NASA SBIR 2009 Phase I Solicitation

S6.05 Software Engineering Tools for Scientific Models

Lead Center: GSFC

This subtopic seeks to improve the productivity and quality of NASA's scientific modeling endeavors through customized tools, which enable and encourage improved software engineering practices. Because many of NASA's principal scientific models have evolved over decades to be hundreds of thousands of lines long with contributions from a wide variety of scientists, much of the software has become "brittle" in the sense that it has become difficult to extend, couple, and optimize. In other software communities (and other programming languages), access to modern software tools has enabled large gains in productivity by providing high-level tools for isolating software defects (bugs) as well as by automating common, albeit tedious, software processes. The goal is to extend these capabilities to support the Fortran programming language so that NASA's scientific models can extract similar benefits.

Target Programs, Missions and Mission Classes

Advances in developer productivity would be of significant benefit to several research and analysis programs within the Science Mission Directorate including:

- High-End Computing Program (http://hec.nasa.gov)
- Modeling, Analysis, and Prediction Program (http://map.nasa.gov)

Technology Areas

The objective is to create a suite of software tools, which directly ameliorate the most significant bottlenecks to productivity in the development of scientific models:

- Tools that assist in the construction of fine-grained unit-level software tests based upon existing functionality in a legacy Fortran application. Although tests written by developers are desirable, such tests are exceedingly difficult to create for legacy numerical software. Suites of these tests could provide a

significant element of risk-reduction for maintenance and extension of these models, and would be incorporated into some sort of unit-testing framework.

- Tools that enable high-level source code transformations ("refactorings"). Although refactoring support for other programming languages, most notably Java, has shown significant gains in productivity, similar support for Fortran is rather limited. (http://www.eclipse.org/photran/).

- Integration of a Fortran unit-testing frameworks within an Integrated Development Environment (IDE). Although multiple Fortran unit-testing frameworks have been developed (http://sourceforge.net/projects/pfunit), adoption by the community has been slow in part due to lack of integration within IDE's. Integration of other Fortran capabilities is also encouraged.

Tools and products developed under this subtopic may be used for broad public dissemination or for use within a narrow scientific community. These tools can be plug-ins or enhancements to existing software or on-line data/computing services. They also can be new stand-alone applications or web services, provided that they are compatible with most widely used computer platforms and exchange information effectively (via standard protocols and file formats) with existing, standard or prevalent applications. To promote interoperability, tools shall use industry standard protocols, formats, and APIs (Application Programming Interfaces).

It is highly desirable that the proposed projects lead to software that is infused into NASA programs and projects.