SANDIA REPORT
SAND2019-11311
Printed Click to enter a date

Sensitivity and Uncertainty Workflow of Full System SIERRA Models Supporting High Consequence Applications

FY19 ASC L2 Milestone (6802) Final Report

George E. Orient, Robert L. Clay, Ernest J. Friedman-Hill, Philippe P. Pebay, Elliott M. Ridgway

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico
87185 and Livermore,
California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: https://classic.ntis.gov/help/order-methods/

ABSTRACT

Credibility of end-to-end CompSim (Computational Simulation) models and their agile execution requires an expressive framework to describe, communicate and execute complex computational tool chains representing the model. All stakeholders from system engineering and customers through model developers and V&V partners need views and functionalities of the workflow representing the model in a manner that is natural to their discipline. In the milestone and in this report we define workflow as a network of computation simulation activities executed autonomously on a distributed set of computational platforms.

The FY19 ASC L2 Milestone (6802) for the Integrated Workflow (IWF) project was designed to integrate and improve existing capabilities or develop new functionalities to provide a wide range of stakeholders a coherent and intuitive platform capable of defining and executing CompSim modeling from analysis workflow definition to complex ensemble calculations. The main goal of the milestone was to advance the integrated workflow capabilities to support the weapon system analysts with a production deployment in FY20. Ensemble calculations supporting program decisions include sensitivity analysis, optimization and uncertainty quantification. The goal of the L2 milestone aligned with the ultimate goal of the IWF project is to foster cultural and technical shift toward and integrated CompSim capability based on automated workflows.

Specific deliverables were defined in five broad categories: 1) Infrastructure, including development of distributed-computing workflow capability, 2) integration of Dakota (Sandia's sensitivity, optimization and UQ engine) with SAW (Sandia Analysis Workbench), 3) ARG (Automatic Report Generator introspecting analysis artifacts and generating human-readable extensible and archivable reports), 4) Libraries and Repositories aiding capability reuse, and 5) Exemplars to support training, capturing best practices and stress testing of the platform. A set of exemplars was defined to represent typical weapon system qualification CompSim projects. Analyzing the required capabilities and using the findings to plan implementation of required capabilities ensured optimal allocation of development resources focused on production deployment after the L2 is completed. It was recognized early that the end-to-end modeling applications pose a considerable number of diverse risks, and a formal risk tracking process was implemented. The project leveraged products, capabilities and development tasks of IWF partners. SAW, Dakota, Cubit, Sierra, Slycat, and NGA (NexGen Analytics, a small business) contributed to the integrated platform developed during this milestone effort. New products delivered include: a) NGW (Next Generation Workflow) for robust workflow definition and execution, b) Dakota wizards, editor and results visualization, and c) the automatic report generator ARG. User engagement was initiated early in the development process eliciting concrete requirements and actionable feedback to assure that the integrated CompSim capability will have high user acceptance and impact.

The current integrated capabilities have been demonstrated and are continually being tested by a set of exemplars ranging from training scenarios to computationally demanding uncertainty analyses. The integrated workflow platform has been deployed on both SRN (Sandia Restricted Network) and SCN (Sandia Classified Network). Computational platforms where the system has been demonstrated span from Windows (Creo the CAD platform chosen by Sandia) to Trinity HPC (Sierra and CTH solvers). Follow up work will focus on deployment at SNL and other sites in the nuclear enterprise (LLNL, KCNSC), training and consulting

support to democratize the analysis agility, process health and knowledge management benefits the NGW platform provides.

ACKNOWLEDGEMENTS

The IWF team would like to acknowledge the consistent support from the ASC sponsors: Scott Hutchinson, Walt Witkowski, Ken Alvin, Tom Klitsner, Jeremy Templeton, Erik Strack, and Amanda Dodd. Without their support this integrated effort would not have been possible.

We would also like to thank the milestone review panel for their insightful feedback and guidance throughout the year: Martin Heinstein, Patty Hough, Jay Dike, Dan Laney (LLNL), and Jay Billings (ORNL).

And of course, without the hard work of the IWF team none of this would have happened.

CONTENTS

1 Next Generation WorkfloW PLATFORM 16
1.1. Requirements, Security Constraints 16
1.1.1. Usability 16
1.1.2. Capability 16
1.1.3. Security 16
1.2. Architecture Overview 17
1.2.1. The Workflow Editor 17
1.2.2. The Workflow File Format 17
1.2.3. The Workflow Engine 18
1.3. Overview of Current Capabilities 18
1.3.1. Workflow Construction and Visualization 19
1.3.2. Distributed and Resilient Workflow Execution 19
1.3.3. Selected Foundational Workflow Nodes 19
1.3.3.1. Creo Assembly Regeneration and Metadata Exploration 19
1.3.3.2. Cubit Workflow Node 20
1.3.3.3. Sierra Workflow Node 20
1.3.3.4. Dakota Workflow Node 20
1.3.3.5. Script Nodes 21
1.4. Plans for Future Work 21
2 DAKOTA usability developments 22
2.1 Study Definition Wizard 22
2.2 Input File Editor 23
2.3 Hierarchical Results Database Structure in HDF5 23
2.4 Results Visualization 23
2.5 Integration of Dakota User Interface Components with NGW 24
2.6 Plans for Future Work 24
3 Automatic Report Generator 25
3.1 Multi-Stage Document Construction 25
3.2 Architecture Overview 26
3.3 Solid Mechanics Capabilities 26
3.4 Structural Dynamics Capabilities 27
3.5 Plans for Future Work 28
4 Exemplars of End-to-End Sensitivity, Optimization and Uncertainty Quantification WorkflowS 29
4.1 Exemplar 2 - Storage Tank 30
4.1.1 Problem Definition 31
4.1.2 Analysis and Documentation Workflow 32
4.1.3 Solution Verification 35
4.1.4 Optimization 37
4.1.5 Deterministic Sensitivity 39
4.1.6 Probabilistic Sensitivity 39
4.1.7 Uncertainty Quantification 41
4.1.8 Plausible Program Decisions 42
4.2 Exemplar 4 - Abnormal Mechanical Crush of an Assembly 43
4.2.1 Problem Definition 43

4.2.2 Analysis and Documentation Workflow 45
4.2.3 Scalability 45
4.2.4 Parallel consistency 46
4.2.5 Implications on Model Credibility 47
5 Libraries of Reusable CompSim and training Resources 49
5.1 Engineering Sciences Tool Repository (ESTR) 49
5.2 NGW End-to-End Exemplar Repository 49
5.3 NGW Unit Training Scenario Repository 49
5.4 ARG Exemplar Repository 50
6 User Engagement 51
7 Follow-on Work 52
8 Summary 53
References 54

LIST OF FIGURES

Figure 1-1: The NGW editor 17
Figure 1-2: The layered architecture of the NGW runtime system. The thin compatibility layer means that the runtime can be easily ported to other workflow engines 18
Figure 2-1: Screenshot of the Dakota study definition wizard 22
Figure 2-2: Human readable tabular input 23
Figure 2-3: Hierarchical Dakota input and output data structure 23
Figure 2-4: Visualization of the optimization method 24
Figure 3-1: Templatized analyst authored results 25
Figure 3-2: Mesh visualization 26
Figure 3-3: Constitutive model curve plot 27
Figure 3-4: FEM mass properties reporting 27
Figure 3-5: Mesh quality reporting 28
Figure 4-1: End-to-end exemplars to support requirement definition, stress testing and training 29
Figure 4-2: CompSim credibility inspired folder structure 30
Figure 4-3: Storage Tank geometry and mechanical boundary condition definition 31
Figure 4-4: Model parameters and responses 32
Figure 4-5: Top level analysis workflow 33
Figure 4-6: Execute Creo 34
Figure 4-7: Cubit-Sierra-ParaView-ARG workflow 34
Figure 4-8: ARG report including analysis results 35
Figure 4-9: Dakota solution verification study workflow 36
Figure 4-10: Mesh convergence study for g_yield 37
Figure 4-11: Evolution of structural mass during the optimization process 37
Figure 4-12: Evolution of the displacement constraint during the optimization process 38
Figure 4-13: Design space coverage 38
Figure 4-14: Deterministic sensitivity, main effects 39
Figure 4-15: Uncertainty inventory 40
Figure 4-16: Uncertainty informed sensitivity 40
Figure 4-17: Histogram of normalized displacement constraint violation 41
Figure 4-18: Incremental LHS convergence 41

Figure 4-19: Sensitivity, partial rank correlation 42
Figure 4-20: Crush of an assembly 43
Figure 4-21: Calculation of SRS 43
Figure 4-22: Gap opening between "Case" and "Plug" 44
Figure 4-23: Model parameters and responses 44
Figure 4-24: Atomic activities implemented as sub-workflows 45
Figure 4-25: Scalability at 50 ft/s impact speed 46
Figure 4-26: Viewing ARG reports for study samples in Slycat 47
Figure 4-27: Viewing deformation and SRS sections of ARG report in Slycat 47
Figure 5-1: ESTR contents 50

This page left blank

EXECUTIVE SUMMARY

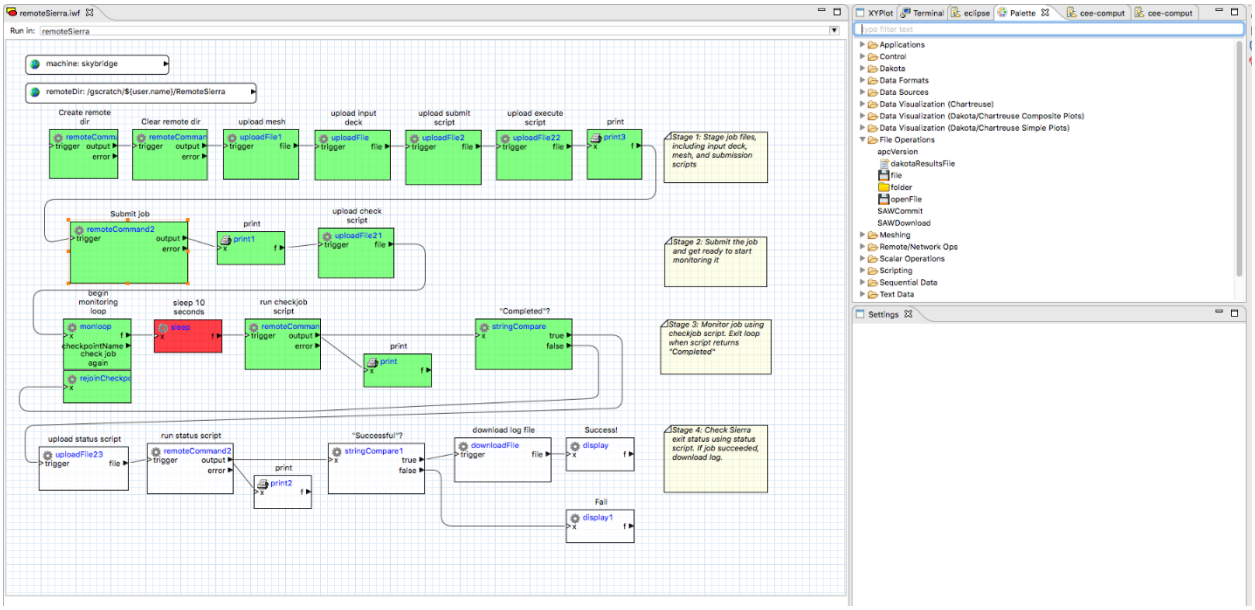
The FY19 ASC L2 Milestone (6802) for the Integrated Workflow (IWF) project was designed to integrate and improve existing capabilities or develop new functionalities to provide a wide range of stakeholders a coherent and intuitive platform capable of defining and executing CompSim (Computational Simulation) modeling from analysis workflow definition to complex ensemble calculations. The main goal of the milestone was to advance the integrated workflow capabilities to support the weapon system analysts with a production deployment in FY20. In the milestone and in this report we define workflow as a network of computation simulation activities executed autonomously on a distributed set of computational platforms. Ensemble calculations supporting program decisions include sensitivity analysis, optimization and uncertainty quantification. The goal of the L2 milestone aligned with the ultimate goal of the IWF project is to foster cultural and technical shift toward and integrated CompSim capability based on automated workflows.

The following specific deliverables were defined for the L2 milestone:

1. Infrastructure to define and execute complex multi-platform engineering workflows in a user friendly, robust and responsible manner.
2. Library mechanism to author and distribute configuration-controlled tools (workflow components) generated by analysts, typically to extract QoIs from large datasets generated by a single analysis.
3. Integration between Dakota and the NextGen Workflow (NGW) infrastructure focused on directly supporting analysis of the parameter and response space.
4. Automatic report generation for each instance of the analysis workflow and each instance of ensembles.
5. Documented end-to-end SM exemplar supporting training and reducing analyst-to-analyst variation by encapsulating best practices.

A summary of accomplishments for each milestone deliverable follows.

Infrastructure: Traditional script-based construction of analysis and ensemble workflows requires specialized skills orthogonal to the engineering disciplines, it may not support rapid design cycles, and it doesn't result in a peer reviewable communication of the CompSim process. The Next-



The NGW editor.

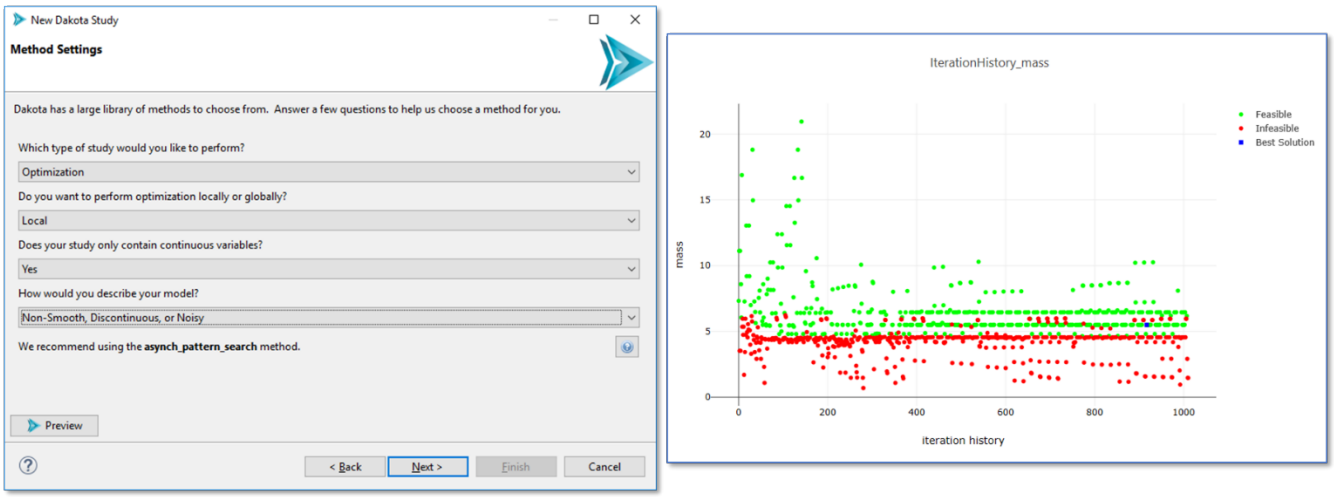
Generation Workflow system (NGW) has been designed to offer solutions to these issues in a pragmatic and flexible way. A graphical editor for building NGW workflows is integrated with the SAW desktop client, and a portable, decoupled, standalone workflow engine that executes these workflows can be distributed from the desktop to high-performance compute clusters. The NGW system leverages open-source components to create a powerful but compact software system which is highly customizable and extensible. A palette of high functionality tools, graphical workflow authoring, robustness, resilience and efficiency features has been implemented to support workflow authoring, resilient distributed execution and inter-team communication. The user-facing NGW editor is illustrated below.

Library Mechanism – Important tools such as quantitative post processors that program decisions are based on are often not under configuration control, and their sharing is usually limited to workgroups. The existing SAW SDM (Simulation Data Management) configuration and access control mechanisms were used to collect a set of reusable CompSim tools in a repository called 'Engineering Sciences Tool Repository'. Tools contributed by several analysts include:

- Creo (CAD) assembly explorer to regenerate parametric assemblies and interrogate metadata as well as mass properties to support credibility communications regarding the adequacy of FEM mesh to represent the mass of each part.
- A set of signal processing tools to post process transient dynamics analyses from Sierra including shock response spectrum generation, and signal conditioning.
- Solid Mechanics data conditioning tools to aid constitutive model calibration.
- Validation metrics library to quantify difference between test data and predicted responses.

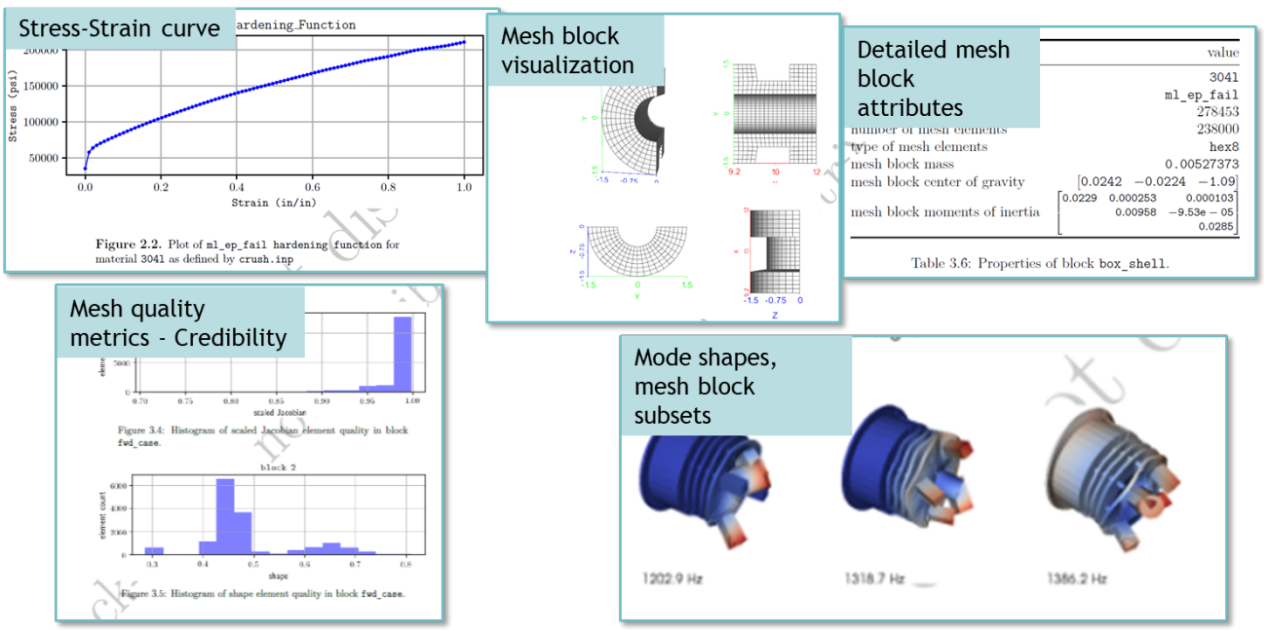
Additionally, repositories of the workflow application exemplars and simple training unit tasks were also constructed.

