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-