[11], [12], [13], [14]) appears to be very promising and WADE fits in it.

The paper is structured as follows: in chapters 2 and 3, we present the applications mentioned above that have a direct influence on the work of thousands of technicians and potentially millions of customers. The first one called "Network Neutral Element Manager" implements a mediation layer between network elements and OSS systems. The second one, known as "Wizard", provides step-by-step guidance to technicians performing maintenance operations in the fields. In chapter 4 and 5 we focus on WADE, the software platform at the basis of both applications actually implementing the agents and workflows related features. Finally in chapter 6 we draw some conclusions and present future activities.

2. NETWORK NEUTRAL ELEMENT MANAGER

One of the major problems in the Operation Support Systems domain is related to the lack of standards in the management interfaces that network elements provide. This is even more critical in large and highly multivendor telecommunication networks such as that of Telecom Italia. Because of this lack of standardization each vendor provides its own Element Manager with a proprietary interface both in terms of protocol (CORBA, SNMP, TL1, XML being the most common ones), mimic of interactions and data modeling. As a consequence all OSS systems that require communicating with the network (such as the activation system, the troubleshooting system, the performance monitoring system and so on) need to embed proper adapters for each vendor and type of device. Moreover every time a new technology, a new vendor or even a new release of a network element is deployed, all OSS systems need to be updated thus multiplicating the effort and slowing down the roll out of new services.

In order to face this problem, a project called "Network Neutral Element Manager" was started four years ago in the OSS Innovation department of Telecom Italia. The focus of the project was the development of a mediation layer decoupling the management systems from the network elements. As depicted in Figure 1 the Network Neutral Element Manager (NNEM) aimed at hiding the diversity of the underlying technologies, vendors and types of device to OSS systems by providing a uniform north-bound interface.

Another important benefit of the NNEM was related to the possibility of controlling the management overload. Having a single entity carrying out all interactions with the network, in facts, allows governing the requests of the different OSS systems. This avoids slowing down the performances of the network due to uncontrolled accesses performed by completely un-coordinated systems.

Considering the challenging goals described above, it was clear that the NNEM had to meet strong requirements in terms of:

Scalability - The NNEM had to be able to manage thousands of network elements and serve several management systems potentially producing each one thousands of requests per minute.

Flexibility - A big telecommunication network is a sort of "living animal" that evolves mostly every day in terms of new services, new technologies, new vendors, new types of device and new firmware releases. Clearly the NNEM had to cope with this continuous evolution providing proper mechanisms to support hot deployment of new/modified system logics.

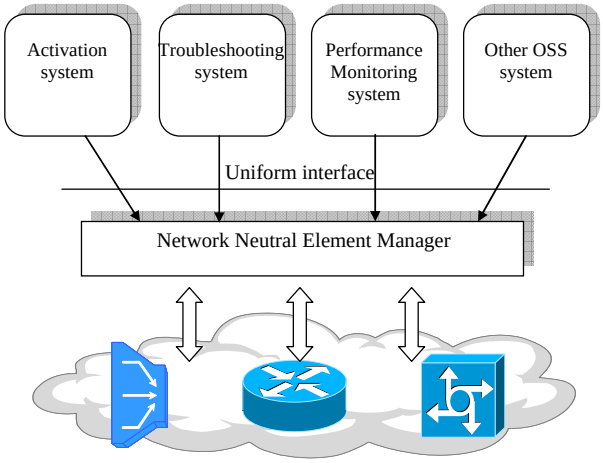

Figure 1. The Network Neutral Element Manager

In order to meet the scalability requirement it was decided to adopt agent technology as the basis for the NNEM. In particular the core of the system was architected with one agent called Resource Proxy for each network element. A Resource Proxy is responsible for virtualizing the related network element and managing all accesses to it. Each Resource Proxy keeps an image of both the physical structure (cards, ports and so on) and the configurations (logical interfaces, cross-connections, profiles) of the virtualized device. This image, called "cache" is normalized according to a vendor independent model closely derived from the SID (Shared Information/Data model) being defined within the scope of the Tele Management Forum [25]. The Resource Proxy also processes the traps issued by the virtualized device and is therefore able to keep its cache constantly up to date.

Flexibility was achieved by describing all the logics of interactions with the network elements to be carried out by Resource Proxy agents as workflows. That is, instead of directly embedding the code implementing these interactions, Resource Proxy agents were designed to include a very light workflow engine. Each time a modification occurs in the network a new/modified workflow reflecting the modification can be deployed at runtime thus making the Resource Proxy agents immediately able to cope with the new situation.

Both the basic agent-related features such as execution model, communication, discovery and life cycle management and the ability to execute possibly long and complex tasks defined according to the workflow metaphor are provided to the NNEM system by a software platform called WADE that will be described in chapter 4.