Dakota Integration – Sandia's Dakota platform and method library provides a proven set of powerful optimization, sensitivity analysis and uncertainty quantification capabilities, but the traditional input file format is difficult to read and author. Additionally, visualization of study results is left to the analyst resulting in multiple visualization tools developed by analysts to meet their particular needs. Extensive usability improvements implemented to support the L2 milestone include study-definition wizards to aid constructing studies from existing analysis workflows, input file editors that represent the sometimes-difficult-to-read Dakota input file in a usable form-based interface and an extensive set of result visualization tools including support for optimization, sensitivity and UQ. The study wizard and visualization of results for an optimization method are illustrated below.



Dakota wizard and results visualization for optimization

ARG (Automatic Report Generator) – ARG is capable of introspecting analysis artifacts and generating human readable extensible and archivable reports. Reporting analysis instances in a complete manner is a tedious error prone process and maintaining association between analysis and report is difficult. ARG is a platform for auto-generating reports from CompSim artifacts enabling integration of analyst authored content. Currently, Sierra SM and SD are supported. The ARG tool examines artifacts produced by a completed analysis and collates the information in a human readable and extensible form. Additionally, Creo (CAD) assemblies are reported to support comparing geometric fidelity through visual examination of CAD parts and FEM blocks, as well as quantitative comparison of their masses based on part material densities obtained from Granta. Model credibility is further supported by reporting provenance metadata embedded in the CAD model. SAND report format with appropriate markings for OUO and classified reports and SAND TK (Test Plan) formats as well as generic reports are available. The figure below illustrates some of the ARG reporting capabilities.



Selected ARG reporting capabilities

End-to-end exemplars – Since rolling out the developed CompSim capabilities to a wide analyst base is a key goal of the SAW development team in this milestone special emphasis was placed on the

Draft

	Analysis Workflow					Ensemble Workflow						Visualization	
	Geometry Preparation	Meshing	Solution	Reporting	Computationally Demanding, Needs Resiliency	Design Optimization	Scalability, Parallel Consistency	Solution Verification	Uncertainty Informed Sensitivity Analysis	Uncertainty Analysis	QMU	Slycat	Dakota Composite Plots
Exemplar 1 -Simple tank partially filled with liquid	N/A	N/A	Analytic	ARG Generic	no	✓	N/A	✓	✓	✓	✓	✓	✓
Exemplar 2 - Tank assembly partially filled with liquid	Creo	Cubit	Sierra imlpicit	ARG SAND	no	✓	✓	✓	✓	✓	✓	✓	✓
Exemplar 3 - Explosion beneath lattice structure	CTH diatoms	CTH grid	CTH	None	yes	✗	✓	✓	✓	✓	✗	✓	✓
Exemplar 4 - Abnormal mechanical crush	Creo	Cubit	Sierra explicit	ARG SAND	yes	✗	✓	✓	✓	✓	✗	✓	✓

End-to-end exemplars to support requirement definition, stress testing and training

requirement that the developed capabilities support realistic use scenarios. A summary of the end-to-end analysis workflow and study workflow developed under the L2 is shown below.

- Simple tank partially filled with liquid: relatively complex closed form solution of a structural engineering problem with simplified boundary conditions. Executes quickly, requires no FEM, supports training with a realistically complex set of workflows. ARG report communicates stress field and structural performance. Structural optimization, solution verification, sensitivity analysis and UQ (Uncertainty Quantification) and QMU (Quantification of Margins and Uncertainties) studies.
- Tank assembly partially filled with liquid: Parametric Creo (CAD) model connected to Cubit meshing and Sierra solution followed by ARG report. Structural optimization, sensitivity analysis and UQ studies.
- Explosion beneath a lattice structure: CTH model illustrating generality of the framework; computationally intensive. Parallel consistency and scalability studies, sensitivity analysis and UQ studies.
- Abnormal mechanical crush: A realistic small assembly (10 parts) subjected to crush representative of scenarios derived from abnormal mechanical system requirements. Parametric Cubit meshing, Sierra explicit solution followed by quantitative Ensight and Python post processing summarized in an ARG report. Computationally intensive. Parallel consistency, scalability and mesh resolution studies.

The new capabilities, usability focused update and integration of existing tools and case studies demonstrate that the milestone met its goals. Early user involvement assured that a core set of requirements was identified, and capabilities required to meet them were successfully implemented. Formal risk management informed deployment of the team's diverse talents in a priority-driven manner. The resulting platform is now ready for deployment to analysts to transform how CompSim models are planned, developed, executed. The developed capabilities are foundational to CompSim credibility, development of models in an agile manner and decision support focused communication of results.

ACRONYMS AND DEFINITIONS

Abbreviation	Definition
ARG	Automatic Report Generator
CAD	Computer Aided Design
CompSim	Computational Simulation
IWF	Integrated Workflow
NGW	Next Generation Workflow
SAW	Sandia Analysis Workbench
SDM	Simulation Data Manager

1 NEXT GENERATION WORKFLOW PLATFORM

Repeatable automated workflows are an essential part of validation, verification and uncertainty quantification (V&V/UQ) in computational simulation-based engineering (1), (2), (3). The Next-Generation Workflow system (NGW) has been designed to meet this need in a pragmatic and flexible way (4). A graphical editor for building workflows is available integrated with the SAW desktop client (5) as well as a stand-alone open source Eclipse application (6), and a portable, decoupled, standalone workflow engine that executes these workflows can be distributed from the desktop to HPC systems (e.g., Trinity). The NGW system leverages open-source components to create a powerful but compact software system which is highly customizable and extensible. In this chapter we will describe the overall system architecture and explore the function of the editor and workflow engine.

1.1. Requirements, Security Constraints

For an engineering workflow system to be successful in our environment, it must simultaneously meet requirements for usability, capability, portability and security. In this section we will briefly enumerate the assumptions we made about these requirements and how our system would meet them.

1.1.1. Usability

First and foremost, workflows should be *built* from components that represent ideas and actions familiar to the user, rather than being programmed in an abstract language. A successful workflow system should impose a *low cognitive load*, permitting users to think more in their application domain and less about the workflow system itself. The spatial organization of a graphical workflow helps the user organize and convey their understanding.

Although a runtime system for executing workflows may be internally complex, the workflows themselves should be *transparent* and *escapable*: it should always be obvious what the system is doing, and it should always be possible for the user to customize or override behavior when needed. Such a system provides assistance to the user without restricting them.

A successful workflow system needs a flexible system architecture. Being able to execute the various components of a workflow on desktop machines, servers, or HPC systems, and to change where components are executed over time, will significantly help engineers when building and debugging a complex computational model.

1.1.2. Capability

Workflows must ideally be able to execute and control all non-interactive analysis codes and tools in use at Sandia. Workflows must be able to invoke other workflows, either on the same computer or on a remote computer. Specifically, NGW must support hierarchical, nested workflows. It must be possible to invoke a workflow from the command line and run it without a graphical interface. The workflow editor must be able to be integrated with the SAW user interface, and portable to all desktop platforms in use at Sandia.

1.1.3. Security

Because the workflow system will be used on a heterogenous network with compartmentalized administrative control, it must be possible for an ordinary user to install the system entirely in their own file space. It must not require the installation of a persistent network server, nor can it require administrative access.

The workflow system must be able to use existing Kerberos infrastructure for authentication. It must be able to negotiate links to remote compute systems accessible only through a "jump host."

1.2. Architecture Overview

The Next Gen Workflow system consists of two major components: a graphical environment for building and monitoring workflows, and a runtime engine for executing them. These two components are completely separable; the runtime engine has no dependencies on anything in the graphical environment, and vice-versa. This is deliberate – the design allows for each piece to evolve separately or be replaced with an independent, alternate implementation (for example, a workflow engine written in Python, or a command-line workflow builder). They are connected by a dedicated file format, described below.

1.2.1. The Workflow Editor

The NGW graphical workflow editor is built on the same Eclipse (7) technology stack that underlies the SAW desktop environment, and so interoperates well with SAW. Nevertheless, it is independent of the rest of SAW and can be used in Eclipse with no other SAW components.

The graph representation used in the editor is provided by the Eclipse Modeling Framework (EMF) (8), while the graphical editor is built on the Graphiti (9) framework developed by SAP.

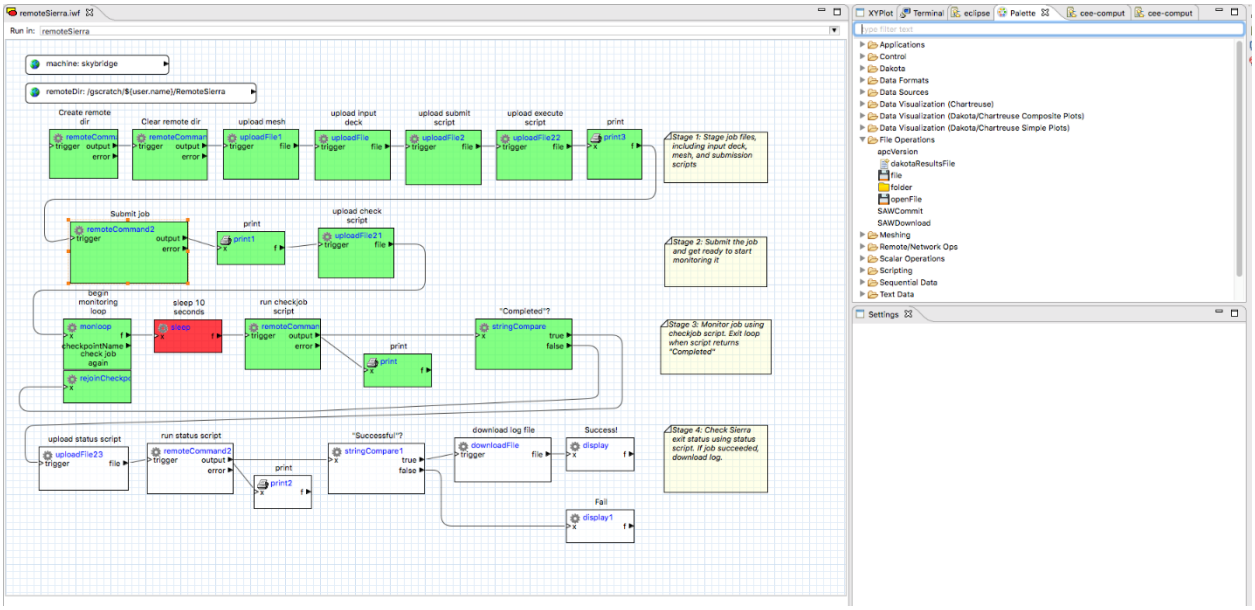


Figure 1-1: The NGW editor

1.2.2. The Workflow File Format

The editor stores workflow definitions in a simple XML file format that the workflow engine can read directly. The files contain information about node configuration, connectivity, and diagram layout, but not about the implementation of the node's functionality, which lies in the workflow engine itself, nor node appearance, which is implemented in the editor. The format is simple enough for a human to understand by inspection and is easily parsed. It is based on the EMF persistence format but can be processed with any XML parser.

1.2.3. The Workflow Engine

The NGW runtime system has a layered architecture designed to be implemented on top of a third-party workflow engine. Our current implementation uses the Sarasvati (10) workflow engine created under the Google Code project. A thin compatibility layer isolates the workflow engine from the bulk of the runtime system – the NGW communications layer, which moves data between workflow nodes, and the node implementations themselves. By using a thin compatibility layer, we can deploy on arbitrary third-party workflow engines with little effort. NGW workflow nodes are implemented with reference only to the portability layer.

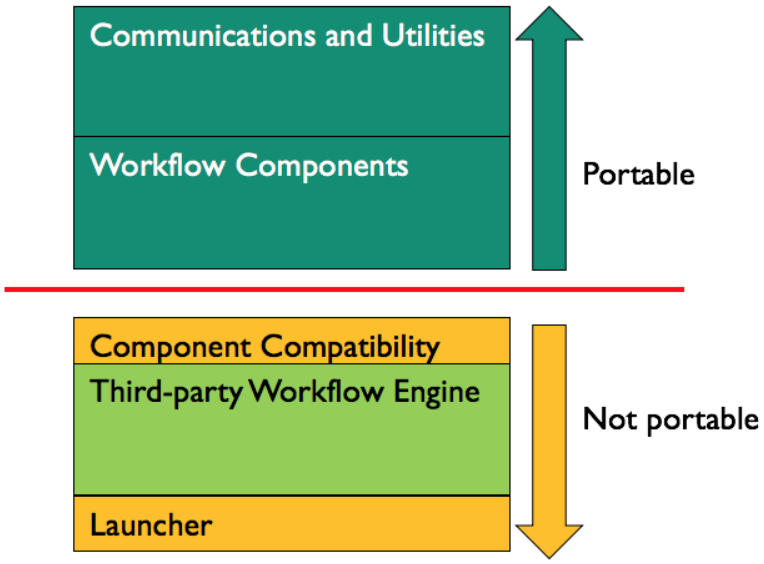


Figure 1-2: The layered architecture of the NGW runtime system. The thin compatibility layer means that the runtime can be easily ported to other workflow engines

1.3. Overview of Current Capabilities

The NGW system is proving to be a powerful addition to the analyst's toolset. In this section we will review some of the major capabilities

1.3.1. Workflow Construction and Visualization

The graphical workflow editor lets the user quickly assemble and configure workflows to execute a wide range of tasks. It is designed to require a minimum number of clicks to compete most tasks. A hierarchical palette of available workflow node types makes it easy to find capabilities. The workflow editor can also monitor the status of a running workflow, or display the status of a completed workflow, using data stored to the working directory of a workflow while it executes. In combination with the SAW Project Navigator, the status of an entire ensemble of workflows (executed via Dakota (11) or with NGW's own iteration mechanisms) can be monitored.

1.3.2. Distributed and Resilient Workflow Execution

Connections are the mechanism for data exchange between workflow nodes. Each connection originates on one node and terminates on another and can carry data between those two nodes.

Properties of the connection let the user control how this data exchange happens; for example, the user can indicate that the exchanged data represents a file path, and the file thus named should be linked, copied, or renamed in transit. Although connections typically transmit data within a single workflow, a workflow can contain nodes that themselves invoke other workflows, either locally or remotely. Depending on the type of connected nodes, therefore, a connection may transparently represent inter-process or network communication. Error handling associated with this data transmission—e.g., remote connection retry—is handled by the underlying system. Using the consistent connection metaphor enables a user to build a distributed workflow as easily as they can build an entirely local one.

Many workflow nodes store state information on successful completion. If a workflow is executed again using the same working directory, that state information will automatically be examined and used, if possible, to avoid computation and provide a previously computed result. This process of *memoization* makes incremental development and debugging of workflows faster and more efficient by preventing redundant computation.

A user-configurable resiliency module—the "Foo Fighter"—can supervise the execution of a nested workflow, responding to anticipated error conditions in a node-dependent manner, typically by retrying or aborting a workflow. In this way, site-specific problems with networks, machines, or codes can be handled in an extensible way.

1.3.3. Selected Foundational Workflow Nodes

As of this writing, over a hundred different kinds of workflow nodes have been implemented, with new ones being created all the time. In this section we will discuss a few of special relevance to the exemplar problems presented in this report.

1.3.3.1. Creo Assembly Regeneration and Metadata Exploration

Sandia's standard CAD program Creo (12) runs only on the Windows OS, and Creo execution is tied to a license server with a limited number of licenses. To facilitate the use of Creo for parametric geometry regeneration—especially as part of an ensemble workflow that includes components running on HPC systems—we developed a Windows-based Creo server which can accept authenticated requests from remote systems, maintain an ordered queue of requests, and return requested results. A set of dedicated workflow nodes were created to send commands to the Creo server, check its status, and upload and download files.

1.3.3.2. Cubit Workflow Node

The CUBIT meshing and geometry toolkit (13) is represented in NGW by a dedicated node that can import arbitrary files, execute CUBIT journal files or Python scripts, and export any generated data. The CUBIT node hides platform dependencies associated with scripting CUBIT across various operating systems. A set of nodes for mesh file manipulation and assembly complement the CUBIT node.

1.3.3.3. Sierra Workflow Node

The Sierra suite of analysis codes (14) is Sandia's premier set of solvers for mechanical and thermal/fluid physics problems. The Sierra workflow node controls Sierra configuration and execution and helps with marshalling inputs and outputs.

The Sierra workflow node manages a set of four scripts, similar to the four scripts used in classic SAW job submission: a *submit script* (which is used to submit a job to a queueing system,) an *execute script* (which executes the actual Sierra code) a *checkjob script* (which queries the queuing system for job status) and a *status script* (which examines the outputs of a job to confirm successful execution.) A local installation of NGW includes a default set of scripts tailored for the local environment: for example, at Sandia, the submit script runs a job using Slurm, while at Goodyear, it uses LSF.) The user has access to the default scripts and can supply a customized version of any or all of them.

The Sierra node can operate in direct or queued mode. In direct mode, Sierra is executed directly, without using a queue; in this case only the execute and status scripts are used. In queued mode, the Sierra job is submitted to a queue on the local machine. If remote submission is desired, then the "*remoteNestedWorkflow*" node can be used to execute the job submission process on a remote machine.

Note that the Sierra workflow node is a specialization of a general "*queueSubmit*" node, which can be used with any HPC code.

1.3.3.4. Dakota Workflow Node

The Dakota[1] toolkit—used to design and run ensembles of evaluations for the purpose of optimization, sensitivity studies, etc.—is well-integrated with NGW. In particular, it is easy to author a workflow in which Dakota appears as a node, and which executes a second workflow as its model. All the complexity of transferring data to and from Dakota is handled transparently. More details are provided in the next chapter.

1.3.3.5. Script Nodes

Nodes have been defined to support execution of user scripts in many languages—Python, Bash, Windows Batch, and more. It is a fact of life that the flexibility of user scripting is needed for many workflows. The design of NGW allows the workflow author to do the minimal amount of scripting necessary. As a result, scripts embedded in NGW script nodes tend to be much shorter and simpler than standalone scripts would be.

1.4. Plans for Future Work

We anticipate that NGW will quickly become a widely-used capability at Sandia. Experiences with that user base will of course guide us as we develop NGW further, but there is a number of enhancements already planned. For example, widening the scope of memoization so that partial results can be shared across an entire ensemble will be well worth the investment.

