



Market Case Study

**Bringing Holland's
Energy Systems to the
Apex of Performance**

RATIONAL
SOFTWARE CORPORATION

EZH Systems and Rational Apex Ada

When it comes to innovation in energy generation and distribution, Electriciteitsbedrijf Zuid-Holland (EZH) is a powerhouse. One of four Dutch energy companies, EZH serves the electricity and heating needs of the economic and industrial heart of the Netherlands, including its capital city, The Hague, and the port city of Rotterdam. But with a European market moving quickly toward deregulation, EZH needed to increase the efficiency of its operations, while

Accomplishing this meant engineering the company's Energy Accounting System (EAS) to more accurately track every aspect of its operations, in real time. **Optimizing the EAS to maximize efficiency and profitability became a key element in the company's strategy for success. To accomplish their goal, EZH**

chose Rational Apex Ada, Rational Software's integrated software engineering environment optimized for development and maintenance of complex Ada systems.

Since 1996, when the European Union passed its directive on deregulating electricity markets, energy companies throughout Europe have been faced with the prospect of maximizing service quality and availability, while decreasing costs to remain competitive. EZH was quick to respond to this trend, recognizing the strategic importance of Energy Accounting Systems (EAS) in controlling and optimizing production costs. According to EZH management, a high-performance EAS is essential to remaining competitive in a deregulated market.

Utilities like EZH must be able to constantly monitor and calculate the amount of power being produced and delivered, and to whom. This information must be recorded and processed in such a way that it is easily and rapidly accessible. This data becomes the framework for billing, load forecasting, generation scheduling, regulation compliance, and other critical business processes.

EZH Systems of Hawthorne, N.Y., EZH’s wholly owned technology subsidiary, was given the task of engineering the EAS to meet the challenges of today’s marketplace.

Innovative Methodology

“The issue we faced was how to migrate at a reasonable cost from older real-time systems and networks to modern technology,” says Leo Verbeek, CIO of EZH and President of EZH Systems. “In developing and maintaining software for the EAS, the challenge for us is controlling complexity in a constantly changing environment. Our systems must provide 100% availability, despite changes to underlying hardware or subsystems.”

To meet this demand, the EZH Systems team created a Generic Real-Time Network (GRTN) methodology for developing mission-critical, real-time, high-reliability systems. The GRTN provides an approach for integrating telecommunications, computer systems, hardware and software that focuses on reliability, expandability and time-critical behavior of the EAS. The methodology supports a wide variety of transmission media and is compatible with existing systems.

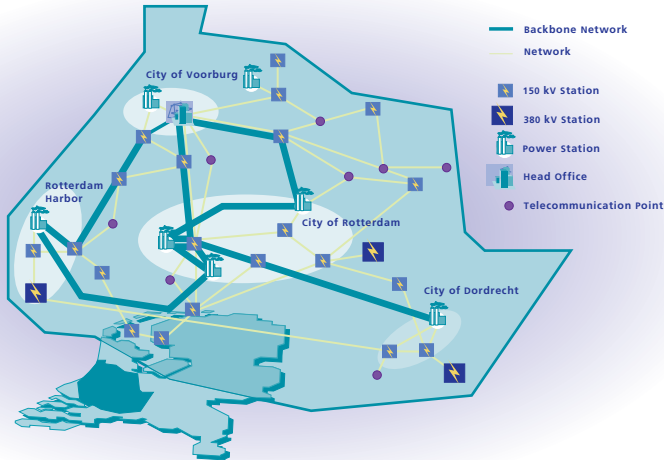
“We chose Rational Apex Ada, in conjunction with Rational’s SoDA and TestMate products, to support the critical needs of the GRTN methodology in developing our EAS,” says Reggie van Leeuwen, Leader of the Technology Support Group for EZH. He noted that Rational Apex Ada and Rational Subsystems enabled EZH engineers to create and share code for their new methodology. EZH developers used Rational

TestMate and SoDA to test and document the models and designs that formed the basis of the EAS.

“We chose Rational Apex Ada, in conjunction with Rational’s SoDA and TestMate products, to support the critical needs of the GRTN methodology in developing our EAS.”

