Rail Technology '99

Use of the Internet for Remote Train Monitoring and Control: the ROSIN Project

ANDREAS FABRI

Information Technology CHCRC.C2

Computer Engineering Dept.

ABB Corporate Research Ltd.

CH-5404 Baden-Dättwil, Switzerland

E-mail: andreas.fabri@ch.abb.com

TXOMIN NIEVA

Institute for computer Communications and Applications (ICA)

Department of Computer Science (DI)

Swiss Federal Institute of Technology (EPFL)

CH-1015 Lausanne, Switzerland

E-mail: txomin.nieva@epfl.ch

PAOLO UMILIACCHI

ROSIN Project Manager

Laboratori Fondazione Guglielmo Marconi

Via Porrettana 123

I-40044 Pontecchio Marconi (Bologna), Italy

E-mail: paolou@labs.it

Keywords: Data Acquisition, Remote Monitoring, Embedded Systems, Internet Technologies

Executive Summary

Railways and mass transit transportation systems are experiencing a new growth that will have an important social impact in the global transport domain. Therefore new important steps are required to advance the urgent needs, such as interoperability at train and vehicle levels, support for commissioning and maintenance, communication with ground infrastructure and introduction of new services.

The ROSIN project aims at the validation of a complete and open platform, which will be the basis for a new generation of vehicles, consisting of an on-board network that interconnects all various on-board systems and subsystems.

Taking advantage of a 10 year effort to specify a standard Train Communication Network (TCN), carried out within the International Electrotechnical Commission (IEC) by Technical Committee 9, Working Group 22, the ROSIN project aims to work out a comprehensive solution that closely addresses users requirements and has been validated and demonstrated extensively, using available technology, in a wide range of applications.

The project covers all aspects of the on-board networking problems, addressing key areas such as passenger trains, freight trains and mass-transit (metros and trams). Communication and application requirements are considered, including maintenance needs and openness toward ground networks. Internet technologies are used to bring ubiquitous and low cost access to train data.

Introduction

The Train Communication Network Standard

The manufacturing of a train, or even a single vehicle for railways, metros or trams, is a very complex task involving a number of different electronic subsystems, often supplied by different manufacturers. Interconnection, cabling and control of these subsystems are not simple, as each manufacturer uses proprietary interfaces and protocols.

Problems also emerge at the train level, as is the case with international trains that consist of vehicles from different countries, with frequent coupling and uncoupling during daily operation. Interoperability between different vehicles can only be achieved by means of international specifications, like those issued by UIC (Union Internationale des Chemins de Fer), which are able to specify a common medium and compatible interfaces, in order to allow equipment to operate together.

In the past, it was sufficient to define a common cable, including many wire pairs, one for each of the required services (lights, doors, and so on).

The new architecture of future trains requires a more efficient, expandable and easily reconfigurable digital communication system, which must be based on a suitable set of standard specifications, to guarantee compatibility among subsystems and among vehicles.

The IEC - International Electrotechnical Commission, in collaboration with the UIC (Pilot Group 5R), started a new Working Group (WG 22) within Technical Committee 9 (TC9) in 1988, with the objective of finalizing a complete network specification, which was code-named TCN (Train Communication Network).

The objective of WG22 was to standardize *"the internal structures of communication networks on trains ensuring data transfer between programmable electronic equipment installed on different vehicles and interfacing between these networks and user equipment"*.

It took many years and a lot of effort to reach a final version, presently in the state of FDIS (First Draft International Standard). In the second quarter of 1999, the FDIS was submitted for the final vote: the result was positive, so TCN will become an International Standard (IS) with the name IEC 61375-1.

In the meantime, many manufacturers have already adopted TCN for new development. It is estimated that TCN has already been installed in more than 1,000 railway vehicles in the world, and many more will be in a very short time.

More information can be found at: http://www.labs.it/rosin/gistanda.htm.

Overview of the TCN

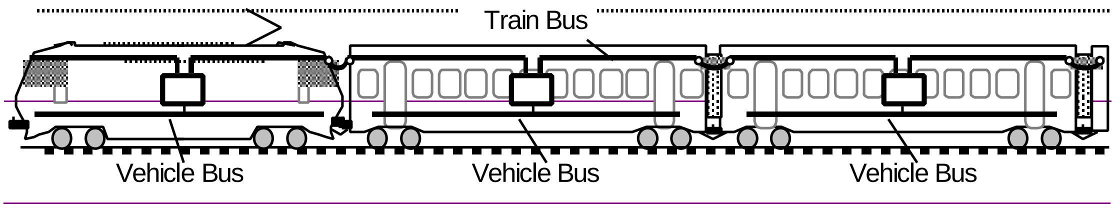

Figure 1: Train Communication Network

A two level hierarchical architecture (train bus and vehicle bus) is required, to comply with the train-vehicle topology. Up to 32 nodes can be networked on the train bus, which satisfies the need of all passenger trains, normally made of up to 22 vehicles. One of the nodes acts as a master, governing access to the bus.

