TEMIC: a New Cooperative Platform for Industrial Tele-Maintenance

Mohamed Brahma[1], Martine Chaudier[2], Eric Garcia[3], Jean-Patrick Gelas[2], Hervé Guyennet[3], Fabien Hantz[3], Laurent Lefèvre[2], Pascal Lorenz[1], Henri Tobiet[4]

1. Université de Haute Alsace IUT
34 rue du Grillenbreit - 68008 Colmar - France
[mohamed.brahma, p.lorenz]@uha.fr

2. INRIA RESO / LIP (UMR CNRS, ENS, INRIA, UCB), Ecole Normale Supérieure
46, allée d'Italie - 69364 LYON Cedex 07- France
[jpgelas, laurent.lefevre]@ens-lyon.fr

3. LIFC Laboratoire d'Informatique de l'Université de Franche-Comté
16, route de Gray - 25030 Besançon Cedex- France
[garcia, guyennet, hantz]@lifc.univ-fcomte.fr

4. SWI – SA
20, rue Salomon Grumbach - 68200 Mulhouse- France
h.tobiet@swinnovation.com

Abstract

Since the year 2000, the performances that are obtained from the point of view of the networks processors make it possible to use real distribution applications. Particularly, in remote maintenance domain, the network quality makes it possible to have several expertises readily available.

As a result, the quality of maintenance increases as it becomes faster, more reliable and highly efficient. The TEMIC Tele-maintenance application will connect mobile users (via their PDA Terminal) to Information Systems in order to perform activities in areas such as corrective, preventive and conditional maintenance.

The TEMIC Tele-maintenance platform is based on a three-level architecture: Collection and Surveillance, Intervention and Reporting, Diagnosis and Collaboration. The use of wireless networks is one of the TEMIC distinctive characteristics, which allows connection between mobile Tele-maintenance qualified actors. Another distinction is the design of active and autonomous industrial network devices that are easily and efficiently deployable in an industrial setup.

Keywords: *Telemaintenance, mobility, wireless networks, active and programmable networks, cooperative work*

1 Introduction

Currently, problems of maintenance for production materials in industry are often pushed into the background in view of the requirements of production [9]. This situation is due mainly to an absence of maintenance policy and to the costs caused by this one, in particular in term of mobility of the qualified personnel carrying out maintenance.

With the maturation of new telecommunication technologies, the industrial companies want to integrate these projections to activate the interventions of the specialists on site and thus to decrease times of inactivity of the machines. The high speed communications make it possible to have quickly several concurrency expertises. Maintenance thus becomes faster, more reliable and of better quality.

In the RNRT TEMIC project*, we propose a hardware and software solution allowing co-operative remote maintenance: not only maintenance technician can carry out his remote work (remote maintenance) but also he can do it in collaboration with other experts (co-operative work). The emphasis is put on the aspects network (heterogeneity, dynamicity) and mobility of the co-operating members for the realization of the action of maintenance. Our system will make it possible to increase, in particular, availability, mobility, quality, and to reduce the costs involved by the failures of the equipment.

TEMIC aims to design and develop a solution of expertise and decision-making aid for technician in charge for the maintenance of the

* This work is supported and carried out within the framework of the French RNRT TEMIC Project.

industrial facilities. from the communication network point of view, these tools will make it possible to the engineers to remotely reach relevant information and to have of indicators and dashboards in order to evaluate the effectiveness of the services concerned with maintenance and to contribute to the decision-making thanks in particular to the co-operative aspect of the application. This last point will support the work of group.

This synthesis paper presents, in the first section, the global architecture of the platform. Then the following section describes the use of the wireless networks in our platform: it allows connection between mobile tele-maintenance qualified actors. In section 4, we present the design of active and autonomous network equipments to be easily and efficiently deployed in an industrial context. Last section concludes this paper.

2. Architecture of the platform

2.1 TEMIC Generic approach

The TEMIC Tele-maintenance application will connect mobile users (via their PDA Terminal) to Information Systems (IS), to perform activities in the area of corrective, preventive and conditional maintenance.

