Case Study

ILTIS- Railway Traffic Control

Siemens Switzerland

Transportation Systems

About Siemens
Siemens supplies leading-edge solutions in information & communications, automation & control, power, transportation, medical and lighting.

Siemens Transportation Systems offers carefully considered and efficient solutions for all transport segments. Its technical expertise and its experience have been proven all around the world. Siemens finds tailor-made and efficient solutions for every requirement in the interests of our customers.

About ILTIS rail automation
ILTIS is a centralized traffic control (CTC) system developed initially for the Swiss National Railways (SBB). The demand for rail automation has grown worldwide in recent years. The ILTIS control and information system enables efficient automatic operations management for main-line and suburban services, as well as private railways. It covers the entire field of operating-orientated railway control, from the local or remote control of individual interlockings to the automation of entire rail networks. ILTIS guarantees safety-critical operator control of interlockings and a reliable display of the operating situation.

The challenge: Selecting a development language that meets requirements for safety, availability, maintainability, portability and scalability.

A CTC system needs to be able to fulfil certain safety-related functions. Indeed it has to ensure that the traffic controller does not make life-endangering decisions based upon faulty information presented to him by the CTC system. It must ensure that critical commands (i.e. the commands which by-pass the safety of an interlocking) cannot be inadvertently executed, and critical commands have to be correctly executed.

The **availability** of a CTC system is becoming an increasingly important factor as railway networks become more and more dependent upon them. Current systems usually tackle this problem by utilizing redundant computers, which can be manually switched into service whenever they are needed. In ILTIS, the necessary redundancy has been built into the software design because all ILTIS tasks can be executed on any available computer in the system. In this way, ILTIS uses a technique of redundant software instead of redundant hardware.

ILTIS was designed to be in service for 25 years. In 1990 (as now), it was impossible to predict how the target platforms would change. It was, therefore, important to choose a development language that would facilitate **porting** to different hardware platforms and operating systems.

In order to satisfy the expectation of a 25-year product life and the necessary related **scalability** and **maintainability** requirements, ILTIS needed an architecture, which would allow ease of extending functionality without degrading existing functionality.

Ada : The Language of Choice

Ada was predestined for ILTIS development. ILTIS is a Safety Integrity Level (SIL) 2 system and Ada is highly recommended for developing such systems. In 1990, it was clear that Ada should be chosen as the development language but even today, Ada is seen as an ideal language for safety critical computing.

Language selection was a priority. ILTIS had several concerns in dealing with the railway's requirements, a few of which were:

- ILTIS must be in service 24 hours a day, 7 days a week. No short breaks in service are permitted. However the redundant computers are not in active service. When a situation arises where they need to be brought into service, they themselves could have developed technical problems, which, as the redundant computers are dormant, have gone unnoticed.
- As each computer in the distributed system needs to have a similarly configured redundant computer, this effectively doubles the number of required computers if a full stand-by capability is needed.
- Switzerland has the densest railway-traffic in Europe. Therefore quality and reliability are also prerequisites for any Swiss CTC
- Since the ILTIS systems deals with 4 million lines of code, a system which offers ease of maintenance is required
- Development for new markets in other countries requires modifications in sensitive parts of the system and the selected language has to allow the adaptation to the local rail-traffic's methods.

With this in mind, ILTIS sought an implementation language that perfectly meets the constraints and requirements of the railway sector.

A Language for a Changing World

All of the software has been written in Ada and was originally targeted onto DEC's OpenVMS operating system using Alpha computers. Development was started in 1990 with each new release providing additional functionality and, at any one time, an average of 20 to 30 engineers is employed on enhancing the system.

Ada is a superior development language when safety is critical. It allows dealing with a high number of lines of code through its constructs that support the developer during required code modifications.

The target operating system for the first ILTIS installations (OpenVMS on Alpha platforms) is approaching the end of its life. Many customers also want to use the more economic Windows platforms for their installations. Because of these market forces, it was decided in 1998 to port ILTIS from OpenVMS to Windows.

ObjectAda : The Environment of Choice

ILTIS was aware in advance that Ada features would meet all of its requirements. A very important decision at this juncture was the choice of compiler and development environment for the Windows platform. After undertaking a comparative analysis of all available products on the market at that time, Aonix ObjectAda for Windows was considered the most suitable environment for ILTIS's needs, not least because of its compatibility with Microsoft developer-tools.

Migration was divided into two phases. The first phase encompassed the safety-critical functions together with their corresponding GUI (ILTIS uses a Motif-GUI for both operating systems). ObjectAda facilitated the relatively painless transfer from OpenVMS to Windows with its stringent static analysis capabilities highlighting many areas of potential problems which had failed to be picked up by the DEC compiler. The first phase of the migration was, therefore, not only completed within budget and on time, but was also delivered with improved software quality.

What is ObjectAda today?

Based on Ada 95, the first and only internationally standardized object-oriented programming language (ANSI/ISO), ObjectAda is a truly multi-lingual environment. ObjectAda allows you to easily integrate Ada components with components written in Java, C, C++, FORTRAN, and other languages for multi-lingual development. And ObjectAda works directly with commercial off-the-shelf libraries, components, and APIs.

For any platform, ObjectAda features a fast, open library model that is fully compatible with other languages, high-speed intelligent compilation, hyperlinked semantic browsers, syntax-directed editors, integration with configuration managers, and instant access to standard APIs. On all platforms, application generation is optimized for reliable, seamless execution of thread-aware applications within a safe and secure operational environment.

Beginning with the ObjectAda 8.2 family of product releases, ObjectAda allows developers to choose between the traditional Aonix IDE and the new AonixADT™ Eclipse plug-in.

One of the main features of ObjectAda for Windows is its interoperability with other languages and technologies (ObjectAda is fully compatible with C, C++, Java, and other languages, and integrates with all standard Microsoft APIs), thus developers can integrate work that has been done in other languages. These features are combined with an easy-to-use environment and efficient, reliable and optimizing compiler technology for a complete and well-rounded development environment.

AonixADT integrates an Ada-language sensitive editor, Ada-language compile and build capabilities, and a complete Ada debugger interface. AonixADT enables Ada developers to enjoy state-of-the-art interface capabilities geared to maximize developer ease and efficiency.

The ILTIS project continues to rely on ObjectAda for Windows

ILTIS under Windows has been successfully running since 2000 in various installations alongside its bigger OpenVMS brother, having been authorized for use in Switzerland by the Bundesamt für Verkehr (BAV) and in Austria by the Bundesministerium für Verkehr, Innovation und Technologie (BMVIT). ILTIS under Windows is scheduled for authorization by both the BAV and BMVIT in summer 2006 and will go into service in autumn 2006.

> *"ObjectAda is being used as a powerful tool in the migration of the final ILTIS-features and has already successfully demonstrated that the development environment using the command line interpreter scales up effortlessly to larger projects."*
>
> Neville Rowden, Siemens, Germany

North America
Phone: (800) 97-AONIX
Fax: (858) 824-0212
E-mail: info@aonix.com
www.aonix.com

Aonix

France
Phone: +33 (0) 1 4148-1000
Fax: +33 (0) 1 4148-1020
E-mail: info@aonix.fr
www.aonix.fr

United Kingdom
Phone: +44 (0) 1491 415000
Fax: +44 (0) 1491 571866
E-mail: info@aonix.co.uk
www.aonix.co.uk

Germany
Phone: +49 (0) 7243 5318-0
Fax: +49 (0) 7243 5318-78
E-mail: info@aonix.de
www.aonix.de