The Network Neutral Element Manager is currently deployed on 15 low cost HP Proliant DL 145 servers each one equipped with 2 ADM Opteron 246 processors (2 GHz) and 4 Mbytes of RAM (cost per unit between 2000 and 3000 euros at the end of 2006). It manages the network elements of the IP broadband and ultra-broadband access (~2000 devices considering IP DSLAM and ONU from 4 different vendors) serving the activation,

troubleshooting and partially the fault management processes. Extensions to new domains (and in particular the GBit Ethernet metropolitan network) and new processes (such as performances monitoring and configuration management) are under evaluation.

As described in [26] laboratory tests carried out in 2007 shown that more than 5400 IPTV service activation requests per hour can be served on just 3 of the servers mentioned above.

3. WIZARD

The costs of the operational processes, such as the ones for providing new services to customers or for removing failures and malfunctions, represent an important percentage of the costs that telecommunications operators must face yearly. Hence the importance of systems aimed at reducing such costs through tools supporting activities of both operator workforce and customers, which can thereby be directly involved in removing troubles related to equipments in the network and in the customer home.

To support the workforce that is directly involved in actions of repairing failures/malfunctions of the network we have realized a software system called Wizard. Wizard guides the technical staff in a complete, integrated and exhaustive way, through all the steps to be followed in problem-solving activities. A complete guide enables both a reduction in the working times for the technicians and a faster insertion of technicians that are new to the job.

In the first place, the system provides a direct interaction with the systems/platforms responsible for network and service management. This significantly reduces the times of execution of problem-solving activities since the correct completion of the jobs performed by the technical staff can be verified in real-time by Wizard that proactively triggers suitable checks with the right data on the relevant OSS systems.

In the second place, the system represents using formal tools (such as the workflows) the operative knowledge to be shared by technicians. This enables a further reduction of the working times of the technicians through an unambiguous and readily understandable description of the activities to be carried out. Such formal representation would also avoid any possible difficulties of interpretation of the supplied indications, which can lead technicians to execute activities that are useless or even harmful for the network.

The Wizard system has been developed on top of WADE (that will be described in chapter 4) and is aimed to support business processes including both automatic tasks (machine to machine interactions) and human task (human workflows, that means the support for human activities and a real-time interaction with the user). One of the features added by Wizard is the concept of Workflow Driven GUI, that means a Graphic User Interface that allows the real-time interaction between the workflow execution and the user (see Figure 2).

This GUI runs also on mobile assets in order to take into account nomadic workplace environments

We have carefully chosen the case study working together with on field technicians in order to understand all details, critical points and bottleneck of their job.

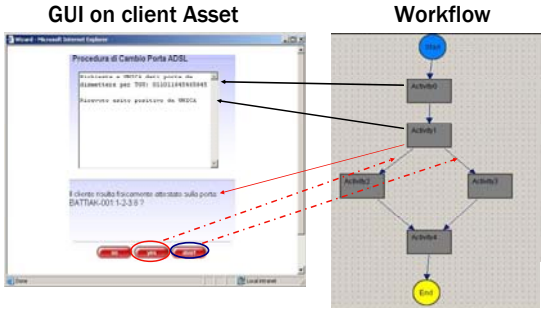

Figure 2. Workflow driven GUI

One result is a new way to perform the ADSL diagnosis that reduce the work for the back office and empower the on field technician.

The technician on field receives the work-request on his mobile asset, drives to the central office (where the network elements are located) and starts through his mobile asset the “ADSL diagnosis and repair“ workflow that drives him through all steps of the whole process interacting with him and with the remote systems when needed (see Figure 3)

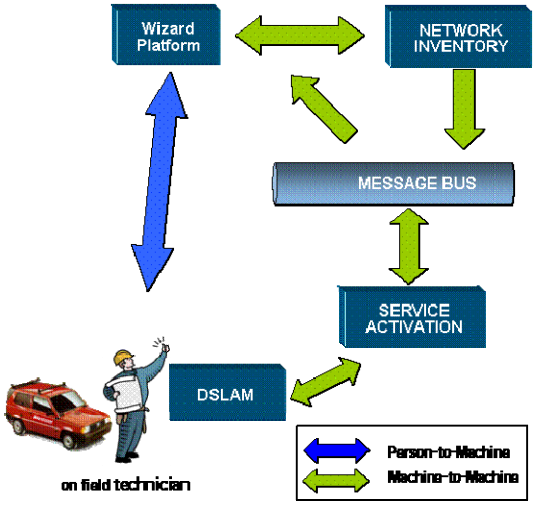

Figure 3. ADSL Diagnosis and repair scenario

At the end of the workflow the technician gets, as a result, the new pair position on MDF (main distribution frame) related to the new DSLAM port to the technician

Finally the technician moves the jumpers on the new MDF positions and closes his work-request

The diagnosis process now is well defined and documented as part of “ADSL Diagnosis and repair workflow”

Technicians of any experience level are able to address the work-request and possibly even learn through the workflow they execute in order to accomplish it

The technician through the Wizard Platform can use the automatic service fulfillment chain in order to reconfigure the circuit saving a lot of time (from hours to minutes)

The feedback from the field is really good because of the time saved by the technician avoiding the phone call to the back office operator and the time saved by the back office operator itself . The overall process now takes only few minutes instead of hours

The solution presented in this section is used in Telecom Italia by hundreds of technicians each day guaranteeing the assurance on the 7.3 million broadband connections.

