SAS Delivers Architectural and Aesthetic Consistency with Eclipse

When a software vendor offers a suite of products for any market, one of the key things that end-users expect to see is a familiar "look and feel" between the components. All the little things, like key commands and menu arrangement, should make it easy for users to move from one application to another without feeling like they need to start at the low end of the learning curve. Today's end users are savvy enough to realize that mastering the use of software is a means to an end, not an end in itself. Their jobs are challenging enough without having to learn numerous combinations of basic commands or look at inconsistent interfaces in order to operate products from a single vendor.

"Users should not have to jump between multiple applications to perform a set of related tasks. The SAS Rich Client Platform gives us the ability to integrate related functions within an application suite even when they are developed and deployed independently."

Cary, NC-based SAS (www.sas.com) is a developer of business analytics software that allows companies to analyze data from multiple sources to predict the course that their enterprises need to take on the road to success. As the world's largest privately held software company—with over 40,000 business, government, and university sites using its products worldwide—SAS has a huge audience to please, including 97 of the top 100 companies on the 2004 Fortune 500 list. To provide its customers with the most consistent user interfaces across its Java-based products, as well as in developing them, SAS is in the process of moving to Eclipse and Eclipse Rich Client Platform (RCP).

"We have three pretty distinct uses of Eclipse within the company," says Rich Main, director of Java development environments for SAS. "The first is the internal use of the IDE for our Java application development purposes. We have over 1,000 Eclipse users within the organization and we've written a set of plug-ins to integrate Eclipse with our internal development processes. The second place we're using Eclipse is in building development tools on top of Eclipse. We have a product called SAS AppDev Studio that provides application development for Java programmers who want to write web applications or Java applications that can tap into the back-end power of the SAS analytics and data capabilities. We're migrating our older proprietary solution onto Eclipse using plug-ins that take advantage of the underlying platform."

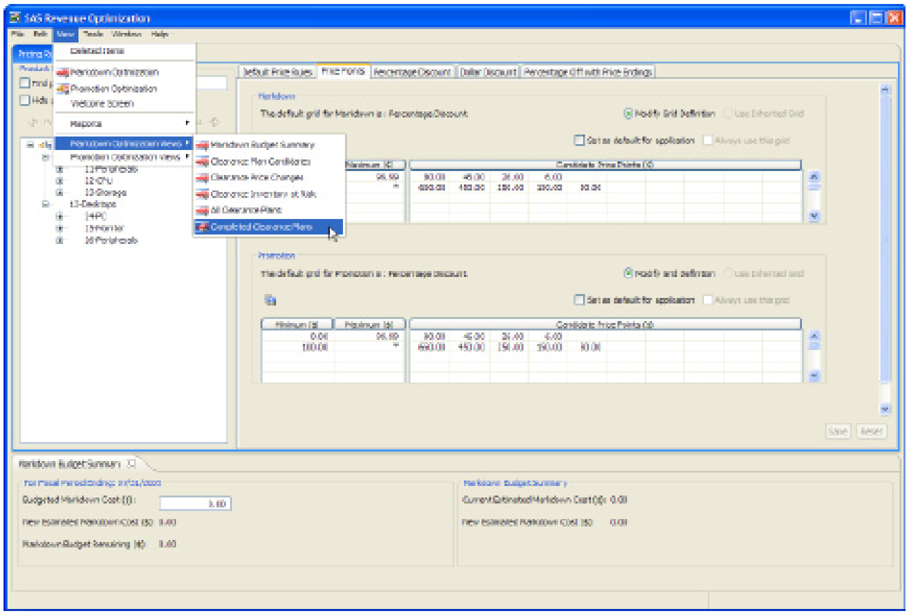
Focusing Development with Eclipse

The third place that SAS is using Eclipse is to build a Java desktop application framework. The Eclipse RCP is the technology base on which the company is building the specific capabilities necessary for its unique application development needs. SAS is using the platform to build a number of different customer-facing products.

"Underlying many of our new Java applications is the SAS Rich Client Platform, which will be installed automatically as a prerequisite, similar to the way the .NET framework is installed on Windows machines," Main says. "The platform is only installed once, and we then deploy multiple independent applications or application suites on top of it. The end user usually doesn't even realize that it exists."

The initial goal for using the Eclipse RCP is visual consistency across applications as well as an underlying integration between the various tools. SAS is supporting things like consistent best practices for accessing application services and consistent use of metadata in areas such as user preferences. “We want the same kind of behavior that users would see in products like the Microsoft Office suite,” Main says. “When they’re using SAS applications that have common user settings and preferences, the applications ought to find these settings consistently within our metadata, rather than asking them for the same information multiple times. That kind of inconsistency just doesn’t sit well when you’re supposed to have an integrated suite. In addition, users should not have to jump between multiple applications to perform a set of related tasks. The SAS Rich Client Platform gives us the ability to integrate related functions within an application suite even when they are developed and deployed independently.”

In the case of SAS’s Java applications, having the underlying consistency of a common framework goes beyond what the end users see. The company wanted to have the same core beneath all of these applications. “Before we started using Eclipse, there was really no Java application framework out there,” Main says. “As applications were being written in-house, we would develop new frameworks. We decided to use the Eclipse RCP to not only standardize our look and feel and achieve application integration, but to codify best practices so that developers could focus on creating value-added features for our customers. It doesn’t benefit our customers at all if we spend our time writing frameworks when we don’t have to.”



This view of SAS’s Revenue Optimization application demonstrates how the company’s various programs are tightly integrated within the Eclipse RCP. Here, we see the app’s Pricing Rules view in the background with its View menu in the foreground. The View menu also shows the availability of SAS’s Promotion Optimization application views as well as reports.

“Just in Time” Development and Deployment

Eclipse and the Eclipse RCP give SAS the ability to create a common framework for development and uniformity across the user interfaces of its Java product offerings. However, some of the benefits of Eclipse are less obvious—but just as important. “Our overall strategy is to support multiplatform clients,” Main says. “Because the Eclipse framework is based on Java, we realize that as an immediate benefit. On top of that, the approach that we’ve taken is to do what we like to think of as a ‘just-in-time’ approach to adding new features to our products. Eclipse gives us the enterprise installation capabilities we need to perform managed updates for certain components without putting out a whole new product release. With the SAS Rich Client Platform, we can deliver integrated application suites incrementally, without having to force our customers to license or deploy the whole suite at once. This gives them the flexibility to control their feature set and to add capabilities only when they see the value in doing so.”

At the enterprise level, Eclipse provides another level of update control. “A developer is going to want to install whatever updates they can find on the net anywhere,” Main says. “However, in a managed IT environment, we want to give administrators the ability to manage updates. The Eclipse RCP’s Update Manager helps to facilitate our efforts to add new functionality as needed while allowing the IT staff to control when those new features are pushed out to the users.”

George Walsh is a veteran tech editor and writer with experience in fields ranging from embedded systems programming to CAD.