

NASA's Johnson Space Center

Constellation™ selected to develop, manage and integrate components for EVA Robot from subsystems to entire Robonaut

“Robonaut, a joint NASA and Defense Advance Research Project Agency project, is advancing the state of the art in anthropomorphic robotic systems, multiple use tool handling end effectors, modular robotic systems components, and autonomous and telepresence control systems.”

An excerpt from the Robot Systems Technology Branch web site. Robonaut is being designed by RSTB at the Johnson Space Center in collaboration with DARPA.

For more information go to http://vesuvius.jsc.nasa.gov/er_er/html/robonaut/robonaut.html

One challenge facing NASA's Robot Systems Technology Branch (RSTB), however, was how to manage the software development. With 45+ degrees of freedom, 150 sensors per arm alone, and multiple control modes, there was going to be a lot of code to develop and test.

Multiple teams would be working on the project, too. Some would be done sooner than others, and over Robonaut's life much of the code would be replaced. How the teams leveraged experience, leveraged code, and integrated subsystems would make the difference between a project finished in a manageable time frame and one that might never get finished.

RSTB chose Constellation as their software platform.

Modular, Hierarchical Control Model

Classical robot control methods would not work for a project of this ambition and scope. Instead, the overall control architecture was designed around self-contained sub-autonomies. Each sub-autonomy combines controller, safety system, low-level intelligence, and sequencing.

The sub-autonomies would be assembled hierarchically; for example, high level sub-autonomies selected services performed by lower level sub-autonomies. Each sub-autonomy interacts with its peers through publicly-defined interfaces.

This model allows each sub-autonomy to be developed and tested individually before integration with other sub-autonomies. It also allows sub-autonomies to be replaced without affecting the others.

The Software Platform for Modular Control Systems

The Constellation software platform is designed specifically for developers building systems with complex control requirements.



Robonaut Anthropomorphic Robotic System

Constellation lets them break down the project into its logical subsystems and in turn break those down into manageable components for incremental development and testing.

Constellation components assist RSTB through all phases of development and test. The software platform provides:

- A graphical development tool for creating components and assembling them into sub-autonomies.
- A run-time framework that takes care of the data flow and event scheduling.
- Repositories of ready-to-use and RSTB-built components.
- Direct integration of Simulink models developed in Matlab.
- Communications middleware for distributing state information over standard networks.
- Debugging tools that capture transient data and present it for easy analysis.

Help in Key Phases of Sub-Autonomy Development

Constellation has made substantial contributions to Robonaut progress in three aspects of the software implementation and integration:

- Rapid application development.
- Automatic sampled data scheduling.
- Simulation and tele-operation.

Constellation's developer tool lists components for easy access. The programmer drags-and-drops them into the diagram editor and connects them through data ports, methods, and interfaces to build higher-level components.

Rapid Application Development

The RSTB engineers use the Constellation Developer graphical development tool to make the relatively abstract object models into executable programs. An application is created as a diagram composed of hierarchical components. The Constellation framework generates and executes the code based on the diagram.

The engineers build their own software components by assembling sampled-data systems and state machines into higher-level components. Constellation provides ready-to-use components or the engineers can build their own. Engineers work individually but have access to other's components, cutting development time for common code. The graphic diagrams are self-explanatory, reducing learning time.

Constellation Does the Scheduling

One of the toughest design problems of the Robonaut software is controlling data flows into the controller from the sensors and remote operators and out to the devices. Managing the sample rate and triggering processing among separately developed sub-autonomies would be a near impossible task.

With the Constellation Developer, RSTB engineers assemble the low-level data flow components and state machines into higher level components. These components are put in the repository and then assembled with the Developer into higher level components, and eventually sub-autonomies.

The Constellation run-time framework builds the component execution list during initialization based on the application's diagrams. At run-time, the framework runs

through the list and executes the sample rate and trigger processing according to "habitat" components defined by the engineers. This lets the engineer use the same components in different execution environments by simply changing habitat properties.

Robonaut uses the NDDS network middleware to transfer data to monitor/headgear and send control messages back to Robonaut.

Publish-Subscribe Simplifies Tele-Operation and Simulation

Two aspects of the Robonaut development make use of the Constellation NDDS publish-subscribe middleware. The simulation interface uses NDDS to transmit data in real-time to a set of libraries developed in-house by NASA for computer animation and simulation. Simulation has been especially useful for testing since there are so many developers and so few Robonauts. It is also useful for debugging; inappropriate movement is easy to see.

The tele-operator is similar; the data indicating Robonaut's position is transmitted over the network to the operators head gear. In addition, the tele-operator uses NDDS to send control information (for example, the data indicating the operators hand actions) back to the Robonaut.

The benefit of using NDDS is that the data consumer can be the simulation front end running on the same machine or a tele-operator display running on another physical node without modifying the publisher code. The application publishes the data to NDDS; it does the distribution.

Constellation End-to-End Tool Chain

The Constellation software platform provides a framework for the run-time modules and a wide variety of ready-to-use components. In addition, it provides tools to help through all phases of application development:

- Constellation Developer: The Constellation Developer tool provides a “drawing board” on which the programmer builds increasingly complex layers of components using easy-to-understand diagrams to represent objects, data flows, and state machines.
- Communications: The NDDS® middleware and NDDS tools simplify network programming and help debug and optimize distributed object communications.

About RTI

Real-Time Innovations, Inc., the expert in real-time information networking, leads the industry with high performance standards-based software solutions for data-critical applications. Its products and consulting services provide the infrastructure for national railways, air traffic control, traffic monitoring, mission-critical combat systems, financial transaction processing, and industrial automation. RTI's flagship product, NDDS, is middleware based on the Object Management Group's (OMG) Data Distribution Service (DDS). NDDS provides the essential foundation for real-time communication in a networked system and enables a new class of embedded to enterprise (e2E) applications. Raytheon, Nikon, Omron, Harmonic, Applied Materials, Schneider Automation, Boeing, Lockheed Martin and the US Military rely on RTI technology for their real-time, data-centric, distributed applications. Headquartered in the heart of Silicon Valley since 1991, RTI is a privately held company.