One of the main features of the train bus is its capability to self-configure, including connecting a node in the network and dynamically assigning an address to it. This process is called inauguration. It is then simple for the Master node to address all the other nodes, giving them a short time slot to send their data.

The main difference between the train bus and the vehicle bus is that the latter is optimized to efficiently handle data traffic made of small packets originating from a high number of devices (up to several hundred). Moreover, the vehicle bus has to cover shorter distances, no more than 200 m compared to the 860 m of the train bus.

One special device, called the gateway, is able to link together the vehicle bus and the train bus. In this way, it is possible to exchange data between any two devices, both located on the same vehicle or in different vehicles of the train.

But what is actually sent through the train network? The data are divided into two categories:

- process data are short periodic data to be sent within precise timing constraints, so as to achieve a deterministic behaviour; process data are continuously refreshed; depending on the data importance, refresh time can span from 100 ms to several seconds
- message data are event driven data to be sent without stringent time constraints; long messages are divided into packets at the source device, sent one packet at a time and then put together at the destination device

The actual data transferred on the TCN depends on the application. TCN transports data from source to destination, regardless of its contents. Application level standards can precisely define the format and semantic of data for a specific application field. For example, UIC, addressing international conventional train applications, describes the data that are expected to be transmitted over the train bus, defining their format, semantics, usage, timing requirements and other prescriptions, all in a suitable document named UIC leaflet 556.

This makes it possible for different devices to co-operate, in a distributed, optimized architecture. For example, data produced by one device can be used by other devices at the same time.

Benefits of the TCN

TCN represents a standard communication platform for all kinds of applications on board trains.

Applications can be simply built, regardless of where the needed data comes from or how to send the data they produce to other devices. All these services, and much more, are automatically provided by the network for all applications, free of charge. Enabling very simple and inexpensive devices to coexist with highly complex on-board computers, TCN can allow for real distributed control systems.

As a standard communication interface, TCN facilitates interoperability between devices from different manufacturers and countries, using the newly defined UIC 558 cable, which includes two wire pairs for inter-coach TCN communication.

The availability of a standard communication infrastructure on board trains can also make it easier to introduce new services, so as to enhance the railway competitiveness.

However, TCN is not confined to a single train. By means of a radio link, train to ground communication can be easily established and the train network becomes a moving segment of a global network.

The ROSIN project

To contribute to TCN development and especially to validate a completely open platform in a wide range of applications, the ROSIN project was founded within the framework of the Telematics Application Programme of DG XIII. The project was started in January 1996.

The ROSIN consortium includes a wide representation of European railway industries, Adtranz, Alstom, Ansaldo, CAF, Firema, Siemens, covering most of the European countries: Germany, Italy, France, Spain, Sweden and Switzerland. Some research centers participate as subcontractors (Laboratori Fondazione Guglielmo Marconi in Italy and ABB Corporate Research in Switzerland). Two major international user organizations, UIC and UITP, participate as sponsoring partners.

Since the beginning of the project, a ROSIN User Group has been founded. In addition to UIC and UITP, a number of Railway and Mass Transit operators joined the User Group and expressed their interest in cooperating with ROSIN.

ROSIN applications can be divided into 4 major application fields:

a. passenger train applications deal with new services for passengers, such as passenger information systems, and with interoperability between vehicles and locomotives;

b. the freight train application deals with basic and application issues related to freight train specific characteristics;

c. mass transit deals with interoperability of devices coming from different manufacturers and integrated in the same on-board system;

d. the standard application interface deals with the standardization of data format and with a new train-ground communication system based on Internet technology and GSM radio links, to support remote maintenance.

In the next chapter, a brief presentation of application fields a, b and c is included, and the rest of this paper is devote to application field d.

More information is also available at: http://www.labs.it/rosin.

Concepts and Results

Passenger Train Applications

Passenger Information System

The Train Communication Network is normally invisible to passengers, but it can bring them useful new services.

ROSIN has explored the possibility of adding a complete information system to existing coaches. A number of displays have been suitably located in the coaches, both inside and outside, so as to show to passengers all the needed information:

- outside train route indicator (outside board)
- information display (in saloon vehicles)
- inside train route indicator (inside board in boarding area)
- seat reservation displays

The displays mentioned above are manufactured with different types of technologies and different resolutions, according to the coach constraints and information needs. A system controller manages the information to be displayed and links the system to the vehicle bus. A man-machine interface allows for input of free text messages, if required. A GSM radio link and a GPS receiver have been added to the system as well.

Relevant information for the passengers, in text and graphics, include:

- train identification and destination
- next station
- seat reservations
- advertisements

The seat reservation displays allow for precise indication of the status of available passenger space, by means of small LED displays located near each seat. Information data, and especially the seat reservation data, can be loaded into the system by means of a floppy disk, or even better, uploaded through the GSM connection. Due to the train bus connection, one GSM link is enough for the whole train.