2 DAKOTA USABILITY DEVELOPMENTS

The Dakota toolkit provides a flexible, extensible interface between simulation codes and a variety of iterative systems analysis methods, including optimization, uncertainty quantification, deterministic/stochastic calibration, and parametric/sensitivity/variance analysis. In our community, it is central to the design, creation and execution of ensembles of computational models. The last few years have seen a renewed effort to provide a rich user experience for Dakota users and seamless integration between Dakota and Sandia's computational simulation capabilities. The present effort significantly advanced the agility and ease of use of Dakota.

2.1 Study Definition Wizard

The New Dakota Study wizard assists analysts in quickly creating Dakota studies by using previously defined input variables and output responses for their simulation model. The analyst need not have a knowledge of Dakota input file syntax to use the wizard. Instead, the wizard guides the analyst through a series of questions to help them select an appropriate Dakota method for their problem. Once the method has been selected, the wizard fills out the remaining details using "best practice" heuristics for that method. The wizard also assists the analyst in making appropriate selections for Dakota's variable, response, and interface blocks.

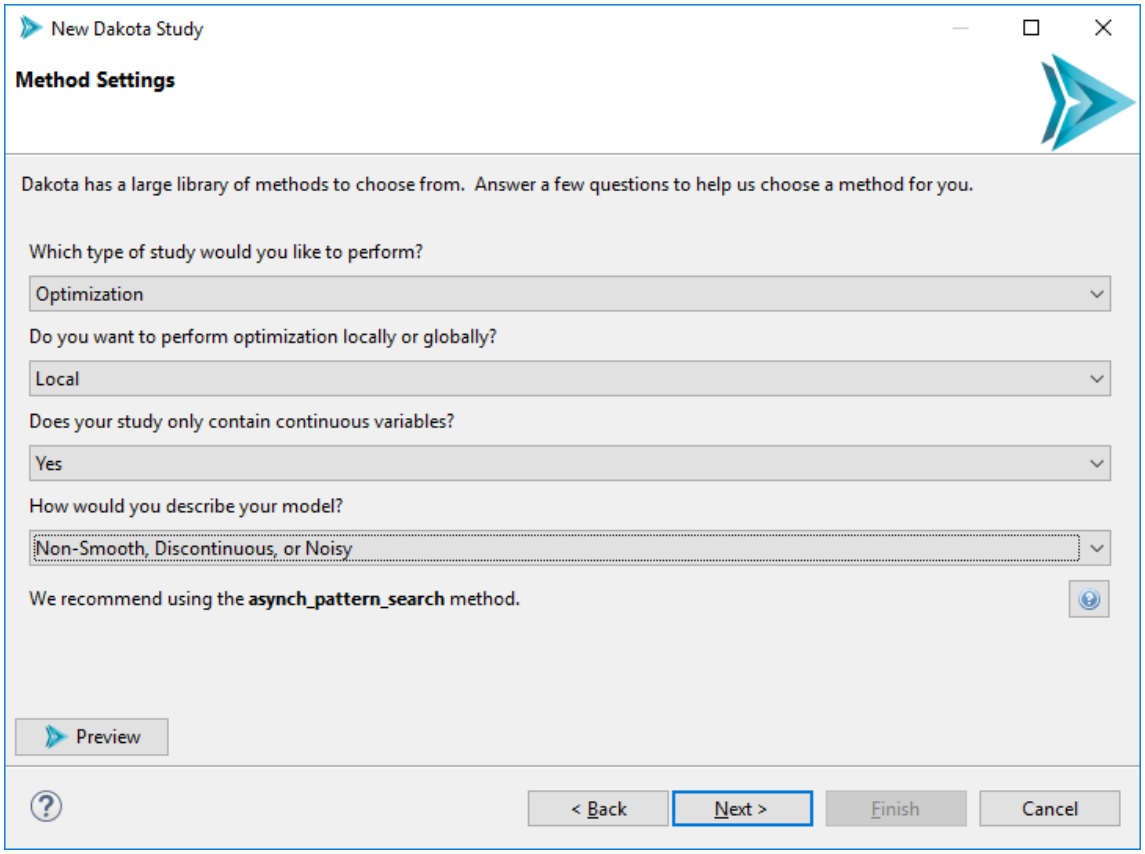


Figure 2-1: Screenshot of the Dakota study definition wizard

2.2 Input File Editor

The latest version of SAW includes custom editor support for many sections of Dakota input files. Previously, analysts who edited Dakota files were required to keep track of long, free-form lists of values. Depending on how the original author formatted their Dakota input file, these values could be extremely hard to read and understand. The custom Dakota editors in SAW eliminate this difficulty by providing labeled, easy-to-understand editors for number data in Dakota files.

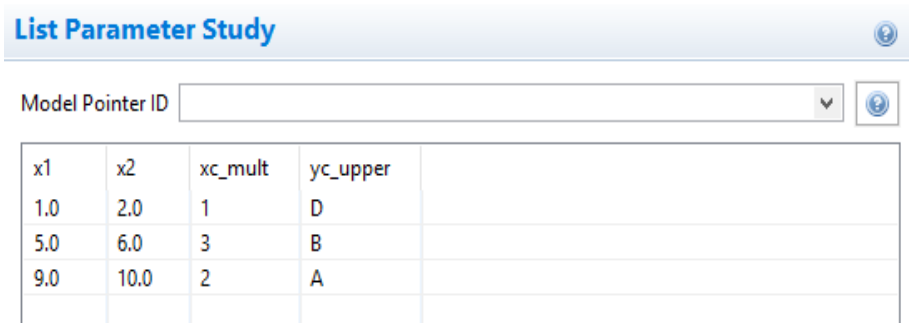


Figure 2-2: Human readable tabular input

2.3 Hierarchical Results Database Structure in HDF5

As of Dakota 6.9, the Dakota team added HDF5 as a supported output format. Prior to 6.9, Dakota's output data was available either through its console output stream or through tabular data files. Neither of these formats provided strong organization of Dakota's output result data, and analysts were always required to write their own custom text parsers. Using HDF5, analysts have access to richer, more organized Dakota output results. In addition, *in situ* HDF5 data browsing is supported in SAW using a modified version of the HDF team's HDFView tool.

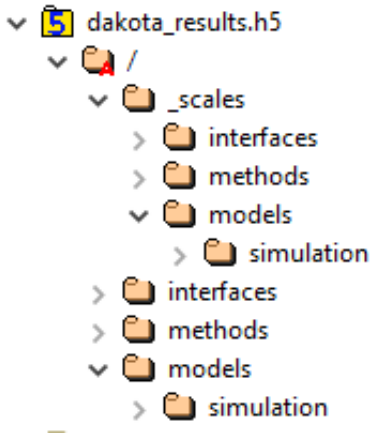


Figure 2-3: Hierarchical Dakota input and output data structure

2.4 Results Visualization

A variety of graphical plots are now supported for Dakota data visualization by using Chartreuse. Chartreuse is a middleware tool that allows Dakota and HDF5 to interface with plotly.js, a powerful JavaScript plotting library. Simple plots – such as scatter plots, bar charts, histograms – are

supported. In addition, specialized plots for Dakota-generated datasets – such as incremental LHS, optimization, and centered parameter studies – are also supported and trivial for analysts to create.

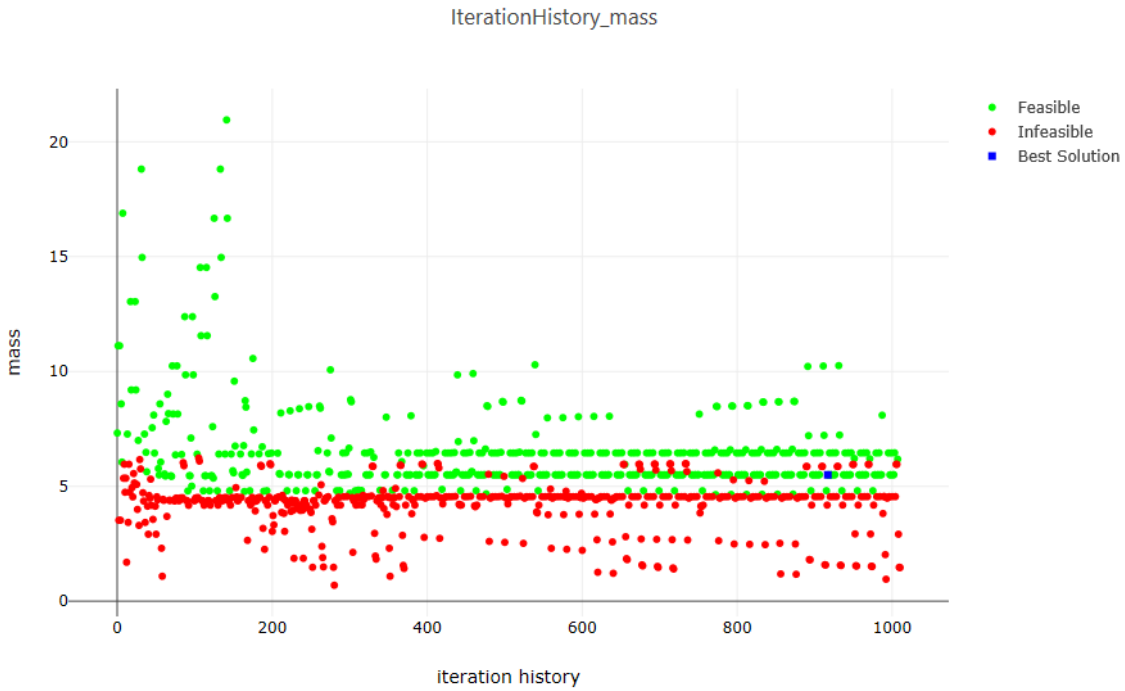


Figure 2-4: Visualization of the optimization method

2.5 Integration of Dakota User Interface Components with NGW

For this milestone, we placed a strong emphasis on seamless integration between Dakota and NGW. A specific example of this integration is demonstrated in the new Dakota Study Wizard, which has the ability to automatically create a Dakota interface block that uses the Next-Gen Workflow engine as its analysis driver. The analyst only needs to select an NGW file for their Dakota study, and all subsequent configuration details are handled automatically.

2.6 Plans for Future Work

To further strengthen Dakota support in SAW and NGW, we plan to continue creating custom editors for the remaining Dakota methods. In addition, there is currently strong interest in bolstering Chartreuse with more plot types and configuration options.

3 AUTOMATIC REPORT GENERATOR

The Sandia Automatic Report Generator (ARG) takes as inputs user configuration files, written content, and data files (with primary emphasis on formats used by Sandia analysts), and using an intermediate abstract representation generates reports in MS-Word or LaTeX, and from the latter PDF through a standard TeX extension. that can be used for report customization by the user (15). For instance, one functional component (the Assembler) of the system extracts Sierra results data, using the Python bindings to the Exodus libraries, and generates tables, plots and visualizations as requested either by the user or by an automatic content analyzer (the Explorator), which forms another functional component. The ARG is written in Python, leverages on Sandia-developed tools as well as open source software libraries, and is deployable on Linux and macOS (16) as well as Windows. Furthermore, by design it is possible to execute the ARG as part of a workflow.

3.1 Multi-Stage Document Construction

With a case integrated in the ARG's text harness ("Thick Pipe"), we have demonstrated this year the ability of ARG to discover, generate, and assemble content coming from multiple sources:

- raw simulation data automatically summarized by ARG's Explorator component;
- human-authored content in the form of text templates;
- simulation input values stored in simulation pre-processor files;
- other artifacts (e.g., images) generated by third-party libraries or applications.

This new capability allows for the smooth integration of ARG into a workflow and in particular to automatically document ensembles of parametric runs, as illustrated in Figure 3-1 by the output of

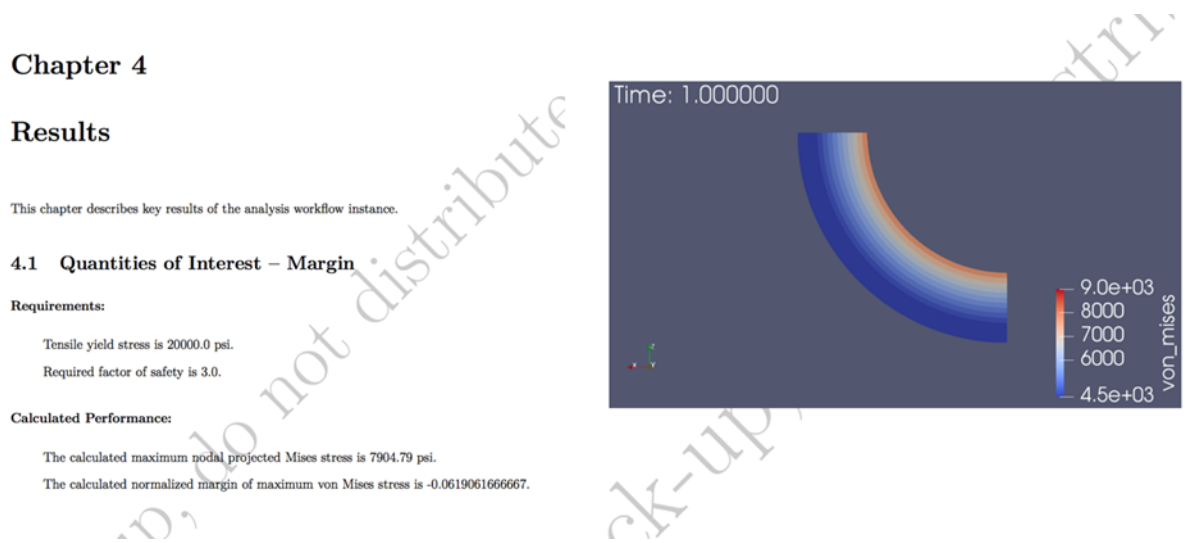


Figure 3-1: Templatized analyst authored results

this multi-stage exemplar.

3.2 Architecture Overview

The Automatic Report Generator comprises the following three functional components:

- **Explorator**: an application to automatically discover content from raw simulation data and create an abstract report structure from it.
- **Generator:** an application to create "artifacts" such as charts, 3D visualizations, histograms, etc. as required by abstract report structure.
- **Assembler:** an application to create a concrete report from an abstract description thereof and containing the necessary artifacts.

These components may be all used in the a single ARG process but need not be: these are written as independent applications and can be used as such. Furthermore, each of these components can take into account a number of user-specified variables ("constants") driving their behavior.

ARG currently supports a variety of "data interfaces", allowing for the ingestion of various types of datasets that may exist in Sandia simulation, as illustrated below, with the original CAD/geometry description of mechanical part (left) and its corresponding mesh or part thereof mesh (right) of Figure 3-2.

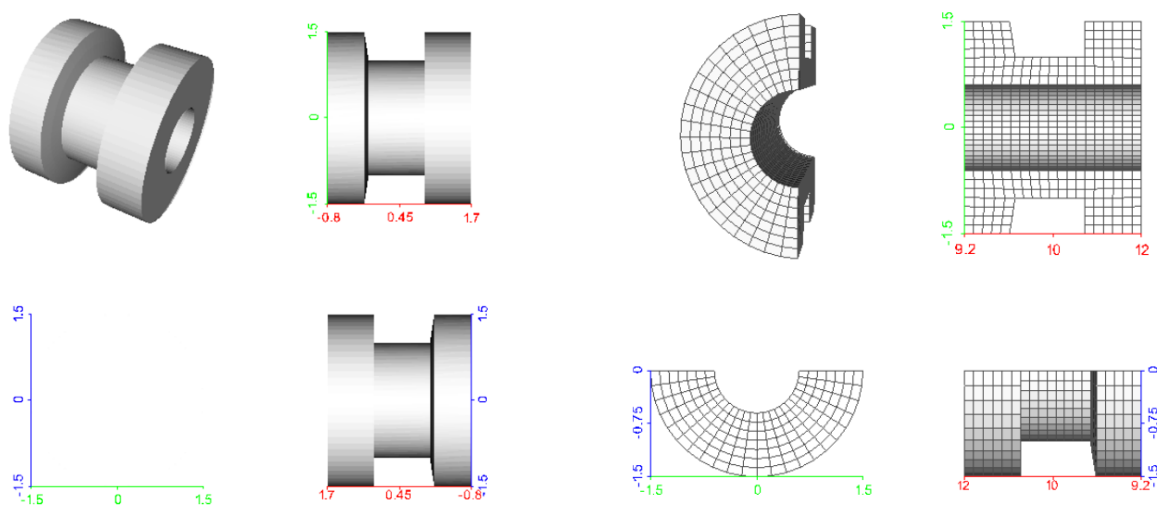


Figure 3-2: Mesh visualization

We have implemented the ability for the Assembler to indifferently use a LaTeX/PDF or a MS-Word "backend", depending on the user's choice, combined with three possible types of outputs: a) SAND report format with appropriate markings for OUO and classified reports and b) SAND TK (Test Plan) formats, and c) generic, user-defined reports.

3.3 Solid Mechanics Capabilities

During FY19, we have collected feedback from SNL production model users related to Solid Mechanics (SM) simulations and, as a result, added multiple new features, such as full plotting of all SM material plotting capabilities as illustrated in Figure 3-3 for the case of a Stress/Strain response function.

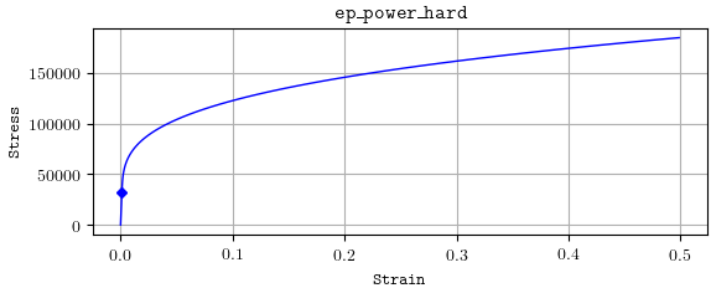


Figure 3-3: Constitutive model curve plot

Also, based on user feedback, we reorganized mesh per-block pages functionality, separating out primary and derived data in order to make things both clear and epistemologically correct for efficiency and extensibility, in particular for SM simulations for which we know extensively mine the log data (e.g., Mass and Moments stored there). In particular, we developed a SM Log reader module, whose capabilities were driven by user feedback. An example of how log data is fused with data from other inputs is shown in the automatically-generated table shown in Figure 3-4.

property	value
number of nodes	52563
number of elements	43986
prescribed material name	3041
prescribed material model	ml_ep_fail
type of first element in block	HEX8
original mass	0.00105082
relative mass change due to mass scaling	0.00000e+00 %
center of gravity	$\begin{bmatrix} -0.0841 & 0.0983 & -1.9 \end{bmatrix}$
moments of inertia	$\begin{bmatrix} 0.00126 & -0.000174 & 1.32e-13 \\ & 0.000348 & -1.21e-13 \\ & & 0.0016 \end{bmatrix}$

Table 3.10: Properties of block lid.

