>
  <import namespace="http://tempuri.org/services/loandefinitions"
  location="http://localhost:8080/bpws-samples/loanapproval/loandefinitions.wsdl"/>
  <slnk:serviceLinkType name="loanApprovalLinkType">
   <slnk:role name="approver">
     <portType name="apns:loanApprovalPT"/>
   </slnk:role>
  </slnk:serviceLinkType>
  <service name="loanapprovalServiceBP"/>
 </definitions>
```

Listing 3: servicelinktype definition

Creating the process.

The definition begins with the <process> element(listing 4), and includes the namespaces that will allow it to refer to the required WSDL information. The next step is to declare the parties involved. Named partners are defined, each characterized by a WSDL serviceLinkType. For this example, the partners are the customer and the financial institution. The myRole/partnerRole attribute on a partner specifies how the partner and the process will interact given the serviceLinkType. The myRole attribute refers to the role in the serviceLinkType that the process will play, whereas the partnerRole specifies the role that the partner will play. This is illustrated in the partner definitions below. The loan approval process offers the functionality of the loanApprovalPT to the customer, and the financial institution in turn offers that functionality to the process. After defining the partners, we add the activities that form the composition. In order to ask for a loan, the customer sends the process a message, the process asks the financial institution whether it will accept the loan application, and replies to the customer with another message either accepting or refusing the application. We need to put the incoming message where a BPEL activity can access it. In BPEL, data is written to and accessed from data containers which can hold instances of specific WSDL message types. From the definition of the customer partner and the loanApprovalPT, it is clear that the customer will send a message of type creditInformationMessage and get a reply of type approvalMessage. Therefore, some containers are added and are illustrated in Figure 3 as gray cylinders.

Interacting with the process: Receive, invoke, reply.

A process may contain only one activity, which in this case will be the <sequence>. Now you can add to the sequence a receive activity that can take the customer's message and put it in the appropriate container. The definition of a receive activity must include the partner that will send it its message, and the port type and operation of the process that the partner is targeting this message to. Based on this information, once the process gets a message, it searches for an active receive activity that has a matching partner-portType-operation triplet and hands it the message. In order to avoid confusion, the specification states that there may not be multiple receive activities with the same partner-portType-operation triplet that are active at the same time. The activity will then place the message in the specified container and end. You start the sequence activity, and add the "receive" to it.

```xml
<process name="loanApprovalProcess"
        targetNamespace="http://acme.com/simpleloanprocessing"
        xmlns="http://schemas.xmlsoap.org/ws/2002/07/business-process/"
        xmlns:lns="http://loans.org/wsdl/loan-approval"
        xmlns:loandef="http://tempuri.org/services/loandefinitions"
        xmlns:apns="http://tempuri.org/services/loanapprover">
<partners>
   <partner name="customer"
           serviceLinkType="lns:loanApproveLinkType"
           myRole="approver"/>
   <partner name="approver"
           serviceLinkType="lns:loanApprovalLinkType"
           partnerRole="approver"/>
 </partners>
<containers>
   <container name="request" messageType="loandef:CreditInformationMessage"/>
   <container name="approvalInfo"  messageType="apns:approvalMessage"/>
 </containers>
 <sequence>
   <receive name="receive1" partner="customer"
           portType="apns:loanApprovalPT"
           operation="approve" container="request"
           createInstance="yes">
   </receive>
<invoke name="invokeapprover"
          partner="approver"
          portType="apns:loanApprovalPT"
          operation="approve"
          inputContainer="request"
          outputContainer="approvalInfo">
   </invoke>
   <reply name="reply" partner="customer" portType="apns:loanApprovalPT"
          operation="approve" container="approvalInfo">
   </reply>
 </sequence>
</process>
```

Listing 4 : the WSDL definition of the Web process

The next step is to ask the Web-Services-enabled financial institution whether or not it will accept the loan. This is done with a regular Web-Services invocation, defined in the process by an Invoke activity. When this activity runs it will make the specified invocation to the Web-Service using the message in its input container, put the answer it gets into its output container, and end. Note that the call will be made on the "approver" partner to perform the approve operation.

In order for the process to respond to the customer's request, it uses a Reply activity. Once a reply activity is reached, the partner-portType-operation triplet it has is used to figure out whom to send the reply to. Therefore, in order to reply to the message that arrived through the Receive activity, you would need a Reply activity with the same triplet. In this case, you want to tell the customer what the financial institution decided, so the message to be sent will be found in the output container of the invoke: approvalInfo. After the reply, the process ends.

I.2 Formal specification of BPEL : state of the art.

A BPEL process may cause non desirable effects due to the eventual interactions between different partners and the complex interrelated patterns composing it, thus a BPEL process must be analysed before being deployed to ensure that orchestration and different synchronisations are done properly.

For this purpose formal tools are more and more used to give BPEL processes more accuracy and reliability. Formalization is the process of changing informal requirements into specifications along with mathematical precision. Thus, it is expected to reveal the loose ends, ambiguities and inconsistencies. Of course, formalization is a necessary prerequisite for executability. Executable specifications provide a behavioural model of the system and help to solve the reliability and correctness problem by using experimental validation. Executable specifications let us experiment with the formal definition, check the conformance of the formal definition to the informal definition, and check the correctness of the formal definition to some extends.

Several works treated the formalization of BPEL processes each one with its tools and techniques: [Lucchi and Mazzara 07] have proposed a mapping from a BPEL process to a π-based calculus which they call *Webπ* and which focuses on transactional aspects of the BPEL language.

A two-way mapping from Lotos to BPEL, is presented in [Chirichielo & Salan 05] [Ferrara 04]. COWS [Lapadula *et al*. 07] is a new foundational language for service-oriented computing whose design has been influenced by WS-BPEL. COWS allows for the encoding of more specific languages such as the WS-calculus [Lapadula *et al*. 06]. Petri nets have been extensively used to model and verify business processes. In [Schmidt & stahl 04], a mapping from BPEL to Petri nets is presented by giving several examples. Each BPEL construct is mapped into a Petri net pattern. The complete transformation from BPEL to Petri nets is given in [Stahl 05]. The authors in [Hinz *et al*. 05] describe the tool BPEL2PN that implements the transformation when abstracting from data. In [Reisig 05], BPEL is modelled by means of a special type of Petri nets called business process nets. In [Verbeek & Aalst 05], authors focus on the structured activities of BPEL. They present a mapping of these structured activities to a kind of Petri nets called WF-nets (workflow nets). For this class of Petri nets, a verification tool named Woflan [Verbeek *et al*. 01] is used. Later in [Aalst & lassen 06] the authors have presented a first attempt at mapping WF-nets onto BPEL processes. Their objective is to use a graphical formal language to create BPEL specifications, in order to facilitate the design and verification of composite Web-Services In [Yang *et al*. 05] coloured Petri nets are used since they provide a more compact way to model business processes than ordinary Petri nets. in [FU *et al*. 05] the authors present a framework to verify properties of a Web-Service composition consisting of multiple BPEL processes that communicate asynchronously. Each BPEL process is translated to a guarded automaton. Subsequently, these guarded automata are mapped to Promela. [Magee & Kramer 06] present a process algebra named FSP (Finite State Process). Each FSP represents a finite labelled transition system that can be used to verify BPEL processes. A tool named LTSA-WS is developed to allow the translation of the activities of BPEL into FSPs. In [Wombacher *et al*. 04] a translation of most BPEL activities into annotated deterministic finite automata is proposed.

I.3 Specifying BPEL patterns with REACTnets.

When we have to build a Web-process, generally this can done not only by constructing a new Web-Service but often by integrating old existing ones, new and legacy Web-Services are then composed as a BPEL process.

For the sake of efficiency and simplicity and to reduce costs it is better to specify the BPEL process by a suitable formal tool and to verify before finally to translate the model to an actual executable BPEL Process.

We are particularly interested by Petri-nets based models for BPEL, in this context several works tried to propose solution to the formalisation of BPEL processes with Petri-nets as presented above. The solution proposed in [Verbeek & Aalst 05] [Aalst & Lassen 06] is to use a number of specific rules to translate respectively BPEL constructs to WF-Nets and workflow specifications in terms of WF-nets to BPEL processes; but this approach is not complete since the more complex and classical specific constructs such as compensation, correlation, related-links and fault handling are not supported by this approach. Instead In [Stahl 05] a formal semantic based on simple Petri nets (PT/Nets) is given but the model obtained is very complex and unreadable in our sense. Finally [Youang *et al.* 07] succeed to formalize all the BPEL constructs but this approach deals with behavioural aspects only and not with data aspects of the system in any way.

With RECATnets, we can handle not only behavioural aspects but also data aspects thanks to the algebraic structure of the Net, data aspects are missing in works cited above although very important, the data structures are present not only in the Net but also at the level of the rewriting system derived such as we can verify the behaviour and data together.

Manipulating data in REACTNets allows us to consider in an elegant way specific constructs such as the correlation sets which can be fully formalized. Furthermore the PortTypes and interaction between partners are naturally handled through REACTNets ports and external events, finally The rewriting system derived from WS-REACTNets can be encoded in MAUDE and thus verified.

I.3.1 The BPEL process life-cycle.

BPEL provides an important construct known as control links which, together with the associated notions of join condition and transition condition, support the definition of precedence, synchronization and conditional dependencies. A control link expresses that this activity can not start before a previous activity has either completed or has been skipped.