For the same reason, one of the coach controllers includes a GPS receiver, which makes it possible to determine the train position, as well as precise time/date and train speed. This is useful for the automatic announcement of the next station stop, but also for updating the seat reservation data on the displays (passengers can reserve seats for a trip shorter than the complete train route).

The demonstration has enabled the checking of system functionalities, technical problems and economic impact and it is likely to be widely used in the future to retrofit existing coaches.

UIC Interoperability Testbed

Interoperability is a major issue for railways. ROSIN has investigated interoperability between vehicles from the networked applications point of view (i.e. interoperability between gateways), which link vehicles together on the train bus.

The first step was to contribute to the standardization of signals and procedures for data exchange, especially regarding traction control. The results are part of suitable specification documents, such as leaflets UIC 556 (version 12) and UIC 647.

The next step was to develop tools that can quickly check the conformance of gateways to such specifications. A testbed was developed to check gateway interoperability at the UIC 556 level.

Upon connecting a gateway to the testbed, an automatic test procedure starts, which stimulates the gateway under test, simulating all possible conditions as they are defined by the specification. After test completion, a clear response is issued. If all tests have been passed, the gateway can be certified for compliance, otherwise the incorrect behaviour clearly indicates where adjustments are needed.

Further future work will include the extension of test coverage both on the lower (TCN conformance test) and the upper levels (application specific definitions, like UIC 647).

TCN on existing UIC cable

To work properly, TCN requires a suitable transmission medium, included in new cables as prescribed in leaflet UIC 558. Unfortunately, this medium is not available on existing coaches. The retrofitting of all coaches will require many years and it is not economically feasible.

ROSIN has investigated a provisional solution that allows for train bus operation over the old 13 pole UIC cable. Even with lower performances, this solution makes it possible to have standard networked applications on board trains, which include coaches that have not yet been retrofitted.

For validation purposes, a real application was developed and implemented as a test system, dealing with the on board diagnostic functions, as specified in leaflet UIC 557. It includes three gateways representing a locomotive, a driving trailer and a passenger coach.

On the vehicle bus of each gateway, diagnostic stations generate messages while other devices simulate traction data traffic.

A typical TCN* composition includes 2 locos and 1 driving trailer, possibly with redundant nodes, and up to 11 coaches. Each coach includes a diagnostic station, connected to the others through the TCN network. If a fault is detected, an event is generated and stored in the local database. At the same time, a message is generated and sent to the other diagnostic stations on the train. The train staff can acknowledge the event by simply pressing a button in any of the diagnostic stations. The operator can then be informed about the event, in clear text.

The application was demonstrated on board a real coach using a Z1 type driving trailer, kindly made available by Italian railways FS.

Diagnostic events are notified to train staff, driver and maintenance personnel, according to their classification. The system can concentrate diagnostic data from all coaches for downloading. This is also useful for the development of the Romain system (see next chapter).

TCN* will be adopted in a number of new trains, allowing benefits and savings during the transition phase towards full TCN trains.

Freight Train Application

Within Europe, the railway market share for freight transportation is very low (about 16%) with respect to trucks (70%). This is a problem from both the economic and environmental point of view. New solutions are needed to increase the competitiveness of freight transportation by train.

TCN will help, offering new services, faster handling, and reducing costs.

Presently, freight vehicles are simpler than passenger ones: they have no autonomous power supply and often no connecting cables. Also, train composition changes more frequently. Moreover, freight trains include a higher number of vehicles, which brings the number of nodes and the total bus length in excess of TCN's current possibilities.

ROSIN solves these problems, but also demonstrates how TCN can support specific freight train applications required by railway operators, such as the automatic brake test.

The brake test is a frequent operation, which requires much time and human resources. ROSIN developed a prototype and tested it in the laboratory with a suitable testbed, including 16 nodes.

TCN features also allow for the automatic checking of train composition, orientation and integrity.

This demonstration shows how network technology can bring real economic benefits to Railways and promotes the use of freight vehicles.

Mass Transit Application

Mass Transit applications put specific requirements on the on-board system, due to vehicle typologies, train configuration and operational procedures.

ROSIN investigated a methodology to allow the building of a complete working system, which integrates devices from different suppliers and shows their interoperability on the TCN bus.

The target was to equip a Mass Transit unit, kindly made available by the Basque railway operator Eusko Tren. Different subsystems were needed: MMI (man-machine interface), door controllers, a number of connections to sensors and actuators (input-output) and some networking devices (bus administrators).

All the devices were linked by means of the TCN vehicle bus (MVB): no train bus was required here.

Six ROSIN partners closely co-operated in the system integration:

- Adtranz provided the MMI in the motor coach cab
- Alstom provided a door controller
- CAF provided the command CPU
- Siemens provided several I/O modules
- Ansaldo provided a second door controller
- Firema provided a second MMI in the trailing coach cab

The tough co-ordination task and overall responsibility was taken on by CAF. For each device, two units were needed, one for the system in the laboratory and one for the installation on the real train.

