



Leading provider of telecommunications equipment calls in IBM and IBM Advanced Business Partner Pathfinder Solutions to help improve code quality.

Overview

■ **Challenge**

A world-leading provider of telecommunications equipment sought to advance its competitive position and improve the quality of its embedded systems by applying development best practices, improving communication, reducing defects and increasing developer productivity.

■ **Solution**

The company worked with IBM Advanced Business Partner Pathfinder Solutions to complete a pilot project using IBM Rational Systems Developer and Pathfinder Solutions PathMATE software. The two products enable the team to build, execute, debug and test models of embedded systems in a single environment and at a high level of abstraction.

■ **Key Benefits**

Following the success of the pilot project, the company expects to break through the code-level developer productivity barrier with more widespread adoption of Rational Systems Developer and PathMATE software. In addition, the simplicity and flexibility of the models, coupled with advanced model transformation technology, have improved code quality.

More than a third of all mobile calls are made through a single company's systems. That company, a global provider of telecommunications equipment and related services to mobile and fixed network operators, controls more than 1,000 networks in more than 140 countries.

As part of its ongoing efforts to improve quality, increase product development efficiency and drive innovation, the company recently engaged IBM Advanced Business Partner Pathfinder Solutions. Pathfinder Solutions provides embedded software developers with methodologies, tools and services required to advance their development processes and gain competitive time-to-market and product quality advantages.

The telecommunications company's embedded system developers had been using IBM Rational® Rose® Technical Developer software for several years to accelerate model-driven development with the Unified Modeling Language (UML). Looking to build on this success, the company asked Pathfinder Solutions to work on a pilot project to evaluate IBM Rational Systems Developer and Pathfinder Solutions PathMATE software. Rational Systems Developer is an Eclipse-based design and development solution that enables developers to create well-architected C/C++, Java™/Java Platform, Standard Edition (Java SE), and CORBA-based applications leveraging version 2 of the UML. Fully integrated with Rational Systems Developer, PathMATE is an open environment for developing, debugging and deploying embedded systems from platform independent models (PIMs).

Leading provider of telecommunications equipment calls in IBM and IBM Advanced Business Partner Pathfinder Solutions to help improve code quality.

Key Components

Software

- *IBM Rational ClearCase*
- *IBM Rational Rose Technical Developer*
- *IBM Rational Systems Developer*

“For our customer, Rational Rose Technical Developer has been very effective in the embedded realtime space. For some of their products, they used it for virtually the entire effort. They have developed millions of lines of code with it very successfully,” explains Peter Fontana, chief technology officer (CTO) at Pathfinder Solutions. “While they are still using it very heavily, they are also looking for a way to further improve quality and productivity by elevating the level of abstraction in their modeling and development efforts.”

Defining goals

Pathfinder Solutions teamed with a group of developers that primarily worked on embedded software for communications between the radio base station (RBS, or cell tower) and the main telephone network. Previously, these systems had all been implemented in C++ using Rational Rose Technical Developer.

Fontana recalls, “While the models developed with Rational Rose Technical Developer were a very helpful step above coding systems by hand, these models were still very much at the level of C++, and exhibited many of the productivity and quality limitations at this level.” As a result, the development team wanted to elevate the level of abstraction from a code-oriented model to a more abstract model focused on a problem space. Using Model Driven Architecture (MDA)—a powerful set of Object Management Group (OMG) standards that includes the UML—the team wanted to facilitate platform independent modeling. This approach allows developers to build models that are above the level of implementation languages yet still complete and executable. These models are much simpler than code-driven models, and they enable substantial gains in productivity, quality and manageability.

At the same time, the company was also seeking a standardized yet flexible development environment that it could use to integrate many aspects of the development process, including modeling, coding and software configuration management. “They wanted to move to Rational Systems Developer and the Eclipse environment—for all the right reasons,” notes Fontana. “They liked the extensibility of Eclipse, and because it is a standard environment, many developers are comfortable with Eclipse even if they do not have much modeling experience. Their goal was to use Rational Systems Developer to create models and use PathMATE, integrated with Rational Systems Developer, to generate code and enable execution and debug at the platform independent model level.”

Developing and debugging at the model level

Together with Pathfinder Solutions, an RBS development team of two junior-level engineers and one senior investigator used Rational Systems Developer to build complete models of one component of the RBS embedded system, including architectural models and detailed PIMs. The UML MDA models included classes, state machines and interfaces with parameters and data types. The team also used PathMATE to specify behavior and state actions at the UML level with platform independent action language (PAL).

“With PAL, you can debug at the model level, set breakpoints, step through, look at variables and examine the state of the system. You can stop at any line of action language and examine lots of different types of model-level run-time data,” explains Greg Eakman, a PhD and Pathfinder Fellow at Pathfinder Solutions. “By maintaining the model level of abstraction, the developer’s focus stays on the models they have built, even through test and integration.”

“Their goal was to use Rational Systems Developer to create models and use PathMATE, integrated with Rational Systems Developer, to generate code and enable execution and debug at the platform independent model level.”

—Peter Fontana, CTO, Pathfinder Solutions

Leading provider of telecommunications equipment calls in IBM and IBM Advanced Business Partner Pathfinder Solutions to help improve code quality.