Moreover, the activity can only be executed if its associated join condition evaluates to true (JC-value place). This join condition is expressed in terms of the tokens carried by control links leading to the activity. These tokens may take either a positive (JC) or a negative (~ JC) value. Transition conditions are boolean expressions over the process variables (V place). IF the join condition evaluates to true, the activity can start as normal. Otherwise, a fault called join failure occurs. A join failure can be handled in two different ways, as determined by a so-called suppress Join Failure attribute associated with X (Suppress-JC place). If this attribute is set to yes, the join failure will be suppressed. In this case, the activity will not be performed. An activity for which a join failure is suppressed, will end up in the finished state as if it is completed as normal, and thus the processing of any following activity will not be affected. Otherwise, if suppress Join Failure is set to no, a join failure needs to be thrown, which triggers a standard fault handling procedure

I.3.2 The SEQUENCE pattern.

In a sequence activities are performed sequentially, figure 4 is the REACTnet based pattern of such a construct, we limit the pattern to two activities which can be generalized.

I.3.3 The Switch pattern.

According to [BPEL 03] : "The case branches of the switch are considered in the order in which they appear; the first branch whose condition holds true is taken and provides the activity performed for the switch. If no branch with a condition is taken, then the otherwise branch is taken." [Ouyang *et al.*07] proposes a non deterministic solution for the switch, which is in our sense not in conformance with BPEL constructs. In BPEL processes the choice is deterministic.

With REACTnets we can model this case very easily by forcing the conditions to be evaluated in the correct order, the first activity whose condition is true is performed when the others are skipped (figure 5).

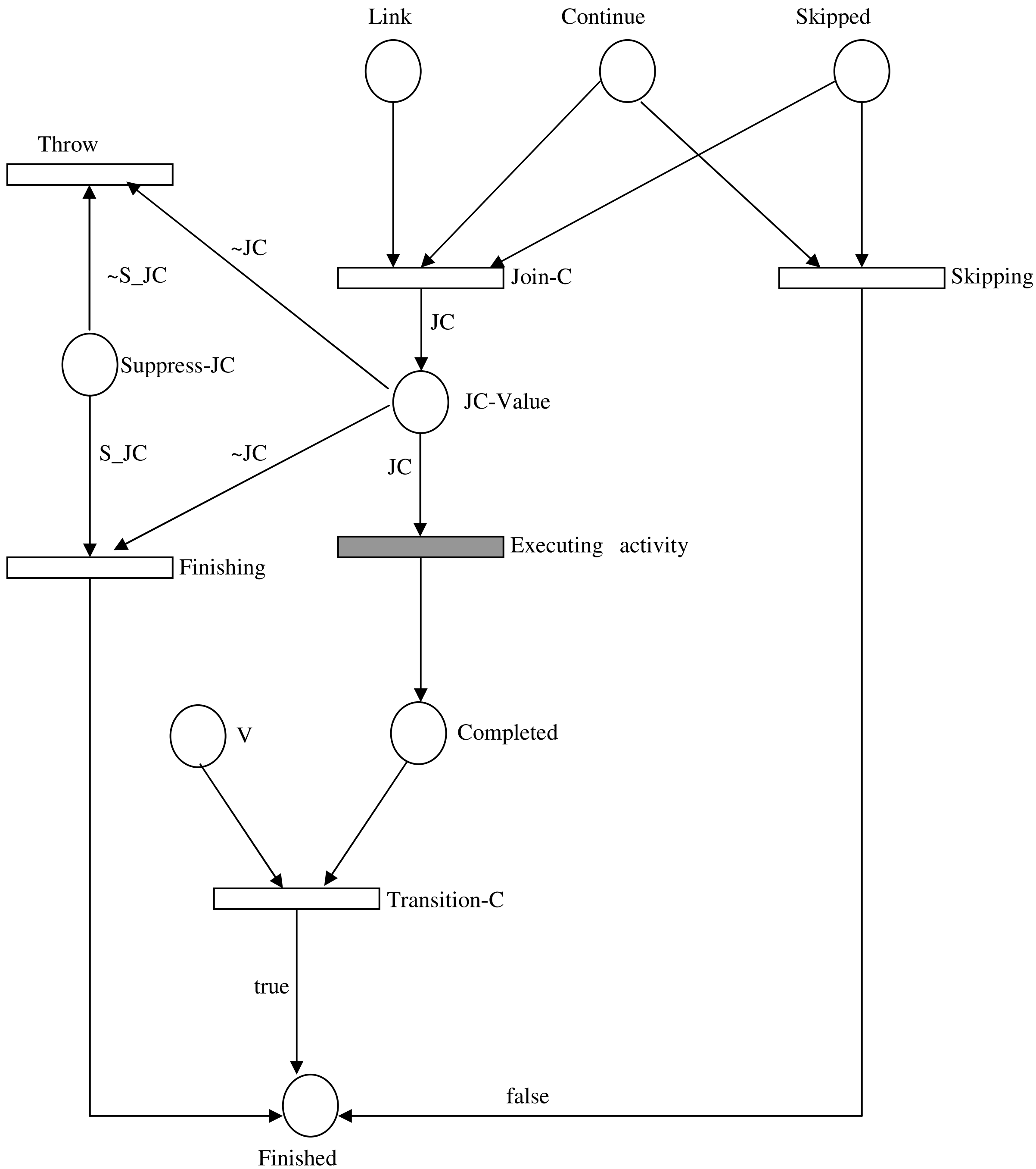
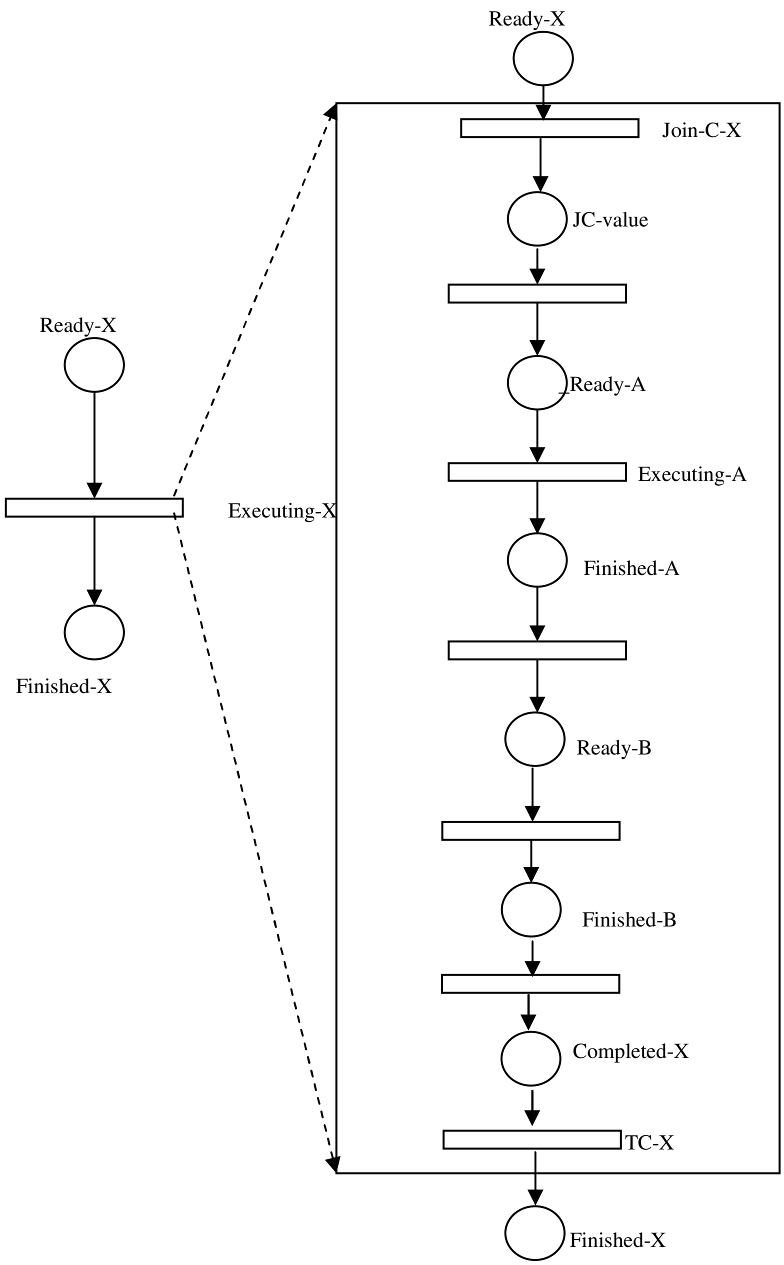


Figure 4. The BPEL process life-cycle pattern




```
<SEQUENCE Name='X'>
        Activity A
        Activity  B
</SEQUENCE>
```

Figure 5. The Sequence pattern

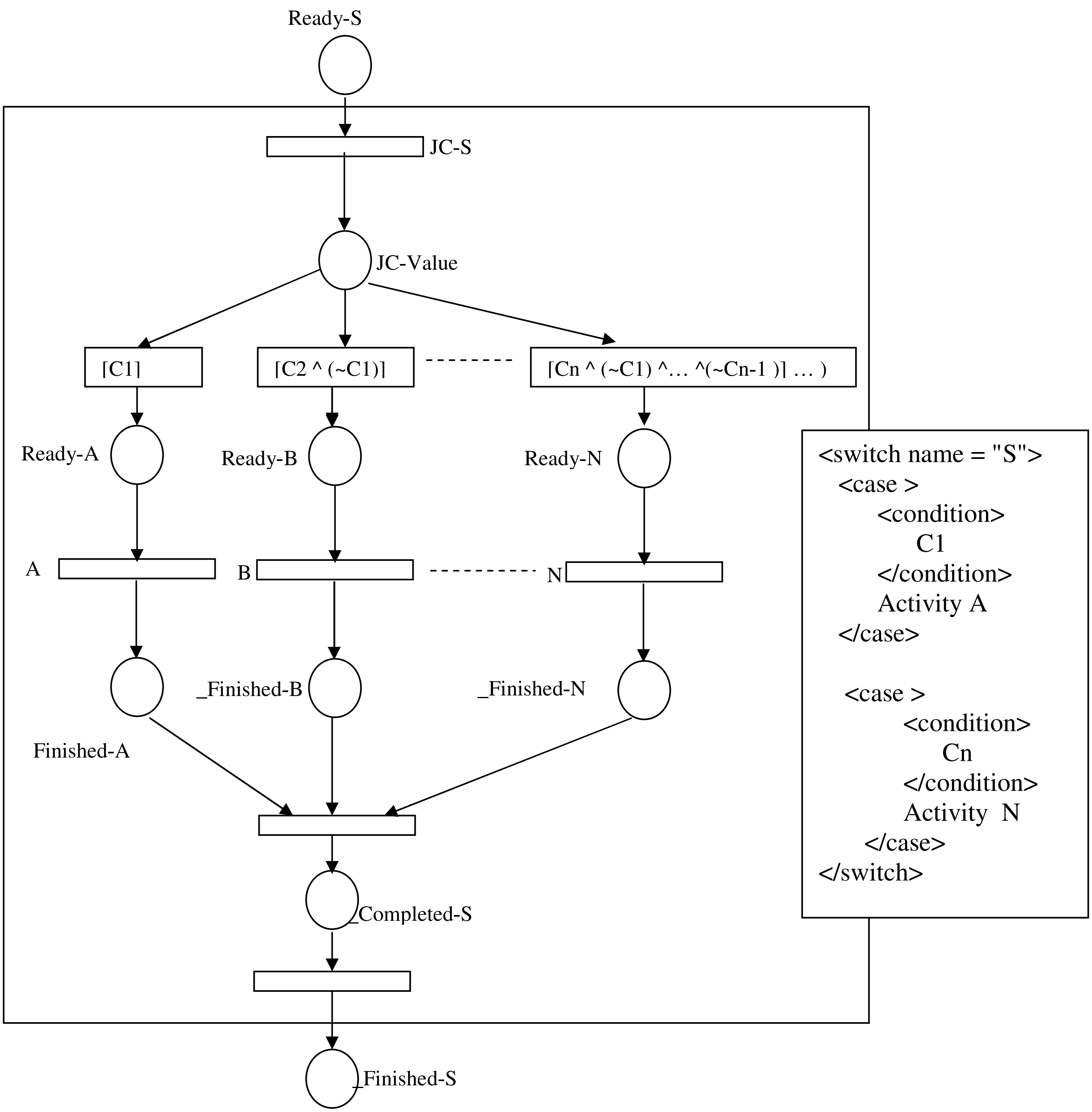
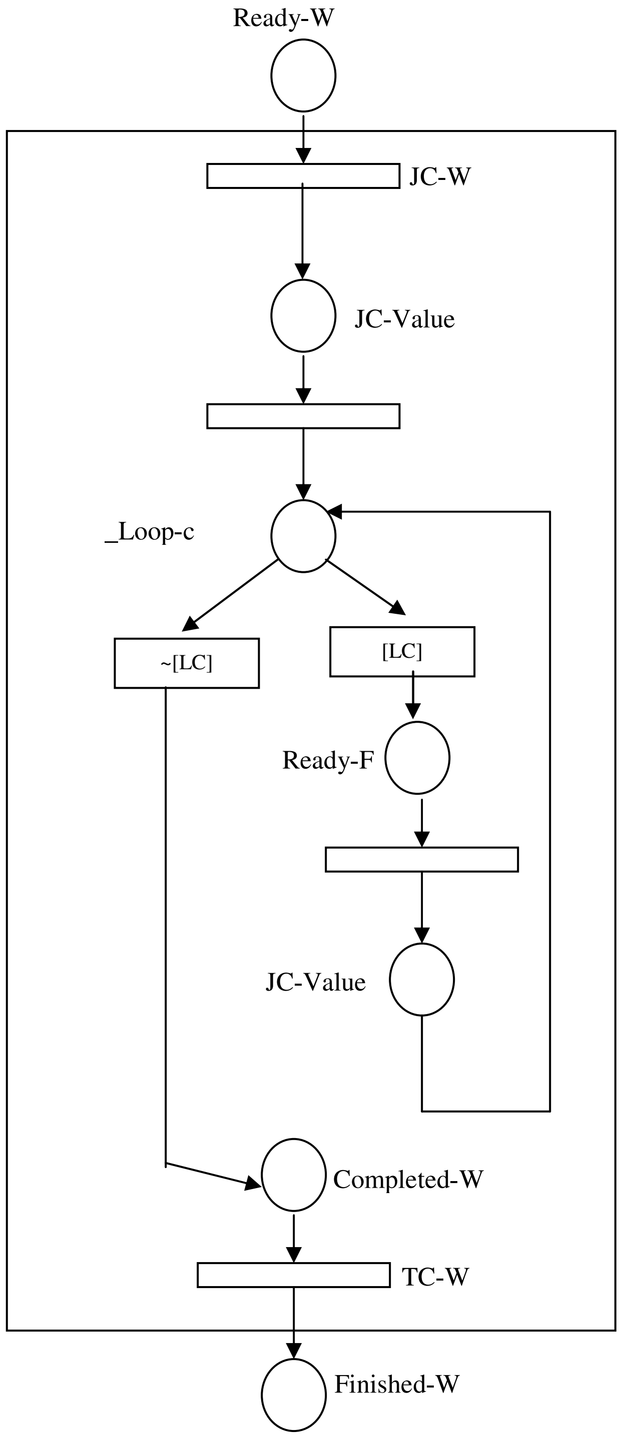


Figure 6. The Switch pattern

I.3.4 The WHILE pattern.

By definition in [BPEL 03], Section 12.3: "The while activity supports repeated execution of a specified iterative activity. The iterative activity is executed until the given Boolean while condition no longer holds true."

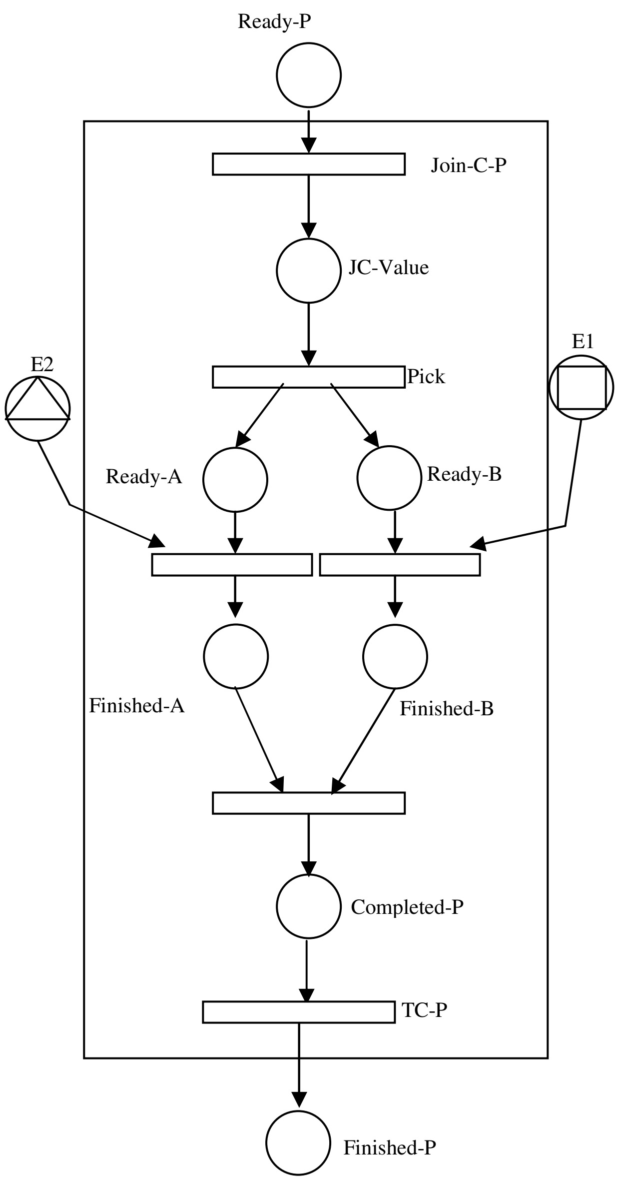


<while name="W">
<condition>
Loop-c
</condition>
Activity A
</while>

Figure 7. The While pattern

I.3.5 The PICK pattern.