—Reggie van Leeuwen

Figure 1



A Powerful Challenge

Due to the unique market conditions in the Netherlands, reliable real-time EAS performance is absolutely critical for EZH.

“In Holland, electricity production is divided among four utilities, with the National Control Center determining how much electricity each is allowed to produce. Exceeding these production allotments can result in considerable fines,” explains Reggie van Leeuwen. “Lost or misinterpreted data could result in tremendous financial losses to the utility.”

Complying with these production limits creates a substantial technology challenge. Every 60 seconds, EZH must gather power measurements from 24 different locations spread across southern Holland and send this data to EZH’s control center in Voorburg, where it is collected. From here, data on the utility’s total load is sent to the Dutch Energy Control Center, which totals the figures for all four state utility companies and sends an updated figure on total national load back to each utility to ensure production compliance.

“This process happens once every 60 seconds and the utilities must provide their production values within a 14-second window,” van Leeuwen says.

A Flexible Solution

To respond to these automation challenges, the EZH team developed an EAS infrastructure (see Fig. 1) utilizing the company’s existing GRTN backbone network (see Fig. 2) to reliably transmit measurement data in real time from the company’s far-flung substations to the EZH control center.

Each EZH substation has its own local digital network consisting of two redundant rings to which the kilowatt/hour meters are connected. These networks are connected to the GRTN-network via a redundant link (see Fig. 3). The GRTN-network is in turn connected to EZH’s enterprise computer network, offering access to the data by users across the company. The GRTN-network also has a dedicated, fault-tolerant link to the Dutch Energy Control Center.

The nodes of the GRTN-network are synchronized using the radio signal of the European atomic clock. The data is also transferred redundantly with signaling of lost data packets. Each minute, when a new measurement is sent over the network, 15 buffered previous values are re-sent along with it, ensuring all data is sent 16 times over a period of 16 days. This prevents loss of information in the event of a data storage failure.

Application-specific logic is implemented through the system’s mission-critical nucleus. In designing the EAS, care was taken to isolate any system dependency in separate software modules. This enables specific parts of the nucleus to be modified to keep pace with changes in the environment (e.g. adding a substation, phasing out old equipment, etc.), without bringing the system down.

“As utilities replace their existing equipment with new technologies, modification of the mission-critical portion of the system must be a manageable process,” says Joe Marasco, Director of Rational Software’s Programming Environments business unit. “Modifications must be implemented systematically, at the lowest cost and with the lowest possible risk of error. The GRTN methodology, in combination with Rational Apex and other Rational software development tools, simplifies system modification, while effectively combining equipment of various types and ages.”

The EZH team developed the new EAS system with Rational Apex Ada on an IBM RS/6000. The target systems were Digital VAX stations running VMS and DEC Ada. Because system-specific parts are isolated, the software can be re-targeted easily if, for any reason, another target system is desired. The connection between the GRTN-network and the enterprise network is accomplished with stand-alone VAX/ELN computers also programmed in Ada. The protocol on the enterprise network is DECnet. A small part of the software is in FORTRAN, which consists of the library of standard data conversion routines for the European Elcom standard. This standard was created for data exchange between electric utility companies.

Building in Quality

Minimizing software bugs was a high priority throughout the development process. To accomplish this, the GRTN methodology makes extensive use of modern software practices. Software designs are based on a single software architecture tailored for real-time, distributed systems design. This architecture offers some important advantages, including:

- The highly disciplined standards of the architectural method give developers guidance in making their work reliable and extendible. Developers can leverage previous uses of the method as a knowledge base, improving quality.

- Development of large systems is always a team effort, and the GRTN method acts as an effective means of communication among developers, improving design quality and integrity.
- The method makes extensive use of large-scale software reusability.
- The software architecture recognizes 22 categories of software modules. These modules are designed to keep application-dependent and an application-independent parts as loosely coupled as possible. Reusable software modules implement parts that are application-independent, ensuring a high degree of reliability.
- The GRTN resource concept separates the hardware-dependent part from the application-independent part of a software design. Machine dependencies are minimized and isolated, enabling developers to make designs that are nearly independent of the underlying hardware platforms. This makes GRTN software designs extremely flexible.

