Software

Marsviewer 2008

Marsviewer 2008 is designed for quality control, browsing, and operational and science analysis of images and derived image products returned by spacecraft. This program allows all derived products (reduced data records, or RDRs) associated with each original image (experiment data record, or EDR) to be viewed in various ways, including in stereo, depending on the type of image.

The program features a pluggable interface called a "file finder." This encapsulates knowledge of a specific mission's filename and directory conventions, hiding the complexity behind each mission from the user, and allowing new missions to be added easily. Within a mission, different directory conventions can also be supported. This file-finder interface presents a similar interface to the user for all these missions and directory structures. All EDRs found for a given Sol are displayed in a list (optionally with thumbnail images) for the user to pick from.

Once an image is picked, a primary (vertical) tab pane allows the user to select the left or right image, left or right thumbnail, or stereo views. A secondary (horizontal) tab pane allows the EDR, or any of its RDRs, to be viewed. Most RDRs may be viewed independently, or as colored overlays on a background image. Each of the 41 RDR types has a display method appropriate for that type, and most have display parameters that can be adjusted.

The program understands two different image geometries (raw and linearized), and can show the actual pixel values under the cursor for every EDR and RDR matching the geometry type at once. Various display manipulations, such as zoom, data range, contrast enhancement, interval selection, and contour controls are available. Metadata (image labels) may be displayed and searched as well. The stereo display shows both left and right images simultaneously. It works either in anaglyph mode (red/blue glasses), or by using dedicated display hardware.

This innovation also covers the applications "jadeviewer" and "jade_overlayer," which are closely related derivatives from Marsviewer. The "jadeviewer" application reuses the image display and visualization portions of Marsviewer without the file finder. The user directly specifies filenames and RDR type, and can then view the product as with Marsviewer.

This work was done by Nicholas T. Toole and Robert G. Deen of Caltech for NASA's Jet Propulsion Laboratory. For more information, contact iaoffice@jpl.nasa.gov.

The software used in this innovation is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-46698.

Mission Services Evolution Center Message Bus

The Goddard Mission Services Evolution Center (GMSEC) Message Bus is a robust, lightweight, fault-tolerant middleware implementation that supports all messaging capabilities of the GMSEC API, including publish/subscribe and request/reply. The Message Bus enables NASA to provide an open-source middleware solution, for no additional cost, that is self-configuring, easy to install, and can be used for the development of GMSEC-compliant components. Some professional capabilities provided by this software include failover and fault tolerance, good performance, compression, debugging, and wide platform support.

This architecture is a distributed software system that routes messages based on message subject names and knowledge of the locations in the network of the interested software components. Functional software components register with the message bus, so that a location directory can be maintained. The functional applications then send messages onto the bus with an indication of the message type/subject/etc. Other applications that want to receive data register with the message bus and indicate what message types/subjects they want to receive. The message bus maintains a routing table where routes publish messages to the applications that have requested them. One message may be delivered to many different applications. Use of the message bus eliminates the need for each application to create separate communications paths with each application to which it interfaces.

The nature of the GMSEC Message Bus enables any project or user to quickly take the initial steps for creating or connecting GMSEC-compliant components, and for developing small systems without high license fees and learning curves. This software uses middleware to facilitate cross application or component communication on a software bus.

This work was done by Arturo Mayorga and John O. Bristow of Goddard Space Flight Center and Mike Butschky of Interface and Control Systems. Further information is contained in a TSP (see page 1). ***GSC-15575-1***

Major Constituents Analysis for the Vehicle Cabin Atmosphere Monitor

Vehicle Cabin Atmosphere Monitor (VCAM) can provide a means for monitoring the air within enclosed environments such as the International Space Station, the Crew Exploration Vehicle (CEV), a Lunar habitat, or another vehicle traveling to Mars. Its miniature pre-concentrator, gas chromatograph (GC), and mass spectrometer can provide unbiased detection of a large number of organic species. VCAM's software can identify whether the chemicals are on a targeted list of hazardous compounds and their concentration. Its performance and reliability on orbit, along with the ground team's assessment of its raw data and analysis results, will validate its technology for future use and development.

The software processes a sum total spectra (counts vs. mass channel) with the intention of computing abundance ratios for N_2, O_2, CO_2, Ar_2, and H_2O. A brute-force powerset expansion compares a library of expected mass lines with those found within the data. Least squares error is combined with a penalty term for using small peaks. This permits calibration even in the presence of unexpected/unknown system contamination or unknown/novel ratios of atmospheric constituents.

Automated, reliable mass calibration is a substantial improvement beyond other comparable systems. A method of compensation for variable response component spectra has been utilized via a weighted sum based on the central peak for each expected component.

It is fully autonomous, non-GUI-based, self-calibrating, and compliant with the VXWORKS flight software system.