"The pick activity awaits the occurrence of one of a set of events and then executes the activity associated with the event that occurred." [BPEL 03]. "If more than one of the events occurs, then the selection of the activity to execute depends on which event occurred first. Each pick activity MUST include at least one onMessage event." Events must be system timeout or external triggers, in our pattern, triggers are explicitly exhibited through input-ports, to each transition belonging to an activity requiring a trigger, an extra-input place is added, a token in this place model the event occurring. In BPEL two types of triggers are considered : arrival of a message or a timeout, we associate to these two events two special places as depicted in figure 8.




```
<Pick  Name="P">
   <onMessage E1
        Activity A
   </onMessage>
   <onTimer E2
        Activity B
   </onMessage>
</Pick>
```

Figure 8. The pick handler pattern

I.3.6 The FLOW Pattern.

The most fundamental semantic effect of grouping a set of activities in a flow is to enable them for concurrent execution. A flow completes when all of the activities in the flow have completed.

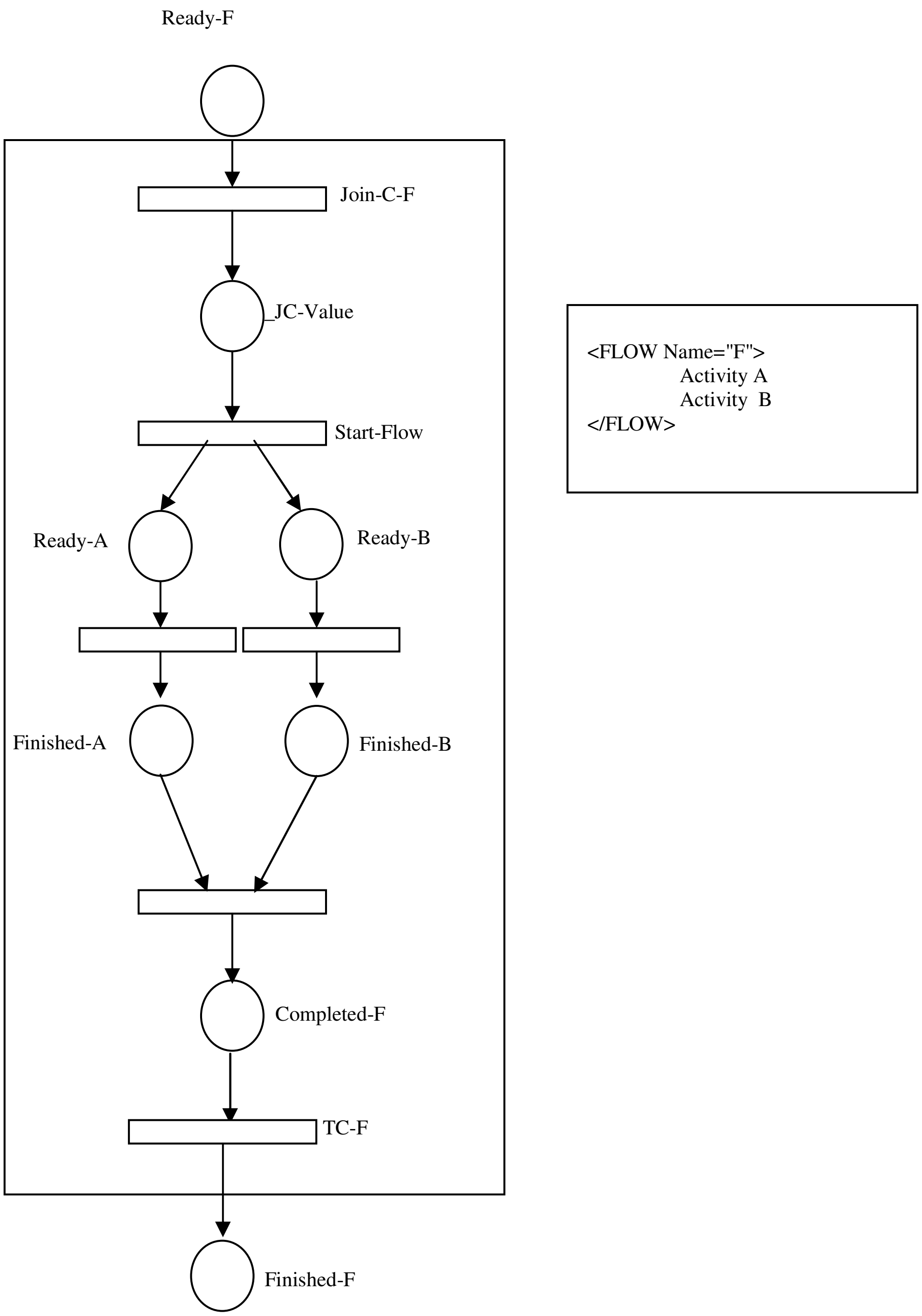


Figure 9. The Flow pattern

I.3.7 The SCOPE patterns.

Scope defines a logical unit of work, for which a Compensation Handler or a set of Fault Handlers can be defined. Compensation Handler defines the compensating behaviour of the logical unit in case of an error. Fault Handler defines the reaction of the logical unit to an error. A scope in practice is a set of activities grouped together, the scope contains one main activity and allows the specification of fault and compensation handlers.

Fault handler

A fault handler contains an activity that will run in case on error occurs, for example it may contain a reply activity that notifies a partner that an error has occurred. When a scope receives a fault, and in the default case, it stops its activities and rethrows the fault to its parent activity and so on until the top-level process is reached and it sends out all its outgoing links with negative values, but if there is a handler, the scope takes appropriate actions and ends normally with the values of its outgoing links being evaluated as usual.

```
<scope name = "X">
<faultHandlers>
 <catch faultname  =   "fault">
        Activity FH
     </catch>
</faultHandlers>
 Activity A
</scope>
```

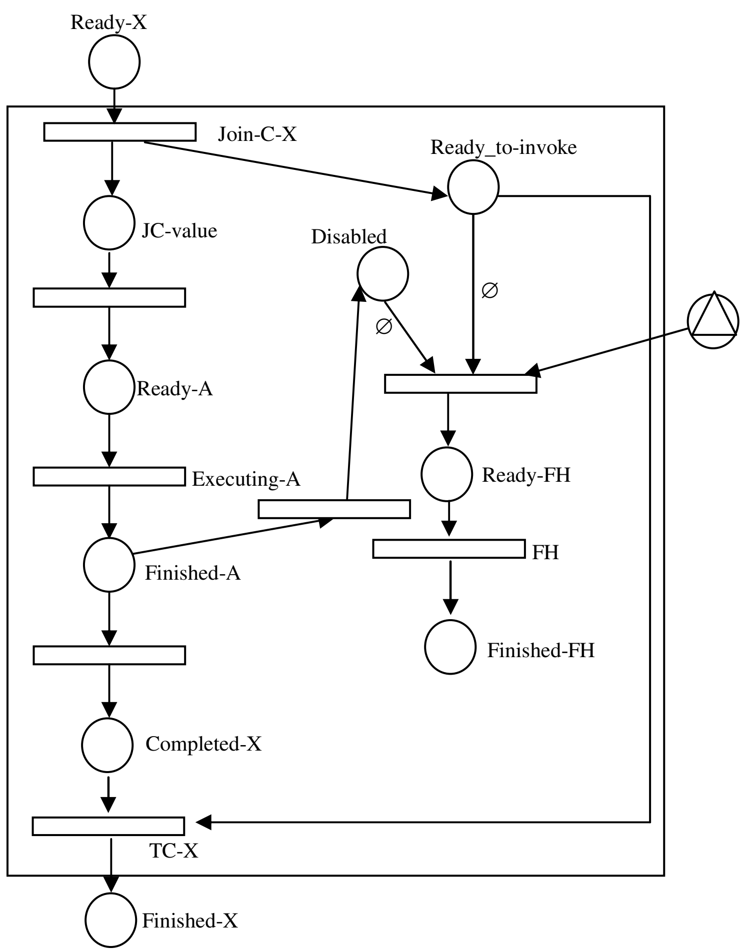


Figure 10. The event handler pattern

Event handler

Event handlers are responsible of handling all normal events (messages or alarme vents) that occur concurrently while the scope is running

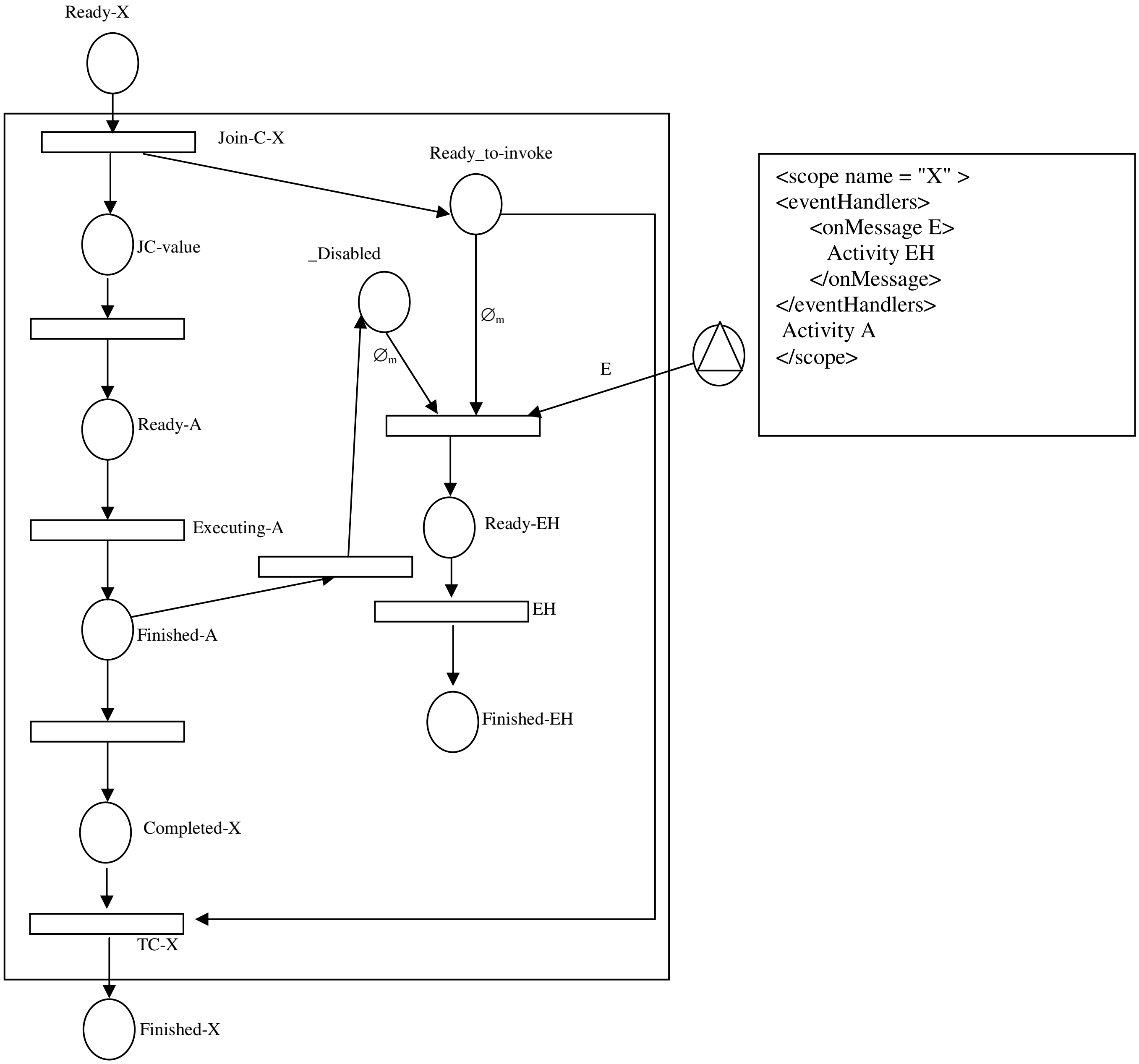


Figure 11. The event handler pattern

I.3.8 The Correlation Set pattern.

