

# NASA Uses Eclipse for Interplanetary Operations

In software development, as in just about any occupation, there are certain types of jobs that just sound more impressive to outsiders than others. To anyone interested in science, having a hand in telling a Mars exploration rover what it needs to do sounds very impressive. Programmers, however, would probably be more interested in the types of tools that are used to get the job done. In the case of the software development done at NASA's Jet Propulsion Laboratory ([www.jpl.nasa.gov](http://www.jpl.nasa.gov)) in Pasadena CA, open-source software—including Eclipse—is used in the application programming efforts.

The work done by JPL's Maestro Team encompasses the high-level operations tools for NASA's Mars surface missions. Mars rover and lander missions use JPL's tools here on Earth to produce the plans that the spacecraft will execute. These tools tell the spacecraft things like where they need to go and how many pictures they need to take.

The JPL Maestro Team develops applications for projects that include the Mars Exploration Rovers currently on the red planet, the Phoenix Mars Lander that NASA plans to launch in 2007, and the Mars Science Laboratory Rover that launches in 2009. Until very recently, the JPL Maestro Team was primarily using proprietary software written in Java. However, in early 2004, the decision was made to make use of Eclipse, including the Eclipse Rich Client Platform (RCP). The transition has been very successful. JPL expected that the very first mission its Eclipse-based software would be used on would be the Phoenix mission in 2007. However, to the group's surprise, some of the new software is ready to be used in the current rover mission that's on

**JPL expected that the very first mission its Eclipse-based software would be used on would be the Phoenix mission in 2007. To the group's surprise, some of the new software is ready to be used in the current exploration rover mission that's on Mars.**



Mars. It will be the first time JPL has used Eclipse-based software to help operate vehicles, and they've made it happen in only a year.

## **The Benefits of Open Source**

JPL's Maestro Team has a history of being open-minded about considering open source tools. The version of its software used for the Mars Exploration Rover Mission was not based on Eclipse but used open-source software

in many areas, including Castor, a data-binding framework; Java Expression Parser, a system for parsing and evaluating mathematical expressions; Xerces-J, a validating XML parser; MySQL; and other components. Using Eclipse as a framework was a matter of moving further along a path that the group had been traveling for some time. In fact, many of the mission operations workstations for the Mars Exploration Rover Mission use Linux as their operating system.

Among the reasons for using open source software is the philosophical idea that spending taxpayer money to do things that have already been done is just not justifiable. Many development groups within NASA believe that they are being paid by the taxpayers to do space exploration and that that's really where the money should be spent. Developing software that could be purchased from a commercial company or found in the open-source community is something that is now being avoided, whenever possible, by many NASA software development groups. However, when evaluating open source applications

or components, criteria such as the maturity of the projects, the amount of time the development community has invested, the number of releases the development community has made, and the results of an in-house evaluation process are taken into consideration. Open source is not preferable to software developed in-house or commercial products if it is not comparable in both quality and stability. Saving taxpayer money initially, then having to scrap a project or begin the development process using different tools isn't a benefit to anyone. All software used in mission-critical applications is carefully scrutinized and the decision of whether or not to use it is based on a number of considerations to be sure that the right tool is being used for the right job.

### **Eclipse for Developers**

In 2004, several members of the Maestro Team transferred off of active operations of the Mars Exploration Rovers and back into development. At that time, they went through a detailed analysis of how they wanted to write their software in the future. The group had always written software entirely in Java but didn't use an IDE of any kind. They also didn't use any kind of application framework, but wrote the user interface themselves on top of the Java runtime environment.

While doing its assessment, the group was looking for any way they could safely retire some of the code they had written in

## **In addition to having to work on the “rocket science,” that operates the space craft, JPL’s Planning Software Systems Group had to build an application that users could understand. Eclipse provided solutions.**

the past. Part of this decision, as in any software development effort, was to reduce cost. Another key issue was that

