Software

Automated Computer Access Request System

The Automated Computer Access Request (AutoCAR) system is a Web-based account provisioning application that replaces the time-consuming paper-based computer-access request process at Johnson Space Center (JSC). AutoCAR combines rules-based and role-based functionality in one application to provide a centralized system that is easily and widely accessible. The system features a work-flow engine that facilitates request routing, a user registration directory containing contact information and user metadata, an access request submission and tracking process, and a system administrator account management component. This provides full, end-to-end disposition approval chain accountability from the moment a request is submitted.

By blending both rules-based and role-based functionality, AutoCAR has the flexibility to route requests based on a user's nationality, JSC affiliation status, and other export-control requirements, while ensuring a user's request is addressed by either a primary or backup approver. All user accounts that are tracked in AutoCAR are recorded and mapped to the native operating system schema on the target platform where user accounts reside. This allows for future extensibility for supporting creation, deletion, and account management directly on the target platforms by way of AutoCAR. The system's directory-based lookup and day-to-day change analysis of directory information determines personnel moves, deletions, and additions, and automatically notifies a user via e-mail to revalidate his/her account access as a result of such changes. AutoCAR is a Microsoft classic active server page (ASP) application hosted on a Microsoft Internet Information Server (IIS).

This program was written by Bryan E. Snook from Johnson Space Center. Further information is contained in a TSP (see page 1). MSC-24174-1

Range Safety for an Autonomous Flight Safety System

The Range Safety Algorithm software encapsulates the various constructs and algorithms required to accomplish Time Space Position Information (TSPI) data management from multiple tracking sources, autonomous mission mode detection and management, and flight-termination mission rule evaluation. The software evaluates various user-configurable rule sets that govern the qualification of TSPI data sources, provides a pre-launch autonomous hold-launch function, performs the flight-monitoring-and-termination functions, and performs end-of-mission safing.

Rule types are tailored to the Range Safety problem domain and are based on existing range practices with human-in-the-loop flight-termination systems. This module provides a cleanly deployable software library for autonomously executing a number of real-time range safety decisional functions. Its key strength is its ability to emulate the substantial variety of human-in-the-loop flight safety mission rules using a comparatively small set of flight-termination rule types. These four rules are:

1. Generic Parameter Threshold Limit – This rule is used to fire a terminate condition in response to one or more threshold conditions or Boolean truth conditions. This rule may carry with it an arbitrary number of interpolative look-up tables. It can implement flight azimuth constraints, erratic flight and attitude rate limits, and vehicle performance limits.
2. Coordinate Boundary Rule – This rule is used to determine whether or not a specified point is contained within a simple closed boundary represented by a table of coordinates. The state of a variable is evaluated using a ray crossing point-in-polygon algorithm. During each update cycle, the closest distance between the specified coordinate point and coordinate boundary is computed. These can be applied to present position, instantaneous impact point coordinates, flight corridor inclusion limits, protected area exclusion enforcement limits, and upper-stage commit limits.
3. Two-Point Gate Rule – This rule is used to determine whether or not a specified point has crossed a gate formed by a line between two points, and whether or not a specified point is ahead of or behind the gate at any point in time. These gates may be statistically located, or dynamically assigned via interpolative lookup tables. The state of a variable is computed by looking for intersections between the gate and a line formed from the previous and current locations of the target point. Another variable is computed by comparing the relative location of the specified coordinate set to the gate.
4. Trajectory Adaptable Green-Time Rule – This rule is used to establish the permissible time of flight with no valid data, and to flag a destruct condition when this time has been exceeded. Algorithms are used to compute the worst-case instantaneous impact point velocity as a function of the current state, and of vehicle acceleration.

The software is configured by the user via an XML configuration file. The algorithms are implemented within a C++ container class that can be embedded in and serviced by host application. Presently, the software possesses approximately 5,500 lines of code distributed over 28 source files.

This work was done by Raymond J. Lanzi of Goddard Space Flight Center and James C. Simpson of Kennedy Space Center. Further information is contained in a TSP (see page 1). GSC-15594-1

Fast and Easy Searching of Files in Unisys 2200 Computers

A program has been written to enable (1) fast and easy searching of symbolic files for one or more strings of characters, dates, or numerical values in specific fields or columns and (2) summarizing results of searching other fields or columns. Intended for use in Unisys 2200-series computers under OS 2200, the program implements a simplified version of a UNIX AWK command implemented in Unisys macro-assembler. (AWK is an interpreted programming language, included in most versions of UNIX, for filtering and manipulating textual data.) The program is given the name of a file or an element thereof to scan, the numbers of the fields or columns that would contain the strings of characters or numerical values that are sought, and options to control the type of search and display results. The program uses standard Unisys library routines for reading files that have standard structures and for editing and printing output. The program can run in a batch or an interactive mode.

This program was written by James S. Sarp of United Space Alliance for Johnson Space Center. For further information, contact the Johnson Commercial Technology Office at (281) 483-3809. MSC-23855-1

Parachute Drag Model

DTV-SIM is a computer program that implements a mathematical model of the flight dynamics of a missile-shaped drop test vehicle (DTV) equipped with a multistage parachute system that includes two simultaneously deployed drogue parachutes and three main parachutes deployed subsequently and simultaneously by use of pilot parachutes. DTV-SIM was written to support air-drop tests of the DTV/parachute system, which serves a simplified prototype of a proposed crew capsule/parachute landing system.

The DTV-SIM model is of a point-mass trajectory-integrator type and includes detailed submodels of the staged deployment of, inflation of, and aerodynamic drag on, the parachutes. The model simulates (1) the forces on the parachutes and the DTV and (2) the motion of the DTV/parachute system from release until landing. Before a planned test, DTV-SIM is used to predict the flight of the DTV/parachute system in order to develop a flight envelope for the test. After the test, DTV-SIM is used to reconstruct the flight on the basis of data acquired during the test and, while so doing, to optimize parameters in the parachute-inflation simulation submodels.

This program was written by Peter Cuthbert of Johnson Space Center. For further information, contact the Johnson Commercial Technology Office at (281) 483-3809. MSC-24361-1

Evolutionary Scheduler for the Deep Space Network

A computer program assists human schedulers in satisfying, to the maximum extent possible, competing demands from multiple spacecraft missions for utilization of the transmitting/receiving Earth stations of NASA's Deep Space Network. The program embodies a concept of optimal scheduling to attain multiple objectives in the presence of multiple constraints. Optimization of schedules is performed through a selection-and-reproduction process inspired by a biological evolution process. A genome (a representation of design parameters in a genetic algorithm) is encoded such that a subset of the scheduling constraints (e.g., the times when a given spacecraft lies within the field of view of a given antenna) are automatically satisfied. Several fitness functions are formulated to emphasize different aspects of the scheduling problem, and multi-fitness functions are optimized simultaneously by use of multi-objective optimization algorithms.

The output of the program consists of a population of Pareto-optimal schedules that demonstrate the compromises made in solving the scheduling problem and provide insight into a conflict resolution process. These schedules are used by human schedulers to choose the simplest paths to resolution of conflicts as items on schedules are changed and as new items are added to schedules.

This program was written by Alexandre Guillaume, Seungwon Lee, Yeou-Fang Wang, Hua Zheng, Savio Chau, Yu-Wen Tung, Richard J. Terrile, and Robert Hovden of Caltech for NASA's Jet Propulsion Laboratory. For more information, contact iaoffice@jpl.nasa.gov.

This software is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-44821.