The TEMIC Tele-maintenance platform is based on three-level architecture (see Figure 1) :

- Level 1: **Collection and Surveillance (C+S)**
    - o allows data collection from sensors and different types of industrial interfaces,
    - o performs wireless data acquisition via Wi-Fi adhoc networks,
    - o is dedicated for real-time process supervision and control.
- Level 2: **Intervention and Reporting (I+R)**
    - o allows maintenance management via mobile terminals (e.g. PDA = Personal Digital Assistant),
    - o performs wireless information reporting on maintenance activities,
    - o is dedicated for real-time system maintenance (CAD/CAM maintenance management software).
- Level 3: **Diagnostic and Collaboration (D+C)**
    - o allows collaborative actions via extended networks and communication facilities (e.g. videoconferencing via the Internet) [8, 2],
    - o performs decisional diagnostics and forwards corrective actions to recover the process.

The three architecture levels are interacting as described below:

2.2 Proposed platform

The three levels are connected together to perform a complex diagnostic centre.

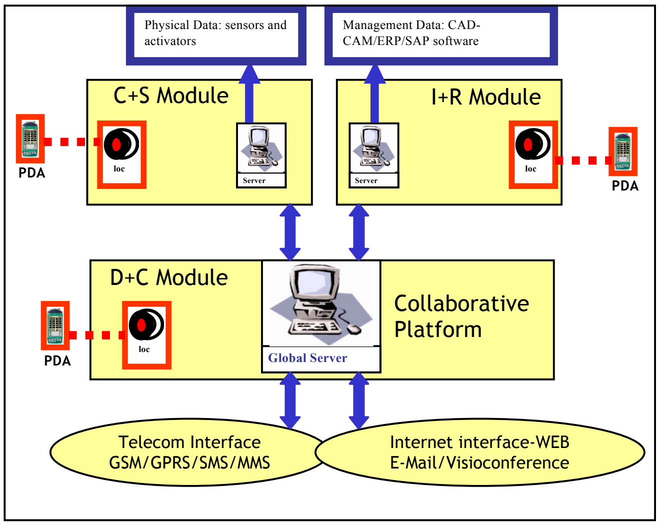

Figure 1: Diagnostic centre

The proposed platform combines data acquisition from both hardware and software information systems:

- hardware data are mainly information coming from process sensors or automatic measurement stations,
- software data are extracted from information systems (servers) like CAD/CAM systems.

Collaboration will be effective:

- locally via wire-line or wireless LANs locally on site,
- remotely via extended telecom networks.

Maintenance operations can be executed:

- instantaneously for corrective maintenance (alarms)
- cyclically for preventive maintenance
- on-demand for conditional maintenance.

2.3 The SWAP process and the concept of Empathy

The connection between the mobile terminal and the information systems is performed via the SWAP (SWI Wireless Application Process) protocol on an OFFER (server) – DEMAND (terminal) procedure.

The connection is effective when three conditions are simultaneously achieved (Empathy). These conditions are based on multi-key recognition:

- a qualitative (Q) key based on user profile, identifier, password, etc …
- a geographic (G) key based on the distance between server and terminal (localization)

- an environmental (E) key based on external conditions (date, hour, alarm, SMS….)

The Empathic process (Q/G/E) allows secured and personalized access to the process information.

2.4 Collection and Surveillance architecture

Data issued from various sensors are collected in real-time via a Wi-Fi proximity network (Figure 2). The data are stored in an industrial SWAP server, situated near the process to be controlled. This server is able to perform a first level of diagnosis (alarms, data calibration…) before making these data available to mobile users. A second Wi-Fi ad-hoc network informs the mobile user that data are available.

When Empathy is performed (key recognition), then a high-speed connection is established between the mobile user and the industrial server; and data are then transferred to the nomadic terminal (PDA).

Data issued from sensors can be temperature, pressure, power consumption, etc…. or multimedia streams coming from IP webcams, microphones..

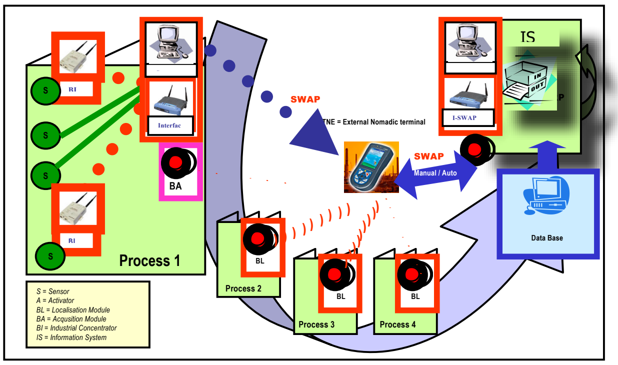

Figure 2: Architecture of Collection and Surveillance

This scenario applies in situations when automatic data collection is required, especially on places where no wired network is available.

