

Sprint Accelerates Integration of Enterprise Applications with IBM Rational Rose®

Overview

■ The Challenge

To improve its competitiveness and grow market share, Sprint was in the process of transforming the way it operates its business and needed to accelerate the integration of enterprise applications in support of business strategies.

■ The Solution

Using IBM Rational Rose®, Sprint developed a process that helps Sprint's development teams integrate existing systems, applications and databases in less time. Rational Rose helped the Sprint team increase efficiency, improve communication, and automate many aspects of analysis and design.

■ The Benefit

While accelerating development on major integration initiatives, Sprint was able to simultaneously reduce costs, mitigate project risk, and improve software quality. Quality issues found during integration testing and in production were reduced by an estimated 40 to 50 percent.

As a leading global integrated communications provider, Sprint serves more than 26 million customers in over 100 countries. Sprint is widely recognized for its history of innovation in developing, engineering and deploying state-of-the-art network technologies, including the United States' first nationwide all-digital, fiber-optic network and an award-winning Tier 1 Internet backbone. Never a company that rests on past accomplishments, Sprint is in the process of transforming the way it operates its business to improve IT competitiveness in the marketplace and grow market share. Sprint's transformation depends, in part, on its ability to integrate enterprise applications in support of business strategies. Using IBM Rational Rose, the industry-leading visual modeling and development tool, Sprint has developed a process that accelerates and simplifies the integration of Sprint's existing systems, applications and databases. Rational Rose enables architects and developers at Sprint to produce language-independent Unified Modeling Language (UML) models of software architecture, business needs, reusable assets and system interactions. Vinod Krishnan, Technical Architect for Sprint, explains, "Sprint recognized that Enterprise Application Integration (EAI) initiatives, typically

"Rational Rose has been a major success story for us in modeling complex data, transactions, and interfaces."

*-- Bill Branch, Vice President
Enterprise Architecture & Strategy
for Sprint's Information Technology
Services division*

large and complex projects, require enhanced analysis processes to enable the integration of data and applications. This led to the creation of Application Integration Modeling, or AIM. Sprint created AIM as a standardized modeling activity using IBM Rational Rose. It is used to capture system interactions in Interface Control Documents (ICD) which are deliverables in Sprint's Enterprise Development Process. Sprint also created the AIM Add-In for Rational Rose. This add-in is a set of tools that parses the information modeled in the ICD and automatically validates the model to generate reports that aid in accomplishing data integration, application integration, and integration testing to a name a few. AIM has proved to be the driving factor in eliminating data integration issues in Sprint's most challenging EAI initiatives. We are now more efficient in developing and applying solutions for projects which involve system integration."

Looking for a better way

Sprint began looking for a more reliable and manageable approach to integrate enterprise applications almost four years ago. When business decision-makers wanted to implement a strategic vision that required the integration of Sprint's IT systems, it was up to the IT team to find a way to do it. In the past there was no established process, and each such project started without the ability to effectively leverage existing artifacts or information. Krishnan recalls, "We began about three or four years ago. Sprint had a lot of mainframe systems and a lot of silo systems. We were working on certain strategic initiatives and we were trying to play the role of integrator -- doing major integration with middleware. We were developing software along these lines, but we didn't have a common process to use, or tools, or a well-defined set of artifacts. There wasn't a lot of turmoil, but there were problems. The development team discovered the first road block was getting a process in place that would make us successful in developing integration software. We quickly realized that Sprint needed a common process that not only captured the integration artifacts, but that would also help us take the artifacts to the next level as development assets that can be directly delivered to the developers -- so the developers didn't have to start creating IDLs (Interface Definition Language) or schemas. One of the goals was to set up a central repository for integration, analysis and design, and development artifacts."

