Software

Natural-Language Parser for PBEM

A computer program called "Hunter" accepts, as input, a colloquial-English description of a set of policy-based-management rules, and parses that description into a form useable by policy-based enterprise management (PBEM) software.

PBEM is a rules-based approach suitable for automating some management tasks. PBEM simplifies the management of a given enterprise through establishment of policies addressing situations that are likely to occur. PBEM provides a way of managing configurations of network elements, applications, and processes via a set of high-level rules or business policies rather than managing individual elements. Thus, PBEM enables abstraction of the capabilities of the individual elements and switching of control to higher levels.

Development of a system that understands colloquial English is an extremely difficult problem. Because most people do not write perfect English, such a system must be very robust in order to understand what has been written. Hunter is such a system. Recognizing that all possible dialects and variants thereof cannot be anticipated in advance, Hunter was developed to have a unique capability to extract the intended meaning instead of focusing on parsing the exact ways in which individual words are used.

This program was written by Mark James of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

In accordance with Public Law 96-517, the contractor has elected to retain title to this invention. Inquiries concerning rights for its commercial use should be addressed to:

Innovative Technology Assets Management
JPL
Mail Stop 202-233
4800 Oak Grove Drive
Pasadena, CA 91109-8099
E-mail: iaoffice@jpl.nasa.gov

Refer to NPO-45791, volume and number of this NASA Tech Briefs *issue, and the page number.*

Policy Process Editor for P^{3}BM Software

A computer program enables generation, in the form of graphical representations of process flows with embedded natural-language policy statements, input to a suite of policy-, process-, and performance-based management (P^{3}BM) software developed at NASA's Jet Propulsion Laboratory. Like the program described in the immediately preceding article, this program (1) serves as an interface between users and the Hunter software, which translates the input into machine-readable form; and (2) enables users to initialize and monitor the policy-implementation process.

This program provides an intuitive graphical interface for incorporating natural-language policy statements into business-process flow diagrams. Thus, the program enables users who dictate policies to intuitively embed their intended process flows as they state the policies, reducing the likelihood of errors and reducing the time between declaration and execution of policy.

This program was written by Mark James, Hsin-Ping Chang, Edward T. Chow, and Gerald A. Crichton of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

In accordance with Public Law 96-517, the contractor has elected to retain title to this invention. Inquiries concerning rights for its commercial use should be addressed to:

Innovative Technology Assets Management
JPL
Mail Stop 202-233
4800 Oak Grove Drive
Pasadena, CA 91109-8099
E-mail: iaoffice@jpl.nasa.gov

Refer to NPO-45821, volume and number of this NASA Tech Briefs *issue, and the page number.*

A Quality System Database

A quality system database (QSD), and software to administer the database, were developed to support recording of administrative nonconformance activities that involve requirements for documentation of corrective and/or preventive actions, which can include ISO 9000 internal quality audits and customer complaints. ["ISO 9000" denotes a series of standards, published by the International Organization for Standardization (ISO), for implementation of quality systems to be used in contractual situations.] The software provides for recording and storage of data, enables tracking, and provides status information.

The current version of this QSD software was written in the Microsoft Access software system and is server-based. Because of an increase in the number of users and the need for data security and reduction in response time, conversion to a Web-based version supported by Oracle software was investigated. As a result, a prototype Web-based version was developed and found to satisfy the aforementioned needs. Efforts are being made to determine the modifications necessary to serve the expanded user base. In addition, the Johnson Space Center ISO auditing group has expressed an interest in adopting this software.

This program was written by William H. Snell, Anne M. Turner, Luther Gifford, and William Stites of United Space Alliance for Johnson Space Center. Further information is contained in a TSP (see page 1). MSC-23447-1

Trajectory Optimization: OTIS 4

The latest release of the Optimal Trajectories by Implicit Simulation (OTIS4) allows users to simulate and optimize aerospace vehicle trajectories. With OTIS4, one can seamlessly generate optimal trajectories and parametric vehicle designs simultaneously. New features also allow OTIS4 to solve non-aerospace continuous time optimal control problems.

The inputs and outputs of OTIS4 have been updated extensively from previous versions. Inputs now make use of object-oriented constructs, including one called a metastring. Metastrings use a greatly improved calculator and common nomenclature to reduce the user's workload. They allow for more flexibility in specifying vehicle physical models, boundary conditions, and path constraints. The OTIS4 calculator supports common mathematical functions, Boolean operations, and conditional statements. This allows users to define their own variables for use as outputs, constraints, or objective functions.

The user-defined outputs can directly interface with other programs, such as spreadsheets, plotting packages, and visualization programs.

Internally, OTIS4 has more explicit and implicit integration procedures, including high-order collocation methods, the pseudo-spectral method, and several variations of multiple shooting. Users may switch easily between the various methods. Several unique numerical techniques, such as automated variable scaling and implicit integration grid refinement, support the integration methods.

OTIS4 is also significantly more user friendly than previous versions. The installation process is nearly identical on various platforms, including Microsoft Windows, Apple OS X, and Linux operating systems. Cross-platform scripts also help make the execution of OTIS and post-processing of data easier.

OTIS4 is supplied free by NASA and is subject to ITAR (International Traffic in Arms Regulations) restrictions. Users must have a Fortran compiler, and a Python interpreter is highly recommended.

This work was done by John P. Riehl, Waldy K. Sjauw, and Robert D. Falck of Glenn Research Center and Stephen W. Paris of Boeing Phantom Works. Further information is contained in a TSP (see page 1).

Inquiries concerning rights for the commercial use of this invention should be addressed to NASA Glenn Research Center, Innovative Partnerships Office, Attn: Steve Fedor, Mail Stop 4–8, 21000 Brookpark Road, Cleveland, Ohio 44135. Refer to LEW-18319-1.

Computer Software Configuration Item-Specific Flight Software Image Transfer Script Generator

A K-shell UNIX script enables the International Space Station (ISS) Flight Control Team (FCT) operators in NASA's Mission Control Center (MCC) in Houston to transfer an entire or partial computer software configuration item (CSCI) from a flight software compact disk (CD) to the onboard Portable Computer System (PCS). The tool is designed to read the content stored on a flight software CD and generate individual CSCI transfer scripts that are capable of transferring the flight software content in a given subdirectory on the CD to the scratch directory on the PCS. The flight control team can then transfer the flight software from the PCS scratch directory to the Electronically Erasable Programmable Read Only Memory (EEPROM) of an ISS Multiplexer/Demultiplexer (MDM) via the Indirect File Transfer capability.

The individual CSCI scripts and the CSCI Specific Flight Software Image Transfer Script Generator (CFITSG), when executed a second time, will remove all components from their original execution. The tool will identify errors in the transfer process and create logs of the transferred software for the purposes of configuration management.

This work was done by Kenny Bolen and Ronald Greenlaw of The Boeing Company for Johnson Space Center. For further information, contact the JSC Innovation Partnerships Office at (281) 483-3809. MSC-23631-1