# The Mars Exploration Rover / Collaborative Information Portal

Joan Walton*, Robert E. Filman†, John Schreiner*
* National Aeronautics and Space Administration
† Research Institute for Advanced Computer Science

NASA Ames Research Center, MS 269-2
Moffet Field, CA 94035
U.S.A.

Astrology has long argued that the alignment of the planets governs human affairs. Science usually scoffs at this. There is, however, an important exception: sending spacecraft for planetary exploration. In late May and early June, 2003, Mars will be in position for Earth launch. Two Mars Exploration Rovers (MER) (Figure 1) will rocket towards the red planet. The rovers will perform a series of geological and meteorological experiments, seeking to examine geological evidence for water and conditions once favorable for life [1, 2].

Back on earth, a small army of surface operations staff will work to keep the rovers running, sending directions for each day's operations and receiving the files encoding the outputs of the Rover's six instruments. (Mars is twenty light minutes from Earth. The rovers must be robots.) The fundamental purpose of the project is, after all, Science. Scientists have experiments they want to run. Ideally, scientists want to be immediately notified when the data products of their experiments have been received, so that they can examine their data and (collaboratively) deduce results.

Figure 1. An artist's drawing of a Mars Exploration Rover

Mars is an unpredictable environment. We may issue commands to the rovers but there is considerable uncertainty in how the commands will be executed and whether what the rovers sense will be worthy of further pursuit. The steps of what is, to a scientist, conceptually an individual experiment may be scattered over a large number of activities. While the scientific staff has an overall strategic idea of what it would like to accomplish, activities are planned daily. The data and surprises of the previous day need to be integrated into the negotiations for the next day's activities, all synchronized to a schedule of transmission windows . Negotiations is the operative term, as different scientists want the resources to run possibly incompatible experiments. Many meetings plan each day's activities.

The Mars Exploration Rover/Collaborative Information Portal (MER/CIP) provides a centralized, one-stop delivery platform integrating science and engineering data from several distributed heterogeneous data sources. Key issues that MER/CIP addresses include:

- **Scheduling and schedule reminders**. Operations planning is driven by meetings. Participants need dynamic information about where they need to be, and cross-correlation to activities on Mars. Rather short-sightedly, all current calendar tools presume a 24 hour day. A Martian Sol is 24.66 hours. Mars time is critical, for the Rover is a diurnal creature, powered by sunlight. For scheduling, both time-scales must be visible.
- **Tracking the status of daily predicted outputs.** The outputs of experiments are radio-transmitted to earth daily. Experimenters want know what is planned to be done on a given Sol, what actually happened, and want to be informed immediately when their data products have arrived. However, tracking the path from scientific command through the interleaving of command execution on to the data products for that command is non-trivial.
- **Finding and analyzing data products**. This includes searching through the data-products space and analyzing the data files found there. Such examinations can be as simple as viewing pictures or as complex as running scientist-created data analysis software. The data is stored in existing heterogeneous structures, developed independently and obliviously to the needs of the portal.
- **Collaborating**. Scientists and operations managers want to share information including not only data products, but also the results of analyses and annotations of these products and analyses.
- **Personalization.** User interfaces, data product awareness and access rights to data must be personalized to the preferences and rights of each user. In particular, MER/CIP servers two very different communities: Scientists, primarily interested in the results of particular experiments, and operations staff, interested in maintaining the "health and safety" of the rovers.

Our goal in developing MER/CIP (and related projects, an emerging technology we call the *InfoCore Information Infrastructure* [4, 5]) has been to create a generic information infrastructure for integrating scientific and engineering data. Key elements of this domain include:

- Integrating heterogeneous data sources
- Managing large amounts of data
- Supporting the use of unstructured data
- Controlling access to data in a distributed and possibly federated environment according to the rights and privileges of particular users
- Facilitating collaboration
- Providing tools for browsing and analyzing a range of data
- Presenting quality interfaces for the above tasks
- Doing all this in an environment that is familiar and easy for users to install and manipulate.

Figure 2 shows a screen-shot from an early prototype of MER/CIP. Key elements of this interface include the simultaneous access to a variety of different information sources using a variety of GUI themes; the integration of numeric, structured, and photography information; scheduling tools based on Mars time, and the implementation of the system within a standard web browser.