First, the complete system was set up in the laboratory where integration, configuration and testing of the devices could be better carried out. Then, the on-board installation was begun, to check problems in the real environment.

Finally, the system was able to perform an automatic startup procedure, controlling the different devices through the vehicle bus: check battery, raise the pantograph, close circuit breaker, check static converter and brakes, close the doors and start moving.

During the project, it was decided to include a TCP/IP to TCN gateway and a GSM link in the system so as to have a joint demonstration of the ROMAIN application. This application will be presented in more detail in the next chapter.

Standard Application Interface and ROMAIN

The aim of the *Work Package 4 (WP4*) of the ROSIN project was to develop a remote monitoring tool for trains that supports maintenance work. The user requirements for such a monitoring tool were:

- **Universal and ubiquitous access**: a user of this monitoring tool, typically maintenance staff, should be able to access train data from anywhere at anytime.
- **Low-cost**: For getting a wide acceptance in liberalizing markets this is one of the most crucial requirements.
- **On-line documentation**: finding the right documentation for train equipment is currently a time consuming task for maintenance staff. The monitoring tool should address this problem.
- **User friendly interface**: the user interface should be simple and easy to use, and as intuitive as possible. Textual and graphical views of the train data should be available.

This monitoring tool does not address direct **control** of train equipment. For that purpose there exist control networks which satisfy real-time and high security constraints, e.g., TCN.

The RoMain Application

Most of the user requirements can directly be fulfilled by the proper use of Internet technologies.

- **Universal and ubiquitous access**: the Internet communication infrastructure brings universal access to the information. Moreover, there is no need to install proprietary client software to access this information, or more exactly client software is transparently installed by the web browser. The information is accessible from anywhere at any time from a standard web-browser.
- **Low-cost**: it is mainly achieved by: (i) the use of the already existing Internet communication infrastructure; (ii) the use of a web browser as a frame for the *GUI*; this greatly reduces the development cost of this monitoring tool because there is no need to develop different *GUIs* for each different client platform; (iii) the use of *COTS (commercial off the shelf)* software based on Internet

technologies.

- **On-line documentation**: this can be solved by the use of hyperlinks to the documentation. In this way, train equipment manufacturers can maintain their documentation up to date, typically in their own servers. Maintenance staff can always access the latest versions of such documentation.

- **User friendly interface**: people are increasingly used to browsing the web for information, reading e-mail and so on. The use of a web browser as a GUI does not mean that it is automatically user friendly, but at least with the web browser the framework is familiar.

Therefore, the *ROSIN WP4* developed a web-based monitoring tool for trains, called *Railway Open Maintenance tool*, or ROMAIN for short. The idea is to get access, through the Internet, to current train data, using nothing but a standard web browser. This is shown in Figure 2.

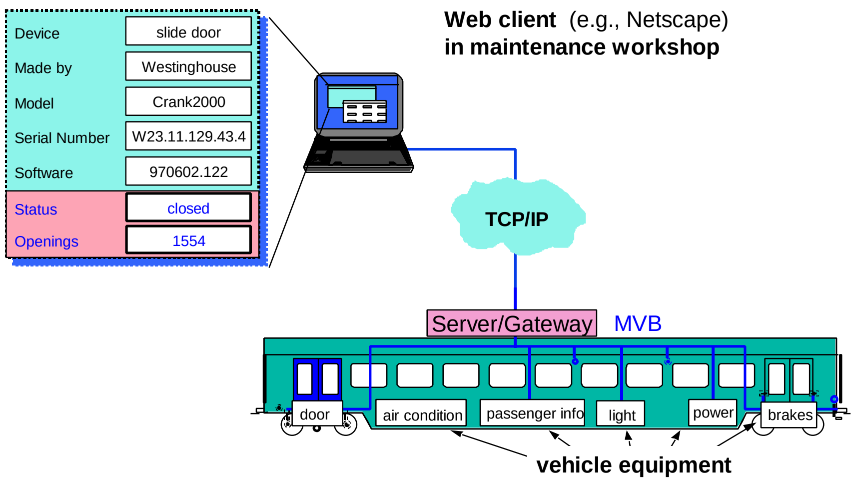

Figure 2: The ROMAIN Application

The kernel of this tool is a data acquisition system which is installed on-board a train. This data acquisition system is based on the experience of the *GLASS* project [3] at ABB Corporate Research, which worked on a general purpose monitoring system based on Internet technology.

ROMAIN covers all the main functional services of a data acquisition system, as defined by the *OMG Data Acquisition from Industrial Systems RFP* [1]:

- **Discovery of Remote System and Device Schema:** it offers mechanisms for discovering accessible remote devices and measurements.

- **Data Access / Retrieval**: it includes mechanisms for retrieving data immediately upon request.