- The user of the mobile terminal will collect successively the required data on selected process servers.
- For each operation a wireless proximity network is established, and authorized data are stored by the mobile terminal.
- After acquisition completion, data is downloaded in the information system.

2.5 Intervention and Reporting architecture

Following documents are processed by the mobile terminal:

- **« Server » -based documents downloaded from the CAD-CAM server to the terminal**
  - Demand for Intervention (DI)
  - Stocks consultation
  - Planning Consultation
  - Agenda Consultation (coordinates of people able to access the process)
- **PDA-based documents constructed on the mobile terminal and downloaded from the terminal to the CAD-CAM server**
  - Intervention Reporting (RI)
  - Demand for Material
- **Process –based documents installed on the terminals**
  - Hysterics of Interventions on the process to be maintained
  - Technical details: plans, photos, graphics, lists, etc……

The empathic process (Q/G/E) will connect the mobile terminal to the information server (Figure 3).

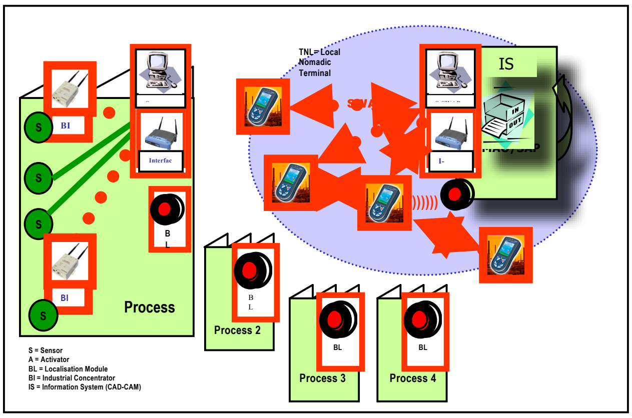

Figure 3: Architecture of Intervention and Reporting

The mobile user will connect to the CAD-CAM information server via the deployed proximity network (Wi-Fi)

- before starting the maintenance activity, he downloads the daily work to be done on his PDA,
- when he has finished (or interrupted) his work he downloads to the server the reports he has noticed on his PDA,
- interactions between user and server can be established in real-time: this needs full Wi-Fi network coverage.

3 Bluetooth/SMS convergence module

3.1 Proposed scheme

In TEMIC Tele-maintenance application, the server is able to perform a first level of diagnosis (alarms, data calibration …) before making these data available to mobile users. A mobile user receives alarms on his mobile phone. We proposed a Bluetooth-SMS convergence module to transfer the data to the mobile user's PDA as described below:

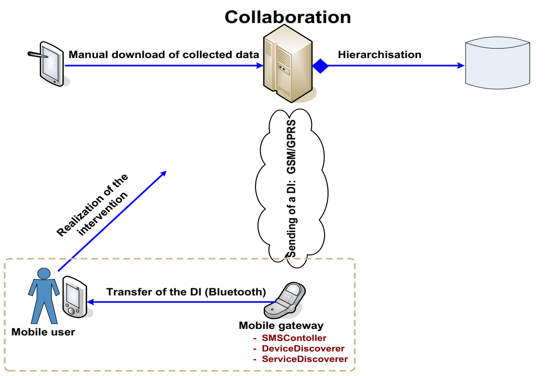

Figure 4: Bluetooth/SMS convergence module

To implement this module we used Java 2 Platform, Micro Edition (J2ME) [1]. This platform offers a certain number of functionalities and APIs that few mobile phones currently support, as JSR-82 API constituted of two significant packages: *javax.obex* (Core Object Exchange) and *javax.bluetooth*.
J2ME Wireless Toolkit is a tool [4] provided by Sun Microsystems. It automates all the operations of the cycle of development of MIDlet. The operations which it deals with go from the compilation of MIDlet, to its execution and the generation of description file (.jad). J2ME is an edition for mobile terminals of the Java technology in which, the environment of execution is adapted to machines "with constraints" or machines whose capacities are limited in comparison with computer or a server: memory extremely limited, small size of screen, and slow processors.

