

idly plotted or re-plotted against different interpretations just by changing pointers to columns.

A key advantage is taking data on a carefully controlled grid, which allowed a single data set to be post-analyzed many different ways. Another innovation was building a software tool to provide very tight coupling between the generator and the recorded data prod-

uct, and the operator's worksheet. Together, these allowed the operator to sweep the jitter stimulus quickly along any of three dimensions and focus on the response of the system under test (response was jitter transfer ratio, or performance degradation to the symbol or codeword error rate). Additionally, managing multi-tone and noise waveforms automated a tedious manual process,

and provided almost instantaneous decision-making control over test flow. The code was written in LabVIEW, and calls Agilent instrument drivers to write to the generator hardware.

This work was done by Chatwin Lansdowne and Adam Schlesinger of Johnson Space Center. Further information is contained in a TSP (see page 1). MSC- 24814-1



µShell Minimalist Shell for Xilinx Microprocessors

NASA's Jet Propulsion Laboratory, Pasadena, California

µShell is a lightweight shell environment for engineers and software developers working with embedded microprocessors in Xilinx FPGAs. (µShell has also been successfully ported to run on ARM Cortex-M1 microprocessors in Actel ProASIC3 FPGAs, but without project-integration support.) µShell decreases the time spent performing initial tests of field-programmable gate array (FPGA) designs, simplifies running customizable one-time-only experiments, and provides a familiar-feeling command-line interface. The program comes with a collection of useful functions and enables the designer to add an unlimited number of custom commands, which are callable from the command-line. The commands are parameterizable (using the C-based command-line parameter idiom), so the designer can use one function to exercise hardware with different values. Also,

since many hardware peripherals instantiated in FPGAs have reasonably simple register-mapped I/O interfaces, the engineer can edit and view hardware parameter settings at any time without stopping the processor.

µShell comes with a set of support scripts that interface seamlessly with Xilinx's EDK tool. Adding an instance of µShell to a project is as simple as marking a check box in a library configuration dialog box and specifying a software project directory. The support scripts then examine the hardware design, build design-specific functions, conditionally include processor-specific functions, and complete the compilation process. For code-size constrained designs, most of the stock functionality can be excluded from the compiled library.

When all of the configurable options are removed from the binary, µShell has an unoptimized memory footprint of

about 4.8 kB and a size-optimized footprint of about 2.3 kB. Since µShell allows unfettered access to all processor-accessible memory locations, it is possible to perform live patching on a running system. This can be useful, for instance, if a bug is discovered in a routine but the system cannot be rebooted: µShell allows a skilled operator to directly edit the binary executable in memory. With some forethought, µShell code can be located in a different memory location from custom code, permitting the custom functionality to be overwritten at any time without stopping the controlling shell.

This work was done by Thomas A. Werne of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

This software is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-47495.



Software Displays Data on Active Regions of the Sun

Lyndon B. Johnson Space Center, Houston, Texas

The Solar Active Region Display System is a computer program that generates, in near real time, a graphical display of parameters indicative of the spatial and temporal variations of activity on the Sun. These parameters include histories and distributions of solar flares, active region growth, coronal mass ejections, size, and magnetic configuration.

By presenting solar-activity data in graphical form, this program accelerates, facilitates, and partly automates what had previously been a time-consuming mental process of interpretation of solar-activity data presented in tabular and textual formats. Intended for origi-

nal use in predicting space weather in order to minimize the exposure of astronauts to ionizing radiation, the program might also be useful on Earth for predicting solar-wind-induced ionospheric effects, electric currents, and potentials that could affect radio-communication systems, navigation systems, pipelines, and long electric-power lines.

Raw data for the display are obtained automatically from the Space Environment Center (SEC) of the National Oceanic and Atmospheric Administration (NOAA). Other data must be obtained from the NOAA SEC by verbal communication and entered manually. The Solar Active Region Display System

automatically accounts for the latitude dependence of the rate of rotation of the Sun, by use of a mathematical model that is corrected with NOAA SEC active-region position data once every 24 hours. The display includes the date, time, and an image of the Sun in H α light overlaid with latitude and longitude coordinate lines, dots that mark locations of active regions identified by NOAA, identifying numbers assigned by NOAA to such regions, and solar-region visual summary (SRVS) indicators associated with some of the active regions.

Each SRVS indicator is a small pie chart containing five equal sectors, each

of which is color-coded to provide a semiquantitative indication of the degree of hazard posed by one aspect of the activity at the indicated location. The five aspects in question are the history of solar flares, the history of coronal mass ejections, the growth or decay of activity, the overall size, and the magnetic configuration.