“[Our customer was] excited to find the models were simpler, because simpler models are easier to understand, prove and verify—which contributes to higher software quality and better run-time performance.”

—Peter Fontana, CTO, Pathfinder Solutions

Creating simpler models and improving flexibility

Though the pilot project team selected a relatively small component to start, one of the most significant benefits became immediately apparent. The state machine for a central class was originally modeled in Rational Rose Technical Developer with 26 states and 64 transitions. The equivalent state machine developed using Rational Systems Developer and PathMATE had 11 states and 20 transitions. “We were able to create a much simpler model, with about half of the states and a third of the transitions in the previous implementation,” says Eakman. As modeling continued throughout the pilot, this trend persisted, surprising the team with how much simpler the PIMs were than code-focused models across the board.

Fontana adds, “When our customer began this project, they expected added productivity from raising the level of abstraction—but they also expected the models to be relatively similar. They were excited to find the models were simpler, because simpler models are easier to understand, prove and verify—which contributes to higher software quality and better run-time performance.”

In addition, the team was able to consolidate and simplify transaction processing. “In the past, each client had to handle its own persistence and transaction processing, and those mechanisms were distributed throughout the code,” notes Eakman. “Now they have pulled that together in one area, abstracting the persistence mechanism to make the model simpler. As a result it is much easier to move to new database types, including flat files, XML or a relational database. They have greater flexibility to handle changing requirements and changing target platforms.”

Building a single development environment

In addition to using Rational Rose Technical Developer, the RBS development team had been using a disconnected chain of tools that made it difficult to improve efficiency. “Because they were experiencing a lot of little productivity losses, there was no question for them that moving to Eclipse and a more integrated tool chain would be more effective,” notes Fontana.

With Rational Systems Developer and PathMATE tightly integrated in Eclipse, the team had seamless access to modeling, model-level execution and debug, and model transformation in a single environment. The team further increased developer productivity by accessing IBM Rational ClearCase® software directly from the Eclipse environment to perform software configuration management tasks.

Pathfinder Solutions Spotlight software also enabled the group to perform automated testing and interactive testing at the model level. Spotlight provides PIM-level debugging and model execution on development and target platforms. By using Spotlight to gain visibility into the system and control of the actual run-time state, the team was able to gain confidence in the model design, to repair problems and to move forward with a fully tested model.

Performing advanced model transformation

A significant benefit of any modeling effort is the ability to accelerate development by automatically generating code. Rational Systems Developer and PathMATE both support automated code generation, and for this project the RBS development team

“There was no question for them that moving to Eclipse and a more integrated tool chain would be more effective.”

—Peter Fontana, CTO, Pathfinder Solutions

**Leading provider of telecommunications equipment
calls in IBM and IBM Advanced Business Partner
Pathfinder Solutions to help improve code quality.**

“Using Rational Systems Developer and PathMATE, they created a PathMATE component and an interface that would allow integration with existing components.”

—Peter Fontana, CTO, Pathfinder Solutions

used the PathMATE Transformation Engine tool. This rules- and template-based engine leverages the PIM to perform self-optimizing transformations and to generate efficient, high-performance C++ code. The same models can readily generate C or Java code as well using transformations provided with PathMATE.

The open, rules-based code generation capabilities of PathMATE were particularly helpful in integrating the newly developed component with the rest of the system, which was developed using Rational Rose Technical Developer. “Using Rational Systems Developer and PathMATE, they created a PathMATE component and an interface that would allow integration with existing components. To create the similar connection at the code level, they extended the base PathMATE code generation with custom transformation rules to generate the integration code they needed. As a result, they were able to simplify integration at the model level as well as the execution level,” explains Fontana.

With Rational Systems Developer and PathMATE, the RBS team established a close link between model and code. “For this project, the model is the source,” says Eakman. “There are just a few handwritten lines of code at the run-time foundation level.”

Moving ahead

For Pathfinder Solutions' customer, the ability to build, execute, debug, test and transform UML 2 models in an open, integrated environment is a significant advantage in developing high-quality, business-critical embedded systems. The company estimates that Rational Systems Developer and PathMATE will double the productivity of its embedded development teams. "They were able to validate their initial premise—that raising the level of abstraction by expressing actions and debugging at the UML MDA level, as opposed to the C++ level, has a positive effect on productivity and quality," says Fontana.

"They were absolutely enthusiastic about the whole effort," concludes Eakman. "Our customer is very methodical in adopting new technology, and the successes to date have the company moving into more areas to apply this approach because this project clearly demonstrated its value."

For more information

To learn more about IBM Rational Systems Developer software, contact your IBM Business Partner or IBM representative, or visit:

ibm.com/software/awdtools/developer/systemsdeveloper

"Our customer is very methodical in adopting new technology, and the successes to date have the company moving into more areas to apply this approach because this project clearly demonstrated its value."

—Greg Eakman, PhD and Pathfinder Fellow, Pathfinder Solutions



© Copyright IBM Corporation 2007

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
11-07
All Rights Reserved.

ClearCase, IBM, the IBM logo, Rational and Rational Rose are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product and service names may be trademarks or registered trademarks or service marks of others.

The information contained in this documentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, it is provided "as is" without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software.