3.2 Implementation

The modeling of SMS/Bluetooth convergence is the entrance point of the implementation. Different parts are defined:

- **SMSController:** it represents the heart of the gateway. This module is responsible for the management of the reception of the messages, the research of the PDA and the sending of SMS. In other term, this module plays the role of a coordinator between the other modules of the gateway.
- **SMSReceiver:** it is the module which deals with the reception of messages (DI) coming from outside (platform of collaboration) via GSM/GPRS. When this module is activated, it is put on standby of a connection on one of the ports (port 5000). As soon as a message arrives, this module deals with the reception. And the message will be stored in a structure of data of the type "String" called receivedText, and which will be used thereafter by the SendMessage module.
- **DeviceDiscovery:** This module is used to launch an agent to seek the destination PDA, and which is in the Bluetooth range of the gateway, in order to relay the message.
- **ServiceDiscovery:** after having found the mobile of destination, it remains to do connect itself to the service. For that it is enough to use an agent which deals with discovered service and which is called DiscoveryAgent. This agent uses a function called searchServices which enables him to find the services offered by the distant mobile.
- **SendMessage:** since the received SMS is stored in receivedText, this module deals with the part of sending of this message via Bluetooth.
- **Messages:** received messages (DI), are stored in an object "Message". Whose contents which can be of text type (TextMessage) or binary type (BinaryMessage) are transformed into character string (String). This character string will be sent via Bluetooth towards the PDA. The size of the messages is equal to the size of a standard SMS (160 characters).

3.3 SMS communication server

The SMS communication server is a flexible application making it possible to send and receive SMS using a modem (mobile phone...) connected to the server via one of the ports. This server allows SMS sending with the same cost as a mobile phone.

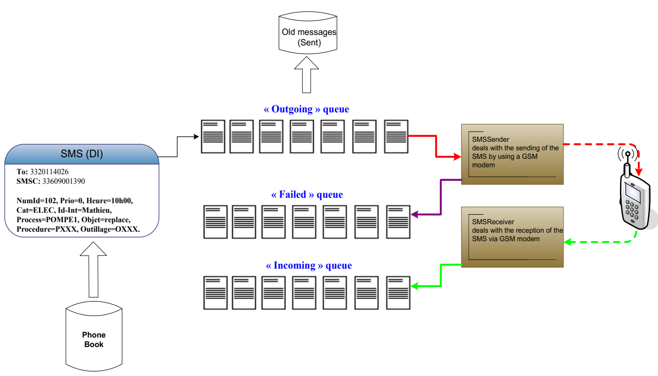

Figure 5: The SMS communication server

Figure 5 describes the server features. The most important part of this server is the SMSD module.

- **SMSD:** represent the most important part of the application. It deals with the management of the modem, and manages two different parts: SMSSender and SMSReceiver which are responsible respectively for the sending and the reception of the messages.

- **SMS format:** SMS messages sent by the server are simple textual files, composed of a heading containing the destination phone number, as well as the service center number, and of a body of the message. These messages which are stored first in an "Outgoing" queue must have unique names. The SMS format is as follows:

To: *3320114026*
SMSC: *33609001390*

NumId=102, Prio=0, Heure=10h00, Cat=ELEC, Id-Int=Mathieu, Process=POMPE1, Objet=replace, Procedure=PXXX, Outillage=OXXX.

Figure 6: SMS format

The "**To**" field indicates the recipient phone number (who can be a technician within the framework of TEMIC). "**SMSC**" field indicates the service center which will be used to send this message. These two fields must appear in the heading of the SMS which can also contain other optional information such as: the name of the "Provider" who can refer to the one of the mobile telephony operators. This field can be used by the SMSD.
Remark: The heading must be separated by a blank line from the body of the message.

- **«Outgoing» queue:** The sending of SMS starts first with the creation of text messages. These messages intended to be sent using the communication server pass by the Outgoing queue, while waiting for their sending. Periodically the SMSD checks the contents of this queue. Each message stored in the "Outgoing" queue is sent to its destination before being removed from this queue. A new checking of this queue is made a few seconds later. If the SMS sending fails, the sending module tries to resend it again. In the event of failure, the message will be moved to the "**Failed**" queue.