- **Defining Data Access Requests:** the set of data to be requested is predefined by the full set of properties of each device.

- **Event Notification for Availability of Data:** it enables clients to subscribe to a notification service, which supervises selected data. Whenever these data become available or change over time all subscribers are notified.

- **Event Driven Data Upload:** data is automatically delivered when notification for the availability of data is given.

Moreover, portability was achieved by the use of the *Java* programming language. As *Java* is independent of a physical machine, the data acquisition system can be installed on any platform running a *Java Virtual Machine (JVM).*

The RoMain Prototypes

In order to investigate different technologies three different prototypes were developed. Demonstrations of these prototypes can be found at http://icapc62.epfl.ch/romain/. The prototypes are described in the following sub-sections.

Monitoring of a Single Device on a Train

The architecture of the system is shown in Figure 3. A dynamic Java applet displays a graphical schema, which represents an air-conditioning system. This applet requests data from a server through *HTTP* requests. A *CGI* application running on the server gathers the current data from the data acquisition system. This system periodically gathers the data from an air-conditioning simulation server. The *CGI* application sends the current data back to the client. The client applet reads these data, and presents the current state of the air-conditioning system graphically.

The remote train station is usually reachable by means of a wireless network, in our case by GSM. A ground station is responsible for transparently establishing a TCP/IP connection to the remote train station via this wireless network. As the bandwidth of GSM is still relatively low, the download of the monitoring *HTML* page, plus the download of the associated monitoring applet, is slow. In order to improve the performance of this download, we investigated an alternative: The monitoring *HTML* page and the applet were moved from the remote train station to a ground based web server, in our case to the ground station. The train station then becomes a pure data server.

This prototype shows a fast-to-develop and elegant way to graphically monitor a single equipment. The main problem with this approach is that the protocol between client and server are proprietary and not even published. This lack of openness does not allow to do anything but displaying the data graphically.

A demonstration of this prototype can be found at http://icapc62.epfl.ch/romainaag/. The air-conditioning data is coming from an air-conditioning simulation server running on the same machine.

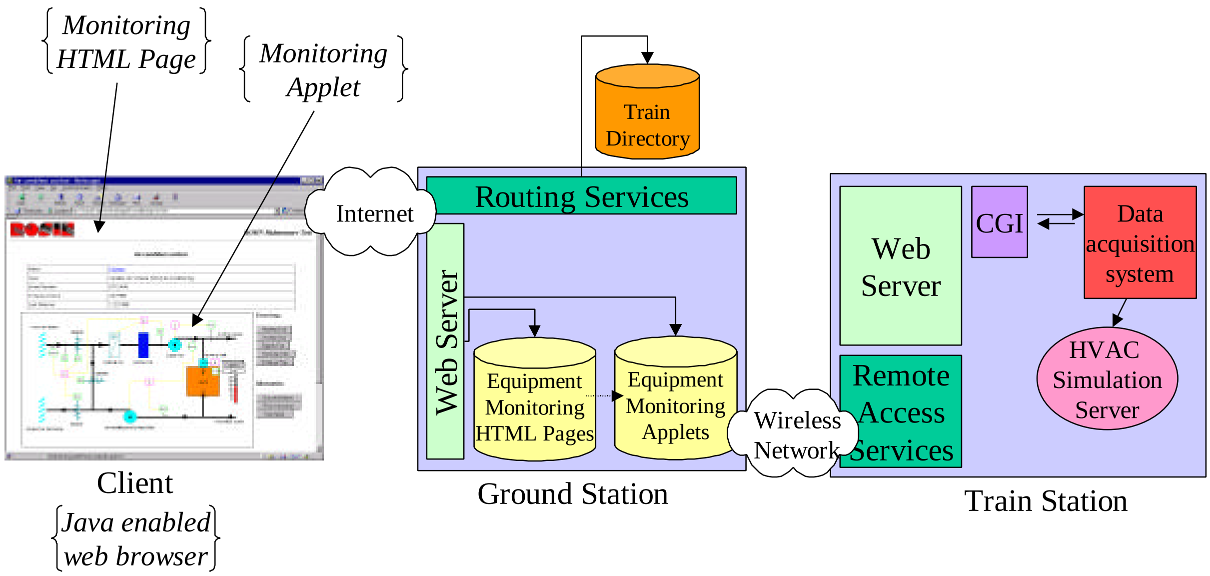

Figure 3: Monitoring of a Single Device (Air-Conditioning System) on a Train

Monitoring of all Devices on a Single Train

The main goal of the second prototype was to specify an application programming interface (API) for data acquisition of train equipment. The API defines the interface between client and server, and it therefore allows to implement new client applications. As the API is object-oriented, and as the applications are distributed we had a choice between different middleware products, as DCOM, Corba, or JavaRMI (Java Remote Method Invocation). We opted for the latter as we strived for a 100% pure Java solution, and built the data acquisition system as a *JavaRMI* server. It offers remote interfaces for, e.g., discovering train configuration, and accessing train, vehicle, and equipment data. Based on the API we developed a client that uses these remote interfaces to display the current state of a train within a web browser.