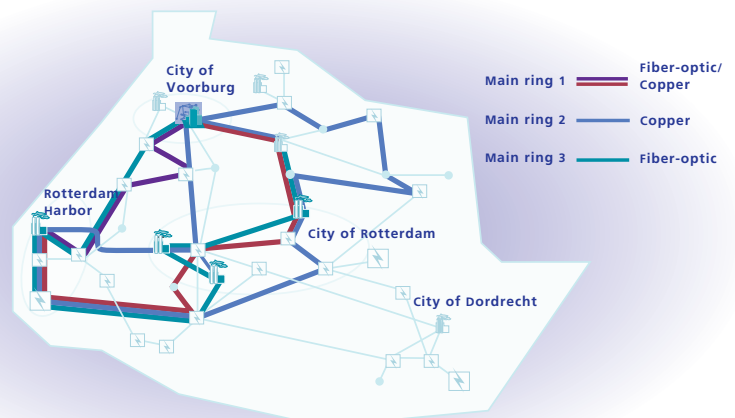
In implementing the GRTN, van Leeuwen says the EZH development team draws heavily from use-case concept of Rational's Objectory Process.

"Keeping the GRTN implementation of the EAS in sync with the ever-changing system specification, while assuring quality, is a major challenge," he says. "To achieve this, we employ use cases extensively, as a kind of 'red thread' cutting through all development phases."

"As utilities replace their existing equipment with new technologies, modification of the mission-critical portion of the system must be a manageable process."

—Joe Marasco

Figure 2



Ada Benefits

In developing their state-of-the-art EAS, EZH Systems chose to program in Ada, an approach that offers important advantages.

"We felt that Ada would provide us with a high degree of reliability and the confidence level that our mission-critical systems require," Verbeek says, noting that readability and maintenance were other key factors for choosing Ada. "Because the system must be maintained over the course of 15 to 25 years, it will likely not be maintained by the original developers." Verbeek noted that Ada's strong typing and segregation of module-interface and implementation simplifies the process of integrating modules written by different people.

Ada's exception mechanism provides an excellent way for incorporating error-handling code without reducing readability, while saving many lines of code otherwise needed for condition testing. The tasking facilities are indispensable because redundant data buffering and processing must take place concurrently.

"Ada also offers improved development productivity and excellent support for reuse," van Leeuwen says. "In developing our EAS, we were able to use a substantial amount of code from previous projects, and produce some new reusable code, as well."

The Apex Advantage

While programming in Ada according to the GRTN methodology offered the EZH team important advantages, the challenges of day-to-day development demanded a toolset as innovative and flexible as the methodology itself.

"We chose Rational Apex Ada for a very simple reason: It provided the best tools for the job," EZH's Verbeek says, adding Rational Apex Ada and its layered products enabled the EZH development team to:

- Enforce the software architecture, without restricting developers' freedom with unnecessary constraints
- Provide a library of system components upon which developers can base their work
- Use an integrated library of source code to generate code for different platforms
- Achieve rapid turn-around on new development

Throughout the development process, the EZH team enjoyed the following key Rational Apex advantages:

Single Development Process

The information model of the Rational SSE is based on a fine-grained, integrated, persistent program representation which captures and manages information about the software programs in development. The development tools that form the environment use this same information, giving them a high degree of integration and interoperability. It also supports the incremental compilation feature, enabling faster production by preventing long compilation times after source code changes.

Configuration Management and Release Management

Because the Configuration Management (CM) system is integrated with the development environment, developers cannot forget to make the necessary decisions for it. Configuration information is automatically updated, giving developers insight on the current state of development. When a software system is released, this is also registered in the CM database and kept under control. As a result, developers can always find out which components are used in a released software system, and how and when they were made.

Library Management

GRTN reusable software components are kept as libraries of Rational Subsystems, which are readily available to each developer. Libraries consist of thoroughly tested components and are kept under strict configuration control. This makes it easy for developers to find needed components, improving quality.

