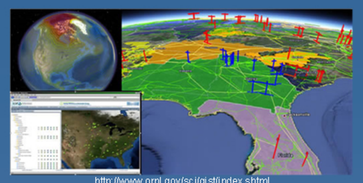
ISPR: A generalized multi-model interface



Christian J. López Méndez
University of Puerto Rico at Mayagüez Campus
Research Alliance in Math and Science
Computational Sciences and Engineering Division, Oak Ridge National Laboratory
Mentor: Robert Stewart

http://sites.google.com/a/g.ornl.gov/c_lopez/clients/poster

Introduction

Scientific investigations are faced with integrating diverse and independently developed models into a single architecture. The Spatial Analysis and Decision Assistance (SADA) freeware package (University of Tennessee) integrates models by using an *Interview-Steps-Parameters-Results* (ISPR) interface. The ISPR decomposes both modeling input and output into a sequential set of steps and parameters specific to the model(s) of interest. Each model is described by an *interview* process comprised of a series of *steps* each of which has a set of *parameters*. The user provides required data and a final *result* is produced. Graphically, the ISPR interface is organized such that each element remains visually consistent regardless of the model(s). Unfortunately, the initial implementation in SADA is embedded in the architecture and cannot be used by outside codes. The goal is to generalize the ISPR and make it usable to a wider audience of scientists and modelers who wish to access a quick interface for integration of their own codes.

Motivation

Scientific model integration is a common practice often requiring the development of a general user Interface (GUI). GUI development efforts are typically redundant, resource intensive efforts. A generalized GUI for model-integration would improve integration efficiency. SADA addresses this problem with the ISPR interface (Figure 1). The ISPR is embedded in SADA architecture and not usable by outside models.

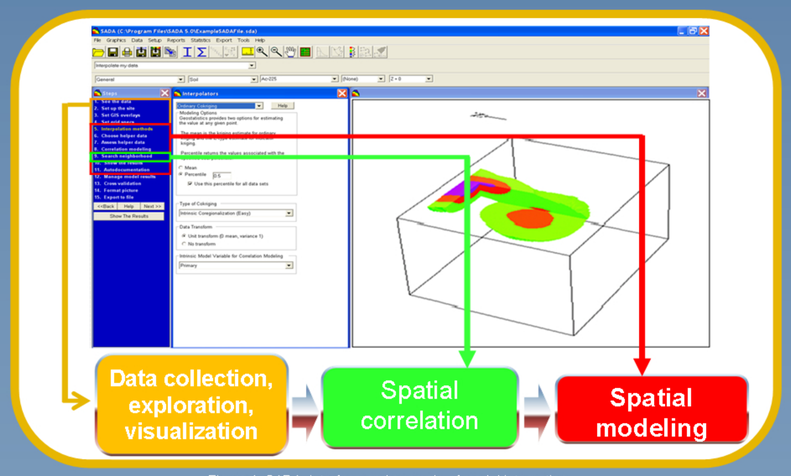
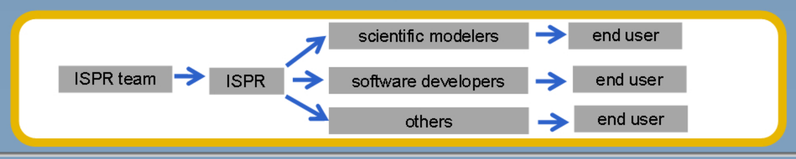


Figure 1. SADA interface and example of model integration.

Purpose

The goal is to implement a stand alone, generalized version of the ISPR. This new ISPR will require no source code modifications to integrate new models. The target audience for this new ISPR is scientific modelers and software engineers.



Method

The new ISPR is built in VB.net. It handles multi-model parameters, error checking, model integration and execution, and displays results. In order to integrate models without programming, a new Interface Markup Language (IML) is created to instruct ISPR organization and ensure proper model execution.