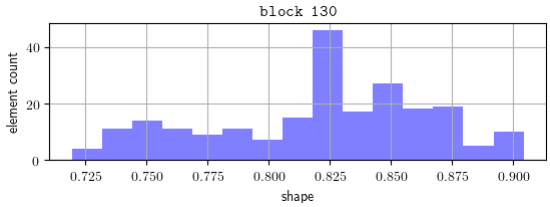
Figure 3-4: FEM mass properties reporting

Several Structural Mechanics exemplars and test cases were added to the test harness, also exercising the two available output formats: LaTeX/PDF and Word, also exercising the three different types of document types that the ARG's Assembler component currently supports.

3.4 Structural Dynamics Capabilities

All new features previously mentioned for SM capabilities, insofar as they are relevant to Structural Dynamics (SD) cases, were also implemented for such simulations, and/or specialized as required. For example, *eigenvalue* solutions are only relevant to SD simulations, and the Explorator component of ARG is now able to look for the appropriate types of solutions, depending on the discipline of interest (currently available: SM, SD, and EMPIRE EM/ES).

Another important capability that was added this year, also based on user feedback, was the computation of mesh quality statistics and histograms. This capability is available for all currently supported disciplines, and leverages Sandia's VERDICT library, as illustrated in Figure 3-5 for mesh block statistics (left) and histogram (right) of SM and SD cases, respectively.



Block 4 (box_shell) element quality

$\mathcal{Q}$	$\min(\mathcal{Q})$	$\mu(\mathcal{Q})$	$\max(\mathcal{Q})$	$\sigma(\mathcal{Q})$	$\sigma/\mu(\mathcal{Q})$
scaled Jacobian	0.6088	0.9951	1	0.01345	0.01352
shape	0.6524	0.9931	0.9998	0.0164	0.01652

Table 3.9: Element quality statistics of block box_shell.

Figure 3-5: Mesh quality reporting

As was done for Structural Mechanics, several Structural Dynamics exemplars and test cases were added to the test harness, also exercising the two available output formats: LaTeX/PDF and Word.

3.5 Plans for Future Work

In the future, we plan in particular to:

- enhance the automatic validation and code coverage of ARG;
- complete transition of ARG to Python-3 and remove all remaining Python-2 dependencies;
- develop and integrate analysis tools for CAD to mesh mapping verification;
- integrate with ongoing Credibility Framework development;
- continue to expand to new disciplines (e.g., thermal, MHD, electrical device models, etc.).

4 EXEMPLARS OF END-TO-END SENSITIVITY, OPTIMIZATION AND UNCERTAINTY QUANTIFICATION WORKFLOWS

A set of exemplars implementing the concepts outlined in (17) has been assembled to support requirements definition, stress testing and training for the IWF components developed for this milestone. The UUR exemplars are published in the SAW project `IWF_L2_Milestone_FY2019-UUR` in the folder `1-Exemplars`. A listing of the exemplars is given in Figure 4-1.

	Analysis Workflow					Ensemble Workflow						Visualization	
	Geometry Definition	Meshing	Solution	Reporting	Computationally Demanding, Needs Resiliency	Design Optimization	Scalability, Parallel Consistency	Solution Verification	Uncertainty Informed Sensitivity Analysis	Uncertainty Analysis	QMU	Slycat	Dakota Composite Plots
Exemplar 1 -Simple tank partially filled with liquid	N/A	N/A	Analytic	ARG Generic	no	✓	N/A	✓	✓	✓	✓	✓	✓
Exemplar 2 - Tank assembly partially filled with liquid	Creo	Cubit	Sierra imlpicit	ARG SAND	no	✓	✓	✓	✓	✓	✓	✓	✓
Exemplar 3 - Explosion beneath lattice structure	CTH diatoms	CTH grid	CTH	None	yes	✗	✓	✓	✓	✓	✗	✓	✓
Exemplar 4 - Abnormal mechanical crush	Creo	Cubit	Sierra explicit	ARG SAND	yes	✗	✓	✓	✓	✓	✗	✓	✓

Figure 4-1: End-to-end exemplars to support requirement definition, stress testing and training

Each workflow exemplar is designed to represent a different use scenario to exercise a distinct set of capabilities. A brief summary of each exemplar is as follows:

- Simple tank partially filled with liquid: non-trivial closed form solution of a structural engineering problem with simplified boundary conditions. Executes quickly, requires no FEM, supports training with a realistically complex set of workflows. ARG report communicates stress field and structural performance. Solution verification, optimization, sensitivity analysis and UQ/QMU studies.
- Tank assembly partially filled with liquid: Parametric Creo (CAD) model connected to parametric conformal hex meshing with Cubit and Sierra implicit solution followed by ARG report. Executes quickly. Solution verification, optimization, sensitivity analysis and UQ studies.
- Explosion beneath a lattice structure: CTH model illustrating generality of the framework. Computationally intensive. Solution verification, parallel consistency and scalability studies, sensitivity analysis and UQ.
- A realistic small assembly (10 parts) subjected to crush representative of scenarios derived from abnormal mechanical system requirements. Parametric Cubit meshing, Sierra explicit solution followed by quantitative Ensight and Python post processing summarized in an ARG report. Computationally intensive. Parallel consistency, scalability and mesh resolution studies.

For each exemplar a folder structure suitable to support credibility evidence compilation was used. The reusable model files including analysis workflow components are kept under `1-Analysis_Model`, and the Dakota studies are in either the `3-Studies` folder or subfolders associated with the appropriate credibility sub-elements in folder `4-Credibility/1-PCMM`.

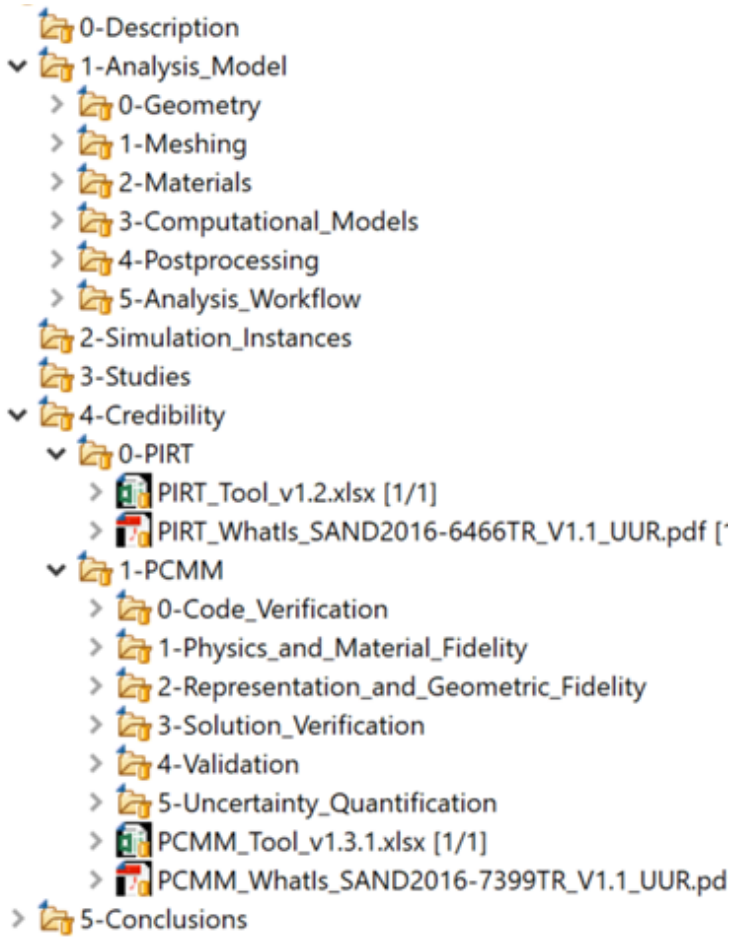


Figure 4-2: CompSim credibility inspired folder structure

In the following sections detailed descriptions of two exemplars are given starting from problem definition through the construction of multi-tiered distributed analysis workflow mapping model parameters to responses and various Dakota studies generating visual presentations of the ensemble results. One of the key goals of the milestone was to support communication of engineering information in a manner accessible to all stakeholders, and such visualization directly supports this goal.

4.1 Exemplar 2 - Storage Tank

This model is an effective demonstration problem for end-to-end credibility-focused analysis, optimization, sensitivity and UQ. Many typical CompSim tools such as Creo, Cubit, Sierra, ParaView, and ARG are involved in the analysis workflow. Each model instance executes relatively quickly making this exemplar suitable for training. Representative workflows are highlighted in this report, and the complete working set is in the SAW project `IWF_L2_Milestone_FY2019-UUR` in the folder `1-Exemplars/2-Storage_Tank`.

4.1.1 Problem Definition

The minimum weight storage tank partially filled with liquid needs to be designed subject to performance constraints expressed as a maximum stress and a maximum deflection constraints. Subsequently, sensitivity analysis is needed to determine the most sensitive and the most important model parameters, which are not necessarily the same as discussed later. Finally, an initial uncertainty analysis is performed to indicate whether the optimized design is vulnerable to known and characterized input uncertainties.

The geometry of the storage tank and design parameters are shown on Figure 4-3. A set of parametric Creo parts representing the storage tank has been build and assembled in a Creo assembly.

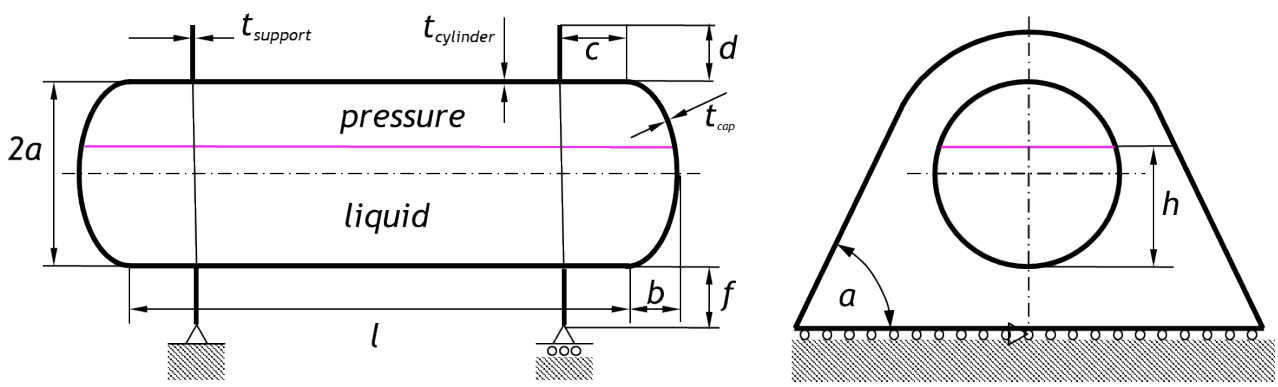


Figure 4-3: Storage Tank geometry and mechanical boundary condition definition

A parametric analysis workflow is needed to map the model parameters to responses also called QoIs (Quantities of Interest). A summary of the model parameters and the responses is given in Figure 4-4. The `g_displ` and `g_yield` model responses are normalized constraints formulated such that a positive value means that the constraint is met, and a negative value means that the constraint is violated. Additionally, normalization allows comparison of model responses with significantly different magnitudes (stress, displacement).

Input	
FSY	Factor of safety on yield
max_displ	Maximum allowed sagging
t_cylinder	Vessel cylinder thickness
t_cap	Vessel cap thickness
a	Vessel radius
l	Vessel length
b	Vessel elliptical cap
c	Support offset
d	Support top web
f	Support bottom web
t_support	Support thickness
p	Pressure
gamma	Liquid specific weight
h	Liquid height
E	Elastic modulus
nu	Poisson's ratio
FTY	Tensile yield stress
accuracy	Model accuracy

Output	
g_yield	Yield constraint
g_displ	Displacement constraint
struct_vol	Structural volume

$$g_displ = \frac{\max_displ - displ}{\max_displ}$$

$$g_yield = \frac{\frac{FTY}{FSY} - \sigma_{eff}}{\frac{FTY}{FSY}}$$

Figure 4-4: Model parameters and responses

4.1.2 Analysis and Documentation Workflow

The workflow encapsulating analysis and documentation is executed on two platforms Windows and Linux. CAD packages such as Creo and SolidWorks are only available on Windows whereas FEM solvers need to run on HPCs for realistic problems.

The top-level analysis workflow on Windows shown on Figure 4-5 has the following visual organization: A table of global configuration parameters allowing easy handoff of a workflow between different analysts is visible on the top. The contents of these global parameter tables are stored in external configuration files ensuring that the workflow itself can be made read-only for distribution to partners that have permission to use but not to edit the workflow. Throughout the exemplars a convention was used that when a workflow has global properties in a tabular manner it is meant to run in studies. Workflows that have individual global parameters may be executed to unit test, but they are meant to be executed as an embedded sub-workflow. Model parameters are listed on the left side. Static input files that don't change with analysis instances are in the lower left corner. Workflow responses are on the right side and workflow nodes performing tasks are in the middle. It his case the important workflow nodes are `1.1-Windows-Creo.iwf` performing Creo model generation for the current set of geometric input parameters on Windows and `2.0-HPC-Analyze.iwf` performing Cubit meshing, Sierra solution, ParaView visualization, QoI

evaluation and ARG reporting of the analysis instance, all executed on HPC in a remote nested workflow. Variables are defined as nominal values and multipliers to support both optimization and UQ. During the optimization process the multipliers are set to unity, and the nominal values are changed. During UQ the nominal values are set to a final optimized design and the multipliers are changed to amounts required by the particular UQ process. Since in this case some of the geometric variables are required by the meshing process visual clarity of the workflow representation is somewhat compromised.

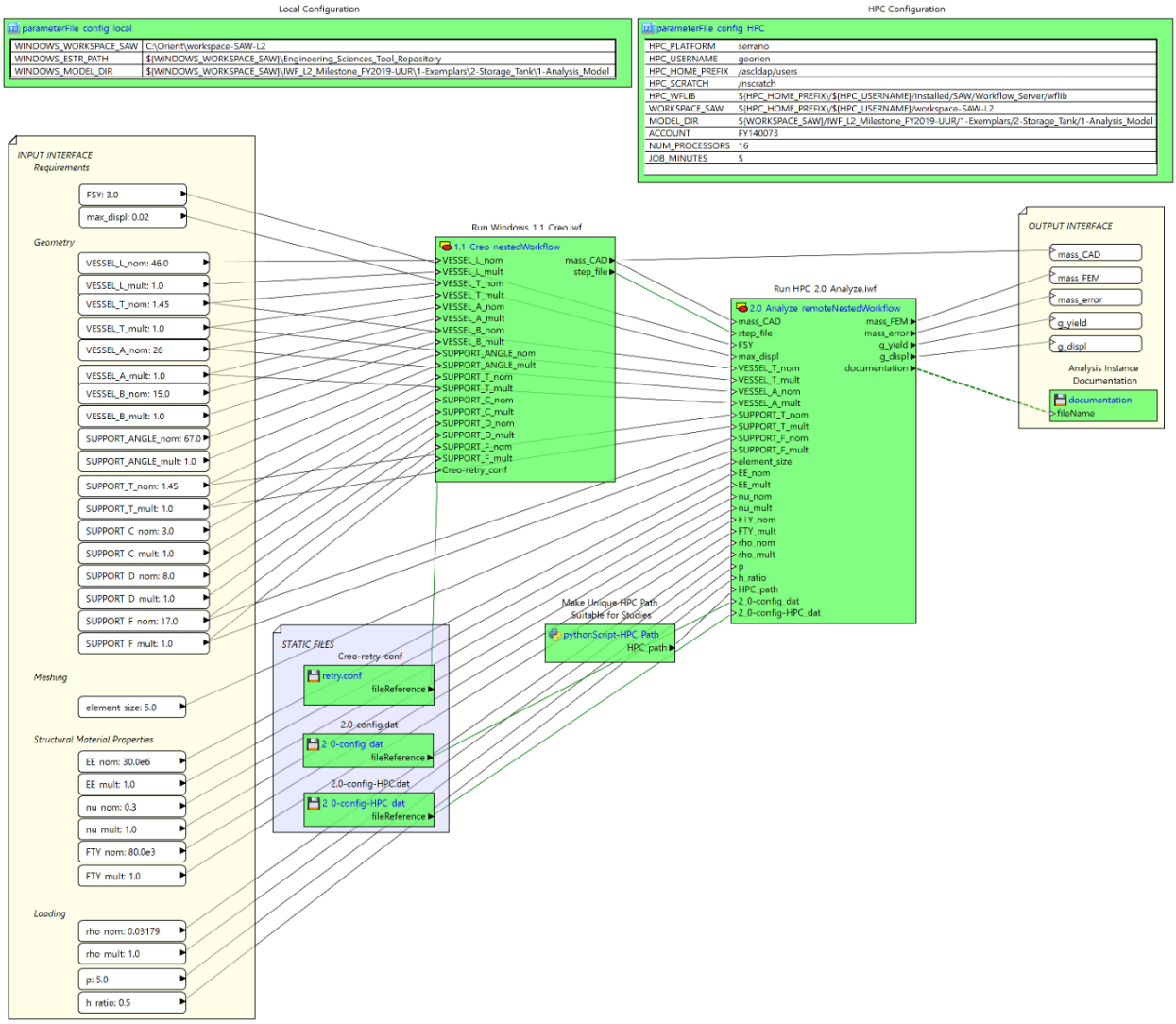


Figure 4-5: Top level analysis workflow

.

<u>Creo execution on Windows</u> – This sub-workflow is shown in Figure 4-6. Notice that unlike the top-level analysis workflow this workflow is not meant to be executed stand-alone, and its global parameters are on the canvas individually, and not in a table associated with a configuration file. The inputs for this subflow are geometric parameters (nominal and multiplier values), and the

responses are assembly volume and mass for FEM comparison downstream and a STEP file that will be imported by Cubit. The static Creo assembly and part files are passed to the Creo execution.

