JERRY: An Interactive Planning Tool for Space Robotics

by Amedeo Cesta, Enrico Giunchiglia, Paola Riccucci, Paolo Traverso

JERRY is a modular system for the interactive design, planning, control and supervision of the operation of autonomous robot systems in space. In such a highly critical environment, JERRY can effectively support the robot operators in both ordinary and emergency situations and make their work easier, safer and faster. JERRY can also provide scientists with no specific competence in robotics with a higher-level support for the automated execution of complex robot activities, with limited contributions from specialized operators.

JERRY has been developed in a collaboration between the Mechanized Reasoning Groups at both the University of Genoa and the Institute for Scientific and Technological Research (IRST), Trento, together with the Institute of Psychology, CNR, Rome, as part of an ongoing and more ambitious project funded by ASI, the Italian Space Agency. In this application, JERRY provides its functionality to different kinds of users who have to design, control and monitor a robot arm performing complex tasks, such as the setting up of experiments in a space workcell.

The system integrates different Artificial Intelligence techniques into an interactive environment to synthesize plans for the robot to execute. The high level goal of the system is to simplify the interaction of users at various levels of expertise with a rather complex robotic device.

JERRY has been designed to enable robot operation in Interactive Autonomy, ie the system can perform all of its tasks autonomously (including recovery from various non-nominal situations), but the user is able to easily monitor and possibly override autonomous operations, in a collaborative fashion. Effectiveness is guaranteed by a set of tightly integrated specialized modules, each dedicated to a specific task. Interactive autonomy is attained through a user-centered architecture, where the user asks for services from the specialized modules.

The key AI feature of the project lies in the definition of an experiment as a planning problem, which is then processed by the Planning and Execution modules.

The Planning Module requests a high-level description of the task to be performed by the robot and performs the synthesis of an equivalent abstract plan (in a user-oriented symbolic language), ie a sequence of high-level actions for the robotic system to execute.

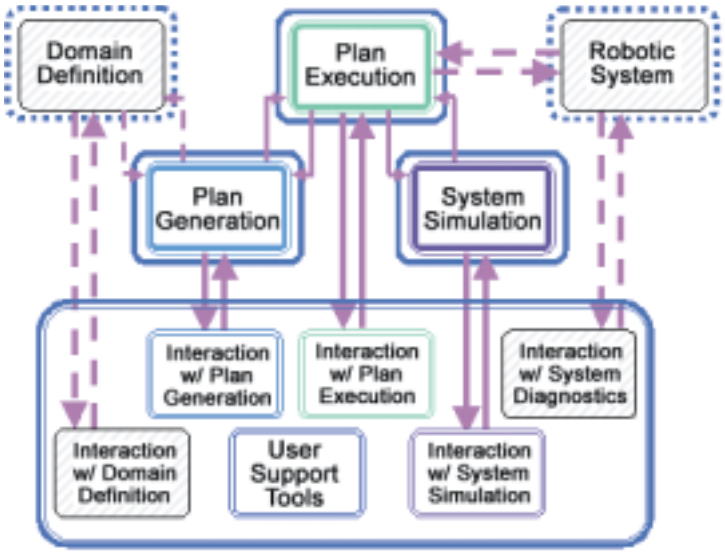

Figure1: Jerry Architecture.

The Execution Module transforms an abstract plan (describing the given task as a sequence of high-level actions) into an executable plan, where high-level actions are expressed in terms of the basic actions the robot system can perform, taking into account constraints related to the geometry and physics of the robot system and its operating domain. The executable plan is then encoded into the robot's control language, thus generating an executable code in system-specific language. The resulting code can be visually validated by submitting it to the Software Simulator of the robotic system.

Prototype

At present, a complete prototype has been developed that considers the SPIDER arm developed by ASI as the target robotic device. The system is able to synthesize low level programs coded in PDL2, the SPIDER command language.

From a software perspective, JERRY features a client/server architecture as shown in Figure 1. Each specialized module (top) is made accessible remotely (via TCP-IP) as a server. The User Interaction Module (bottom) acts as a client, connecting to the appropriate server at each step of the experiment's lifecycle and requesting its processing services.

Such an architecture results in a rather 'lightweight' Interaction Module, thus enabling users to run the earlier stages of development on computers of limited resources, such as laptops; for a wider portability, the module is written in Java.

Figure 2 shows JERRY as seen by the user, through its Interaction Module. In the main window (left), four viewports show the resources of the specialized servers: the planning problem specification window (left); the plan in execution (top-right); the PDL2 code corresponding to the action being executed (middle-right); and the visual simulation of the plan coming from the simulator (bottom-right). At any moment, for better readability, users can move the module on which their attention is focused to the main viewport. At the bottom of

the main window, a Console Area shows information on the status of communications with the remote servers and the data exchanged.