- **« Incoming » queue:** Received Messages are redirected towards this queue. At the beginning when the modem receives a message, this last is stored temporarily in the memory of the modem. The SMSD downloads the message from the modem. All the messages are stored in text files with random file names. The received messages contain two additional fields: the sending date and the "From_SMSC" which indicates the service center of which the message was received:

- **« Failed » queue:** In certain cases, the sending of messages can fail; in this case failed messages are stored in a queue which is called "**Failed**". The messages in the "**Failed**" queue can be recovered and returned once again, if their recipients do not appear in the black list "**Blacklist**".

4 Adaptation services in the network

4.1 Designing an industrial autonomic node for TEMIC

Research works about active and programmable networks and evaluation of the experimental prototypes [5] take place mostly in academic research laboratory. Currently no ``plug and process" active equipments are available on the market place.

In the framework of a cooperative industrial maintenance and monitoring project (TEMIC project), we design devices to be easily and efficiently deployable in an industrial context. Once the hardware deployed and used, it must also be easily removable at the end of the maintenance or monitoring contract. In this project, we deploy our devices in secured industrial departments, restricted areas, or in an out-of-the-way location. These devices must act as auto-configurable and re-programmable network nodes. Thus, the equipments must be *autonomic* and must not require direct human intervention.

The design of autonomic network equipment must take into account specific requirements of active equipments in terms of dynamic service deployment, auto settings, self configuration, monitoring but also in terms of hardware specification (limited resources, limited mechanical parts constraints and size constraints), reliability and fault tolerance.

We propose an adaptation of the high performance active network environment Tamanoir[6] in order to deploy it on limited resources based network boxes and to increase reliability and scalability. The implementation process is based on a hardware solution provided by the Bearstech Company. Through this approach we propose the architecture of an **I**ndustrial **A**utonomic **N**etwork **N**ode (called IAN^2) able to be deployed in industrial platforms. We evaluate the capabilities of IAN^2 in terms of computing and networking resources and dynamic re-programmability.

4.1.1 Hardware platform

This section describes briefly the hardware used to implement the IAN^2 **I**ndustrial **A**utonomic **N**etwork **N**ode. To support a transportable solution, we use a small compact aluminum case which hosts a small motherboard (200x150 mm) featuring a VIA C3 CPU1GHz, 256MB DDR RAM, 3 GigaEthernet LAN ports.

To reduce risk of failure, we choose a fan less hardware solution. Moreover, the box does not embed a mechanical hard disk drive. The operating system, file system and execution environment are stored in a memory card (e.g. Compact Flash).

4.1.2 Software execution environment: Programmable dynamic environment

This section describes the software used on top of the operating system node (described above). This software is called an Execution Environment (EE) which is used to dynamically plug-in and run Active Applications (AA) also called active services. A service is deployed on demand and applied on one or several data streams. Services can run in parallel and are all executed in the EE.

4.1.2.1 IAN^2 software architecture

We propose the IAN^2 Industrial Autonomic Network Node architecture [3] (
Figure 7). This node supports switching and routing protocols through wire and wireless connection hardware. The limited CPU facilities are open to dynamically deploy autonomic services. Some limited storage capabilities are available to support heterogeneous classes of services.

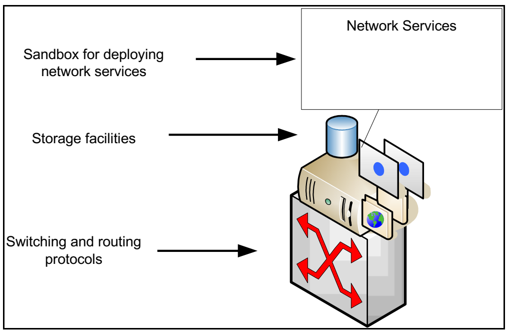

Figure 7: IAN^2 an Industrial Autonomic Network Node

Our Execution Environment called Tamanoir$_{embedded}$ is based on the Tamanoir [6] software suite written by L. Lefèvre and J.P. Gelas. The Tamanoir suite is a high-performance execution environment for active networks designed to be deployed in either local area networks or wide area networks. It is a prototype software with features too complex for an industrial purpose (cluster-based approach, Linux modules, multi-level services…).

Due to some typical industrial constraints (e.g. code maintenance), we reduce the code complexity and remove all unused classes and methods or actually useless for the TEMIC project. It allows us to reduce the overall size of the software suite and make the maintenance and improvement of the code easier for service developers.

