Sandia National Laboratories SAND2014-16869R

Integrated Sensitivity Analysis Workflow
DART Workbench Level 2 Milestone FY14
August 12th, 2014

Robert L. Clay, PM
Ernest J. Friedman-Hill, PI
Edward L. Hoffman, PO

Abstract

Sensitivity analysis is a crucial element of rigorous engineering analysis, but performing such an analysis on a complex model is difficult and time consuming. The mission of the DART Workbench team at Sandia National Laboratories is to lower the barriers to adoption of advanced analysis tools through software integration. In FY14, the DART Workbench team created an integrated capability for building complex finite element models and performing sensitivity analysis that allows the engineer to concentrate more on the analysis problem at hand, and less on the details of using the complex software involved. This new Integrated Sensitivity Analysis Workflow integrates best of breed tools for creating parameterized finite element models, applying advanced statistical techniques, controlling and monitoring model execution on high-performance compute clusters, and collating, presenting, archiving and protecting results. The integrated environment guides the engineer in the use of these integrated tools and greatly reduces the cycle time for engineering analysis.

Introduction

Complex computational models provide a wealth of data but for those data to be trusted the model, and the assumptions built into it, must be verified. A classic technique for model verification is sensitivity analysis, in which the numerical uncertainty in the output of a model is characterized, and its dependence on the values and uncertainties of input parameters is determined. Performing a full sensitivity analysis on a complex finite element model is a daunting task for many reasons, not the least of which is the diverse nature of the model inputs, which include mesh density, material parameters and algorithm choices. In practice these variable inputs are directed to many different software applications involved in creating an executable simulation. Furthermore, the difficulty of managing the large number of compute jobs involved in such a study, and then collating, processing and interpreting the results, is a serious impediment to the widespread adoption of sensitivity analysis in large-scale simulation.

Recent work at Sandia has led to the creation of a fully integrated modeling environment which allows the analyst to prepare, execute and monitor finite element simulations through a single consistent interface. For FY14, the DART

U.S. DEPARTMENT OF ENERGY

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

NNSA National Nuclear Security Administration

Workbench team further extended this environment to enable ubiquitous sensitivity analysis. With the new extensions to the DART toolset, the user can easily designate any variable quantity in the user interface as an input to a sensitivity study. The DAKOTA toolkit is used to choose appropriate input vectors and gather ensemble results for interpretation. The resulting family of simulations is managed via a fault-tolerant job monitor that reports job status and keeps the user aware of progress.

Results

The integrated workflow tools as implemented consist of a number of components, now integrated with the existing DART components for model building, job submission, and data management:

1. A Java library interface to the DAKOTA toolkit, so that bidirectional communication with DAKOTA is via simple function calls to an embedded library. This interface has been donated to the DAKOTA team so it will be generally available.
2. A new simplified DAKOTA Wizard for initial definition of V&V problems.
3. A new graphical user interface for detailed editing of DAKOTA input files. This interface is dynamically built from an XML syntax description of DAKOTA syntax; the same XML file is now used by DAKOTA to parse its input.
4. New graphical tools for defining and managing a set of parameters for a model, and replacing values anywhere in a model with parameters
5. The new Model Response Toolkit, an architecture for Quantity of Interest (QoI) extraction from simulation results.
6. A new Workflow Engine and Model Component Architecture for orchestrating the execution of simulation components.
7. Integration code such that all these pieces work well with each other and with the other components in the DART Workbench.

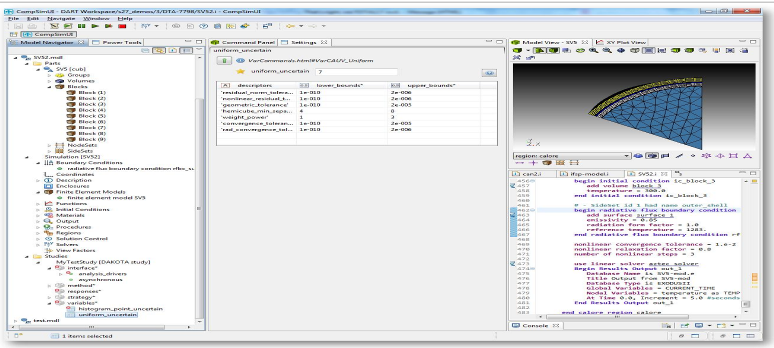
The new capabilities are slated for general release as part of the Sandia Analysis Workbench 2.0 in November 2014.