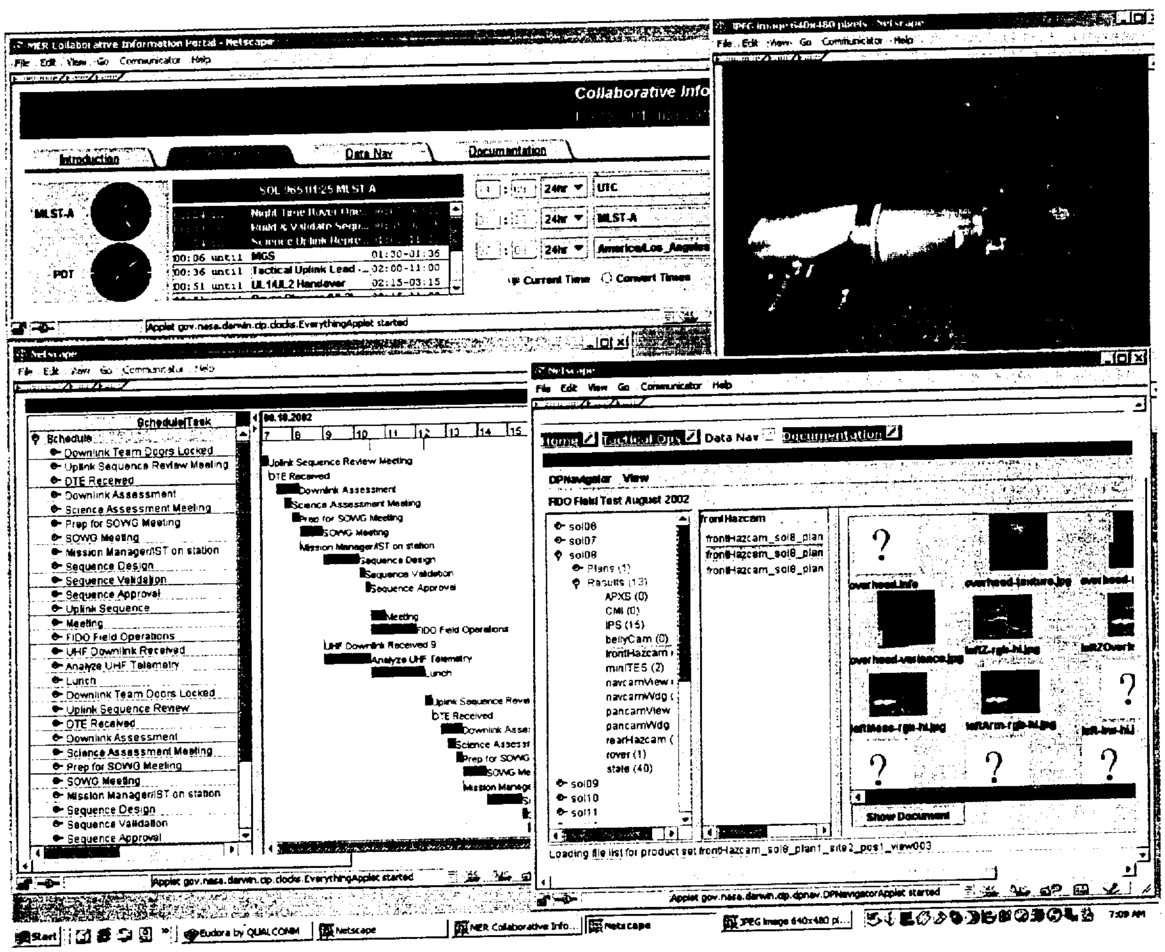


Figure 2. A prototype MER/CIP portal browser window.

Architecturally, MER/CIP is a web-based client-server application (Figure 3). Clients run as applets in browsers, connecting to an application server. The overall client is a collection of applications; the applications are Java applets, with the ability to serve conventional HTML for conventional display. Typical client applications navigate the space of the files in the data repository, view meeting schedules, present Rover camera images, read reports and summaries, send and receive broadcast and directed messages, and plot telemetry data.

The Rovers consume and produce a variety of data files, being presented with compilations of command sequences and returning the pictures and data from a variety of cameras and sensors. Scientists and operations staff want to know which commands have been or will be sent, and what data has returned. This information resides on several different legacy MER Mission Data Servers. MER/CIP needs to know when things have arrived, and where to find them. The monitor process runs asynchronously, and notifies the data acquisition system when new files have appeared. The data acquisition system parses this data and stores it in the CIP data and meta-data databases.

Systems in the InfoCore family keep two different kinds of data. "Small" datasets are kept in a conventional relational database. Large data products usually reside on the systems associated with the instruments that collected them. Typical large data products are files containing pictures or time- or space-series of data values. InfoCore systems keep a meta-database storing the location of these files and information about them. Dominating themes of the meta-database are that (1) Scientific data is often naturally hierarchical. For example, a database may be composed of a series of experiments, each of which has a number of configurations. For each configuration, many identical steps may be performed (e.g., a series of photographs from a specific camera). The actual number of layers in this hierarchy varies among domains (and sometimes even varies within a domain) but the hierarchical nature is common. (2) The attributes of interest for any given experiment are numerous and vary from step to step. Thus, a conventional relational database organization of well-defined columns will prove inadequate. Instead, InfoCore systems use a relational database with mechanisms for expressing both "part-of" and dynamically defined relationships.

MER/CIP applets connect to the MER/CIP server, a BEA WebLogic application server running the JetSpeed portal server. At the heart of the server is the MER/CIP Middleware services architecture. The middleware is responsible for vetting user identity and enforcing access control, managing the movement of data to and from the repositories and CIP databases, and arbitrating the destinations of asynchronous messages. Of particular relevance to the MER/CIP task is caching data: when an interesting data file arrives, 250 users may all want to see it simultaneous. The middleware server is driven by an Enterprise Java Bean model. It's notable in its use of both stateful and stateless session beans, and both container- and bean-managed entity beans [6].