4.1.2.2 Autonomic services deployment

Tamanoir$_{embedded}$ is a dedicated software platform fully written in Java and suitable for heterogeneous services. Tamanoir provides various methods for dynamic service deployment. First method allows services to be downloaded from a service repository to a Tamanoir Active Node (TAN). Second method allows a TAN to request the service from the previous active node crossed by the active data stream (15).

Tamanoir$_{embedded}$ also supports autonomic deployment and services updating through mobile equipments. Inside automatic maintenance projects, we deploy wireless based IAN^2 nodes in remote industrial environments (no wire connections available). In order to download maintenance information, human agents can come near IAN^2 nodes to request information. During this step, mobile equipments (PDA, Tablets, cellular) are also used as mobile repositories to push new services and software inside autonomic nodes (
Figure 8).

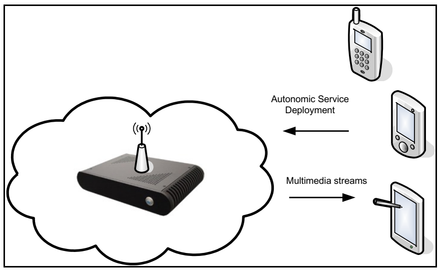

Figure 8: Autonomic Service Deployment through mobile nodes

4.2 Adaptation services

In this paper, we study and test two particular active services that we've developed for the TEMIC project. Both services are designed to adapt multimedia data, but they are intended to be used in different cases. This section describes first the implementation choices for active services and then both services.

Our framework is based on a client server model. The client plays a multimedia stream, sent by a web server. We use classical software on the server and on the client: an Apache web server and various players like VLC, PocketTV, Totem... The media adaptation is performed by the active service while data are transmitted from the server to the client.

Our active services use the Java Media Framework (JMF), version 2.1.1 [1] for the media adaptation. The JMF API enables the display, capture, encoding, decoding and streaming of multimedia data in Java technology-based applications. JMF also provides an API for the RTP (Real Time Protocol) transport protocol, which is adapted for transmission of real time media. JMF includes a number of *codec* to process data streams.

The JMF architecture contains a process unit called a JMF processor. The processing performed by a Processor is split into four main stages:

1. Demultiplexing - an interleaved media stream is first demultiplexed into separate tracks of data streams that can be processed individually (audio stream and video stream).
2. Data transcoding - each track of data can be transcoded from one format to another. If the individual tracks are compressed, they are first decoded.
3. Multiplexing - the separate tracks can be multiplexed to form an interleaved stream of a particular container content type.
4. Rendering - the media is presented to the user.

The active service initiates the processor by giving him the data source, the output media type, the video frame size, the audio sample rate and the output protocol.

Our framework support MJPEG and MPEG-1 as input video formats and MPEG layer II (mp2), PCM as input audio formats. Data are transcoded into the H263 format for the video track and into the LINEAR PCM for the audio track. These formats have been chosen among the JMF supported media formats, and also because H263 format provides a medium quality while needing only medium CPU requirements and low bandwidth requirements, whereas PCM format provides a high quality and needs low CPU requirements and high bandwidth requirements.

4.2.1 The VideoAdaptS service

The VideoAdaptS service is used for the transcoding (if necessary) and transmission of a media file (containing audio and video or only video frames) to the client (the player).

The client connects to the active network node by giving him the URL of the multimedia file he wants to download. The URL also contains the active service to apply to the resulting data stream. This HTTP request is analyzed by the active service in order to extract the client description. The active service then checks if it is necessary to transcode the media format to a less consuming resources media format. If being the case, it gives parameters to the JMF processor which will process data and creates a file, at the specified format, containing the transcoded data. This resulting file is then sent to the client.

If not the case, the request is simply transmitted to the server and then the response (containing the requested data) is sent back to the client. In both cases, data exchanges between the active service on the network node and the web server take place with the HTTP protocol, and the resulting file is transmitted to the client with the HTTP protocol.

The active service runs the JMF processor by giving:

- the input URL (i.e. the data location),
- the target transport video and audio formats,
- the file output content type.

4.2.2 The VideoStreamS service

The VideoStreamS service is used for the transcoding (if needed) and the real time transmission of a video sequence (coming from a source file or from a capture device) with the RTP protocol. The mechanism is similar to the one explained for the previous service. The main difference is in the output protocol.