Clear communication seen as essential

Krishnan points out that effective communication among development teams during the analysis and

design timeframe is key to successful software development, particularly on large-scale integration efforts. He remembers, "At the time, I was leading a development team that was developing integration software and middleware. I had firsthand experience in seeing the problems that we experienced in integration testing and how much miscommunication there was. So the first step was to enable communication between architects, analysts, and developers so that they all look at a problem the same way. We needed a process to do that; and we needed a tool to support that process."

IBM Rational Rose provided all of the capabilities the Sprint team was looking for to support AIM. First, Rational Rose was already being used successfully in various departments within Sprint to visualize, specify, construct and document software systems. As a result, the team was confident in Rational Rose as a tool, and knew that adoption of AIM would be simplified because Sprint development teams were already familiar with Rational Rose. Second, because Rational Rose is based on the Unified Modeling Language, it provides everyone on the Sprint team with a common language to help communicate and understand complex systems. Lastly, the Rational Rose Extensibility Interface enabled Sprint to customize the tool's capabilities and automate many aspects of analysis and design.

Rational Rose Extensibility Interface helps automate analysis tasks

Rational Rose Extensibility Interface helped Sprint extend and customize the capabilities of Rational Rose to meet the specific needs of the

AIM project. For example, Sprint customized menus; automated manual functions using scripts; and activated custom add-ins using the Add-In Manager. One of the first areas that AIM leveraged Rational Rose's Extensibility Interface was in the analysis phase of its integration projects.

Krishnan explains, "We had to put a process in place, and we needed a common work product. To create the common work product, we needed a tool that really lends itself to that kind of activity, and that was Rational Rose." The process was AIM, and the work product was the Interface Control Document or ICD, which is a specialized Rational Rose UML model that shows how applications come together. Lavanya Srinivasan, Software Engineer at Sprint, a key contributor and a thought leader to the AIM effort adds, "Basically, regardless of the implementation or how they interfaced with each other, our applications were represented in an ICD, and the message exchanges between applications were captured."

Krishnan continues, "We created analysis ICDs first. The goal of the analysis ICD was to identify the various integration use cases that applied across the project and to create a logical work product that showed the sequence of events that transpired within the application, the logical attributes exchanged within the application, and the attributes that were stored and retrieved from the database. Once we got there, the next question we looked at was, 'How do we verify that what we have in this work product will turn into a good development artifact?'. Essentially, the question was 'How do we test our

models for validity?'. An example of that would be integrated databases, which store thousands of attributes. I have models showing Application A storing 200 to 300 attributes into this database, and Application B and C retrieving portions of the attributes and updating them. At this point, I can manually check every attribute at the end of analysis to see that all the attributes required by the downstream applications were being produced by the upstream applications, or I can develop tools to do it for me. That's where we started exploring the option of using the APIs exposed by Rational Rose to automate that function."

Finding potential problems early

According to Krishnan, Rational Rose helped eliminate error-prone manual checking during the analysis phase, and also helped Sprint identify and fix problems earlier in the development lifecycle – well before integration testing. He continues, "We developed a data validation tool, which at the analysis level, would check all the ICDs in the project and automatically create a matrix that we called the Data Producer Consumer Matrix. This matrix showed the data that was being produced and consumed. It was a report that would either show that everything was okay, or that there was a consumer exception where we had an application trying to consume a data attribute which was not being produced. Before I had that tool and before I had this process in place, the only way I found out that there was an exception was if I manually sifted through every attribute -- or even worse I discovered it at integration testing time."

Because the data validation tool relied on the fact that ICDs were in a specific,

well-defined format, the Sprint development team also wanted to automate the task of ensuring the ICD complied with established formatting rules. Once again, they used Rational Rose to simplify a tedious and mistake-prone job. "As soon as the modeler completes an ICD they can go back and run the standards exception tool which goes through the model and automatically creates a report showing exceptions that don't meet the standards. And that was all built based on the APIs exposed by Rational Rose." says Krishnan.