Two different updating mechanisms were implemented based on pull and push technologies. Using push technology the update of data is triggered by the server if and only if there are changes of that data. Hence a GSM connection is only open during notification subscription and notification, even if there is an arbitrary long time span in between.

In this prototype we also investigated security issues. As the user must grant the client applet certain privileges for stepping out of the sandbox (the restrictive security policy implemented by the browser), he must be able to check whether this applet is trustworth or not. Therefore, the developer of the applet must sign it with a digital signature obtained from a certificate authority as Verisign. We use the same technology as it is used to make e-commerce applications more secure.

The main problem with the second prototype is the problem of accessing data for a client behind a firewall. This problem can be partially overcome with *HTTP tunneling*, but this slows down the communication and the use of server side push technology is no longer possible.

The architecture of the system is shown in Figure 4. As for the first prototype, a ground station is used in order to get access via a wireless network to the train remote station. Again, the monitoring *HTML* page and monitoring applet are stored on a web server on the ground station.

A demonstration of this prototype can be found at http://icapc62.epfl.ch/romainjava/. The train station is connected to a train network installed in a laboratory at the *CAF* facilities in *Beasain (Spain)*. This train network is exactly the same as the network installed on a real train for a demonstration in *February'99*. The data is collected from real devices, which are interconnected via the *Train Communication Network (TCN)*. The ground station is installed at *ICA* (an institute of *EPFL* in *Lausanne)*. The only difference with the real demonstration is that the connection of the ground station with the train station is not done by a wireless network but by an Internet wired connection. This is done in order to reduce costs. However, this would be totally transparent for any client of the application.

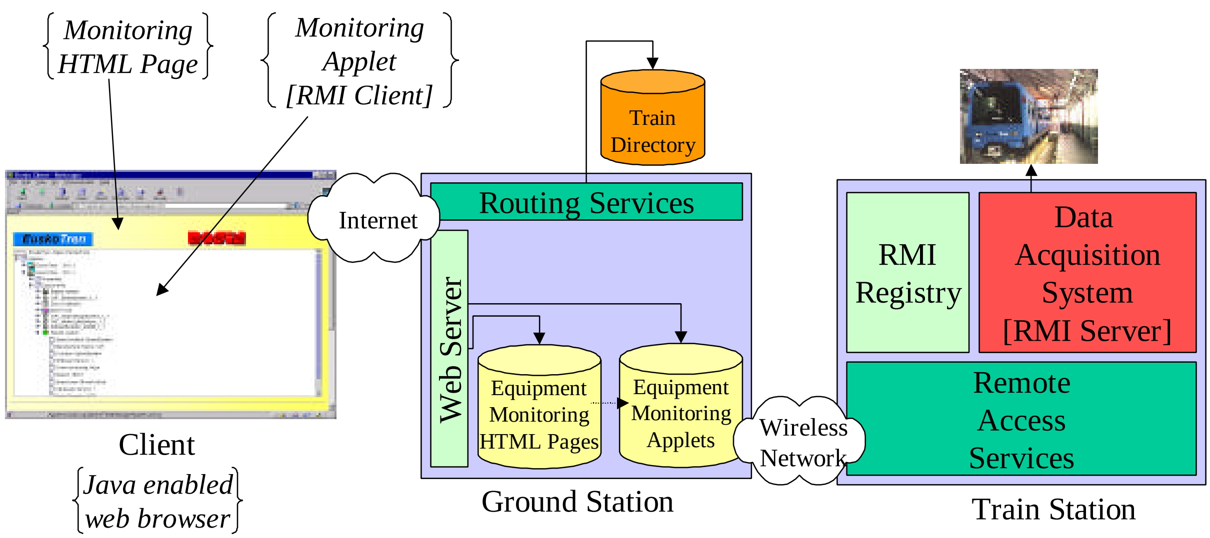

Figure 4: Monitoring of all Devices on a Single Train

Monitoring of all Devices on a Fleet of Trains

In the third prototype we implemented a system with a three-tier architecture. First, to overcome the problem we had with the second prototype, but mainly to investigate how data coming from an entire fleet of trains can be integrated. This would allow a component manufacturer to supervise, say all door controllers, regardless on which trains they are.

We investigated technologies that allow the client, e.g., to choose among different views, and to receive data combined from train stations on a single page. Choosing among different views means that we need a way to separate what is data and what is presentation format. Combining data from different sources means that we have to parse the data and create new data.

All these features are easily implemented by the use of *XML (eXtensible Markup Language). XML* is the de facto standard for exchanging documents over the web. As *XML* also allows to define new domain specific markup languages, data can be transmitted together with some metadata that describe them. This makes it easy to parse, and combine *XML* documents. Moreover, the presentation format can be added by means of a separate *XSL (eXtensile Style Language)* file, which defines how an *XML* document should be displayed. Therefore, it is easy to implement different views of the same *XML* document by providing different *XSL* files.