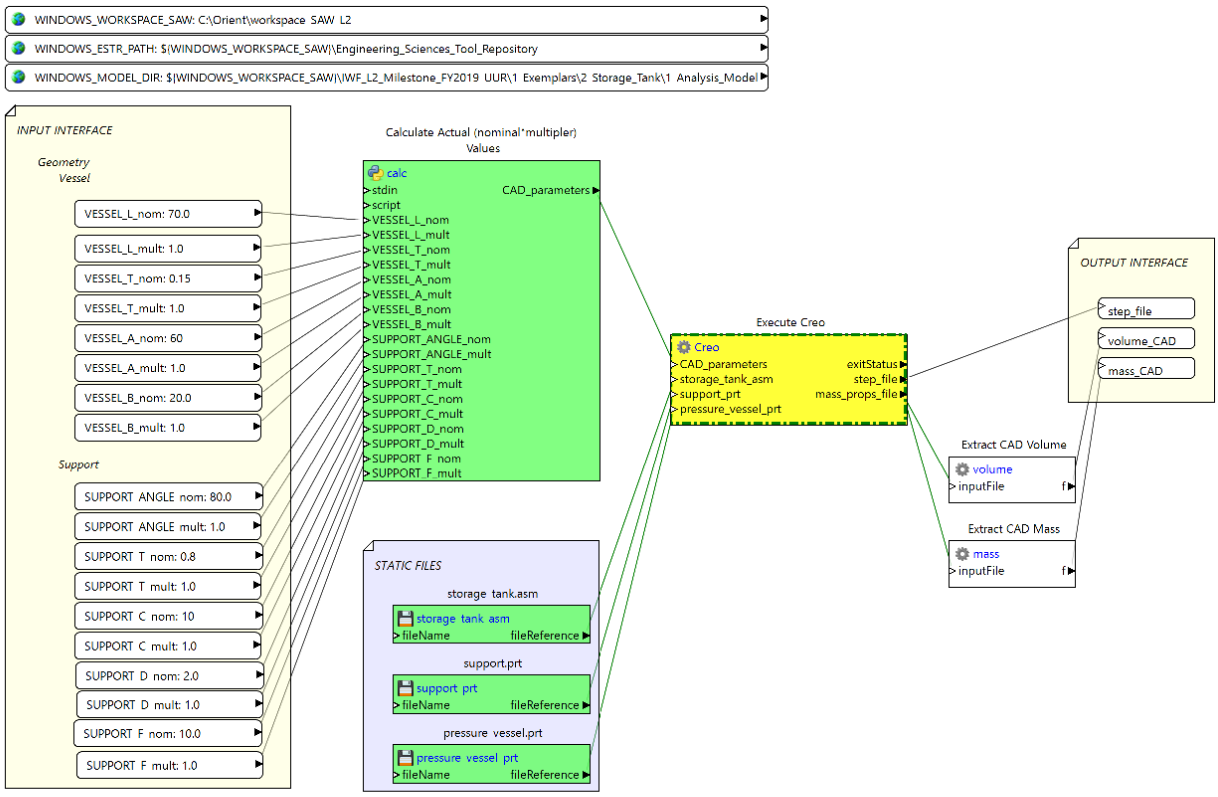


Figure 4-6: Execute Creo

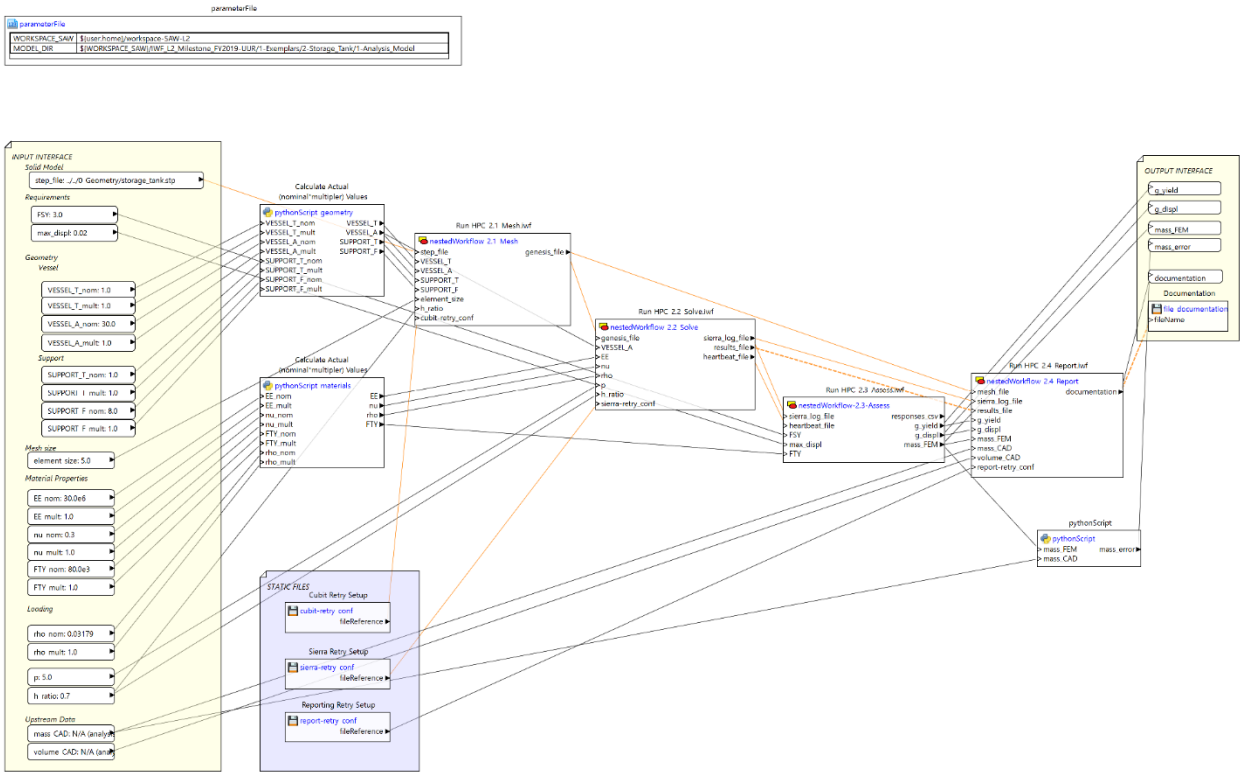


Figure 4-7: Cubit-Sierra-ParaView-ARG workflow

Analysis workflow on HPC– This is a somewhat complex remote nested workflow that implements

the sequence of Cubit, Sierra, ParaView and ARG. It is difficult to represent it clearly in this report, but an image of this workflow is shown in Figure 4-7. Input parameters to the workflow are the STEP file and mass properties generated by the upstream Creo workflow, geometric, meshing, material and structural performance requirement parameters. The workflow returns mass calculated by CAD, the error between FEM and CAD mass, normalized constraint violations (`g_yield`, `g_displ`) and the ARG generated report. Figure 4-8 illustrates the analysis instance report generated by this workflow including human authored templatized results reporting.

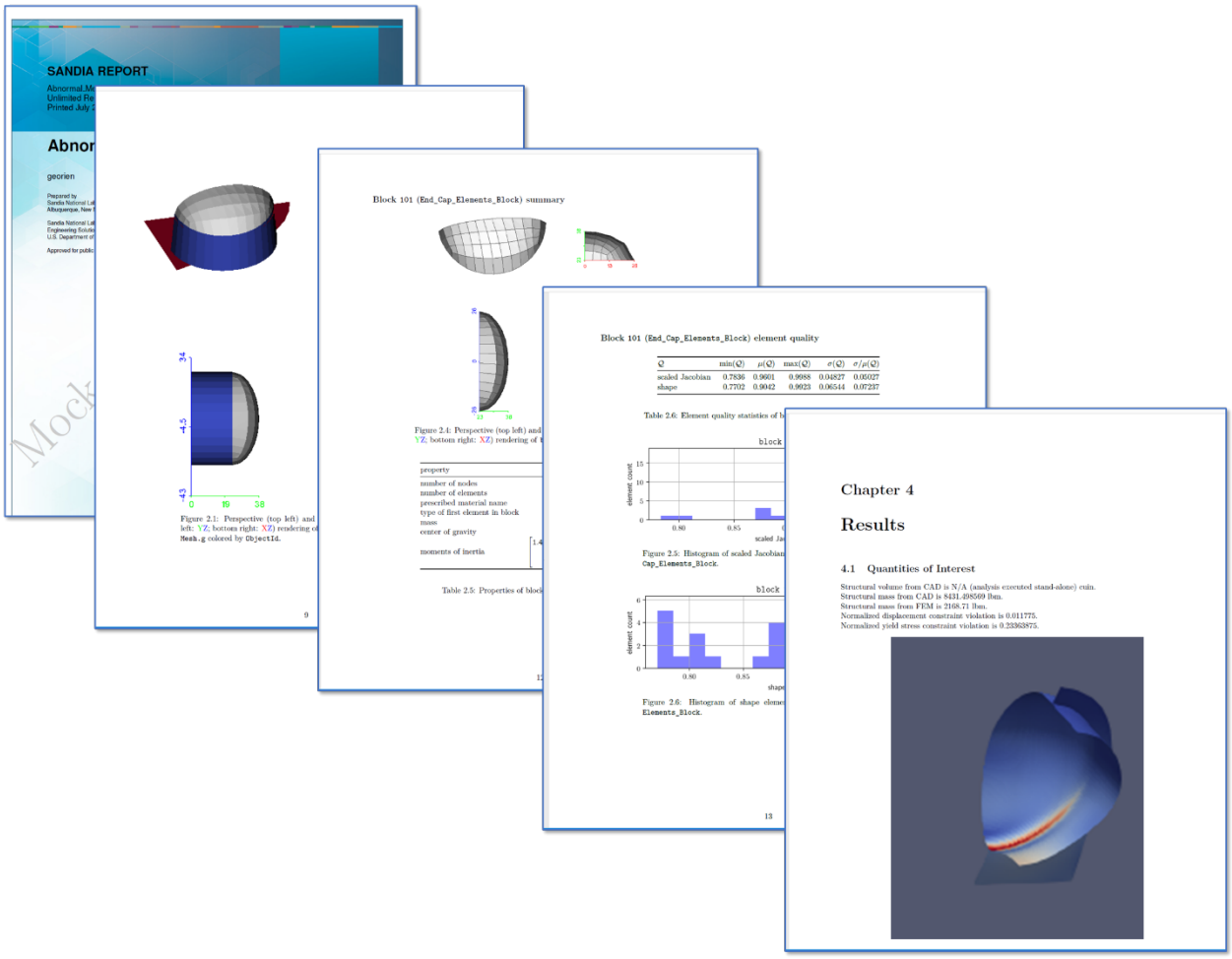


Figure 4-8: ARG report including analysis results

The analysis workflows in this section are in the SAW project `IWF_L2_Milestone_FY2019-UUR` in subfolders of `1-Exemplars\2-Storage_Tank\1-Analysis_Model\5-Analysis_Workflow`.

4.1.3 Solution Verification

Once the end-to-end analysis workflow is implemented it opens the possibility to examine the model, interrogate and adjust it and ultimately derive information decision makers can use. The first action for any discretized model is solution verification to determine the optimal resolution for different intended uses. For example, in an optimization scenario a model that is *comparative* correctly ranking designs is sufficient while for qualification the model must be *predictive*, meaning

(among other attributes) that either it is mesh converged, or the numerical error associated with the discretization used in the evidence package is quantified.

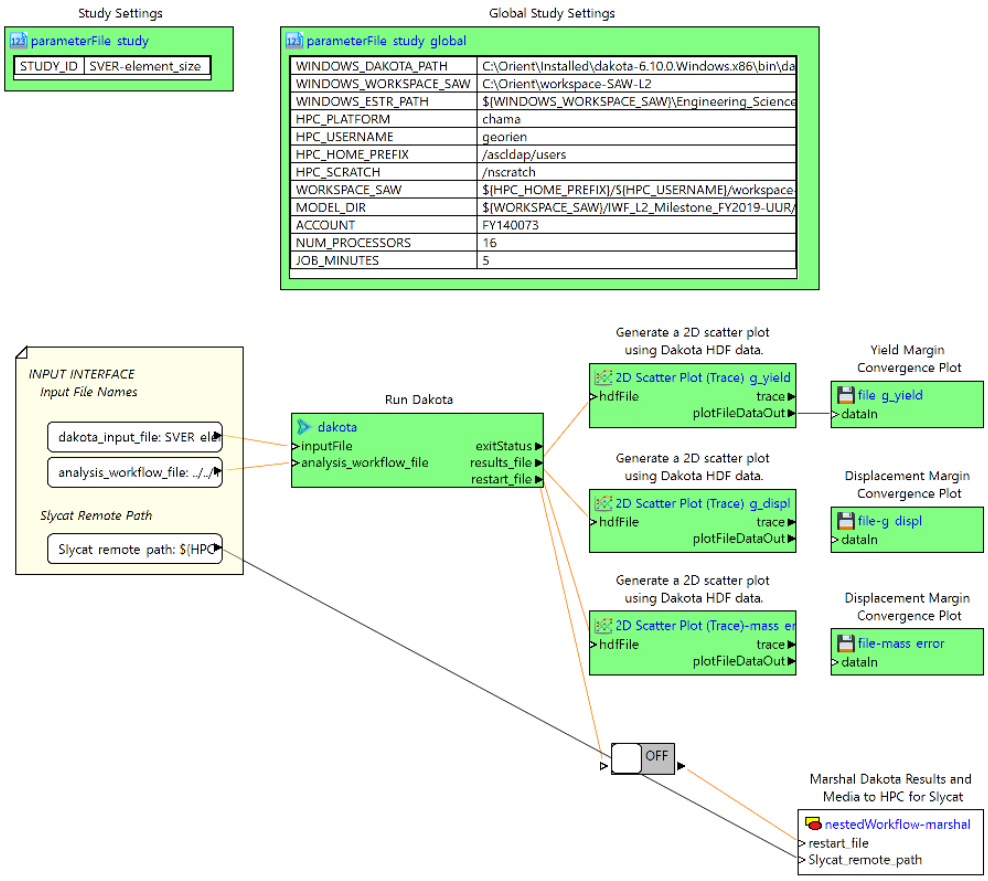


Figure 4-9: Dakota solution verification study workflow

The Dakota workflow for solution verification is illustrated in Figure 4-9. The tabular global parameters on the top indicate that this workflow is intended to run stand-alone. The Dakota node has two input parameters: a) the Dakota input file usually created by the wizard and edited using the syntax-aware editor, and b) workflow that represents the mapping between variables and responses named analysis driver. During the ensemble sample execution other artifacts such as images, videos and .PDF reports are generated, and they are best viewed in Slycat. A working prototype of the Slycat interface was created for this milestone, and it is distributed as a workflow node from the Engineering Sciences Tool Repository.

A user operable on/off switch set not to activate this workflow at this time. While studies that produce large variations in the design such as optimization benefit from having an ARG report available for each sample, UQ studies where the difference between samples may be imperceivable, it is not necessary to generate a report for each sample. Dakota plotting nodes (Chartreuse) create convergence plots such as the one shown in Figure 4-10. The mesh convergence plot displays an important finding: there is a "false convergence" region when `element_size` is in the interval of (1.6,3.0). From this plot it was decided that `element_size=1.0` will be used for

optimization studies, and `element_size=0.5` is appropriate for UQ. For CompSim models used for purposes other than demonstration a formal error estimation would follow.

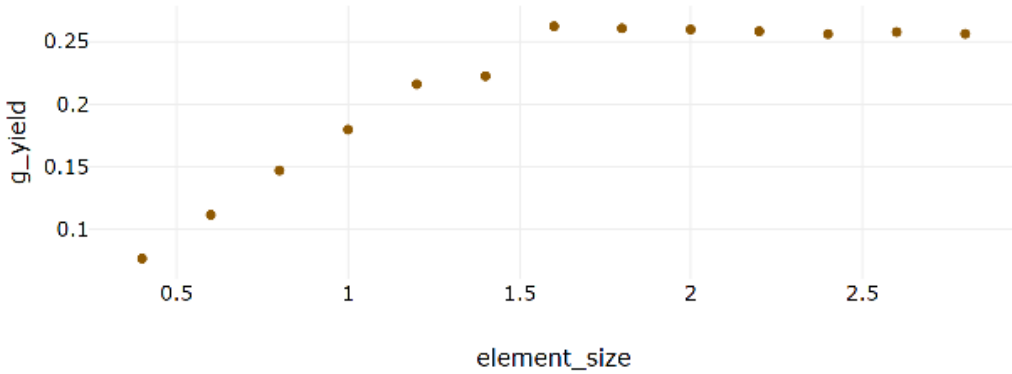


Figure 4-10: Mesh convergence study for g_yield

The workflow in this section is in the SAW project `IWF_L2_Milestone_FY2019-UUR` in folder of `1-Exemplars\2-Storage_Tank\3-Studies\0-Solution_Verification`.

4.1.4 Optimization

Genetic optimization study type was used to define the design from realistic bounds of the geometric parameters using the of CAD mass and the normalized constraint violation workflow responses.

The Dakota workflow is similar to that shown Figure 4-9. With the exception that a different Dakota input file representing an optimization process is attached to it. For this and all subsequent studies the workflow file is identical underlying the return on investment on architecting and implementing the analysis workflow. A single objective genetic algorithm executed 800 samples of

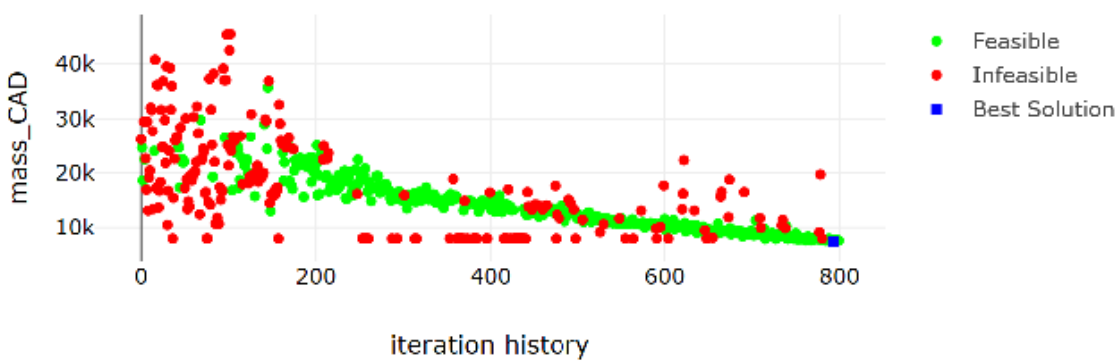


Figure 4-11: Evolution of structural mass during the optimization process

with a concurrency of 4 meaning four evaluations were executed simultaneously.

The history of the CAD evaluated mass, which is the objective in the optimization process is shown on Figure 4-11. Green symbols indicate feasible designs (all the constraints are satisfied), red symbols mean infeasible designs (at least one constraint is violated) and the blue symbol indicates the best design. It can be observed that there are many infeasible designs during the initial stage of the optimization while the design space is being fully explored, and there is a converging trend later. The CAD-Cubit interface failed for about 15% of the samples during the optimization process. For these samples a fictitious weight of 8000 pounds was returned with `g_yield=g_displ=-1.0` indicating infeasible design to steer the optimizer away. The assumption was that the Creo-Cubit failure occurs for unreasonably thin designs, but it was found that wasn't necessarily so. Nevertheless, the Dakota genetic algorithm managed to improve the design despite these random failures. The samples that produces invalid CAD export were collected and sent to PTC as evidence underlying the fact that workflow based CompSim thoroughly tests all elements of the CompSim software stack producing artifacts to support deep conversations regarding reliability of the tools.