Getting closer to code

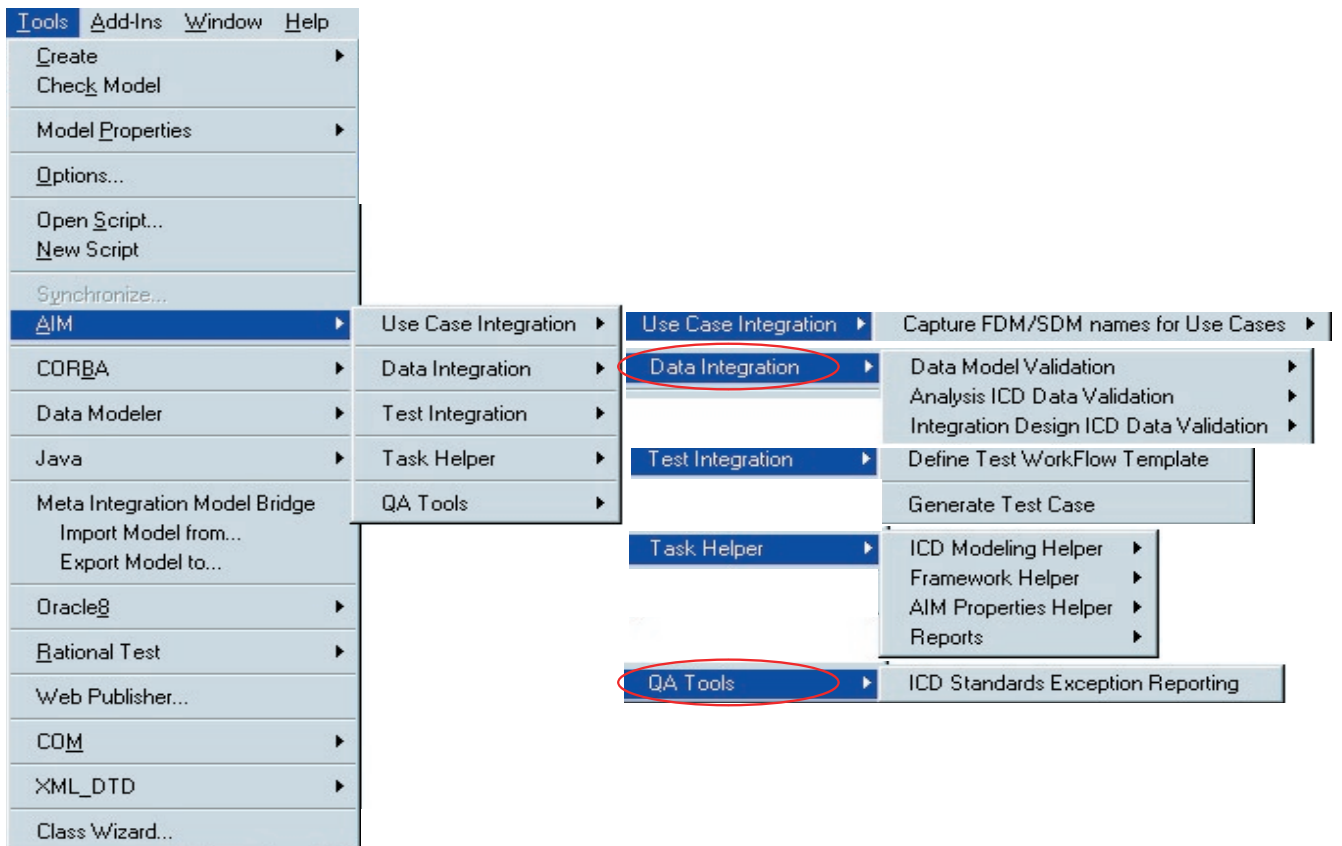
After using Rational Rose to address the analysis phase of integration, Sprint tackled the design phase. "Once we solved the analysis problem, looking at design was the next logical step. The question became, 'How do we take this closer to code?'. People can sit in a room and agree to a diagram, but if there still is a manual step that requires taking something that you see on a screen or a document and converting it manually to another work product, then there is an element of human interpretation and that allows the possibility for human errors. So to close that gap, get our work products more usable, and really add value to developers, we wanted to get closer to code – and that's when we created the design ICD," says Krishnan.