The architecture of the system is shown in Figure 5. It is a three tier architecture composed of: (i) a *Java* servlet on-board a train that gathers data from the *JavaRMI* server developed for the second prototype and that replies to *HTTP* request of data with *XML* documents giving the requested information; (ii) a middle tier that receives data from different sources in *XML* format, combines them into a single *XML* document and adds the style sheet corresponding to the client view; (iii) and the thin client, which is like in the previous prototypes, a web browser. Note that in the first two prototypes the applet also is loaded from the ground station, but after that it connects directly to the data server on the train, that is in the first two prototypes we had a two-tier architecture.

As in the previous prototypes, a ground station is used in order to access, via a wireless network, the remote train station. In this prototype there is no monitoring *HTML* page or applet. Instead there is a document server on the middle-tier, which is responsible for integrating data from different train stations and for adding the corresponding style sheet to make the output readable by the client. In our case the entire middle-tier is on the ground station.

This architecture is scalable, extensible and adaptable for future evolution. It runs over firewalls and the performance is higher than the solution based on object-oriented middleware. The main drawback is that it is not possible to use server side push technology.

A demonstration of this prototype can be found at http://icapc62.epfl.ch/romainxml/. The train data source is the same as in the demonstration of the second prototype. The ground station is installed at *ICA*. The communication with the train station is done by means of an Internet wired connection.

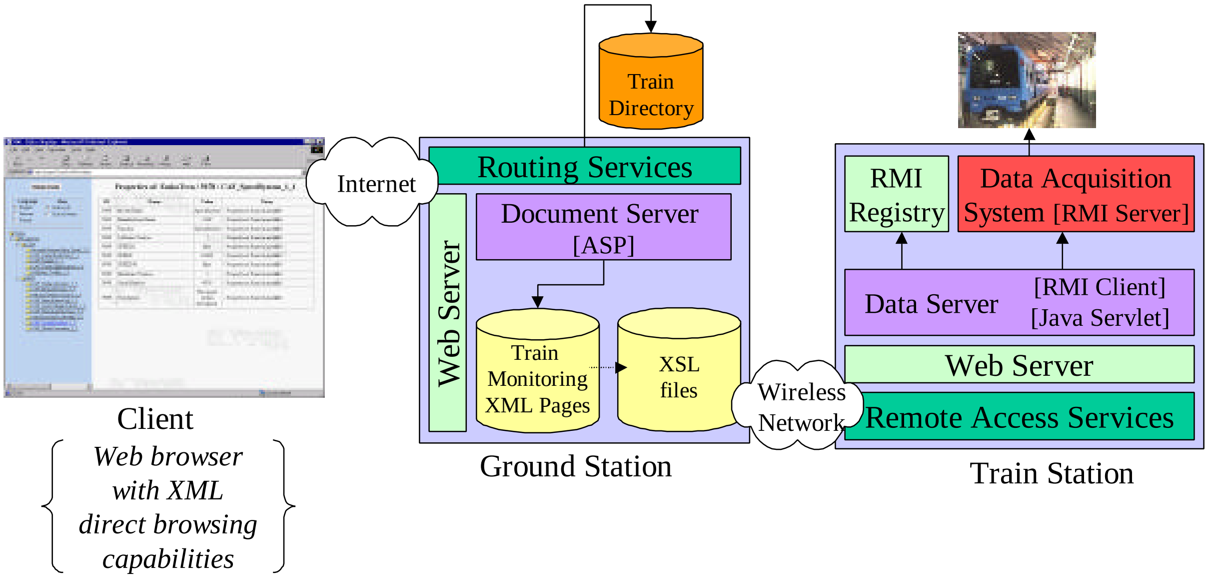

Figure 5: Monitoring of all Devices on a Fleet of Trains

The RoMain Demonstration

The second prototype was presented at the joint demonstration with the mass transit application as described in the previous chapter. We used the ROMAIN application to monitor a mass transit unit, which was composed of two vehicles equipped with equipment from heterogeneous manufacturers. The setup of this train is shown in *Figure 6*.