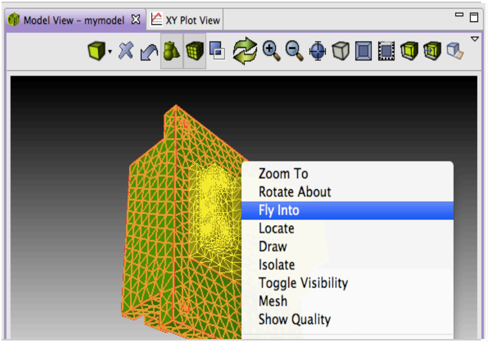
Milestone Evaluation

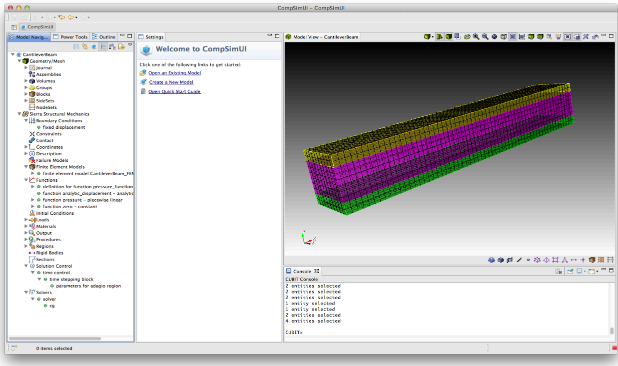
A panel was convened to evaluate milestone completion. The panel was chaired by Walt Witkowski and included Jay Dike, Patty Hough, and Tré Shelton. The milestone was deemed completed when it was determined by testing, using a test plan developed by George Orient, that a user can specify and carry out a sensitivity analysis, starting from a working Sierra model and continuing all the way through execution and results interpretation, as a single workflow in the integrated application. On August 12th, 2014, the panel met with the DART Workbench team to present evidence regarding the successful completion of the milestone.

Exceptional service in the national interest

Sandia National Laboratories







ASC IC V&V L2 – Final Review

August 12, 2014

V&V L2 Project Scope

The purpose of this milestone is to lower the barriers of performing V&V, specifically sensitivity analysis (SA). This will be accomplished by facilitating the setup and execution of DAKOTA from within the SAW Model Builder application.

Specific objectives:

- Assist users with computing model responses for common quantities of interest.
- Enable a basic workflow capability (Cubit → Sierra → Responses) in SAW such that a model can be rebuilt with a single call.
- Integrate DAKOTA UI problem definition into the SAW Model Builder environment as an integral part of the problem definition.
- Create a more robust mechanism for DAKOTA to execute SAW workflows.

The work on this milestone is phase 1 of a multi-year, multi-phase effort to support V&V integrated workflows for engineering sciences analysis.

The primary use-case for this milestone is Sierra Solid Mechanics.

V&V L2 Functional Requirements

As an analyst user of SAW Model Builder, I want/need the software to …

1. ***Create "DAKOTA ready" models in SAW Model Builder***
   a) Manage parameters and responses as an integral part of a model.
   b) Request time dependent history data for specific quantities of interest.
   c) Compute model responses by selecting from a library of "recipes" that parse quantities of interest from tabular ASCII data files (i.e. Sierra heartbeat file) based on common patterns.
2. ***Execute SAW Model Builder workflows***
   a) Execute an entire Cubit → Sierra workflow or individual stages.
   b) Automatically extract parameters and responses from workflow stages (Cubit & Sierra).
   c) Run a workflow stage for computing responses.
   d) Evaluate model robustness by running "what if" studies with user-defined parameter sets.
3. ***Set up a DAKOTA study.***
   a) Run a wizard process that creates a *"run-ready"* DAKOTA study: select a DAKOTA template, characterize model variables from model parameters and tag responses of interest.
   b) Seamlessly create/edit a DAKOTA SA in the SAW model navigator tree, including the method, strategy, responses, and variables.
   c) Support centered-parameter studies and sampling-based sensitivity studies.
4. ***Execute DAKOTA.***
   a) A local DAKOTA process will make call back to CSUI to perform workflow evaluations.
   b) Make resulting response data available for further processing and visualization.
5. ***Be flexible and robust to user process execution.***
   a) Allow deviations from intended process workflow.
   b) Provide feedback to indicate on readiness to proceed to the next step.