The HTML User Guide (right) appears in a separate window. It can be also viewed independently of the User Interaction Module, through any HTML browser.

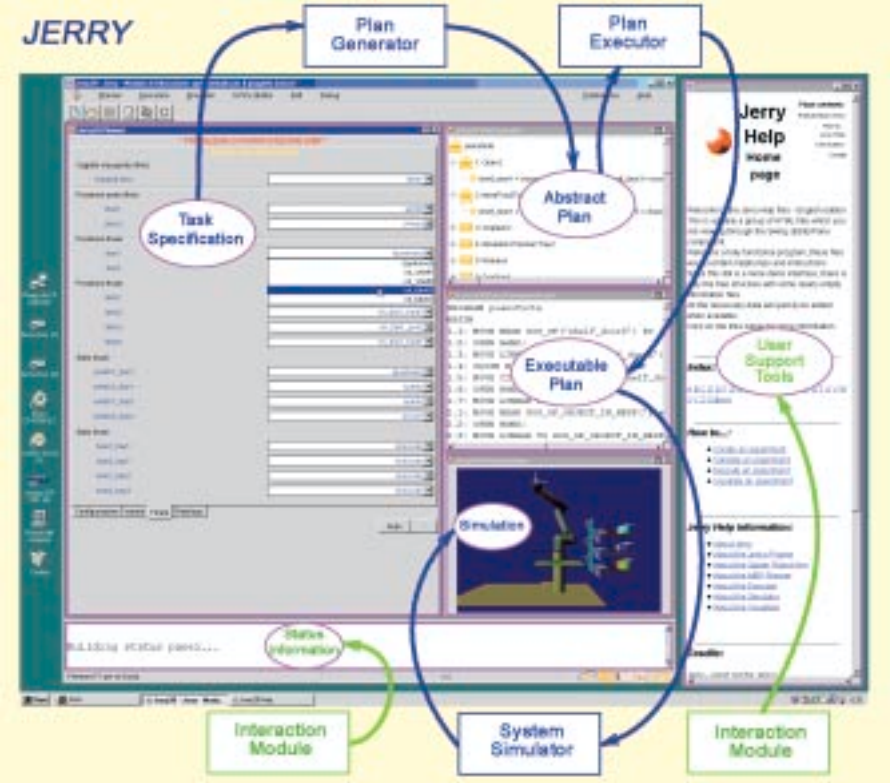

Figure 2: Jerry user interface.

Future Developments

Although the project is still running, a first working prototype is now available for experimentation. The current scenario concerns the SPIDER arm extracting a tray from a shelf, fixing it to one of two tables and then automatically performing experiments moving objects contained in the tray.

Further development will enhance the capabilities of the planning problem specification window. While the current design is centered on the selection of text parameters, the next release will enable users to specify all the settings via direct manipulation of graphical objects. Further work will be aimed at making the whole architecture completely domain-independent, in order to obtain a powerful tool for robot software development and verification, starting from a high level specification language.

Links

http://pst.ip.rm.cnr.it/projects/jerry.htm

Please contact:
Amedeo Cesta – IP-CNR
Fax: +39 06 824 737
E-mail: cesta@ip.rm.cnr.it

Mobile Robots with Dual Dynamics

by Ansgar Bredenfeld, Herbert Jaeger and Thomas Christaller

Speed and complexity of behaviors are key factors for mobile robots acting in unpredictable, dynamic environments. The Behavior Engineering (BE) team of the GMD Institute for Autonomous intelligent Systems (AiS) focuses on the combination of these two research issues. We use soccer playing robots as demonstrator platform since they provide an ideal benchmark environment for interdisciplinary research on mobile robotics.

Our approach to robot programming is based on a mathematical model for robot behaviors which we developed. It integrates central aspects of a behavior-based approach, robust control, and a dynamical systems representation of actions and goals. Robot behaviors are specified through ordinary differential equations, forming a global dynamical system made of behavior subsystems which interact through specific coupling and bifurcation-induction mechanisms. Behaviors are organized in levels where higher levels have a larger time scale than lower levels. Since at the elementary level the activation of behaviors (activation dynamics) is separated from their actuator control laws (target dynamics), we named our approach 'Dual Dynamics'. An important feature of Dual Dynamics is that it allows for robust and smooth changes between different behavior modes, which results in very reactive, fast and natural motions of the robots.

Dual Dynamics Design Environment

The successful design of robot software requires means to specify, implement and simulate as well as to run and debug the robot software in real-time on physical robots. It was a major challenge to make the Dual Dynamics approach productive in a state-of-the-art design flow. The result of our work is the integrated Dual Dynamics Design Environment. It allows to design Dual Dynamics models on a high level of abstraction and to synthesize all code artifacts required to make Dual Dynamics models operative in practice: a documentation, a simulation model, control programs for physical robots and a parameter set for generic test and debug