SANDIA REPORT

SAND2005-4647
Unlimited Release
Printed August 2005

DART System Analysis

Michael F. Hardwick, Sandia National Laboratories
Robert L. Clay, Sandia National Laboratories
Paul T. Boggs, Sandia National Laboratories
Edward J. Walsh, Sandia National Laboratories
Alex R. Larzelere, Exagrid Engineering, Inc.
Alan Altshuler, Exagrid Engineering, Inc.

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

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

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@adonis.osti.gov
Online ordering:	http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online order:	http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SANDIA REPORT
SAND2005-4647
Printed August 2005

DART System Analysis

Sandia National Laboratories
Michael F. Hardwick
Robert L. Clay
Paul T. Boggs
Edward J. Walsh

Exagrid Engineering[1]
Alex R. Larzelere
Alan Altshuler

ABSTRACT

The Design-through-Analysis Realization Team (DART) is chartered with reducing the time Sandia analysts require to complete the engineering analysis process. The DART system analysis team studied the engineering analysis processes employed by analysts in Centers 9100 and 8700 at Sandia to identify opportunities for reducing overall design-through-analysis process time. The team created and implemented a rigorous analysis methodology based on a generic process flow model parameterized by information obtained from analysts. They also collected data from analysis department managers to quantify the problem type and complexity distribution throughout Sandia's analyst community. They then used this information to develop a *community model*, which enables a simple characterization of processes that span the analyst community.

The results indicate that equal opportunity for reducing analysis process time is available both by reducing the "once-through" time required to complete a process step and by reducing the probability of backward iteration. In addition, reducing the rework fraction (i.e., improving the engineering efficiency of subsequent iterations) offers approximately 40% to 80% of the benefit of reducing the "once-through" time or iteration probability, depending upon the process step being considered. Further, the results indicate that geometry manipulation and meshing is the largest portion of an analyst's effort, especially for structural problems, and offers significant opportunity for overall time reduction. Iteration loops initiated late in the process are more costly than others because they increase "inner loop" iterations. Identifying and correcting problems as early as possible in the process offers significant opportunity for time savings.

[1] http://exagridconsulting.net/index.html

Table of Contents

Executive Summary 7

Introduction 11

The Generic DTA Process Model 13

The Analysis Methodology 17

Data Collection and Characterization 19

The Analyst Community Model and Results 23

Summary and Recommendations 31

Appendix 1: The Data Collection Tool 35

Appendix 2: The Analysis Algorithm 43

Distribution 47

List of Figures

Figure 1. Generic DTA Process Model 13

Figure 2. Qualitative transition probability evaluation results 29

Figure 3. Metadata Input Sheet 36

Figure 4. Step Times Input Sheet 37

Figure 5. Transition Probability Matrix Input Sheet 39

Figure 6. Results Sheet 40

Figure 7. Button Bar 40

List of Tables

Table 1. Average normalized engineering time per problem type. 8

Table 2. Average expected number of visits per problem type 9

Table 3. Data set problem type and complexity distribution 20

Table 4. Step times input data averages per problem complexity (engineering time in hours). 21

Table 5. Step times input data averages per problem type (engineering time in hours) 21

Table 6. Baseline problem category weightings according to FTE loads 23

Table 7. Community output engineering time per problem type (engineering time in hours). 24

Table 8. Community output engineering time per problem type (normalized) 25

Table 9. Community output expected number of visits per problem type 25

Table 10. Community model results showing each category weight increased by 20% 26

Table 11. Initial step time and rework fraction sensitivity estimates as calculated using the community model with baseline weights. 27

This page intentionally left blank.