Design ICDs are similar to analysis ICDs, but are technology specific and design-oriented. "We extended some of the tools that we developed in analysis to support design – including the standards exception tool and the data validation tool that provided producer/consumer reports. In analysis, ICDs are

technology agnostic. But in design, we have applications that integrate through EJBs, CORBA, XML, or other technologies so they must be technology specific. And we use Rational Rose to generate EJB code, or IDLs," say Krishnan. The team also used the Rational Rose Extensibility Interface to enable modeling of mainframe copybooks. With this capability, Sprint was able to cover the entire spectrum of its integration technologies within Rational Rose.

Another important benefit of design ICDs was the ability to link the application design to the database designs that were involved in the integration. Krishnan explains, "We had linkage all the way through, so the person developing the data access layer knew, for example, that an attribute that's called X is coming from the application, and it actually maps back to the physical location in database one and field five, for example. We were able to capture that in a tool and in a repository, which of course required input from Rational Rose and input from the physical database. We closed that gap and linked all the way to the database, so the developers had the information they needed and didn't have to map data attributes while coding. The designs and the thought processes were already in place when that artifact – the design ICD – was captured.

Today more than 5,000 Sprint analysts, developers and testers are using AIM and IBM Rational Rose to accelerate enterprise application integration.



Using the AIM Add-In and the Rational Rose Extensibility Interface, the Sprint development team can access AIM features directly from Rational Rose menus

A wide range of benefits

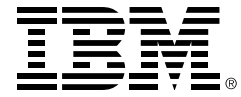
Today more than 5,000 Sprint analysts, developers and testers are using AIM and IBM Rational Rose to accelerate enterprise application integration. Srinivasan notes, “The system has users, such as process analysts, who facilitate in creating the ICDs as well as customers, such as development, database management and test integration teams that use the ICDs. The process enforces rigor and meticulous gathering of message calls and data exchanges between enterprise systems for business scenarios. Development teams benefit from early impact analysis to their code base; validation of data consumed or produced in a business scenario; and detailed, close-to-code artifacts that reduce coding cycles. Database management teams benefit from impact analysis to existing data stores and developing logical data models. Integration test teams can

plan for tests based on the business scenarios defined and the detailed system exchanges that can be tested.”

AIM has also contributed to overall improvement in software quality for Sprint’s enterprise integration initiatives. Krishnan notes, “In talking to people who have worked with AIM, everyone agreed that there was a substantial improvement in the quality of code delivered in projects that involved application and data integration. This improvement was apparent in the estimated 40 to 50 percent reduction in the number of tickets during integration testing and in production for projects that utilized the AIM methodology.”

The success of the AIM project and Rational Rose has been recognized throughout Sprint. Srinivasan notes, “The time savings that Rational Rose, the process, and the tools

provided catapulted AIM to be adopted across the enterprise.” Bill Branch, Vice President of Enterprise Architecture & Strategy for Sprint’s Information Technology Services division, adds, “Rational Rose has been a major success story for us in modeling complex data, transactions, and interfaces.” And Krishnan concludes that AIM’s success has helped it become a mission-critical foundation for future integration efforts at Sprint, “Rational Rose, AIM tools and processes, assisted in accelerating development, reducing costs and mitigating project risk. All major integration initiatives that are supported by our division now use AIM.” As a result, Sprint is better able to integrate enterprise applications and rapidly implement new business strategies – helping the company transform key business operations, gain market share, and improve its competitive position.



© Copyright IBM Corporation 2004

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

Produced in the United States of America
04-04

All Rights Reserved

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

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