Correlation sets are not considered in [Ouyang *et al*. 05] since they need precise data modelling, next figure shows how they can be specified with REACTnets. We consider a message input(M,initiate), "initiate" is a boolean variable taking yes or not, M is a tuple of variables, we suppose predefined a boolean function "match" rending true when M is conform to the correlation-set CS, and a function "new-CS" when a correlation set have to be initialized.

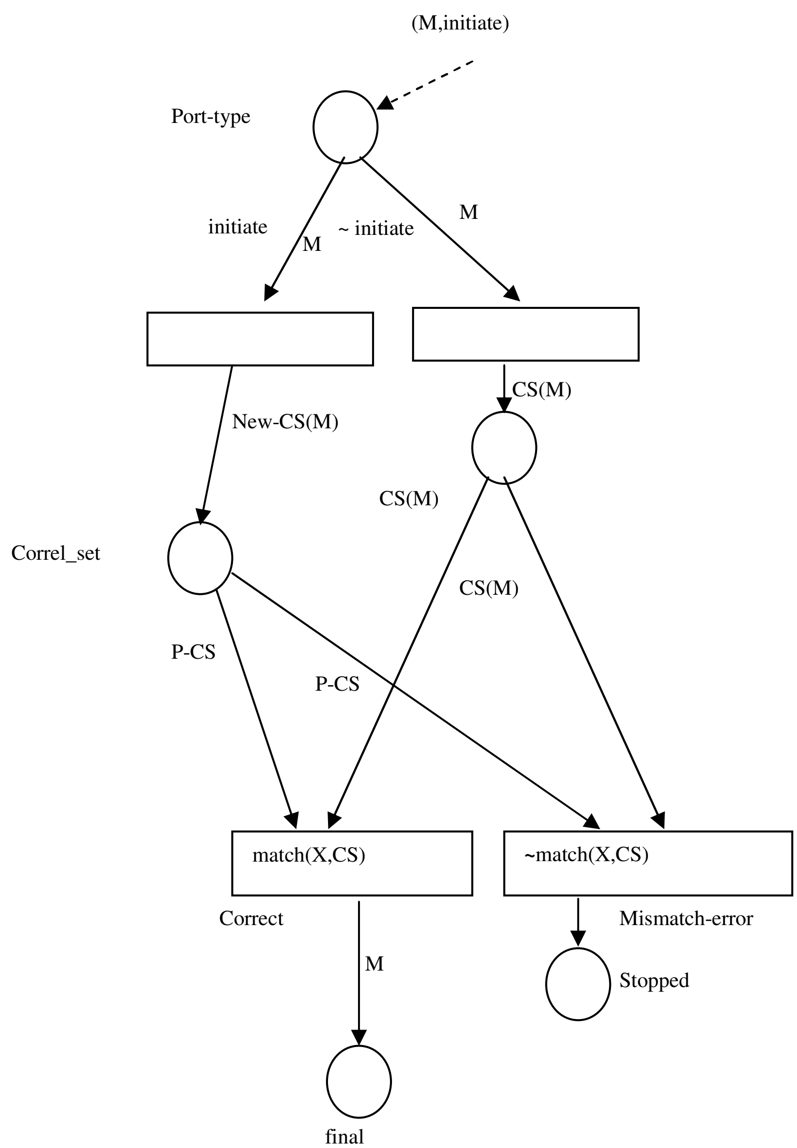


Figure 12. The correlation-set pattern

II. Formalization of the MA_Layer.

The workflow enactment in MOBIFLEX is performed by an M-Worker which is a mobile worker. An approach that can simplify and improve the development of workflows based on mobile agents are the use of design patterns similar to classical workflow patterns [Aalst]. The use of design patterns is an approach to improve the development process of applications and the quality of the final products. Some mobile agent design patterns have already been proposed [Aridor & Lange 98], [Kendal *et al.* 98], [Tahara *et al.* 99], [Tahara & Ohsuga 01],[Ojha *et al.* 07], [Al-Shrouf 08] [Ferrera *et al.* 03].

II.1 An Algebra of Itineraries for the MA_Layer.

This proposition is based on the algebra of itineraries proposed in [Loke *et al.* 99] which we adapt and enhance with an operational semantics in term of MAUDE rules.

Let $\{A^{\{i\}^*}\}$ where {i}* denotes strings of integers identifying an agent or a clone of an agent be a finite set of mobile agents. we use the following convention for generating names for agents and their clones. When an agent $A^{\{i\}}$ is cloned k times, its clones are named $A^{\{i\}1}$, … , $A^{\{i\}k}$.

Let A ={a1,a2 , … ,an} be a finite set of activities and S ={s1,s2 , … ,sn} a finite set of sites to be visited. Itineraries (denoted by *I*) are formed as follows representing the null activity, atomic activity, parallel, sequential, nondeterministic, conditional nondeterministic behaviour, and have the following syntax:

- The null activity : $\varnothing$
- The atomic activity, A_s^a which means the agent *A* moves to site *s* and executes the action *a*
- The parallel behaviour : $I \,\Pi\, I'$ which means the itinerary *I in parallel with the itinerary I'*
- Sequential behaviour : $I \oplus I'$ which means itineray *I* then the itinerary *I'*
- Nondeterministic behaviour : $I \mid I'$ which means *I* or *I'*
- Conditional nondeterministic behaviour : $\dfrac{I \mid I'}{c}$ which means if condition c is true then *I* else *I'* , c is a Boolean expression.

Example the next graph, shows in an abstract manner a complex itinerary built by composing atomic itineraries by the constructs listed above.

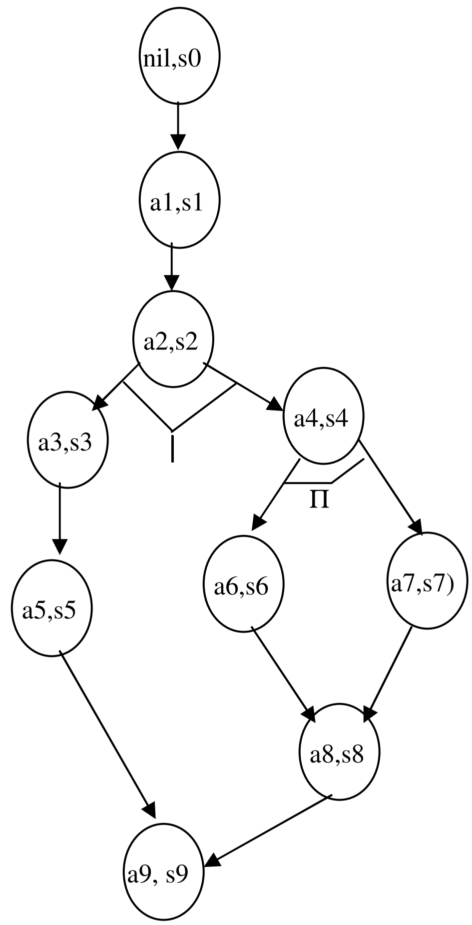


Figure 13. An example of an MWorker itinerary

This itinerary can be described by the next algebraic expression

$$A_{s0}^{nil} \oplus A_{s1}^{a1} \oplus A_{s2}^{a2} \oplus \left(\left(A_{s3}^{a3} \oplus A_{s5}^{a5}\right) \mid \left(A_{s4}^{a4} \oplus \left(A_{s6}^{a6} \prod A_{s7}^{a7}\right) \oplus A_{s8}^{a8}\right)\right)$$

II.2 Operational semantics.