Requirement 1: Create Dakota Ready Models

- **Accomplishments**
  - Implemented an API on model components to query parameters/responses.
  - Created the Model Response Toolkit (MRT), a library of "recipes" that parse quantities of interest from Sierra models.
  - Created the UI to display, edit and reuse model parameters in the SAW Model Builder.
- **Impact**
  - The SAW graphical model building environment allows users to more easily develop parametric models in the UI.
  - The MRT allows users to easily extract response data from complex simulation outputs without tedious scripting, and in a form that can be used in Dakota studies.
  - The MRT architecture provides a framework to develop a richer set of recipes commonly used for extracting model responses.

Requirement 2: Execute SAW Model Builder Workflows

- **Accomplishments**
  - Integrated a 3rd party workflow engine for executing a linear Cubit → Sierra workflow.
  - Created an extensible framework for contributing workflow handlers to SAW model components.
  - Created two types of studies: (1) a parameter sample study for user-defined "what if" studies, and (2) a Dakota study for large ensembles of runs. Both study types execute the same workflow interface.
  - Display study results in the integrated model tree.
- **Impact**
  - Unified model-centric presentation helps users explore parametric models and better understand the effects of uncertainty on their problem domain.
  - Workflow architecture is not Sierra or Cubit specific. Any simulation code can be integrated with a workflow handler.
  - The 3rd party workflow engine can support more sophisticated workflows in a future development effort.

Requirement 3: Set up a Dakota Study

- **Accomplishments**
  - Implemented a next-generation Dakota UI.
    - Highly integrated into the SAW Model Builder. Studies are part of the model definition tree and are edited in the same way the model is edited.
    - Intuitive user interface is automatically generated from the Dakota XML language specification.
    - Contextual links to Dakota documentation.
  - Enhanced the New Dakota Study Wizard
    - Provides user guidance through the creation of a run-ready Dakota study based on the users goal (optimization, SA, UQ) and the model inputs/outputs.
- **Impact**
  - Makes uncertainty an integrated part of the model. Users need not be aware that there is a Dakota input file nor the intricacies of the input syntax.
  - The UI provides guidance to users not familiar with Dakota without limiting power users.
  - Reduces cycle time and errors, allowing users concentrate on problem domain.
  - Dakota UI stays in sync with future versions of Dakota via XML.