Mouse-clicking on an active-region-marking dot, SRVS indicator, or NOAA

region number causes the program to generate a solar-region summary table (SRT) for the active region in question. The SRT contains additional quantitative and qualitative data, beyond those contained in the SRVS: These data include the solar coordinates of the region, the area of the region and its change in area during the past 24 hours, the change in the number of sunspots in the region during the past

24 hours, the magnetic configuration, and the types, dates, and times of the most recent flare and coronal mass ejection.

This program was written by Mike Golithly of Johnson Space Center, Mark Weyland of Lockheed Martin, and Vern Raben of Raben Systems, Inc. For further information, contact the JSC Innovation Partnerships Office at (281) 483-3809. MSC-23300-1

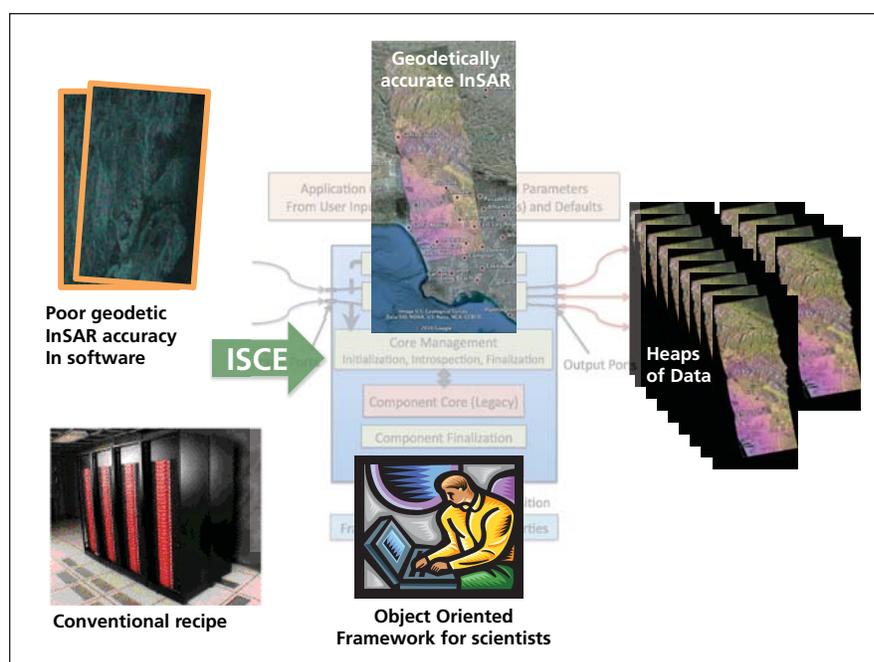
InSAR Scientific Computing Environment

NASA's Jet Propulsion Laboratory, Pasadena, California

This computing environment is the next generation of geodetic image processing technology for repeat-pass Interferometric Synthetic Aperture (InSAR) sensors, identified by the community as a needed capability to provide flexibility and extensibility in reducing measurements from radar satellites and aircraft to new geophysical products. This software allows users of interferometric radar data the flexibility to process from Level 0 to Level 4 products using a variety of algorithms and for a range of available sensors.

There are many radar satellites in orbit today delivering to the science community data of unprecedented quantity and quality, making possible large-scale studies in climate research, natural hazards, and the Earth's ecosystem. The proposed DESDynI mission, now under consideration by NASA for launch later in this decade, would provide time series and multi-image measurements that permit 4D models of Earth surface processes so that, for example, climate-induced changes over time would become apparent and quantifiable. This advanced data processing technology, applied to a global data set such as from the proposed DESDynI mission, enables a new class of analyses at time and spatial scales unavailable using current approaches.

This software implements an accurate, extensible, and modular processing system designed to realize the full potential of InSAR data from future missions such as the proposed DESDynI, existing radar satellite data, as well as data from the NASA UAVSAR (Uninhabited Aerial Vehicle Synthetic Aperture Radar), and other airborne platforms. The processing approach



The **InSAR Scientific Computing Environment (ISCE)** replaces old InSAR processing algorithms and conventional computing paradigms with modern geodetically accurate algorithms embedded at the core of a modern, flexible, and extensible object-oriented computing framework. The framework enables scientists to easily and efficiently process raw data into useful data products for their science models and investigations.

has been re-thought in order to enable multi-scene analysis by adding new algorithms and data interfaces, to permit user-reconfigurable operation and extensibility, and to capitalize on codes already developed by NASA and the science community. The framework incorporates modern programming methods based on recent research, including object-oriented scripts controlling legacy and new codes, abstraction and generalization of the data model for efficient manipulation of objects among modules, and well-designed module interfaces suitable for command-line execution or GUI-program-

ming. The framework is designed to allow users' contributions to promote maximum utility and sophistication of the code, creating an open-source community that could extend the framework into the indefinite future.

This work was done by Paul A. Rosen, Gian Franco Sacco, and Eric M. Gurrola of JPL/Caltech; and Howard A. Zebker of Stanford University for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

This software is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-47557.