Geospatial Authentication

Stennis Space Center, Mississippi

A software package that has been designed to allow authentication for determining if the rover(s) is/are within a set of boundaries or a specific area to access critical geospatial information by using GPS signal structures as a means to authenticate mobile devices into a network wirelessly and in real-time. The advantage lies in that the system only allows those with designated geospatial boundaries or areas into the server.

The Geospatial Authentication software has two parts — Server and Client. The server software is a virtual private network (VPN) developed in Linux operating system using Perl programming language. The server can be a standalone VPN server or can be combined with other applications and services. The client software is a GUI Windows CE software, or Mobile Graphical Software, that allows users to authenticate into a network. The purpose of the client software is to pass the needed satellite information to the server for authentication.

This work was done by Stacey D. Lyle of Geospatial Research Innovation Design for NASA's Stennis Space Center.

Inquiries concerning rights for its commercial use should be addressed to:

Dr. Stacey D. Lyle, RLPS
Conrad Blucher Insitute of Surveying and Science
6300 Ocean Drive
Texas A&M University
Corpus Christi, TX 78412
Phone No. : (361) 825-3712
Fax: (361) 825-5848
E-mail: stacey.lyle@tamucc.edu

Refer to SSC-00282, volume and number of this NASA Tech Briefs *issue, and the page number.*

Mars Science Laboratory Workstation Test Set

NASA's Jet Propulsion Laboratory, Pasadena, California

The Mars Science Laboratory developed the Workstation TestSet (WSTS) is a computer program that enables flight software development on virtual MSL avionics. The WSTS is the non-real-time flight avionics simulator that is designed to be completely software-based and run on a workstation class Linux PC. This provides flight software developers with their own virtual avionics testbed and allows device-level and functional software testing. The WSTS has successfully offloaded many flight software development activities from the project testbeds. Flight software developers can now instantiate as many virtual testbeds as there are available computer resources and also enables device level fault injections that are difficult to achieve on real avionics testbeds.

The WSTS provides peripheral component interface (PCI)-card-level simulation of avionics hardware, enabling testing of all but the lowest layers of the flight software. The WSTS utilizes shared-memory and synchronization provisions of POSIX in a Linux environment to provide high-resolution simulation with synchronization of the interaction between simulation and the flight software.

This program was written by David A. Henriquez, Timothy K. Canham, Johnny T. Chang, and Nathaniel J. Villaume of Caltech for NASA's Jet Propulsion Laboratory.

This software is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (626) 395-2322. Refer to NPO-45690.

Computing Bounds on Resource Levels for Flexible Plans

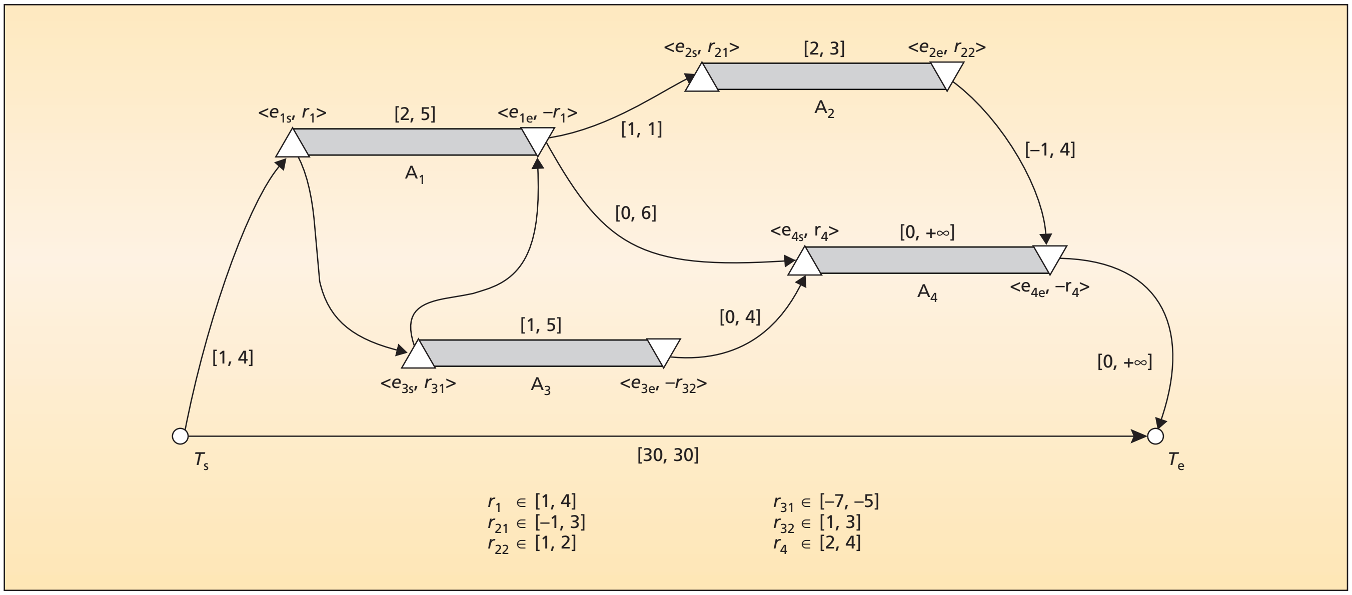
New algorithm entails less computation than previous algorithms.