It was found that the displacement constraint was the most active in this design space. It can be

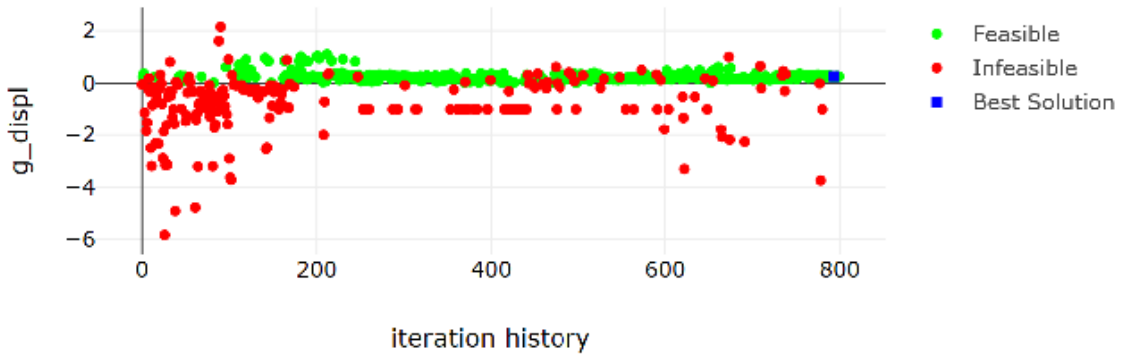


Figure 4-12: Evolution of the displacement constraint during the optimization process

observed in Figure 4-12 that the constraint was driven down to nearly zero which is the boundary

between feasibility and infeasibility. This may have negative implications on the robustness of the

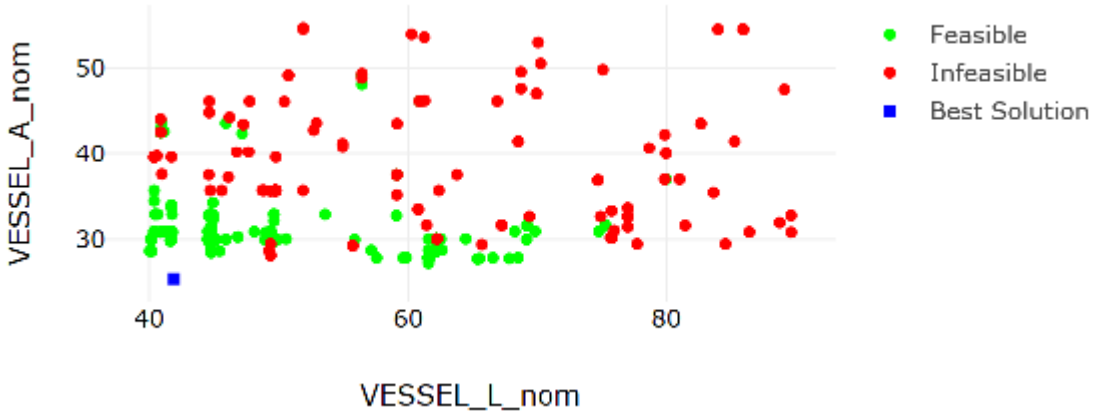


Figure 4-13: Design space coverage

deign against uncertainties. Additionally, the Dakota visualization package for optimization provides evidence whether the optimization process covered the entire design space, and whether the

optimum is in one of the corners of the design space. For example, Figure 4-13 plotting vessel radius against vessel length indicates that the optimum design is at the lower bounds of the vessel radius and length. It can be observed that the design space was covered reasonably well, and shorter smaller vessels tend to be feasible while the long storage tanks with large radii are infeasible. In a preliminary design scenario these findings precipitate discussions between the system engineering and the design team to define the appropriate design window for the product.

The workflow in this section is in the SAW project `IWF_L2_Milestone_FY2019-UUR` in folder of `1-Exemplars\2-Storage_Tank\3-Studies\ 2-Deterministic_Design_Optimization`.

4.1.5 Deterministic Sensitivity

Once the design is obtained through either formal optimization - as it is the case in this exemplar - or by other means, designers want to know which variables have the largest effect on system performance. To this effect each input variable was perturbed by 5% and the Dakota parametric sweep study was executed. Dakota provides a Chartreuse composite plotting tool to quickly visualize main effects from such a study. Figure 4-14 shows that normalized yield stress criterion violation is sensitive to vessel geometric variables, liquid density and height as well as material yield strength. Note that this only represents information of the "slope" of the response with respect to the input variables, and not necessarily ranking of which variable is most important. That statement needs to be made by a probabilistic sensitivity analysis considering the characterized physical bounds of the input variables.

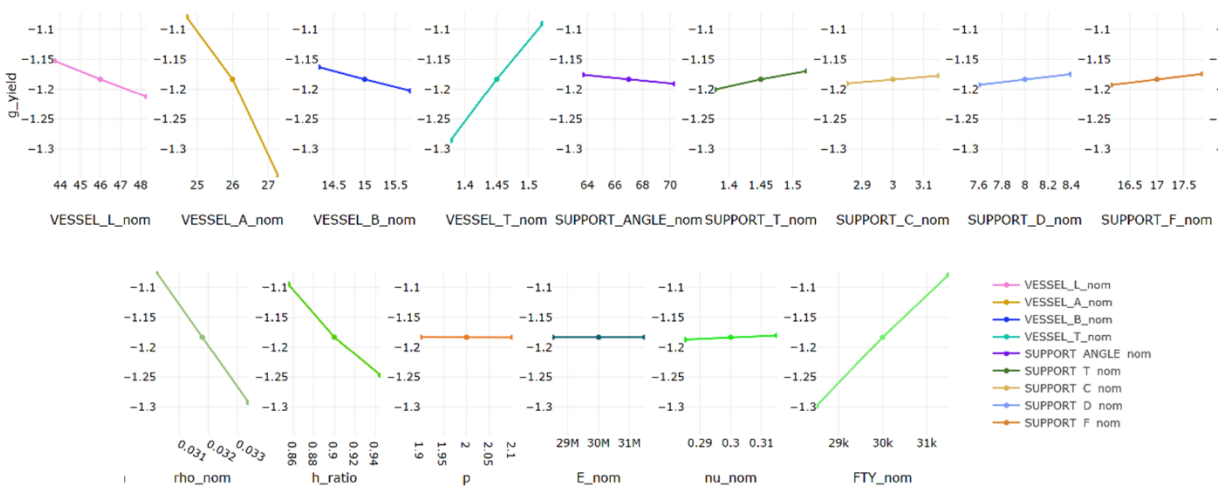


Figure 4-14: Deterministic sensitivity, main effects

The workflow in this section is in the SAW project `IWF_L2_Milestone_FY2019-UUR` in folder of `1-Exemplars\2-Storage_Tank\3-Studies\ 1-Deterministic_Sensitivity`.

4.1.6 Probabilistic Sensitivity

The next step in the credibility assessment is to characterize the input uncertainties irrespective of whether they are model features or data to characterize then is available. The resulting uncertainty

Variable Name	Description	Type	Characterization	Parameters	Model Feature?	Feasible Physical Range	References
				Geometric Parameters			
l	Length of the vessel	aleatory	distributional	normal, C.o.V.=0.0004			http://www.assakkaf.com/Papers/Journals/Unc
a	Inner radius of the vessel	aleatory	distributional	normal, C.o.V.=0.0004			
t	Wall thickness of the vessel	aleatory	distributional	normal, C.o.V.=0.0004			
				Loading			
gamma_unc	Uncertainty multiplier on fluid specific v	epistemic	distributional	uniform, [0.097,1.003]	yes	>0.0	http://www.genesisny.net/Commodity/Oil/ODe
h_ratio	Height of the fluid/inner diameter	epistemic	interval	uniform, [0.5, 0.9]	yes	[0.0, 1.0]	Operating requrements
p	Pressurization	epistemic	interval	uniform, [1.5, 2.5]	yes	>0.0	Operating requirements
				Material Model Parameters			
E	Elastic modulus	aleatory	distributional	normal, C.o.V.=0.076	yes	>0.0	http://www.assakkaf.com/Papers/Journals/Unc
FTY	Tensile yield strength	aleatory	distributional	lognormal, C.o.V=0.124	yes	>0.0	

Figure 4-15: Uncertainty inventory

inventory is a key input to further investigating credibility of the model.

Next, a sensitivity analysis where the amount of perturbation is tied to the quantified uncertainties is performed. The results now identify the most important variables. In this case, since geometric tolerances as well as natural variation in fluid density are relatively small sensitivity of stress with respect to these variables is small. The only variables that are important are fluid level (`h_ratio`) and material yield strength. The slopes of the sensitivities are intuitive; increasing the amount of fluid with more gravity load should reduce margin on yield and increasing the material yield should increase margin on yield. While this analysis is very useful it doesn't account for interaction between variables.

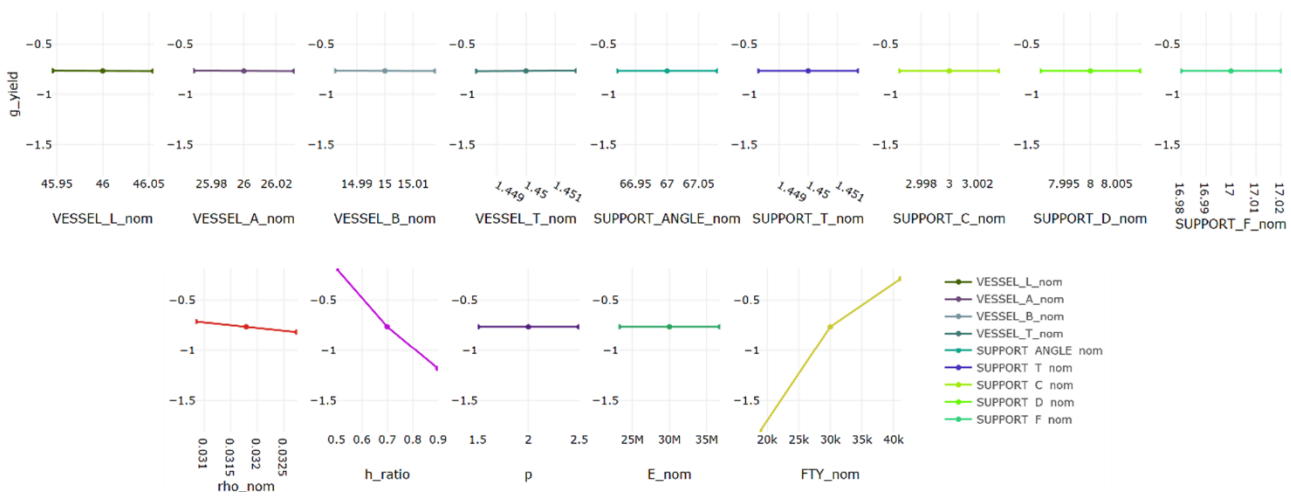


Figure 4-16: Uncertainty informed sensitivity

The workflow in this section is in the SAW project `IWF_L2_Milestone_FY2019-UUR` in folder of `1-Exemplars\2-Storage_Tank\4-Credibility\1-PCMM\5-Uncertainty_Quantification\1-UQ\0-SA-Uncertainty_Informed`.

4.1.7 Uncertainty Quantification

The uncertainty inventory (Figure 4-15) is used in conjunction with the Dakota wizard to define an uncertainty quantification study. Depending on the goal of the UQ study different methods may be

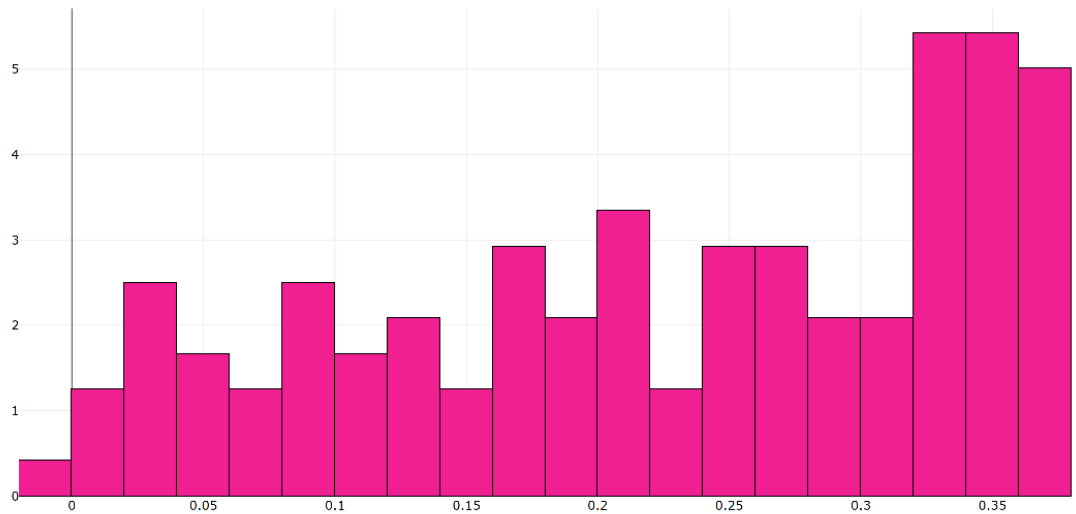


Figure 4-17: Histogram of normalized displacement constraint violation

appropriate, bit it is a good practice to perform an initial sampling study, typically an incremental LHS (Latin Hypercube Sampling). This study generates information regarding the adequacy of the sample size through examination of the first four statistical moments. Figure 4-18 indicates that while the mean and standard deviation are adequately converged with 120 samples the higher statistical moments are not yet converged. From QMU point of view this means that no statement

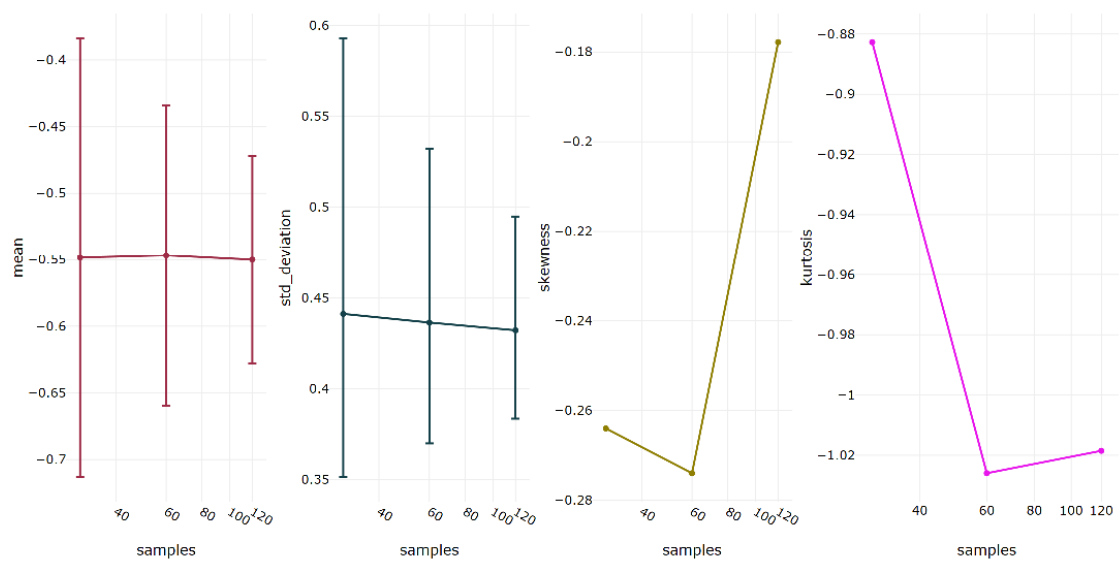


Figure 4-18: Incremental LHS convergence

about low probability events should be made from this study, since the shape of the response distribution is not yet converged.

The histogram of the model response `g_displ` is show on Figure 4-17. There is an indication of non-zero probability of negative values at the left end of the histogram. Since it was recognized that the shape of the response distribution has not converged with respect to sample size no statement may be made about the probability of violating margins at this point.

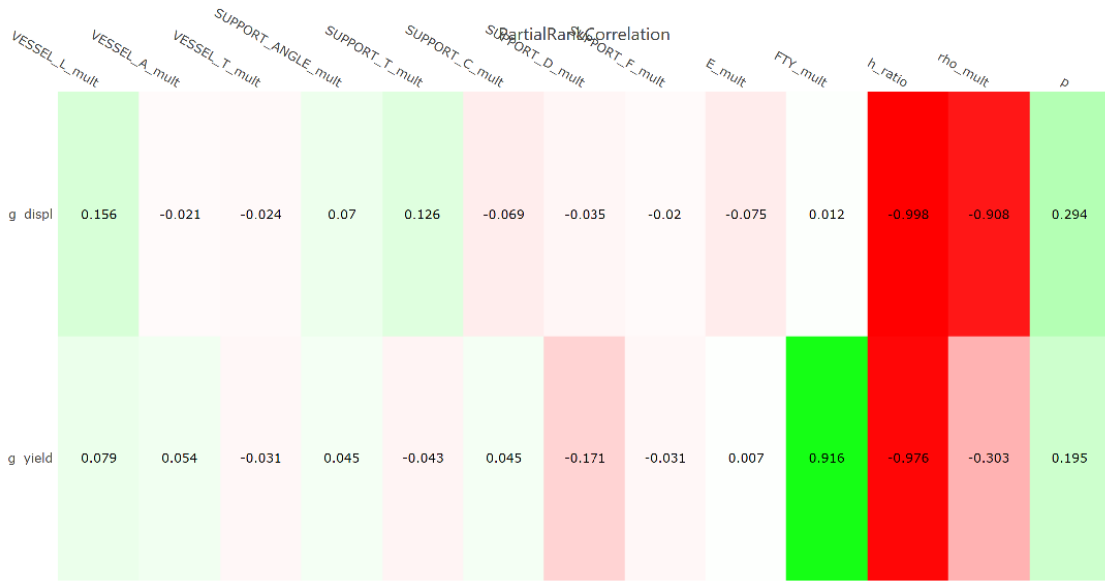


Figure 4-19: Sensitivity, partial rank correlation

Partial rank correlation capable of measuring non-linear sensitivities is shown on Figure 4-19. Positive correlation is shown in green, negative correlation in red. The intensity of the colors indicates the strength of the correlation. For this model the findings are consistent with the uncertainty informed sensitivity results shown in Figure 4-16 both in magnitude and sign.