Instead of sending an entire file over HTTP, this service streams video content over RTP, so that it enables immediate playback by the client (Figure 9).

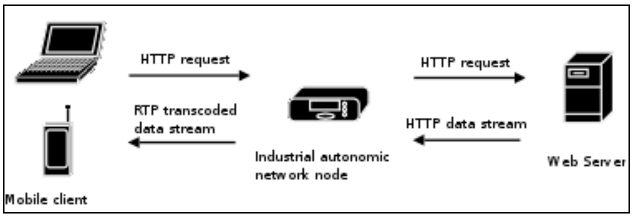

Figure 9: The VideoStreamS service mechanism

This service only transmits the video track, because in the case of an RTP transmission, each media stream is transmitted in a separate RTP session, and RTP players listen only for a single session.

The active service runs the JMF processor by giving:

- the input URL (i.e. the data location),
- the target transport video format,
- the stream destination (identified by its IP address and a port number).

Both services are used in different cases: the first service is preferably used when the user wants to download the file onto his terminal device (so he can read it again in the good format for his terminal capacities with no additional network request). Also the first service will be used in case no RTP player is available on the terminal device. The second service is better used for a real-time view, for instance when the video source is a video camera.

5 Conclusion and future works

Furthermore, this new solution of cooperative maintenance is in the process of deployment, and a demonstrator is currently being deployed. The TEMIC consortium now focuses its future work on innovative & mobile aspects of the application which include mobility nd adaptation amongst others. This advancement implies all layers of this project: adaptable HCI point of view, cooperative middleware point of view and also adaptability for mobile networks point of view.

6 Acknowledgments

We wish to thank the French government which has financed this project within the framework of a project RNRT (National Network of Research in Networks).

References

[1] Sun Java Wireless Toolkit. *Home page:* *http://java.sun.com/products/sjwtoolkit/*, consulted in 2006.

[2] H. Guyennet, D. Saint-Voirin, C. Lang, and N. Zerhouni,. *"Cooperative systems modeling, example of a cooperative e-maintenance system"*. In 6th Int. Symposium on computational intelligence in robotics and automation (CIRA'05), Espoo, Finland, pages 439-444, June 2005. IEEE.

[3] Martine Chaudier, Jean-Patrick Gelas, Laurent Lefèvre. *"Towards the design of an industrial autonomic network node"*, IWAN2005: Seventh Annual International Working Conference on Active Networks, Nice, France, November 21-22, 2005.

[4] Martin de Jode *"Programming Java 2 Micro Edition for Symbian OS: A developer's guide to MIDP 2.0"*. ISBN: 0-470-09223-8, May 2004

[5] L. Lefèvre and J.P. Gelas. *Chapter 14 "High Performance Execution Environments"* (pages 291-321) of the "Programmable Networks for IP Service Deployment", book - ISBN 1-58053-745-6; Galis, A., Denazis, S., Brou, C., Klein, C. (ed), published by Artech House Books, pp 430, UK, May 2004.

[6] Jean-Patrick Gelas and Laurent Lefèvre: *"Tamanoir: A High Performance Active Network Framework"* Active Middleware Services, Ninth IEEE International Symposium on High Performance Distributed Computing , Pittsburgh, Pennsylvania, USA - Aout 2000

[7] H. Tobiet and P. Lorenz *"Performance Measurement Methodologies and Quality of Service Evaluation in VoIP Networks"* (Editeur C.G. Omidyar), Mobile and Wireless Communications Networks, Lecture Notes in Computer Science, LNCS 1818, Springer, International Workshop MWCN'2000, Networking2000, May 14-19, 2000, pp. 92-107, Paris, France.

[8] J.W. Hong, Y.M. Shin, M.S. Kim and Y.H. Suh (1998). *"Design and Implementation of a Distributed Multimedia Collaborative Environment."* Special Issue on Multimedia Collaborative Environments of Cluster Computing: Networks Software Tools, and Applications, Baltzer Science, Fall.

[9] Decréton, M. and De Geeter, J. (1996). *"Remote Maintenance of Future Fusion. Remote Maintenance of Future Fusion Reactors, A Challenge for Rad-Hard Components and Smart Control Strategies"*. Proceedings of the International Conference on Remote Techniques for Hazardous Environments, Leicester.