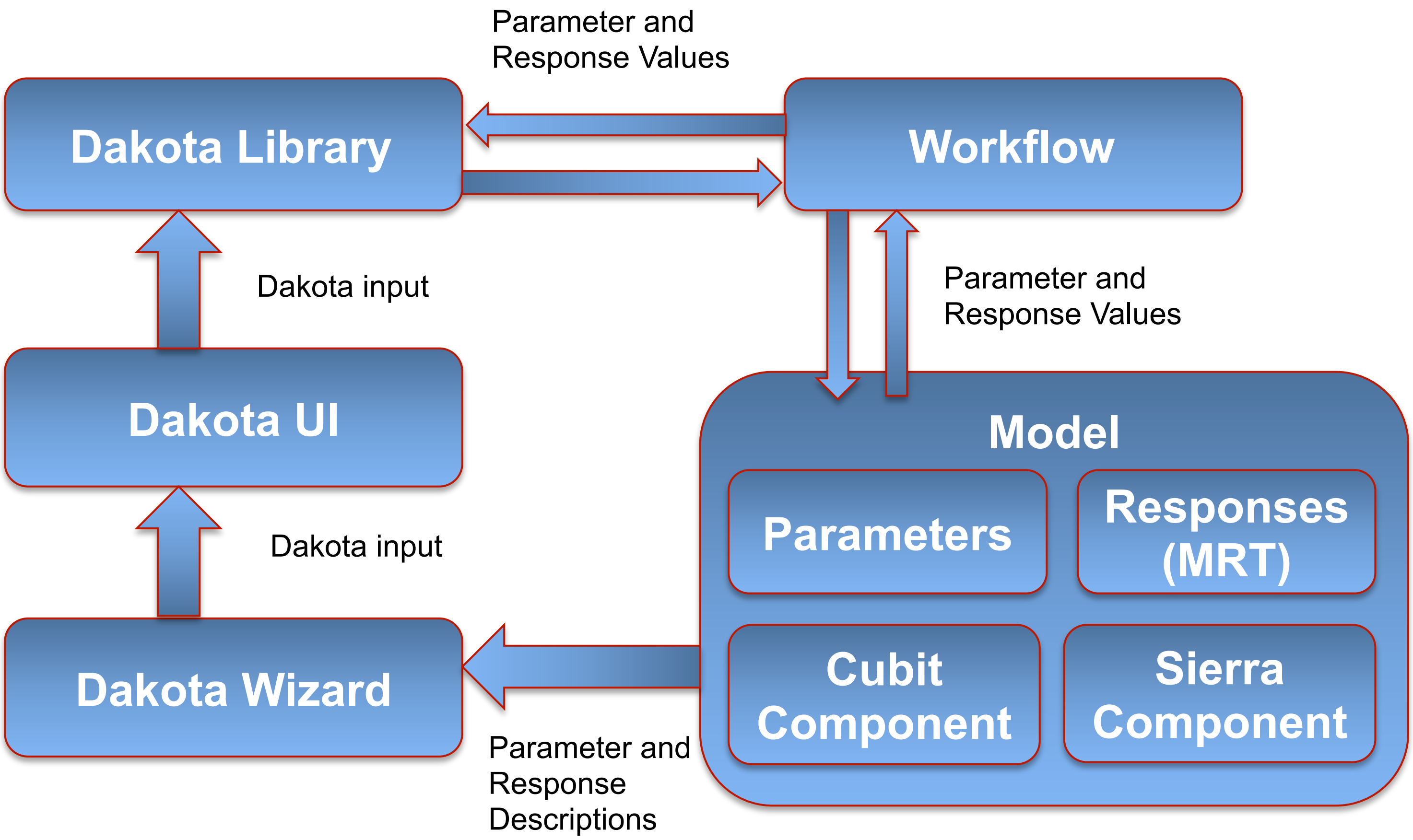
Requirement 4: Execute Dakota

- **Accomplishments**
  - Implemented a library interface to Dakota that allows a user to launch Dakota with provisions for Dakota to make callbacks to execute SAW workflows.
  - Packaged Dakota executable with the SAW application installer.
  - Created a mechanism to catch and report errors when running Dakota.
  - Created visualization tools to view Dakota results data (i.e. scatter plots).
- **Impact**
  - The tighter coupling of the library interface eliminates the tedious scripting of a driver script required for the more traditional loosely coupled mode of running Dakota.
  - The system architecture places the workflow engine in SAW, providing the foundation for integrating a more powerful workflow builder in the future.
  - Users can easily understand the effects of parametric changes on model responses.

Requirement 5: *Be flexible and robust to user process execution*

- **Accomplishments**
  - Provided flexibility to work in a nonlinear fashion, allowing users to change models (e.g. add parameters) and studies (e.g. add variables) after they are created.
  - Created a model validation framework to ensure that a model and its studies are in a consistent state (e.g. study variables match model parameters).
- **Impact**
  - Fits the way users work, allowing users to easily correct problems, make changes to studies and then resubmit.
  - Validating the entire Dakota study package of files greatly reduces the small errors that contribute to the fragility of Dakota studies, thereby reducing the cycle time to developing functional V&V studies.