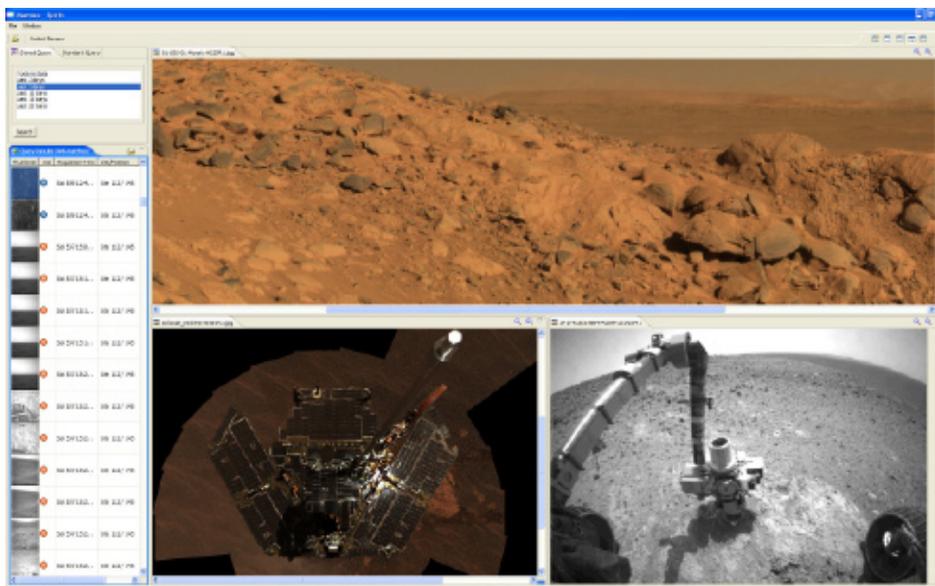
they felt that there were certain aspects of application development that could be done better by others. The realization they came to was that, while they were good at building software to drive spacecraft, a lot of other things were simply not within the realm of their expertise. Relying on an outside source to shoulder the burden of certain tasks helped the group to make the next version of its application better and cheaper while giving it the freedom to work on what it really needed to. In addition to having to work on the “rocket science,” that operates the space craft, JPL’s Maestro Team had to build an application that users could understand. Eclipse provided solutions.

### **RCP: Users See Interfaces, Not Technologies**

A lot of the functionality in previous versions of the Maestro Team’s applications was similar to what they found in Eclipse. The key difference between using in-house software and the Eclipse RCP was that, along with the functionality, the features were more polished. In addition, the group no longer

had to build and support the parts of the application that didn't deal directly with their programming strengths.

Keeping the application up to date is also an important issue for JPL, and the Eclipse RCP’s Update Manager goes a long way toward helping the developers



**The Maestro rover operations application displaying images received from the Spirit and Opportunity Mars Rovers. The Eclipse Rich Client Platform is the core foundation for Maestro and provides features such as a help system, automatic update, and organization of its many data views.**

keep users on the right track. Update Manager allows them to publish changes to their clients in a way that doesn't require the end user to manually download

**The key to JPL's use of Eclipse is all about focusing on their strengths and taking advantage of a community of collaboration. A lot of thought has been put into how RCP applications look and behave, and the Maestro Team's customers reap the benefits of that effort.**

companies get together to develop functionality that benefits their entire community, leaving the field of competition to the unique features

anything. In the past, the group's experience was that it was sometimes necessary to force updates because, as well-intentioned as the users might be, they often didn't update their software when they were asked to. Sometimes they would forget, sometimes they wouldn't have time, and other times they were just reluctant to change, so making it easier for them was a welcome capability.

**An Ensemble that Plays in Harmony**

Eclipse is also beginning to take hold in other development groups within NASA and one of the benefits that it's starting to provide mirrors the Eclipse community itself. Eclipse is serving as a framework for a project called Ensemble, which is something of a consortium for multiple operations tools that are being developed within the organization. The Maestro Team has presented the Eclipse RCP and the Eclipse development environment to the community of operations tools developers within NASA with the idea that they will make it easier to transfer code between teams and connect programs developed in different groups.

The different teams that are now working within the Eclipse framework still have their areas of expertise. However, while each group previously operated as more of an independent entity, there are now certain application functions the groups are no longer keeping to themselves. These components are offered in an effort to take advantage of the huge body of code that's developed within NASA and tend to be what would commonly be termed "middleware," built as plug-ins to the Eclipse framework, that the participating groups are now open-sourcing within NASA.

The idea is very similar to that of open source in general (and Eclipse in particular) where competing

of the applications themselves. At NASA, the groups participating in the Ensemble framework are encouraged to distinguish themselves from other participants based on their stated strengths in different fields, thereby avoiding competition. However, everything that is not considered application-specific is shared between Ensemble's participants to help the various groups avoid reinventing the wheel within NASA as a whole.

**Backing Science with Logic**

The key to JPL's use of Eclipse is all about focusing on their strengths and taking advantage of a community of collaboration. A lot of thought has been put into how RCP applications look and behave, and the Maestro Team's customers reap the benefits of that effort. Eclipse perspectives and the fact that the Eclipse RCP provides an attractive user interface are benefits to the system's users. Applications that use the Eclipse RCP also tend to look a lot more like the shrink-wrapped programs that users might buy. It lets the JPL Maestro Team concern themselves with thinking about how to make a Mars rover move rather than with developing a user interface.

In addition to the benefits of saving development time by using software that already exists and does what they need, JPL has also created a community within NASA that seeks to achieve goals similar to those involved in open source efforts. The idea that taking advantage of collective intelligence is a better approach to software development than building programs in isolation benefits NASA in the same way it does the Eclipse community. As it is, both the open source development model and Eclipse have proven very beneficial within NASA.