In [Loke *et al*. 99] it is shown that itineraries are associative with an element identity $\varnothing$, so we can conclude that itineraries are a special case of strings rewriting theory modulo a set of axioms *E=AI.* In this section we shall propose a rewriting logic based operational semantics in term of MAUDE rules. The configuration is in this case a set of agents and messages, messages will be used to mark important steps of the workflow such as the achievement of activities.

Additionally we assume predefined two sets (Site) for places to be visited by the M-worker and (Activity) for the activities to be performed by the M-Worker and we define a class M-Worker with two attributes site and activity. a mobile agent will be specified by the term < *A:MWorker s:s0 , a:a0*> which at the right-side of a rule means : The M-Worker A moves to the site s0 to perform the activity a0, and at the left-side of a rule it means that the activity a0 is already performed by the agent A in the side s0. The activity a0 is essentially built upon the invocation of a Web-process. Since we are interested by the behavioural aspect of the workflow enactment and for the sake of simplicity, we don't take into account the state of the agent which can be modelled in MAUDE by a set of additional attributes.

A workflow instance enactment is performed by one agent and starts at a source site and ends at a target site, at intermediary steps, the agent may clone itself to perform some parallel tasks so that the configuration may involve more than one agent, after achieving parallel tasks, clones have to declone themselves so that the workflow terminate with only one agent which is exactly the original one who launches the workflow at the source site. In what follows we show how we can specify the different itinerary constructs as MAUDE rules.

II.2.1 Agent basic Movement.

Different cases have to be envisaged for agent movements : the agent is in site s1 with no activities to perform here and has to move to site s2 to perform activity a1

<*A: MWorker* |*s:s1, a: nil*>=> <*A: MWorker*|*s: s2, a:a1*>

After achieving activity a1 in site s1, the agent moves to site s2 to perform the activity a2

<*A: MWorker*| *s1:site, a1: activity*>=> <*A: MWorker*| *s2:site, a2: activity*>

The agent have to perform sequentially two activities a1 then a2 in the same site s1

<*A: MWorker*| *s1:site, a1: activity*>=> <*A: MWorker*| *s1:site, a2: activity*>

II.2.2 Sequential behaviour.

The basic rule presented above can be used to specify a sequence of more than one activity, so that for n steps (n sequential activities), we need n rules, for example an agent *A* is in a site s0 and have to perform sequentially 3 activities (a1,a2, a3) in three sites respectively s1,s2 and s3 ; this can be expressed as follows: $A_{s0}^{nil} \oplus A_{s1}^{a1} \oplus A_{s2}^{a2} \oplus A_{s3}^{a3}$

We can model this sequence in MAUDE like this

<A: MWorker| s:s0, a: nil>=> <A: MWorker|s: s1, a: a1>

<A: MWorker| s:s1, a: a1>=> <A: MWorker|s: s2, a: a2>

<A: MWorker| s:s2, a: a2>=> <A: MWorker|s: s3, a: a3>

II.2.3 Parallel composition.

Two itineraries composed by "Π" are executed in parallel. $A_{s1}^{a1}\Pi A_{s2}^{a2}$ means that the agent *A* has to perform activities a1 in site s1 and a2 in site s2 concurrently. Parallelism imply cloning of agents. since agent A has to perform actions at both s1 and s2 in parallel. we assume that the agent *A* is in site s0 and have to perform the itinerary $A_{s0}^{nil} \oplus (A_{s1}^{a1}\Pi A_{s2}^{a2})$

The MAUDE rules for this itinerary are as follows :

<A: MWorker| s:s0, a: nil> => <A1: MWorker| s:s1, a:a1><A2: MWorker| s:s2,a: a2>

<A1: MWorker| s:s1, a:a1><A2: MWorker| s:s2,a: a2> =><A: MWorker| s:s0, a:nil>

The second rule is the decloning rule , it may be used in different manners and to represent different situations, the clones can before decloning themselves perform complex itineraries, when these latter achieved they declone themselves in a site which can one of the two sites where the clones reside or a third site where the rebuilt agent can continue its itinerary.

II.2.4 Nondeterministic behaviour.

We extend the nondeterminism to more than two choices, since MAUDE has an interleaving semantics, we can specify nondeterministic behaviour in a very natural manner, as example: we assume the next itinerary : $A_{s0}^{nil} \oplus (A_{s1}^{a1}|A_{s2}^{a2}|A_{s3}^{a3})$. The MAUDE specification can be like this

<A: MWorker| s:s0, a: nil> => <A1: MWorker| s:s1, a:a1>

<A: MWorker| s:s0, a: nil> => <A1: MWorker| s:s2, a:a2>

<A: MWorker| s:s0, a: nil> => <A1: MWorker| s:s3, a:a3>

Executing any rule of these three rules in a nondeterministic manner will disable automatically the two others.

II.2.5 Conditional nondeterministic behaviour.

A conditional nondeterministic choice between two activities a1 and a2 can be for example expressed like this :

$$A_{s0}^{nil} \oplus \left(\frac{A_{s1}^{a1} \mid A_{s2}^{a2}}{c} \right)$$

Since MAUDE rules are conditional this expression can be specified easily as follows:

<A: MWorker| s:s0, a: nil> => <A1: MWorker| s:s1, a:a1>   if c

<A: MWorker| s:s0, a: nil> => <A1: MWorker| s:s2, a:a2> if not c

II.3 Verifying in MAUDE.

Based on the operational semantics proposed above, we shall show how the workflow of the example presented in figure can be verified in MAUDE, next listing is the source of the specification in MAUDE as an object oriented module of this workflow . MAUDE offers a number of commands for verifying properties of the modelled system, the most powerful are rewrite command and search command, rewrite command causes a specified term to be rewritten using the rules and equations, in the given module. Figure 14 shows how we use the rewrite command to verify that an enactment of the workflow of the previous example specified in MAUDE, is correct i.e. the correct activities are performed in the right order and at the final stage we have the original M-Worker that launched the enactment at the home site (s0). In figure 16 we tried with a wrong specification (see figure 15 for the itinerary graph) where a decloning of two clones is omitted, the workflow is launched by an M-Worker 'ma' but at the last stage not only some activities are not performed but we end the workflow with its two clones 'ma1' and 'ma2' rather than the original one.