System Architecture Overview



Dakota Library

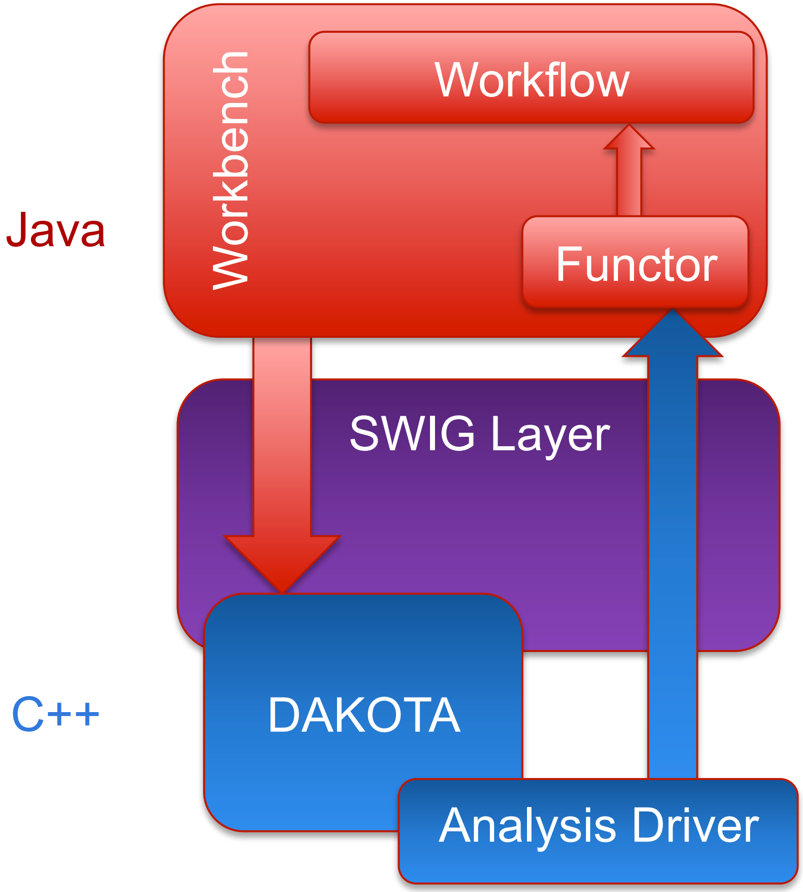
Java API for DAKOTA 6.0

Really two separate APIs:

1. Control DAKOTA from Java
    - Java "wrapper" for DAKOTA
    - Generated using SWIG
    - Utility methods in C++
2. Function evaluation in Java
    - C++ driver implementation
    - Java functor interface
        - Marshalling parameters
        - Returning responses
    - Java functor base class

Each API can be used independently.

In this implementation, everything is in one process, but this is not essential. Next year, we'll break this apart for distributed operation.



Model Response Toolkit (MRT)

General framework for representing workflow component responses. Several initial implementations are available:

1. Local Python
   Write custom scripts with access to components in running workflow. Execute in Jython inside Workbench.
2. Remote Python
   Write custom scripts with access to remote resources (Sierra outputs). Execute in Python interpreter on remote machine.