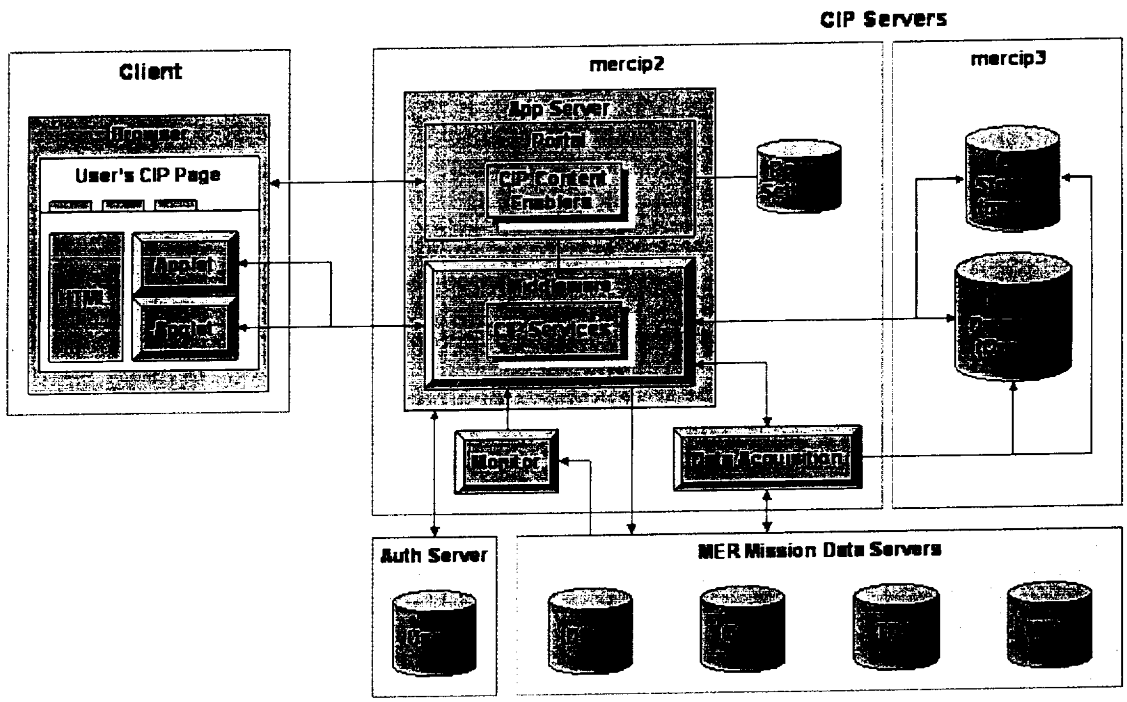


Figure 3. The MER/CIP software architecture.

And astrology? Unlike most software projects, MER/CIP must be finished when the planets align. It's a hard deadline, and is proving to be an ambitious deadline. Our response has been to try to incorporate as much off-the-shelf software as possible into the MER/CIP system. For example, rather than building a scheduler that reflects Mars and Earth timescales, we have wrapped conventional scheduling tools in a dual-timescale interface. And we have employed commercial and open-source servers like JetSpeed and WebLogic, despite the sharp learning curve associated with these products. These have been both rewarding and frustrating decisions. We have benefited from fundamentally sound commercial implementations while having little control over idiosyncrasies and the quirks of these implementations. For example, our original architecture relies on the Java Message Service (JMS) [3] to notify client applets of important events, such as the arrival of a data file or the rescheduling of a meeting. WebLogic is kind enough to provide a client JMS service. However, we were forced to reorganize our architecture to a polling client pattern when we discovered that the applet would need to download a 59 megabyte jar to take direct advantage of message notification.

MER/CIP is currently nearing it's Version 1 release. We expect to be able to report on the pre-launch lifecycle experience with this system at the conference.

**References**

1. Cornell/Athena Team, "Mission to Mars," http://athena.cornell.edu/the_mission/index.html
2. Jet Propulsion Laboratory, "2003 Mars Exploration Rover Mission," http://mars.jpl.nasa.gov/missions/future/2003.html
3. Richard Monson-Haefel, David A. Chappell, and Mike Loukid, *Java Message Service* Sebastapol: O'Reilly & Associates, 2000.
4. Joan Walton, Robert E. Filman, Chris Knight, David J. Korsmeyer, and Diana D. Lee. D3: A Collaborative Infrastructure for Aerospace Design. Workshop on Advanced Collaborative Environments, San Francisco, August 2001.
5. Joan Walton, Robert E. Filman, and David J. Korsmeyer. The Evolution of the DARWIN System." *2000 ACM Symposium on Applied Computing*, March, 2000, Como, Italy, pp. 971–977.
6. Vinoski, S., "Web services interaction models. I. Current practice," *IEEE Internet Computing, Vol.6*, No.3, 2002, pp. 89–91.