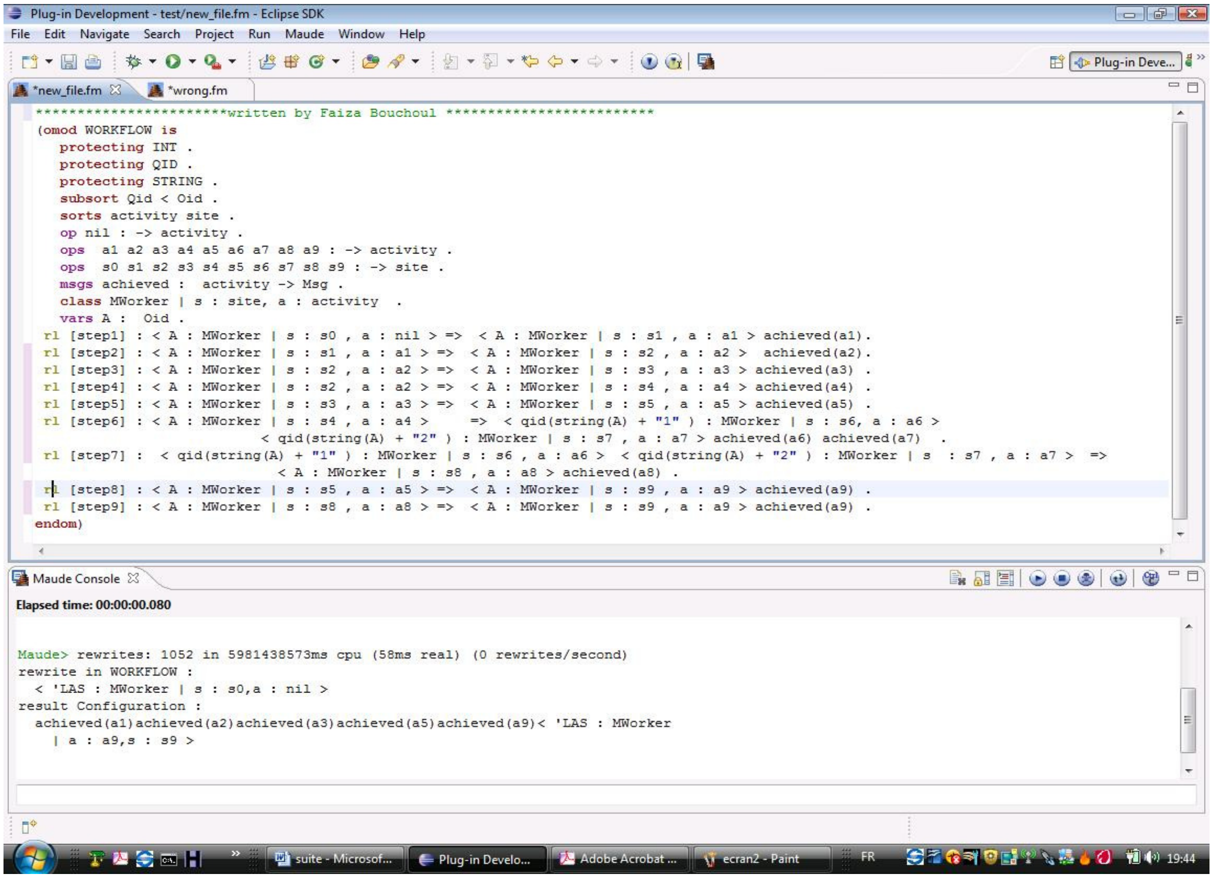


Figure 14 . A MAUDE specification of a mobile workflow

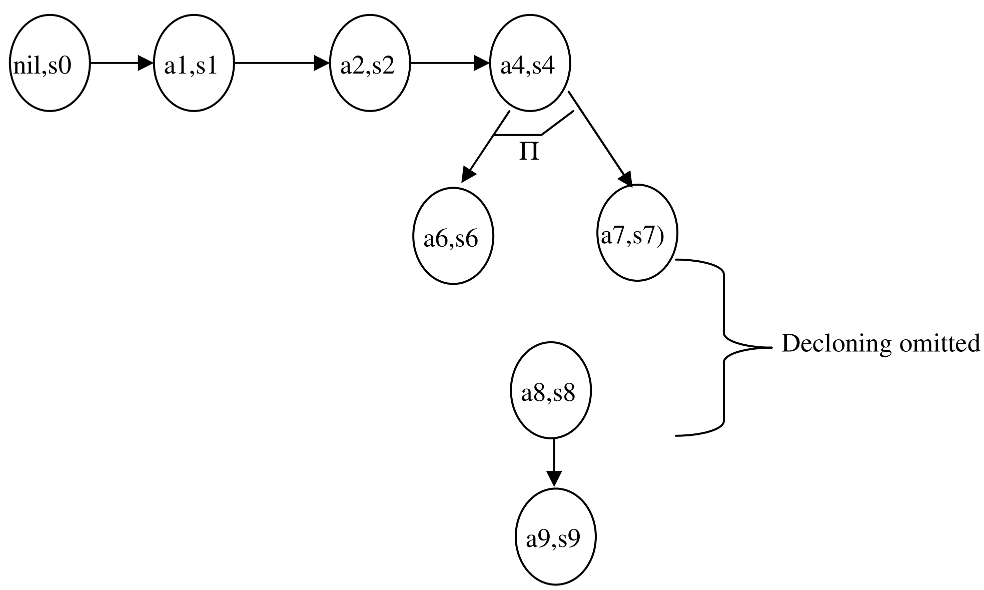


Figure 15 . A wrong itinerary

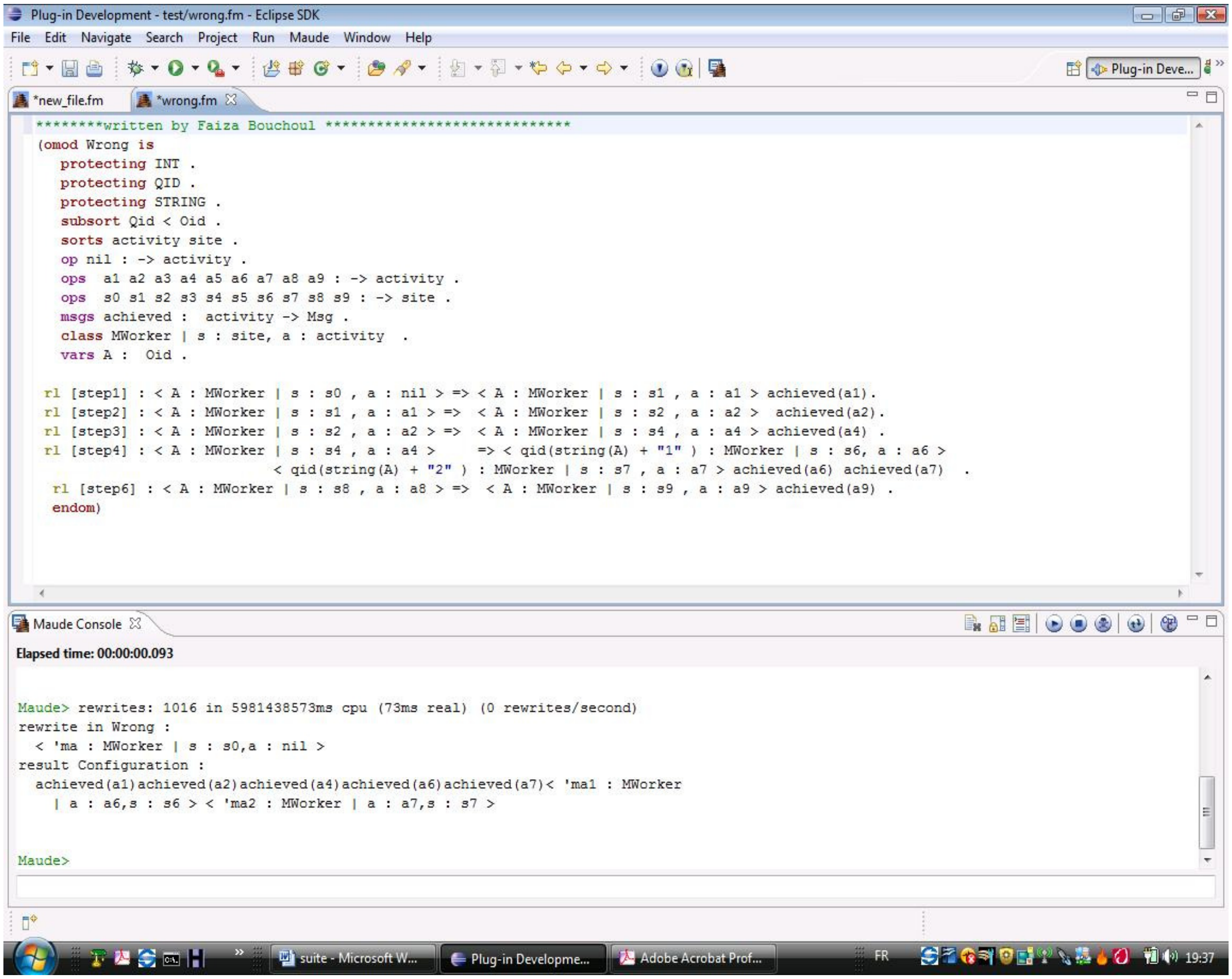


Figure 16 . Verification of a wrong M-Worker itinerary in MAUDE

REFERENCES

[Aalst & Lassen 06] W. M. P. van der Aalst, and K. B. Lassen, "Translating unstructured workflow processes to readable BPEL: Theory and implementation"., the International Journal of Information and Software Technology (INFSOF) (2006).

[Al-Shrouf 08] F. M. Al-Shrouf, "Facilitator Agent Design Pattern of Procurement Business Systems," compsac, pp505-510, 2008 32nd Annual IEEE International Computer Software and Applications Conference, 2008

[Ojha *et al*. 07] A. C. Ojha ; S. K. Pradhan and M. R. Patra " Pattern-Based Design for Intelligent Mobile Agents" 4th International Conference on Innovations in Information Technology, 2007. IIT apos;07. , 18-20 Nov. 2007 pp :501 - 505

[Arias *et al*. 04] J. Arias-Fisteus, L.S. Fernandez, and C.D. Kloos. "Formal verification of BPEL4WS business collaborations". In K. Bauknecht, M. Bichler, and B. Pr oll, editors, Proc. of the 5th international Conference on Electronic Commerce and Web Technologies, LNCS 3182, pp 76-85, Zaragoza, Spain, August/September 2004. Springer-Verlag.

[Aridor & Lange 98] Y. Aridor and D.B. Lange. "Agent design patterns: Elements of agent application design". In Proc. of the Second International Conference on Autonomous Agents. ACM Press, 1998.

[BPEL 03] Business Process Execution Language for Web-Services Version 1.1, 2002, 2003 BEA Systems, International Business Machines Corporation, Microsoft Corporation, SAP AG, Siebel Systems.

[Chirichiello & Salan 05] A. Chirichiello and G. Salan, "Encoding abstract descriptions into executable Web-Services: Towards a formal development", in: 2005 IEEE/WIC/ACM International Conference on Web IntelligenceWI'2005, Compine, France, 2005.