3. Sierra Heartbeat
   1. Choose built-in ("point and click") operations
   2. Write custom scripts, which execute in Python on remote machine.

Framework allows for more implementations, including more built-in operations. Convenience library will also be expanded.

Local Python Responses are for Cubit and Workbench scripting, and general local scripting

Remote Python Responses are for Sierra result extraction. Exomerge library is available to scripts.

Sierra Heartbeat Responses offer "point and click" access to various QOIs, or can be scripted. Pre-parsed heartbeat content is available to scripts.

Workflow

Leveraging a 3rd party workflow engine based on graph execution.

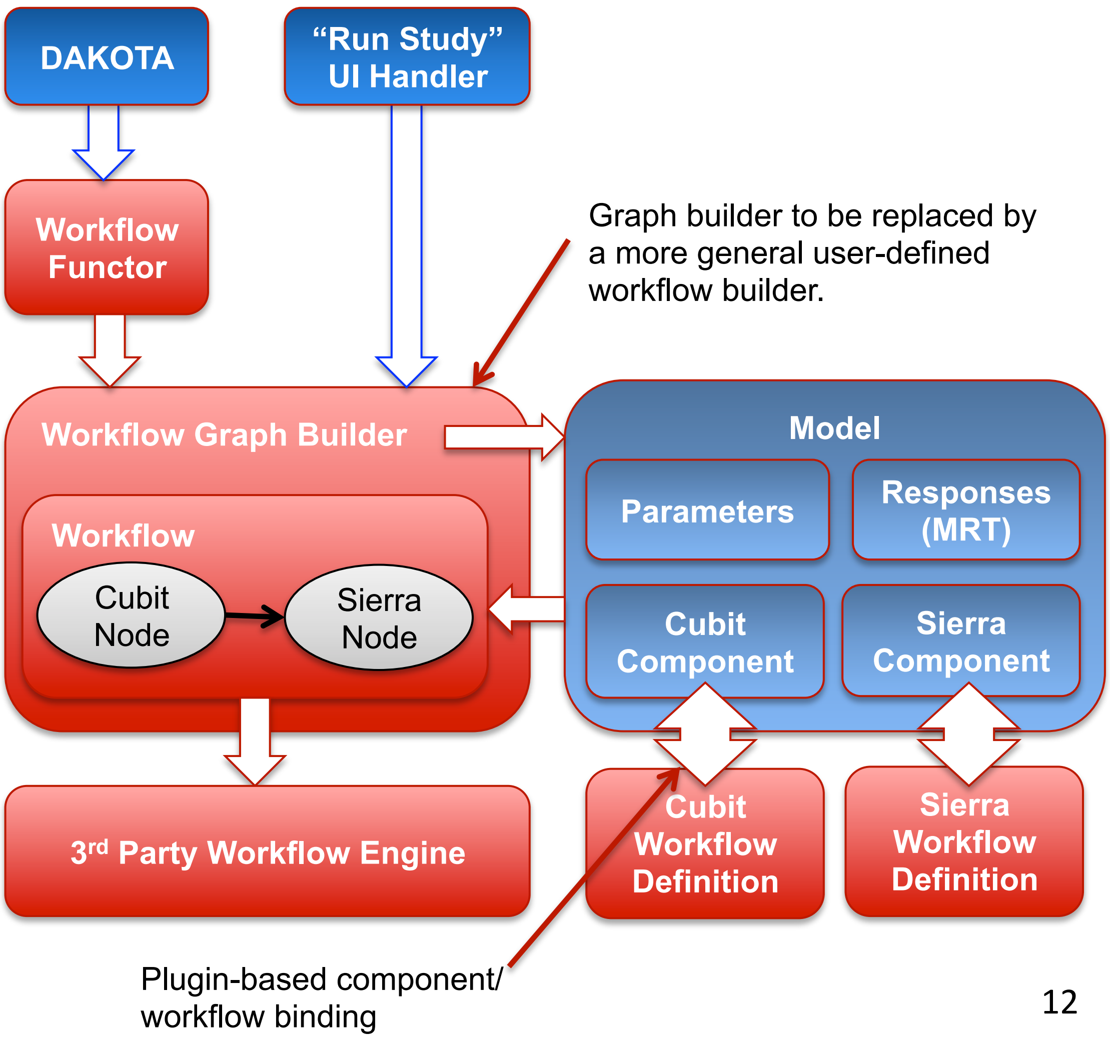
- Java based core.
- Highly customizable.
  - Node guards allow bypassing nodes under certain conditions (e.g. Skip Cubit node if geometry/mesh is not parameterized).
- Platform for a more general user-defined workflow builder.
  - XML based persistence of process definitions.
  - Visual editors and process monitor.