The workflow in this section is in the SAW project `IWF_L2_Milestone_FY2019-UUR` in folder of `1-Exemplars\2-Storage_Tank\4-Credibility\1-PCMM\5-Uncertainty_Quantification\1-UQ\1-Incremental_LHS`.

4.1.8 Plausible Program Decisions

These studies suggest several program decisions regarding the operation of the storage tank as follows:

- Negotiate acceptance criteria: Since the LHS study suggests a small probability of not meeting margin requirements, examine where the requirements came from. Do we really need to maintain a factor of safety of 3.0? Can we tolerate more displacement?
- Use deterministic sensitivities to tune the design. For example, Figure 4-14 indicates that increasing vessel thickness or decreasing the radius may be effective ways of increasing margins.
- Restrict operations. Limit the height of the liquid the tank is allowed to take on, although this may not be practical.
- Quantify the probability of not meeting margin and use this information in risk assessment. This requires using a more accurate UQ method from the Dakota toolset.

- Live with the plausibility of not meeting margins. This may not be a desirable decision, since risk (loosely defined as the probability of occurrence x cost of occurrence) is not quantified.

4.2 Exemplar 4 - Abnormal Mechanical Crush of an Assembly

This model is a demonstration problem for end-to-end credibility focused initial assessment of a non-trivial SM (Solid Mechanics) problem. Many typical CompSim tools such as Cubit, Sierra, Ensight, post processing tools from the Engineering Sciences Tool Repository (ESTR) and ARG are involved in the analysis workflow, and each instance executes for hours on hundreds of processors. Representative workflows are highlighted in this report, and the complete working set is in the SAW project `IWF_L2_Milestone_FY2019-UUR` in the folder `1-Exemplars/4-Abnormal_Mechanical_Impact`.

4.2.1 Problem Definition

A small assembly is crushed between a steel target and a steel impactor, and the system engineering organization is asking at what impact velocity will loss of function occur. Before answering this a credibility assessment addressing parallel consistency and scalability of Sierra SM for this model

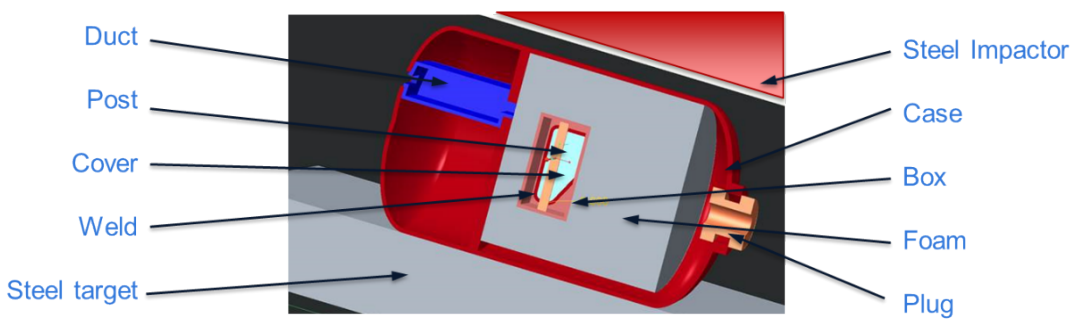


Figure 4-20: Crush of an assembly

needs to be assessed.

Loss of function is defined by mapping high level system requirements to QoIs that can be directly measured in experiments and evaluated from the model. In this demonstration problem two QoIs derived from use scenarios within Sandia's mission are considered. Shock environment – Maximum Shock Response Spectrum transformation of mass averaged vertical acceleration in the part "Post".

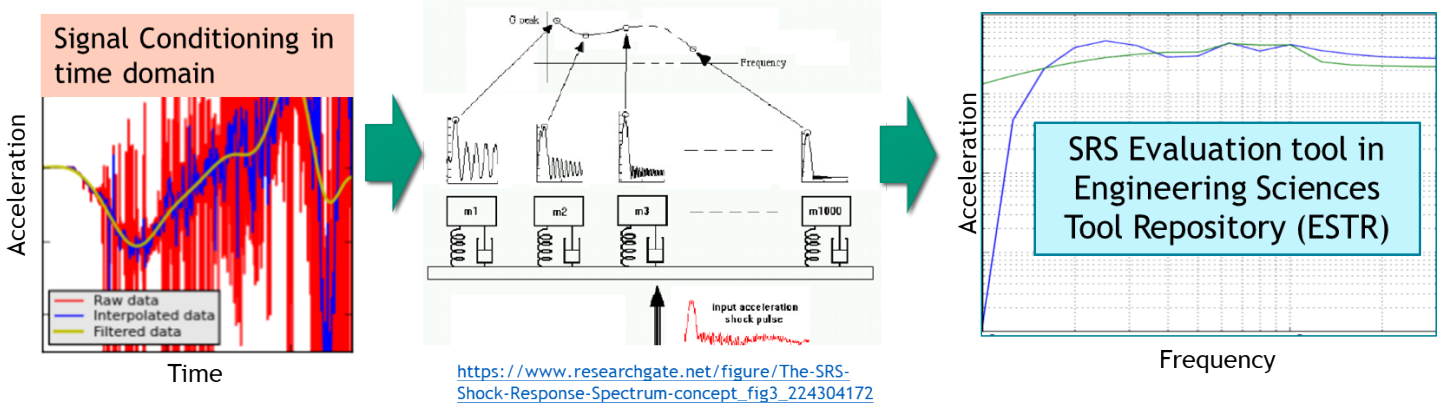


Figure 4-21: Calculation of SRS

The process is illustrated in Figure 4-21. Signal conditioning and SRS tools from Engineering Sciences Tool Repository are invoked in the analysis workflow.

Gap between the parts "Case" and "Plug" – During the impact event a gap between these two parts may open. The gap opening is transient, and it is evaluated using a custom Ensight macro not yet generalized enough to be added to ESTR. Figure 4-22 shows the time history of the gap and the spatial distribution of the gap at the end of the impact event on the cylindrical surface of the hole in "Case".

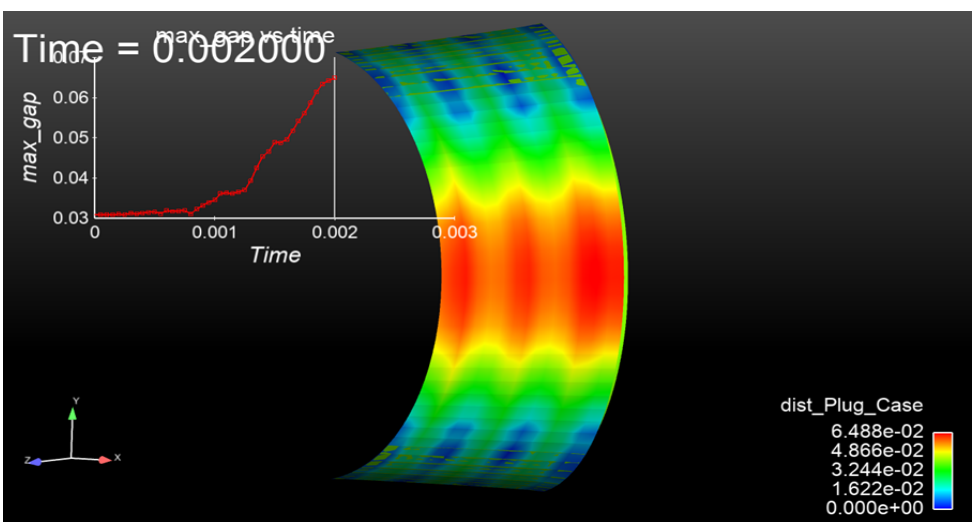


Figure 4-22: Gap opening between "Case" and "Plug"

velocity	Crusher impact velocity
elementsThroughThickness	Number of elements through case thickness
num_processors	Number of processors

Output	
gap	Gap between "Plug and Case"
SRS_max	Maximum SRS

Figure 4-23: Model parameters and responses

Model parameters in this initial study are summarized in Figure 4-23, and they are focused on solution verification and scalability aspects. In order to simplify mesh resolution studies, the model has a conformal hex mesh where all the mesh sizes in each part are scaled in relation to the number of elements through the case thickness.

4.2.2 Analysis and Documentation Workflow

The top-level analysis workflow is submitted to HPC from a workstation (Windows or Linux), and it has a switch to enable/disable ARG report generation. For the studies documented in this report obtaining a report for each sample is useful but in other scenarios such as UQ studies the reports may not be sufficiently different to be of value.

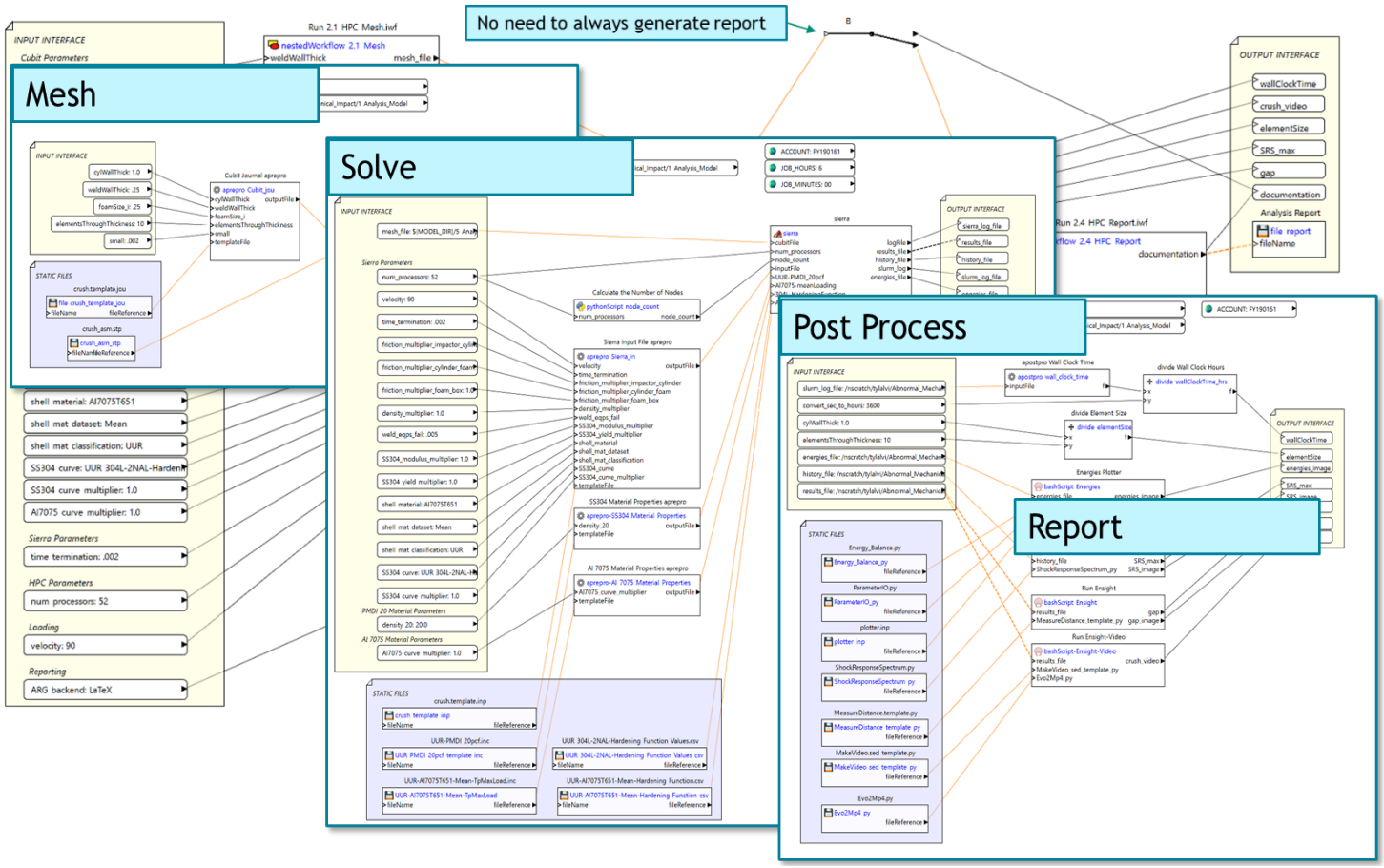


Figure 4-24: Atomic activities implemented as sub-workflows

This workflow may be viewed as the contract with the CompSim team describing the parametric interfaces between the embedded workflows. Having such clear interfaces enable agile model development by a team implementing the defined sub-workflows in parallel.

A large Dakota study of the analysis workflow described above running for four days changing impact velocity, mesh resolution and processor count has been performed. Execution of such a computationally demanding ensemble demonstrates robustness and resiliency of the entire NGW system.

4.2.3 Scalability

Figure 4-25 shows that for any mesh size the running this problem beyond 380 processors will not result in reduction of wall clock time. While independence of the optimal number of processors on model size (mesh resolution) is surprising it is a useful guideline for setting up larger UQ studies in the future. Examining other impact speeds, it was found that in all cases the point beyond which this model doesn't scale is also about 380 processors.

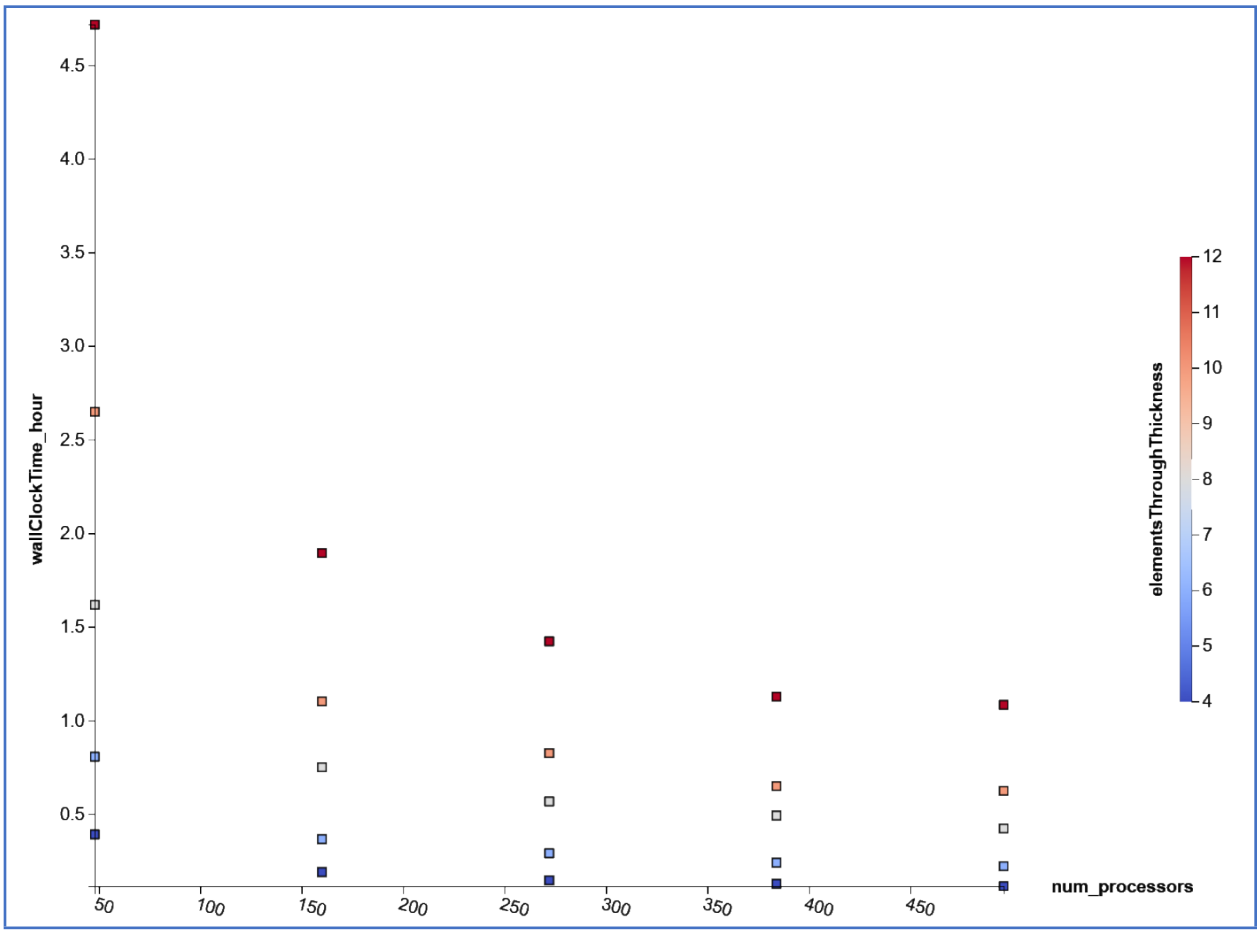


Figure 4-25: Scalability at 50 ft/s impact speed

4.2.4 Parallel consistency

At 50 ft/s impact speed it was found that irrespective of the mesh size the "gap" QoI was consistent is a sense that the variation in the numerical value was less than 0.5% when running the same problem on different number of processors. However, at higher impact speeds this consistency no longer held. The dataset (variables, responses, reports and videos) was sent to Slycat for detailed examination. In a process illustrated in Figure 4-26, reviewing the ARG reports for the individual samples revealed significant differences in the solutions at different processor counts.

The maximum SRS values exhibited similar behavior. Parallel consistency was found to be excellent at lower impact speeds and rapidly deteriorated as the impact velocity was increased. Figure 4-27 illustrates such investigation where is was found that the fracture patterns and, therefore, shock environments were at different processor counts.