This work was done by Lukas Mandrake, Benjamin J. Bornstein, Stojan Madzunkov, and John A. Macaskill of Caltech for NASA's Jet Propulsion Laboratory. For more information, contact iaoffice@jpl.nasa.gov.

The software used in this innovation is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-46956.

Astronaut Health Participant Summary Application

The Longitudinal Study of Astronaut Health (LSAH) Participant Summary software captures data based on a custom information model designed to gather all relevant, discrete medical events for its study participants. This software provides a summarized view of the study participant's entire medical record. The manual collapsing of all the data in a participant's medical record into a summarized form eliminates redundancy, and allows for the capture of entire medical events. The coding tool could be incorporated into commercial electronic medical record software for use in areas like public health surveillance, hospital systems, clinics, and medical research programs.

The software also enables structured coding that enforces a custom set of rules, as well as captures the context of the coded term. The terminology used is SNOMED CT, which is a massive terminology consisting of over 366,000 concepts with unique meanings and formal, logic-based definitions that are organized into 18 hierarchies. In addition, it contains more than 993,000 descriptions or synonyms for flexibility in expressing clinical concepts. SNOMED CT is also a compositional terminology, so multiple concepts can be grouped together to create an expression that has a totally different logical definition. By using some custom composition rules along with the context within the Participant Summary, a user can greatly reduce the number of candidate concepts, which not only improves productivity, it ensures that only legal SNOMED expressions can be created.

LSAH defines the line between the terminology and the information model. It takes a middle road between putting all the structure in a complex coded term and putting all the structure in numerous database fields.

This work was done by Kathy Johnson-Throop of Johnson Space Center; Ralph Krog of National Space Biomedical Research Institute; Deborah Eudy and Diane Parisian of EASI; Seth Rodriguez and John Rogers of Barrios Technology; and Mary Wear, Robert Volpe, and Gina Trevino of Wyle Laboratories. Further information is contained in a TSP (see page 1). ***MSC-24172-1***

Adaption of the AMDIS Method to Flight Status on the VCAM Instrument

Software has been developed to function onboard the International Space Station (ISS) to help safeguard human health by detecting compounds of concern in the cabin atmosphere, both in identity and concentration. This software calibrates and processes a standard 2D dataset (mass spectrum versus time) output from a gas chromatogram/mass spectrometer by identifying temporal events, including the possibility for near simultaneous event overlap, reducing the mass spectra for each event and comparing to an arbitrary library of known compounds. The level of autonomy, adjustment of parameters for the VCAM devices' specific data characteristics, and adaptive mass resolution to ease requirement of precision mass calibration are three unique features of this design. The estimation of concentration is also a significant addition to the standard AMDIS (NIST) implementation. Solution filtration based on elution time, and an arbitration algorithm for similar matches, provide the user with a more succinct, single-valued estimate in comparison to algorithms designed to merely augment expert hand analysis.

This work was done by Lukas Mandrake, Benjamin J. Bornstein, Seungwon Lee, and Brian D. Bue of Caltech for NASA's Jet Propulsion Laboratory. For more information, contact iaoffice@jpl.nasa.gov.

The software used in this innovation is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-46563.

Natural Language Interface for Safety Certification of Safety-Critical Software

Model-based design and automated code generation are being used increasingly at NASA. The trend is to move beyond simulation and prototyping to actual flight code, particularly in the guidance, navigation, and control domain. However, there are substantial obstacles to more widespread adoption of code generators in such safety-critical domains. Since code generators are typically not qualified, there is no guarantee that their output is correct, and consequently the generated code still needs to be fully tested and certified.

The AutoCert generator plug-in supports the certification of automatically generated code by formally verifying that the generated code is free of different safety violations, by constructing an independently verifiable certificate, and by explaining its analysis in a textual form suitable for code reviews. This enables missions to obtain assurance about the safety and reliability of the code without excessive manual effort. The key technical idea is to exploit the idiomatic nature of auto-generated code in order to automatically infer logical annotations that describe properties of the code. These allow the automatic formal verification of the safety properties without requiring access to the internals of the code generator. The approach is therefore independent of the particular generator used. The use of a combined generation/analysis tool can allow system engineers to concentrate on the modeling and design, rather than worrying about low-level software details. By providing tracing between code and verification artifacts, and customizable safety reports, the tool supports both certification and debugging. Although integrated with the code generator, AutoCert is functionally independent in the sense that it does not rely on the correctness of any generator components. The tool has two main benefits: (1) it helps catch bugs in autocoders, and (2) it helps with the certification process for the autogenerated code, thus mitigating the risk of using COTS autocoders that lack a trusted heritage.

The AutoCert technology also has a number of advantages over other approaches to formal verification. It can handle code with arbitrary loops, and can handle code generated from both continuous and discrete models. Moreover, the certification system based on annotation inference is more flexible and extensible than decentralized architectures where certification information is distributed throughout the code generator. Identifying the patterns that are used to infer the annotations is an iterative process, but by allowing tracing between VCs (verification conditions) and state-