Implemented an extension-based workflow contribution mechanism.

- Model components are bound to a contributed workflow node class.

Workflow is created at runtime based on hardcoded rules.

- To be replaced with XML persisted process definition.

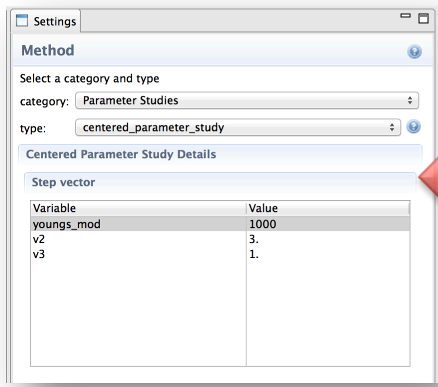


Dakota UI

Like Sierra UI, the Dakota UI is automatically generated from an XML language specification (Dakota XML).

- Extension point for contributing custom UI panels where required.
- UI will automatically reconfigure to new versions of Dakota XML ➔ sustainable solution.

Dakota UI Panel

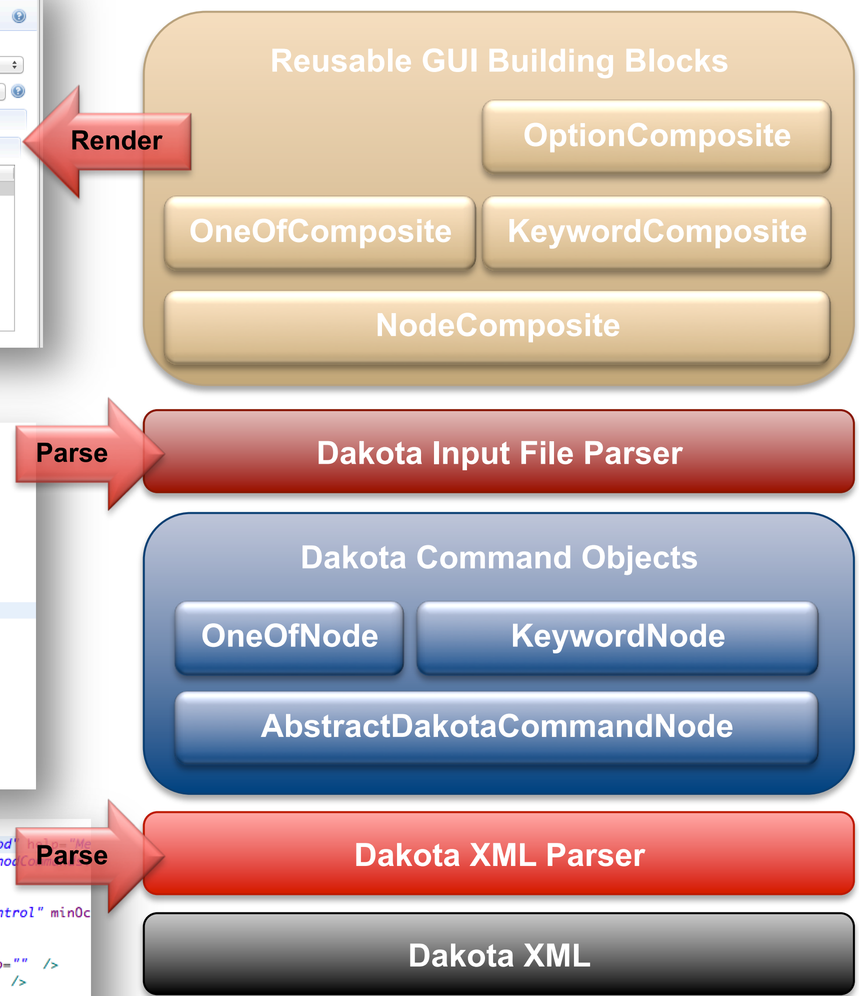


Dakota Input File

```
responses
interface
    analysis_drivers 'plugin_dart'
        direct
method
centered_parameter_study
    step_vector 1000 3. 1.
    steps_per_variable 4 4 4
model
    single
environment
    output_file 'dakota.out'
    tabular_data
        tabular_data_file 'tabular_graphics.out'
    results_output
        results_output_file 'results.out'
variables
uniform_uncertain 3
    lower_bounds 28.e6 2 3
    upper_bounds 32.e6 3 4
    initial_point 30.e6 1 2
    descriptors 'youngs_mod' 'v2' 'v3'
```