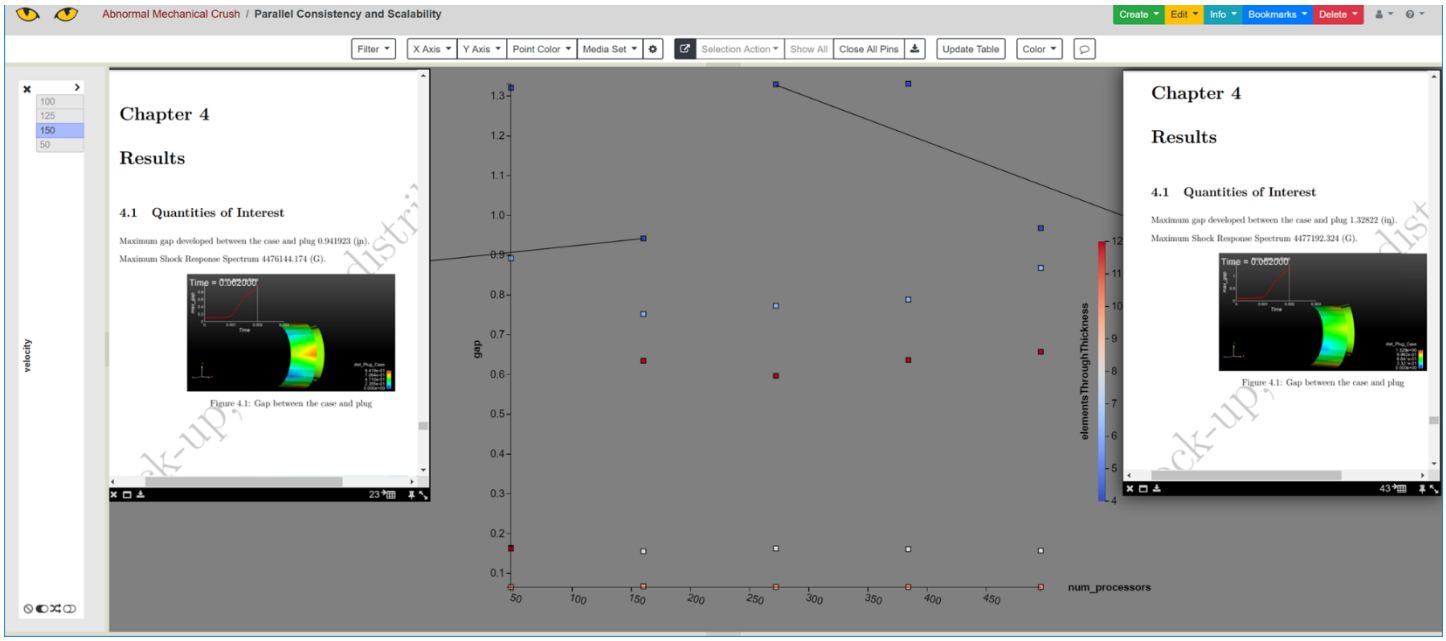


Figure 4-26: Viewing ARG reports for study samples in Slycat

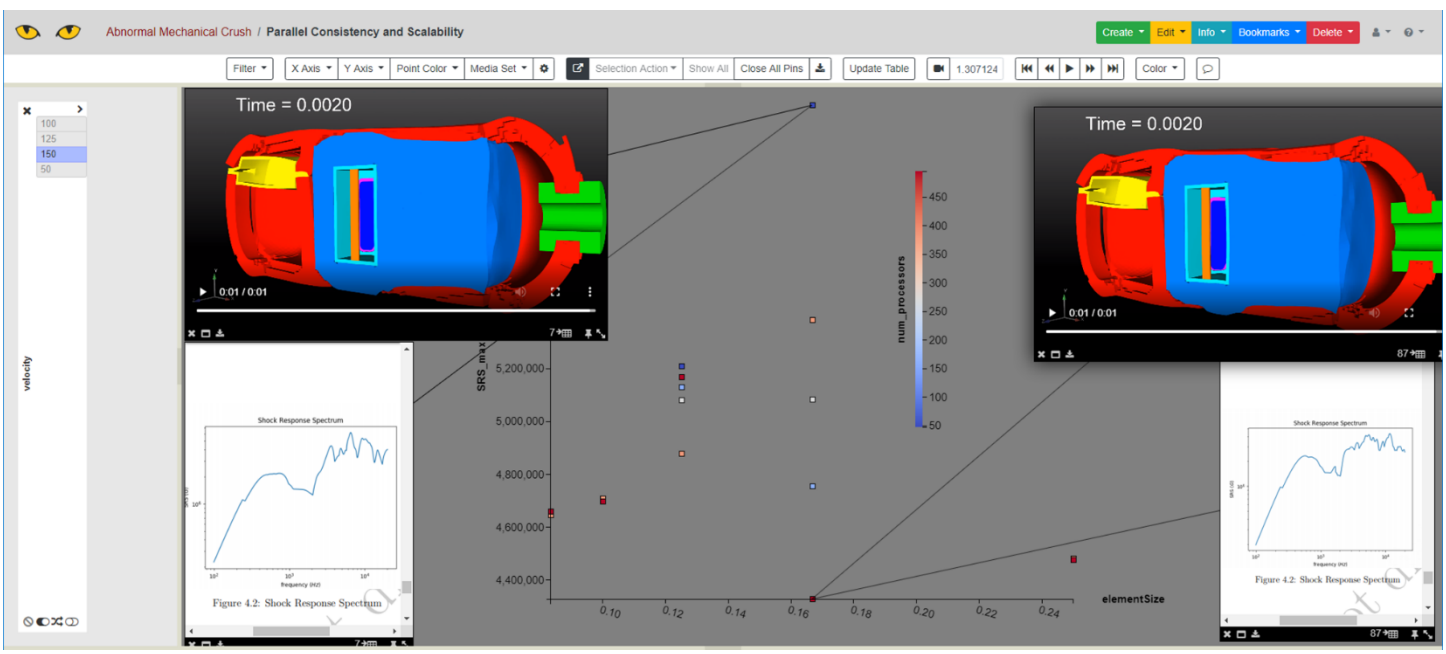


Figure 4-27: Viewing deformation and SRS sections of ARG report in Slycat

4.2.5 Implications on Model Credibility

NGW enabled a rare in-depth initial look at the interaction between scalability, parallel consistency and mesh resolution. It was found that: a) the model doesn't scale past 380 processors, b) significant parallel inconsistency is present at impact velocities beyond 50 ft/s, and c) mesh convergence is not yet indicated with the studied meshes at higher impact speeds.

These findings open more questions than they answer but this is exactly where the power of the toolset developed in this milestone is revealed. Analysts can now interrogate their models much more effectively than before, and ask deep probing questions from system engineering customers (what are the QoIs? what are the acceptance criteria?) as well as code developers (what are the limits of model scalability? how can parallel consistency be achieved?).

5 LIBRARIES OF REUSABLE COMPSIM AND TRAINING RESOURCES

During the L2 milestone existing configuration-controlled repository and access control capabilities provided by SAW SDM were used to develop CompSim library mechanisms.

5.1 Engineering Sciences Tool Repository (ESTR)

Important tools such as quantitative post processors that program decisions are based on are often not under configuration control, and their sharing is usually limited to workgroups. The existing SAW SDM configuration and access control mechanisms were used to collect a set of reusable CompSim tools in a repository called 'Engineering Sciences Tool Repository". Best practices requirements describing basic software quality assurance have been recorded in a document in the repository. These practices are: a) configuration control b) documentation, and c) regression testing. The repository is periodically synced from SRN to SCN to ensure a consistent set of capabilities on the two platforms. Tools meeting these requirements contributed by several analysts include:

- Creo (CAD) assembly explorer to regenerate parametric assemblies and interrogate metadata as well as mass properties to support credibility communications regarding the adequacy of FEM mesh to represent the mass of each part. The Creo assembly explorer tool was used in Exemplar 2, Storage tank.
- A set of signal processing tools to post process transient dynamics analyses from Sierra including SRS (Shock Response Spectrum) generation, and signal conditioning. The SRS tool was used in Exemplar 4, Abnormal Mechanical Impact.
- Dakota_to_Slycat is a pathfinder tool to connect Dakota studies, the media artifacts generated during execution of the samples to Slycat. It is used in most of the exemplars.
- Solid Mechanics data conditioning tools to aid material constitutive model calibration.
- Validation metrics library to quantify the difference between test data and predicted responses.

A complete listing of tools in ESTR is given in Figure 5-1. The SAW SDM repository is UUR, and it is named `Engineering_Sciences_Tool_Repository`, available on the SRN and SCN networks.

5.2 NGW End-to-End Exemplar Repository

A set of end-to-end CompSim exemplars described in Section 4, available on the SRN and SCN networks.

5.3 NGW Unit Training Scenario Repository

This repository OUO, and it is named `WorkflowSandbox`. This repository contains numerous exemplars of unit activities supporting training and one-on-one consulting. These unit test were instrumental on helping early adopters of the NGW system to construct their own workflows. This repository is available on the SRN network.

Creo		
	CreoWorkflowNode	Export Creo parts assemblies and metadata for meshing and reporting
HPC		
	SLURM_monitor	Monitor SLURM jobs
	SLURM_cancel	Cancel all SLURM jobs
	MOAB_monitor	Monitor MOAB jobs
	MOAB_cancel	Cancel all MOAB jobs
Signal Processing		
	Signal Synthesizer	Construct time series signals from various signal components (sine, wavelet, haversine, etc.)
	Signal Conditioner	Sample and filter time series signals, optionally take numerical derivatives
	SRS	Generates shock response spectrum from time series signal
SM (Solid Mechanics)		
	Least Squares Error	Calculates LSE between two curves and plots the two datasets
	Load-Displacement Interpolating	Smoothes noise data using the lowess (locally weighted scatterplot smoothing) approach. Interpolates/extrapolates smoothed data to key points to support calibration
	Solution Quality Assessment	Plots energy measures over multiple restarts
	Time Series UQ	Calculates and plots C.o.V as a function of time (multiple uses; load-displacement, SRS etc.)
V&V		
	DAKOTA	
	ParameterIO	Read and write Dakota parameters
	Visualization	Generates common plots for Dakota studies
	Dakota_to_Slycat	Exposes images and videos generated in a Dakota run to Slycat; Stand-alone and NGW node
	Validation Metrics	Evaluates various standard validation metrics
Video Processing		
	Evo2Avi, Evo2Mp4	Convert a set of Ensight .evo animation files to commmon formats

Figure 5-1: ESTR contents

5.4 ARG Exemplar Repository

This repository is OUO, and it is called `ARG_Exemplars-OUO`. It contains SM and SD exemplars with detailed instructions on how to generate reports for them, available on the SRN and SCN networks.

6 USER ENGAGEMENT

Based on suggestions by the milestone review panel, user engagement was formalized and enhanced midway through the year to assure smooth transition of the developed capabilities to production deployment in FY20. In addition to asking early adopters whether the right capabilities were being developed, the engagement helped developers to construct well-defined, actionable requirements – acknowledging that requirements are fluid, and with increasing use of the tools additional definition is likely.

User engagement by the IWF team started years before commencing the current milestone. A formal strategic meeting with stakeholders in a major program identified pain points such as manually authoring reports and reliably executing studies.

As near production level capabilities came on line in the middle of FY19 a one-on-one consultation based engagement model was adopted. Members of the IWF team spent time with early adaptors for a few weeks to help them to construct their workflows and studies, then stepped back to let the users experiment with the system. Early adopters ranging from summer interns to experienced analysts were asked to present their feedback as a regular "voice of the user" feature of the IWF sprint meetings. Their presentations were recorded, and their slides were archived and subsequently feature enhancements entered in the project tracker, Jira. Additionally, several dozen formal tickets were sent to both the SAW and the Dakota help sites. Most of these issues have been addressed while a few tickets were placed in the backlogs. Several early adopters became mentors to additional analysts justifying the consulting investments by the IWF team.

Among the enhancements initiated by early user engagement are extended features in graphical workflow definition and execution. Dakota study visualization, and context sensitive help. Both NGW and the Dakota usability features are in the respective production releases, and their regular user help systems will provide ongoing production-level support.

7 FOLLOW-ON WORK

Capabilities such as NGW, Dakota usability features and ARG developed in the milestone have reached production levels. The key mission for IWF participants for FY20 is to introduce these tools to the analyst community beyond the group of early adopters. The task "Democratization of Qualification and Design through Analysis – Production Hardened Platform and Training" is aiming to spread the use of these new tools to insert CompSim early in the design cycle and to promote credible CompSim model building and use practices.

Through implementing analysis and ensemble workflows for the exemplars, the L2 milestone has identified technical tasks that were entered in the IWF FY20 plan. These include data transfer between workflow nodes through hierarchical data containers to enhance visual and organizational clarity, enhanced library mechanisms providing search capabilities and separation of the roles of author and user, integration of new ATSS platforms, ARG extension and initial implementation of the Credibility Framework.

Additionally, findings about parallel consistency and scalability of Sierra Solid Mechanics models in the abnormal mechanical crush exemplar motivated a proposing an ASC V&V task to investigate these effects. Outcomes from this task include guidelines for analysts regarding parallel consistency of different QoIs under a variety of loading scenarios and providing the Sierra development team UUR exemplars they can examine in detail to find the root cause of this often significant numerical uncertainty.

8 SUMMARY

The new capabilities, usability focused update and integration of existing tools and case studies demonstrate that the milestone met its goals. Early user involvement assured that a core set of requirements was identified, and capabilities required to meet them were successfully implemented. Formal risk management informed deployment of the team's diverse talents in a priority driven manner. The resulting platform is now ready for deployment to analysts to transform how CompSim models are planned, developed, executed. The developed capabilities are foundational to CompSim credibility, development of models in an agile manner and decision-support-focused communication of results.

The milestone motivated several follow-on activities including a productionizing and training task "Democratization of Qualification and Design through Analysis – Production Hardened Platform and Training" and several technical tasks delivering capabilities needed to support construction, dissemination and use of large full system models the enterprise increasingly relies on.

REFERENCES

[1] Orient, G. E., Babuska, V., Lo, C. S., Mersch, J. P. (2017). An Approach Integrating PCMM, UQ and Evidence Theory for System Requirements Verification by Analysis with Solid Mechanics Models, presented at ASME V&V Symposium, Las Vegas, 2017
[2] Orient, G. E., Friedman-Hill, J., Hoffman, E. (2017). The Role of Computational Simulation Workflow in Decision Making for High Consequence Engineering Systems, presented ad NAFEMS Congress, Stockholm, 2017
[3] Orient, G. E., Babuska, V., Lo, C. S., Mersch, J., Wapman, P., W. (2018). A Case Study for Integrating CompSim Credibility and Convolved UQ and Evidence Theory Results to Support Risk Informed Decision Making, presented at IMAC-XXXVI, Orlando, 2018
[4] Friedman-Hill, E. J. (2019). The SAW Next Generation Workflow System, presented at NAFEMS Congress, Quebec City, 2019
[5] Sandia National Laboratories: The Sandia Analysis Workbench https://www.sandia.gov/saw/
[6] Sandia National Laboratories: The Next Generation Workflow project https://gitlab.com/iwf/ngw
[7] Eclipse Foundation: The Platform for Open Innovation and Collaboration https://www.eclipse.org
[8] Eclipse Foundation: Eclipse Modeling Framework (EMF) https://www.eclipse.org/modeling/emf/
[9] Eclipse Foundation: Graphiti - a Graphical Tooling Infrastructure https://www.eclipse.org/graphiti/
[10] Sarasvati Workflow Engine https://github.com/kmmbvnr/sarasvati
[11] Sandia National Laboratories: Dakota https://dakota.sandia.gov
[12] Parametric Technologies: Product Design 3D CAD Software
[13] Sandia National Laboratories: Cubit https://cubit.sandia.gov
[14] Sandia National Laboratories: Sierra suite https://sierradist.sandia.gov https://www.ptc.com/en/products/cad/creo
[15] Aidan Hendrickson, Philippe P. Pebay: Recommendations on a Document Structure Format for Automatic Report Generation, SAND2017-8890
[16] Philippe P. Pebay, Robert L. Clay: Installation and Testing Instructions for the Sandia Automatic Report Generator (ARG), SAND2018-4421
[17] Orient, G. E., Clay, R., Friedman-Hill, E. J., Hoffman, E. (2019). Credible Product Design and Qualification Workflow, presented at NAFEMS Congress Quebec City, 2019

DISTRIBUTION

Email—External (encrypt for OUO)

Name	Company Email Address	Company Name
Billings, Jay Jay	billingsjj@ornl.gov	Oak Ridge National Laboratory
Laney, Dan	laney1@llnl.gov	Lawrence Livermore National Laboratory

Email—Internal

Name	Org.	Sandia Email Address
Adams, Brian M	1463	briadam@sandia.gov
Akin Heitman, Lili A	1553	laakin@sandia.gov
Alvin, Kenneth F	1420	kfalvin@sandia.gov
Bishop, Joseph E	1556	jebisho@sandia.gov
Bennett, Janine C	8759	jcbenne@sandia.gov
Blecke, Jill	1512	jblecke@sandia.gov
Brightwell, Ronald B	1423	rbbrigh@sandia.gov
Burns, Shawn P	1514	spburns@sandia.gov
Clay, Robert	9152	rlclay@sandia.gov
Dike, Jay	8752	jjdike@sandia.gov
Dodd, Amanda B	9159	ajbarra@sandia.gov
Fang, Eliot H	1558	hefang@sandia.gov
Friedman-Hill, Ernest J	8753	ejfried@sandia.gov
Feddema, John T	1460	jtfedde@sandia.gov
Galpin, Terri L	1545	tlgalpi@sandia.gov
Glass, Micheal W	1545	mwglass@sandia.gov
Glickman, Matthew	1461	mrglick@sandia.gov
Glover, Steven F	1353	sfglove@sandia.gov
Heger, Arlen S	1350	aheger@sandia.gov
Heinstein, Martin	1545	mwheins@sandia.gov
Heroux, Michael A.	1400	maherou@sandia.gov
Hoffman, Edward L	8753	elhoffm@sandia.gov
Hough, Patricia	8754	pdhough@sandia.gov
Howe, Christina	1355	chowe@sandia.gov
Hutchinson, Scott A	1350	sahutch@sandia.gov

Name	Org.	Sandia Email Address
Klenke, Scott E	1550	seklenk@sandia.gov
Klitsner, Tom	9320	tklitsn@sandia.gov
Kramer, Michael R J	1351	rmkrame@sandia.gov
Lam, Cheryl	8751	clam@sandia.gov
Langston, William L	1352	wllangs@sandia.gov
Lefantzi, Sophia	1516	slefant@sandia.gov
Manginell, Ronald P	1513	rpmangi@sandia.gov
McNeish, Jerry	8754	jmcneis@sandia.gov
Mish, Kyran D	1555	kdmish@sandia.gov
Nelson, Stacy M	8752	smnelso@sandia.gov
Oldfield, Ron A	1461	raoldfi@sandia.gov
Ostien, Jakob	8363	jtostie@sandia.gov
Owens, Brian C	1554	bcowens@sandia.gov
Smith, Justin	1515	jussmit@sandia.gov
Stewart, James R	1440	jrstewa@sandia.gov
Strack, Erik O	1150	oestrac@sandia.gov
Payne, Jeffrey L	1510	jlpayne@sandia.gov
Pott, John	1557	jpott@sandia.gov
Templeton, Jeremy A	1151	jatempl@sandia.gov
Turner, Daniel Z	1463	dzturne@sandia.gov
Urbina, Angel	1544	aurbina@sandia.gov
Vigil, Dena	1424	dmvigi@sandia.gov
Witkowski, Walter R	1540	wrwitko@sandia.gov
Wix, Steven D	1356	sdwix@sandia.gov
Technical Library	9536	libref@sandia.gov

This page left blank