



NASA STTR 2014 Phase I Solicitation

T11.02 Computational Simulation and Engineering

Lead Center: JPL

Virtual Worlds - We solicit proposals for development of computational tools that enable rapid demonstration of mission concepts. The intent of such a tool is to enable non-experts in animation to rapidly build mission scenarios and visually express their concepts in a virtual world. These tools should enable full 3D visualization by importing of CAD parts of electromechanical systems (e.g., rovers, landers, orbiters), environment models (height field maps with textures for terrain, star maps and planetary bodies), animation functionality to show temporal progression and movement of appropriate objects in the scene. The tool should support animation of flexible bodies (e.g., solar panel vibrations) along with articulation of components. The tool should feature a ray-tracing engine for good quality visualization with shadowing, ambient lighting, etc. The tool should also be able to demonstrate terrain artifacts such as rocks, dust and ejecta as both static and dynamic objects. An example of a static artifact may be a rock pile that does not move during the animation while a dynamic artifact may be dust rising from a lander thruster interaction with terrain. Note that the emphasis is on visualization and not necessarily on the physics of the problem. However, the tool should have a standard API for integration with physics engines (e.g., ODE, Bullet, Proprietary Code) so that physics simulations can be used to control temporal progression of a scene. While it is preferred to have a representative physics engine coupled to the tool, it is not required. There should also be a functionality to write simple scripts for animating the virtual entities. There should be an avenue for developing a library of animation objects (e.g., rovers, terrains and locations) for re-use in later concept developments. The tools should be cross platform and enable development of animations or movies. The tool should take advantage of graphics processors or enable use of cluster computers for fast rendering of complex scenes. Alternately, the tool could feature a server-based functionality where the front-end user-interactions are through a webpage (using Java, HTML or other alternatives) and the computations are remotely conducted. Support for multiple concurrent users for content creating is desired. Ease of user interaction is key to the success of the tools. It is expected that at the end of Phase I, the performer will deliver an architecture document that captures the full intent of the tool. Similarly, performer will deliver software prototype of the implementation of the tool. It is expected that the software at the end of Phase I will be a prototype and may not have all features implemented or debugged. Performer will identify options for desired licensing options for the software to be developed for Phase II. At the end of Phase II, the performer will deliver all source code associated with the tool and verification test cases demonstrating all the proposed features within the software. The performer will also deliver a document summarizing the installation and usage of the tool and appropriate licensing options. In case of use of any third party software (e.g., open-source code) in this effort, the performer will deliver an acknowledgement that they have complied with appropriate licensing agreements. The anticipated TRL level at the end of Phase II is 5-6.

Computational Optimization - We solicit proposals for developing numerical tools that enable robust continuous and discrete optimization as well as sensitivity analysis for physics based computational models. There are a number of open-source and proprietary tools that are capable of meeting this objective at various levels of success. We are interested in proposals that develop a standard API for using the various open-source tools for different kinds of numerical problems. This would be in the form of a cross-platform abstraction API that enables users to have a standard API set for interacting with different optimization engines. We are also interested in software for autonomous optimization (genetic algorithms, simulated annealing, etc.), mixed (discrete and continuous) optimization problems and human-in-the-loop optimization / minimization. Methods for classifying problems

(potentially in a catalogue) and associating them with solution methods are also of interest. Methods for measuring the similarity between a new problem with those solved in the past (e.g., those in a catalogue) and hence identifying associated solution method(s) are of high interest. Phase I deliverables should include:

- A document summarizing the different numerical methods that would be implemented in Phase II and representative numerical examples of these methods developed in Matlab or similar program.
- Pseudo-code of the abstraction API.
- Architecture for classification (or cataloguing) problems with their solution methods and measuring similarity between problems.

Performer will identify options for desired licensing options for the software to be developed for Phase II. At the end of Phase II, the performer will deliver all source code associated with the abstraction API and as well as the software tool for interfacing with different numerical engines for solving optimization or sensitivity analysis problems, mechanisms for cataloguing problems and measuring similarity between problems. Performer will deliver verification test cases demonstrating the proposed features within the software. The performer will also deliver a document summarizing the installation and usage of the software and appropriate licensing options. In case of use of any third party software (e.g., open-source code) in this effort, the performer will deliver an acknowledgement that they have complied with appropriate licensing agreements. The anticipated TRL level at the end of Phase II is 4-6.