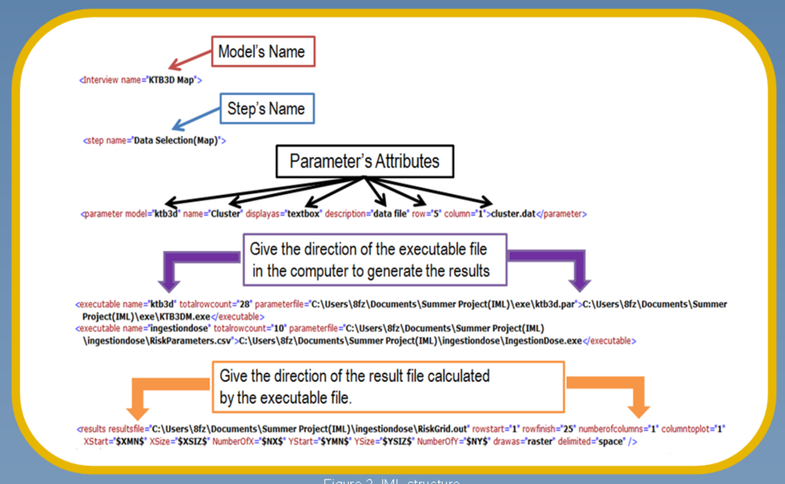
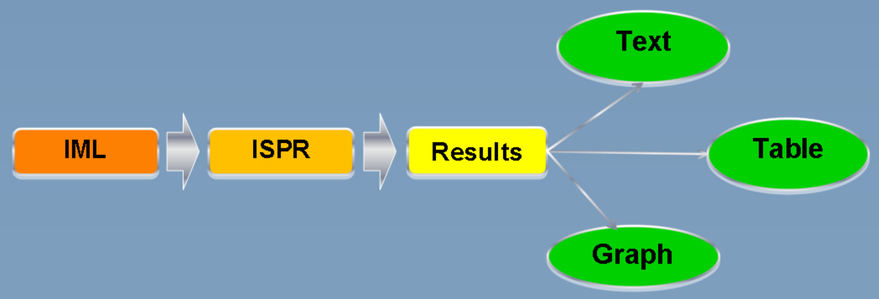


Figure 2. IML structure.



Results

Two major results arose from this effort. First, a complete IML specification was fully developed. Secondly, a desktop ISPR application for Windows was created. To demonstrate this approach, two models were integrated using only the IML and ISPR.

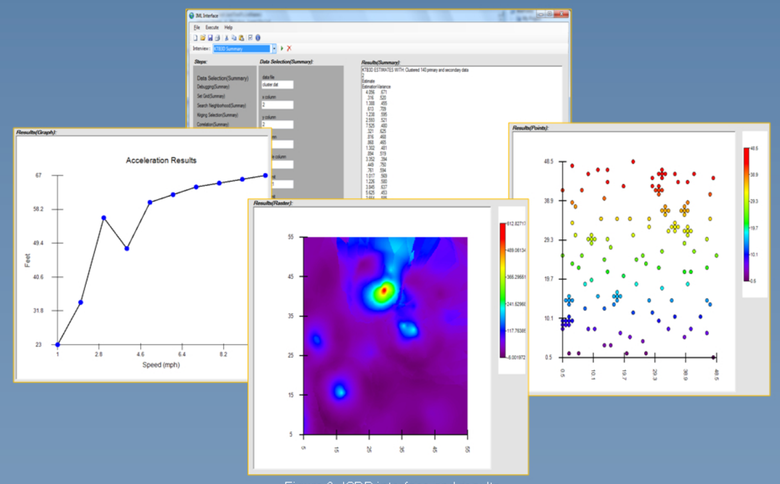


Figure 3. ISPR interface and results.

Conclusion

The first standalone ISPR interface was successfully implemented as a desktop application. Using only the new IML, scientists can access this application for their own projects. This first version represents a starting point where both ISPR and IML can evolve to accommodate more complex integrations.

Future Work

Next steps include publishing this work in the open literature, releasing the code as open source, and continuing to evolve both IML and ISPR as a sound method for model integration.

The Research Alliance in Math and Science program is sponsored by the Office of Advanced Scientific Computing Research, U.S. Department of Energy. The work was performed at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC under Contract No. De-AC05-00OR22725. This work has been authored by a contractor of the U.S. Government, accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. The author would like to thank Robert Stewart, Debbie McCoy, Rashida Askia, the RAMS program and the entire Computational Sciences and Engineering Division for this opportunity.