Ames Research Center, Moffett Field, California

A new algorithm efficiently computes the tightest exact bound on the levels of resources induced by a flexible activity plan (see figure). Tightness of bounds is extremely important for computations involved in planning because tight bounds can save potentially exponential amounts of search (through early backtracking and detection of solutions), relative to looser bounds.

The bound computed by the new algorithm, denoted the resource-level envelope, constitutes the measure of maximum and minimum consumption of resources at any time for all fixed-time schedules in the flexible plan. At each time, the envelope guarantees that there are two fixed-time instantiations — one that produces the minimum level and one that produces the maximum level. Therefore, the resource-level envelope is the tightest possible resource-level bound for a flexible plan because any tighter bound would exclude the contribution of at least one fixed-time schedule. If the resource-level envelope can be computed efficiently, one could substitute looser bounds that are currently used in the inner cores of constraint-posting scheduling algorithms, with the potential for great improvements in performance.

What is needed to reduce the cost of computation is an algorithm, the measure of complexity of which is no greater than a low-degree polynomial in N (where N is the number of activities). The new algorithm satisfies this need. In this algorithm, the computation of resource-level envelopes is based on a novel combination of (1) the theory of shortest paths in the temporal-constraint network for the flexible plan and (2) the theory of maximum flows for a flow network derived from the temporal and resource constraints. The measure of asymptotic complexity of the algorithm is $O(N \cdot O(\text{maxflow}(N))$, where $O(x)$ denotes an amount of computing time or a number of arithmetic operations proportional to a number of the order of x and $O(\text{maxflow}(N))$ is the measure of com-



An **Activity Network With Resource Allocations** constitutes a graphical representation of a flexible activity plan to which the instant algorithm applies. Each activity time interval (A_i) is characterized by (1) time variables e_{is} and e_{ie} for start and end events, respectively; (2) a non negative flexible activity-duration link (e.g., [2,5] for activity A_1); and flexible separation links between events (e.g., [0,4] from e_{3e} to e_{4s}). Associated with each event is a resource-allocation variable (e.g., r_{31} with event e_{3s}). It is assumed that all events occur after starting time T_s and before an ending event time T_e rigidly connected to T_s. The interval $[T_s, T_e]$ is denoted the time horizon of the network.

plexity (and thus of cost) of a maximum-flow algorithm applied to an auxiliary flow network of $2N$ nodes. The algorithm is believed to be efficient in practice; experimental analysis shows the practical cost of maxflow to be as low as $O(N^{1.5})$.

The algorithm could be enhanced following at least two approaches. In the first approach, incremental subalgorithms for the computation of the envelope could be developed. By use of temporal scanning of the events in the temporal network, it may be possible to significantly reduce the size of the networks on which it is necessary to run the maximum-flow subalgorithm, thereby significantly reducing the time required for envelope calculation. In the second approach, the practical effectiveness of resource envelopes in the inner loops of search algorithms could be tested for multi-capacity resource scheduling. This testing would include inner-loop backtracking and termination tests and variable and value-ordering heuristics that exploit the properties of resource envelopes more directly.

This work was done by Nicola Muscvettola of Ames Research Center and David Rijsman of Mission Critical Technologies Inc. For further information contact the Technology Partnerships Division, Ames Research Center, (650) 604-2954. ARC-14948-1

MSLICE Science Activity Planner for the Mars Science Laboratory Mission

NASA's Jet Propulsion Laboratory, Pasadena, California

MSLICE (Mars Science Laboratory InterfaCE) is the tool used by scientists and engineers on the Mars Science Laboratory rover mission to visualize the data returned by the rover and collaboratively plan its activities. It enables users to efficiently and effectively search all mission data to find applicable products (e.g., images, targets, activity plans, sequences, etc.), view and plan the traverse of the rover in HiRISE (High Resolution Imaging Science Experiment) images, visualize data acquired by the rover, and develop, model, and validate the activities the rover will perform. MSLICE enables users to securely contribute to the mission's activity planning process from their home institutions using off-the-shelf laptop computers.

This software has made use of several plug-ins (software components) developed for previous missions [e.g., Mars Exploration Rover (MER), Phoenix Mars Lander (PHX)] and other technology tasks. It has a simple, intuitive, and powerful search capability. For any given mission, there is a huge amount of data and associated metadata that is generated. To help users sort through this information, MSLICE's search interface is provided in a similar fashion as major Internet search engines.

With regard to the HiRISE visualization of the rover's traverse, this view is a map of the mission that allows scientists to easily gauge where the rover has been and where it is likely to go. The map also provides the ability to correct or adjust the known position of the rover through the overlaying of images acquired from the rover on top of the HiRISE image. A user can then correct the rover's position by collocating the visible features in the overlays with the same features in the underlying HiRISE image. MSLICE users can also rapidly search all mission data for images that contain a point specified by the user in another image or panoramic mosaic.

MSLICE allows the creation of targets, which provides a way for scientists to collaboratively name features on the surface of Mars. These targets can also be