The process of transferring the right source code versions to the target platform is completely automated, resulting in higher productivity and preventing incorrect combinations of source code files.

Testing Environment

All component and completed design testing is performed in TestMate, a structured environment which standardizes testing. All testing activities are registered in the central information-repository, offering developers a complete overview of the reliability of both software components and entire designs. Registration of test information also offers future advantages, enabling more rapid testing of components when changes are required.

Documentation Generation

While documenting development processes, models, designs, architecture, and code is critical, documentation is time-consuming to write and difficult to keep up-to-date. Rational's SoDA product documentation generator automates the process of creating and maintaining documentation. SoDA extracts information from the central repository of code and configuration management to produce an initial document. Users add descriptions of the extracted information to the document and, as the project information changes, SoDA is used to refresh the document.

By automatically producing consistent documentation at each step of the project, SoDA ensures documentation is always current, saving many hours of review time. SoDA's flexible approach to automating documentation enables customized document generation in accordance with the GRTN software development methods and demands.

Key Contribution

As the European energy marketplace transitions to a competitive environment, efficient information management isn't merely a convenience; it's an essential ingredient for success. EZH's state-of-the-art EAS enables the effective and reliable integration of energy accounting resources, while enabling the real-time data access and transmission that enable regulation compliance, as well as profitability.

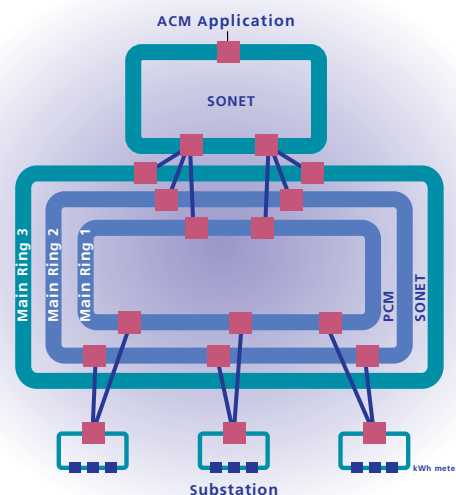
With Rational Apex Ada, the EZH Systems development team was able to implement its innovative GRTN technology to create an EAS with the capacity, flexibility and availability to meet the needs of a demanding energy marketplace.

"Without Rational Apex, our advanced Energy Accounting System would have taken considerably longer to complete," EZH's Verbeek says, noting that the EZH development team continues to use Rational Apex Ada for maintenance of EAS applications, including configuration-management activities. "The quality of the code, the ability to clearly express the design in subsystems, automatic enforcement of the GRTN requirements, and support of activities like testing and documentation generation made Rational Apex a key contributor to the project."

"We chose Rational Apex Ada for a very simple reason:
It provided the best tools for the job."

—Leo Verbeek

Figure 3



For more information on Rational Apex, or the full line of advanced Rational development tools and solutions, contact Rational Software at 800.728.1212 or www.rational.com.

Rational Software Corporation

18880 Homestead Road
Cupertino, CA 95014-0721

Toll-free: 800.728.1212 (US and Canada only)
Tel: 408.863.9900
Fax: 408.863.4120
Email: info@rational.com
Web site: www.rational.com

International phone numbers

Australia: +61-2-9419-8455
Belgium : +32-16-46-24-11
Brazil: +55-21-84-1841
Canada: 613-599-8581
France: +33-1-30-12-09-50
Germany: +49-89-613-7690
India: +91-80-553 8082/9864
Japan: +81-3-5423-3611
Korea: +82-2-556-9420
The Netherlands: +31-23-569-4300
New Zealand +64-4-568-9591
Sweden: +46-8-703-4530
Taiwan: +886-2-720-1938
UK: +44-1344-462-500

Rational, the Rational logo, Rational Apex, SoDA, and TestMate are trademarks or registered trademarks of Rational Software Corporation in the United States and in other countries. All other names are used for identification purposes only and are trademarks or registered trademarks of their respective companies.

© Copyright 1997 by Rational Software Corporation. D-180; effective 12/97.
Subject to change without notice.

RATIONAL
SOFTWARE CORPORATION