[Cicirello *et al*. 05] V. A Cicirello, A. Mroczkowski and W. Regli "Designing Decentralized Software for a Wireless Network Environment: Evaluating Patterns of Mobility for a Mobile Agent Swarm" IEEE 2nd Symposium on Multi-Agent Security and Survivability, 30-31 Aug. 2005 pp 49 – 57

[Ferrara 04] A. Ferrara, "Web-Services: a process algebra approach", in: Proc. of the 2nd international conference on Service oriented computing, New York, NY, USA, 2004, pp 242–251.

[Fu *et al*. 04] . X. Fu, T. Bultan, and J. Su. "Analysis of interacting BPEL Web-Services". In S.I. Feldman, M. Uretsky, M. Najork, and C.E. Wills, editors, Proc. of the 13th International World Wide Web Conference, pp 621-630, New York, NY, USA, May 2004. ACM.

[Fu *et al*. 05] X. Fu, T. Bultan, and J. Su. "Synchronizability of conversations among Web-Services". IEEE Transactions on Software Engineering, 31(12) pp 1042-1055, December 2005.

[Fuxman *et al*. 04] A. Fuxman, L. Liu, J. Mylopoulos, M. Pistore, M. Roveri, and P. Traverso. "Specifying and analyzing early requirements in Tropos". Requirements Engineering, Vol. 9[2], pp 132-150, May 2004.

[Hinz *et al*. 05] S. Hinz, K. Schmidt, and C. Stahl. "Transforming BPEL to Petri nets". In W.M.P. van der Aalst, B. Benatallah, F. Casati, and F. Curbera, editors, Proc. of the 3rd International Conference on Business Process Management, LNCS 2649, pp 220-235, Nancy, France, September 2005. Springer-Verlag.

[Kendall *et al*. 98] E.A. Kendall, P.V.M. Krishna, C.V. Pathak, and C.B. Suresh. "Patterns of intelligent and mobile agents". Proc. of the 2nd International Conference on Autonomous Agents (Agents'98), New York, 1998. ACM Press.

[Khalaf 04] R. khalaf "Business Process with BPEL4WS: Learning BPEL4WS, Part 2" Available at http://www.ibm.com/developerworks/Web-Services/library/ws-BPELcol2/ 2004

[Lapadula *et al*. 06] A. Lapadula, R. Pugliese and F. Tiezzi, "A wsdl-based type system for ws-BPEL., in: Proc. Of COORDINATION'06", LNCS 4038 , pp 145–163. 2006

[Lapadula *et al*. 07] A. Lapadula, R. Pugliese and F. Tiezzi, "A Calculus for Orchestration of Web-Services"., in: Proc.of 16th European Symposium on Programming (ESOP'07), LNCS 4421, pp 33–47. 2007

[Lohmann *et al*. 06] N. Lohmann, P. Massuthe, C. Stahl, and D. Weinberg. "Analyzing interacting BPEL processes". In Proc. of the 4th International Conference on Business Process Management, LNCS 4102 pp 17-32 Vienna, Austria, September 2006. Springer-Verlag.

[Loke *et al*. 99] S. W. Loke, H. Schmidt and A. Zaslavsky "Programming the Mobility Behaviour of Agents by Composing Itineraries" : Proc. of the 5th Asian Computing Science Conference on Advances in Computing Science 1999 pp 214 - 226

[Lucchi & Mazzara 07] R. Lucchi and M. Mazzara, "A pi-calculus based semantics for ws-BPEL", Journal of Logic and Algebraic Programming, Elsevier press (2007).

[Magee & Kramer 06] J. Magee and J. Kramer. "Concurrency: State Models and Java Programs". Wiley, New York, NY, USA, second edition, 2006.

[Nakajima 02] . S. Nakajima. "On verifying Web-Service ows". In Proc. of the Symposium on Applications and the Internet Workshops, pp 223-224, Nara City, Japan, January/February 2002. IEEE.

[Oasis 07] Oasis, Web-Service business process execution language version 2.0 specification, oasis standard, http : //docs.oasis – open.org/wsBPEL/2.0/wsBPEL – v2.0.pdf (2007).

[Ouyang *et al*. 07] C. Ouyang, H.M.W. Verbeek, W.M.P. van der Aalst, S. Breutel, M. Dumas, and A.H.M. ter Hofstede. "Formal Semantics and Analysis of Control Flow in WS-BPEL". Science of Computer Programming Vol. 67[2,3] pp 125-332, July 2007. Elsevier Science Publications.

[Reisig 05] W. Reisig. "Modelling and analysis techniques for Web-Services and business processes". In M. Ste_en and G. Zavattaro, editors, Proc. of the 7th IFIP WG 6.1 International Conference on Formal Methods for Open Object-Based Distributed Systems, LNCS 3535 , pp 243-258, Athens, Greece, June 2005. Springer-Verlag.

[Schmidt & Stahl 04] K. Schmidt and C. Stahl. "A Petri net semantic for BPEL4WS validation and application". In E. Kindler, editor, Proc. of the 11th Workshop on Algorithms and Tools for Petri Nets, pp 1-6, Paderborn, Germany, September/October 2004. Universit at Paderborn.

[Stahl 05] C. Stahl "A Petri Net Semantics for BPEL". Technical Report 188, Humboldt-Universität zu Berlin (2005)

[Tahara & ohsuga 01] Y. Tahara and A. Ohsuga "Beheviour Patterns for Mobile Agent Systems from the Development Process Viewpoint " Proc. of the 5th International Symposium on Autonomous Decentralized Systems, pp 239 – 242. 2001

[Tahara *et al*. 99] Y. Tahara, A. Ohsuga and S. Honiden, "Agent system development method based on agent patterns" in: Proc. of the 21st international conference on Software engineering , pp 356–367. 1999

[verbeek & Aalst 05]. H.M.W. Verbeek and W.M.P. van der Aalst. Analyzing BPEL processes using Petri nets". In D. Marinescu, editor, Proc. of the Second International Workshop on Applications of Petri Nets to Coordination, Workflow and Business Process Management, pp 59-78, Miami, FL, USA, October 2005.

[Verbeek *et al*. 01] . H.M.W. Verbeek, T. Basten, and W.M.P. van der Aalst. "Diagnosing workow processes using Woan". The Computer Journal, Vol. 44[4] pp 246-279, July 2001.

[Wombacher *et al*. 04] A. Wombacher, P. Fankhauser, and E. Neuhold. "Transforming BPEL into annotated deterministic finite state automata for service discovery". In Proc. of the 2nd IEEE International Conference on Web-Services, pp 316-323, San Diego, CA, USA, July 2004. IEEE.

[Yang *et al*. 05] Y. liu Q. Yang, J. Y.Tan and F. Liu. "Transformation BPEL to CP-nets for verifying Web-Services composition". In Proc. of the International Conference on Next Generation Web-Services Practices, Seoul, Korea, August 2005. IEEE.

[Yang *et al*. 05] Y. Yang, Q. Tan, and Y. Xiao. "Verifying Web-Services composition". In J. Akoka, S.W. Liddle, I. Song, M. Bertolotto, I. Comyn-Wattiau, S.S. Cher_, W. vanden Heuvel, . Thalheim, M. Kolp, P. Bresciani, J. Trujillo, C. Kop, and H.C. Mayr, editors, Proc. of the ER 2005 Workshops, LNCS 3770, pp 354-363, Klagenfurt, Austria, October 2005. Springer-Verlag.

[Yi & Kochut 04] X. Yi and K.J. Kochut. "A CP-nets-based design and verification framework for Web-Services composition". In Proc. of the 2nd IEEE International Conference on Web-Services, pp 756-760, San Diego, CA, USA, July 2004. IEEE.

[Yi & Kochut 04bis]. X. Yi and K.J. Kochut. "Process composition of Web-Services with complex conversation protocols: a colored Petri nets based approach". In Proc. of the Design, Analysis, and Simulation of Distributed Systems Symposium, pp 141-148, Arlington, VA,

Conclusion and further works.

By this work we have proposed a new architecture for mobile workflows called MOBIFLEX, Comparing to related works, MOBIFLEX architecture brings together some features that seems very promising. First of all MOBIFLEX combine advantages of multiagent technology and Web-Services technology. In the other hand MOBIFLEX is powered by a fault tolerance mechanism which is missing in similar works .In MOBIFLEX we tried to resolve limitations of mobile devices by a judicious combination of PCs and mobile devices since hard tasks are not executed on PDAs but on fixed nodes in the network and the PDA is used to launch the execution of the workflow and receive the results. MOBIFLEX combines in a convenient manner PCs and mobile devices so that the mobile device is used with respect to its limited resources.

We proposed a formal framework for specifying and verifying MOBIFLEXMOBIFLEX architectures, the framework is based on rewriting logic. To overcome the complexity of the system we distinguished between the Web-Services layer and the workflow enactment by mobile agents. so we proposed to formalize each one by itself. This was done by a judicious combination of formal tools to deal with all facets of the system. For the first layer we propose and use the REACTnets which combine abstract data type, algebraic Petri nets and rewriting logic, for the second layer we use MAUDE language and especially object orient MAUDE modules which combine object oriented paradigm with rewriting logic.

For further works, we intend first of all to achieve the implementation of the MOBIFLEX prototype, and secondly to work on an integrated tool handling the specification and verification of the two layers of MOBIFLEX by REACTNets and MAUDE.

CONTENTS