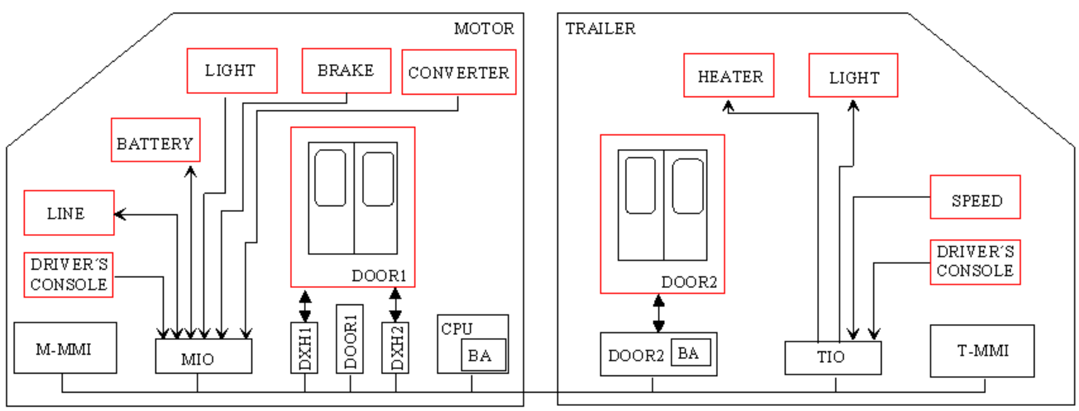

Figure 6: Mass Transit Unit

We installed the train station, a normal *PC* running *Microsoft Windows NT 4.0 WorkStation*, on-board the mass transit unit.. The train station was connected simultaneously to the train network and to an industrial *GSM*. It ran the data acquisition system to gather data from the train network, and a remote access server supporting the Point to Point Protocol *(PPP)* to be reachable remotely via *GSM*. The ground station, a normal *PC* running *Microsoft Windows NT 4.0 Server*, was installed at the *CAF* facilities. It was connected simultaneously to a wired Internet connection and to a modem and a phone line. It also ran a remote access server supporting PPP to transparently establish a TCP/IP connection to the train station. The client was another *PC*, at the *CAF* facilities, connected to the Internet and running *Netscape Communicator 4.5*, but it virtually could have been any PC connected to the Internet.

Conclusions and Remarks

Cost-effective maintenance of globally distributed devices is an important issue in industrial applications. The main conclusion from the ROMAIN project is that this goal can be achieved by using Internet technology. Internet technolgy here means, the world wide TCP/IP network, protocols as HTTP and PPP, markup languages as HTML and XML, Internet browsers capable of downloading and running Java applets, web servers and application servers, architectural patterns reused from business applications, Java, which is not only a programming language, but also a vast collection of libraries.

This article does not explain how the train station automatically detects the installed devices. The information modeling of the devices, necessary for such a *plug&play* architecture, is discussed in more detail in [2].

Acknowledgments

The authors would like to thank the Basque railways – *EuskoTren* - for making available a mass transit unit for the *RoMain* demonstration. And to the Italian railways – *FS* – for making available a trailer for the *"TCN on existing cable"* demonstration. Special thanks to all the *ROSIN* partners for the great collaboration and the efforts made during the whole project.

Acronyms

API: Application Programming Interface

CGI: Common Gateway Interface

CORBA: Common Object Request Broker Architecture

COTS: Component Off-The-Shelf

CPU: Central Process Unit

DCOM: Distributed Component Object Model

GSM: Global System for Mobile

JVM: Java Virtual Machine

MMI: Man Machine Interface

MVB: Multifunction Vehicle Bus

PPP: Point to Point Protocol

RMI: Remote Method Invocation

TCN: Train Communication Network

XML: eXtensible Markup Language

telecommunications

GUI: Graphical User Interface

HTML: HyperText Markup Language

HTTP: HyperText Transfer Protocol

XSL: eXtensible Style Language

References

[1] Objet Management Group. *Data Acquisition from Industrial Systems - Request For Proposal*. OMG Document: dtc/99-01-02. January 15, 1999. http://www.omg.org/techprocess/meetings/schedule/Data_Acquisition_RFP.html

[2] T.Nieva. *Automatic Configuration for Remote Diagnosis and Monitoring of Railway Equipments*. Proceedings of 17th IASTED International Conference - Applied Informatics, Innsbruck, Austria, February 1999. http://icawww.epfl.ch/nieva/thesis/Conferences/ai99/article/ai99.pdf

[3] R.Itschner, C.Pommerell and M.Rutishauser. *GLASS: Remote Monitoring of Embedded Systems in Power Engineering*. IEEE Internet Computing, Volume 2, Number 3, May-June 1998. http://www.crc.ch.abb.com/chcrc/highlights/orbit/ieee-glass.pdf

[4] Paolo Umiliacchi. *The role of European research in the railways modernisation process: the ROSIN project* WCRR '97, Florence (Italy), 16-19 November 1997 http://www.labs.it/rosin/docs/wcrr97.pdf

[5] Salvatore Longobardo, Paolo Umiliacchi. *Developing a standard train communication network to interconnect all on-board services and devices: the ROSIN project* Telecom Rail '98, London (UK), 7-8 December 1998 http://www.labs.it/rosin/docs/TelecomRail98.pdf

[6] G. Fadin, H. Kirrmann, P. Umiliacchi *ROSIN - Web Technologies for railways* Proceedings of Automation in Transportation '98 - KOREMA - 25-28 November 1998 - Stubicke (Croatia), Gratz, Wien (Austria) http://www.labs.it/rosin/docs/rosinwt.pdf