Dakota XML

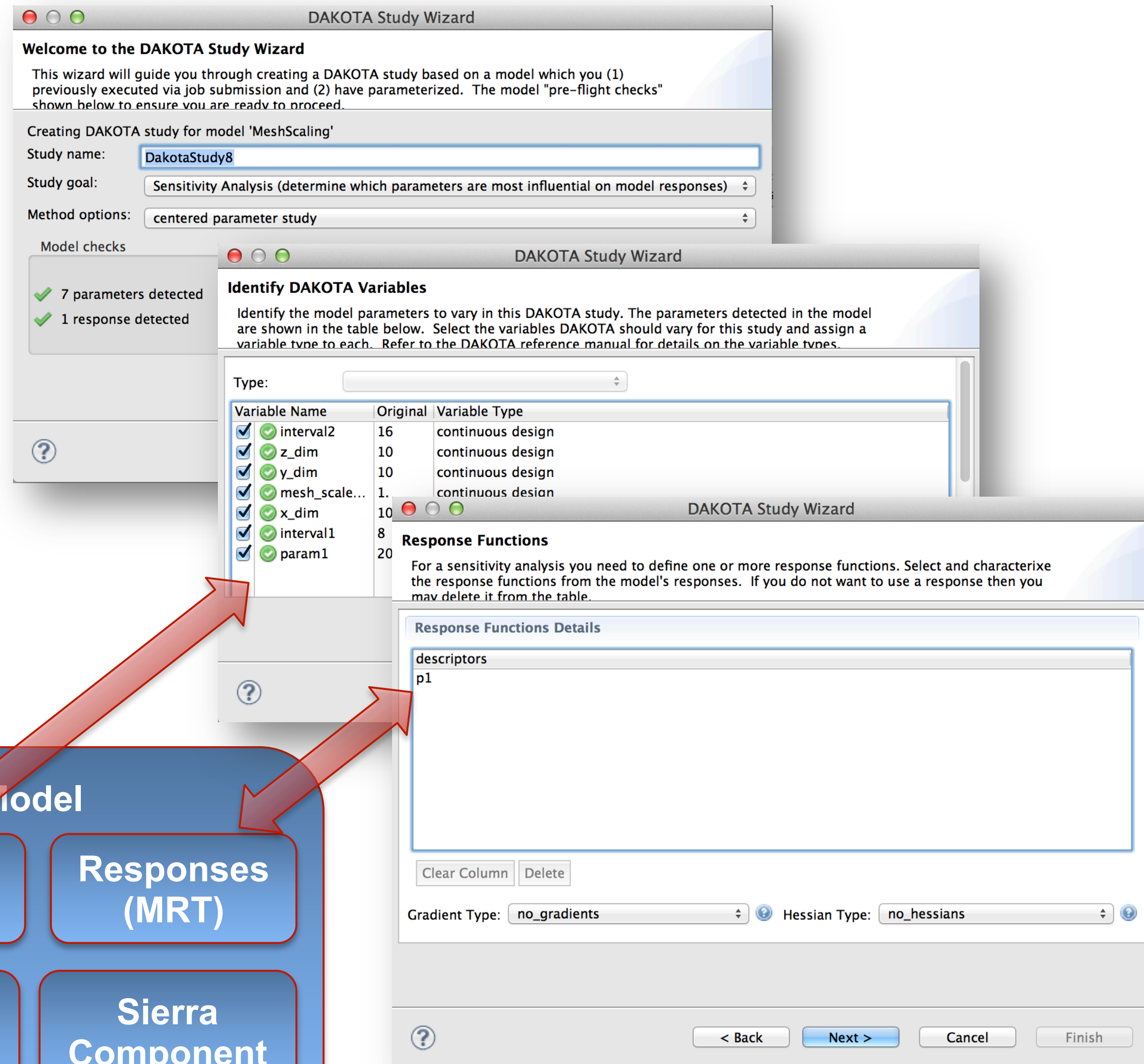
```
<!-- **** TOPLEVEL *** -->
<keyword id="method" name="method" minOccurs="1" maxOccurs="unbounded" code="{N_mdm3(start,0,stop)}" label="Method"
  <keyword  id="id_method" name="id_method" code="{N_mdm(str,idMethod)}" label="Method set identifier" help="Metho
    <param type="STRING" taglist="method" />
  </keyword>
  <keyword  id="output3" name="output" code="{0}" label="Output verbosity" help="MethodCommands.html#MethodIndControl" minOc
    <oneOf label="output level">
      <keyword  id="debug" name="debug" code="{N_mdm(type,methodOutput_DEBUG_OUTPUT)}" label="debug" help=""  />
      <keyword  id="verbose" name="verbose" code="{N_mdm(type,methodOutput_VERBOSE_OUTPUT)}" label="verbose" help=""  />
      <keyword  id="normal" name="normal" code="{N_mdm(type,methodOutput_NORMAL_OUTPUT)}" label="normal" help=""  />
      <keyword  id="quiet" name="quiet" code="{N_mdm(type,methodOutput_QUIET_OUTPUT)}" label="quiet" help=""  />
      <keyword  id="silent" name="silent" code="{N_mdm(type,methodOutput_SILENT_OUTPUT)}" label="silent" help=""  />
    </oneOf>
  </keyword>
```

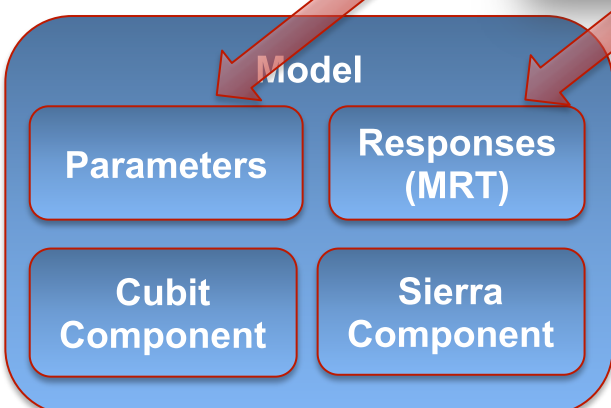


Dakota Wizard

The New Dakota Study Wizard was enhanced as follows:

- Goal driven studies
  - Goals: optimization, calibration, sensitivity analysis & uncertainty quantification.
  - Options for methods, variables and responses are tailored to the users goal.
- Wizard queries the model component interfaces for parameters and responses.
- Capable of producing a run-ready Dakota study.
- Wizard UI Panels are XML driven





What's Next?

- Harden and productionize V&V capabilities in upcoming release of Sandia Analysis Workbench.
  - Windows deployment of Dakota libraries
  - Add concurrency to embedded Dakota libraries
- Extend the packaged recipes for response extraction in the model response toolkit (MRT)
  - On-going effort performed in collaboration with the Dakota and V&V
  - Collaboration with Dakota and V&V groups to identify most common and useful recipes.
- Provide user guidance on other Dakota study types: calibration, optimization, and uncertainty quantification.
- Develop a more general workflow builder capability that allows users to create arbitrary workflows.