4. THE WADE PLATFORM

Though addressing different domains and showing opposite characteristics in terms of user interactivity, the applications described in sections 2 and 3 are both built on top of a common software platform called WADE. WADE (Workflow and Agent Development Environment) represents the main evolution of ***JADE*** [5], a popular open source middleware conceived to facilitate the development of distributed applications based on the ***agent-oriented paradigm***.

As depicted in Figure 4, JADE provides a distributed runtime environment, the "agent" and "behaviour" abstractions, peer to peer communication between agents and basic agent lifecycle management and discovery mechanisms. WADE adds to JADE the support for the execution of tasks defined according to the ***workflow*** metaphor and a number of mechanisms that help ***managing the complexity of the distribution*** both in terms of administration and fault tolerance. This paper in particular focuses on the aspects related to workflow based development that we consider WADE most characterizing feature.

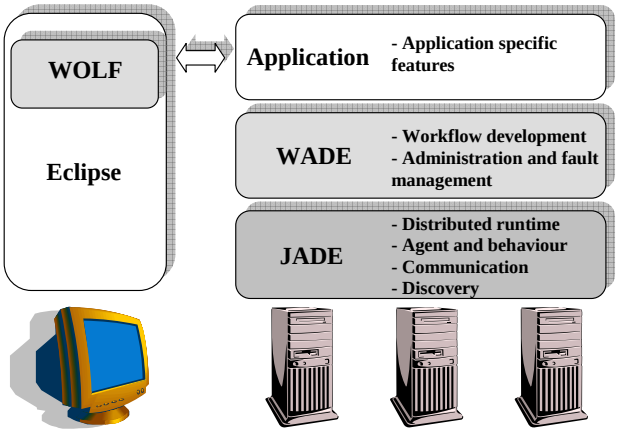

Figure 4. The WADE platform

In principle WADE supports "notepad-programming" in the sense that there is no hidden stuff that developers can't control. However, especially considering that one of the main advantages of the workflow approach is the possibility of representing processes in a friendly graphical form, WADE comes with a development environment called ***WOLF*** that facilitates the creation of WADE-based application. WOLF is an Eclipse [15] plug-in and as a consequence allows WADE developers to exploit the full power of the Eclipse IDE plus additional WADE-specific features.

4.1 Workflow based development

A workflow is the definition of a process in terms of activities to be executed, relations between them, criteria that specify the activation and termination and additional information such as the participants, the software tools to be invoked, required inputs and expected outputs and internal data manipulated during the execution.

The main advantage of implementing a process as a workflow is the expressiveness of the workflow metaphor. ***A workflow in facts can be represented in a purely graphical form*** that is understandable by domain experts as well as by programmers. Domain experts can therefore validate system logics directly and not only on documents that most of the time are not perfectly up to date. In some cases they could even contribute to the actual development of the system without the need for any programming skill.

Another important characteristic is that the steps that compose the process are explicitly identified. This makes it possible to create ***automatic mechanisms that trace the execution of a workflow*** thus facilitating system monitoring and problem investigation. Additionally, when processes have to be executed within the scope of a transaction, ***semi-automatic rollback procedures*** can be activated in case of unexpected fault.

Finally, since ***workflows are fully self-documented***, workflow-based development releases the development team of the burden of keeping documentation aligned each time design choices must be revisited to face implementation details.

4.1.1 Scope

Nowadays the workflow metaphor is mostly used in BPM (Business Process Management) environments where a workflow represents a business process and orchestrates a number of existing systems typically (but not necessarily) accessible by means of Web Services-based interfaces.

The main challenge in WADE is to ***bring the workflow approach from the business process level to the level of system internal logics***. That is, even if it could be used for that purpose too, WADE does not target high level orchestration of services provided by different systems, but the implementation of the internal behaviour of each single system.

A direct consequence of the described approach is that WADE is expected to be particularly suitable for applications that imply the execution of ***possibly long and fairly complex tasks***.

Furthermore, unlike the majority of existing workflow systems that provide a powerful centralized engine, in WADE each agent can embed a "micro workflow engine" and a complex process can be carried out by a set of cooperating agents each one executing a piece of the process.

From an industrial point of view one advantage in using WADE is the possibility to develop mission critical applications that can work on grid of blade servers (or PC) with great scalability, and big savings in hardware [16]. For example, one impressive success of PC-derived components, harnessed in parallel by open-source-based software, is the Google search engine, implemented on a massive cluster comprising more than 15,000 commodity-class PCs as described in an paper published in 2003 [17]. The