Executive Summary

The Design-through-Analysis Realization Team (DART) is a group chartered with reducing the time required for Sandia analysts to complete the engineering analysis process. The DART system analysis team studied the analysis processes used by analysts in centers 9100 and 8700. With the study results, DART leadership is better able to identify opportunities for process improvement, both to decrease the overall analysis time and to improve the quality of the analysis results. This study employs a generic 10-step design-through-analysis process model parameterized with information acquired directly from analysts. The analysts provided information for combinations of problem type (modal, linear structural, non-linear structural, heat transfer, fluid flow, and radiation transport) and complexity (simple, medium, and complex) that included estimates of both the time required to complete individual process steps and the likelihood of having to iterate through various portions of the process. The input data were analyzed to produce results that include the expected time spent in each step and the expected number of visits to each step. In addition to calculating such results for each combination of problem type and complexity, this study developed a *community* model that averages the results using analyst workload data for the different problem types and complexities. The community model represents a weighted average analysis for the analyst community. Because this study did not include a formal uncertainty analysis, the results are not quantifiably justified. Nevertheless, they provide useful insight and can augment engineering judgment when making programmatic decisions.

This scope of this study is limited to the processes employed by the analyst and does not necessarily account for all activities required to complete an engineering analysis. For example, the study does not account for the time a design engineer and drafter spend to develop a CAD model.

Table 1 presents the normalized engineering times associated with each process step for individual problem types and for the community model. The results in Table 1 are calculated only through the first simulation; that is, no subsequent runs for parameter, sensitivity, or optimization studies are considered. The analytical model does, however, enable one to account for these additional runs. Note that the values for each problem type are computed as weighted averages across the three problem complexity categories. Geometry manipulation, represented by steps B and C, consume the majority of the overall analysis time for structural problems and represent a substantial opportunity for overall process improvement. Table 2 shows the expected number of visits to each process step and also indicates a high degree of iteration in geometry and meshing steps for modal and nonlinear structural problems.

Table 1. Average normalized engineering time per problem type.

		Modal	Linear Struct.	Non-Lin Struct.	Heat Trans.	Fluid Flow	Rad. Trans.	Comm. Model
A	Design Solid Model Creation and/or Edit	0.00	0.00	0.05	0.13	0.00	0.15	0.04
B	Analysis Solid Model Creation and/or Edit	0.37	0.26	0.10	0.12	0.49	0.05	0.21
C	Geometry Decomposition	0.24	0.34	0.41	0.07	0.03	0.03	0.32
D	Meshing	0.12	0.11	0.16	0.06	0.13	0.25	0.14
E	Mesh Manipulation	0.08	0.05	0.05	0.03	0.02	0.08	0.06
F	Model Parameter Assignment	0.06	0.04	0.06	0.04	0.04	0.05	0.06
G	Simulation Model Assembly	0.10	0.02	0.05	0.14	0.02	0.02	0.08
H	Run Simulation	0.02	0.09	0.05	0.03	0.03	0.26	0.04
I	Results Post-Processing	0.01	0.09	0.04	0.37	0.22	0.07	0.05
J	Archiving of Analysis Artifacts	0.00	0.01	0.01	0.01	0.01	0.04	0.01

In addition to the baseline step time and visit outputs, the study also evaluates the analysis time sensitivity to perturbations in individual input parameters, which is useful for identifying process steps that might produce large time savings for small investments. Since a single set of baseline input parameters doesn't exist (we created these only for the outputs), calculating a single set of baseline input parameter sensitivities is difficult at best. Because the use of a synthesized input set that results in the baseline output results is not mathematically defensible for calculating the desired sensitivities, a more basic finite difference method is used to effectively construct a set of community sensitivities for two sets of model parameters, the initial step times and subsequent visit work fractions. These sensitivities, which are discussed in more detail in the body of this report, indicate the degree to which improvements in basic process capability manifest as overall design-through-analysis time improvements. By estimating a cost associated with the improvement, a reasonable return-on-investment prediction can be made. Such insight helps predict where funding may most effectively be allocated. For example, if the CUBIT project estimates that a proposed geometry tool will reduce the average time to complete the geometry decomposition step by 25% and will reduce the likelihood of returning to the geometry decomposition step after meshing by 33%, the process sensitivities might indicate an overall time savings of 10%. If the cost to implement the proposed improvement is $500K, then the return on investment is 10% process improvement/$500K, a metric directly comparable with other proposed improvements.

Table 2. Average expected number of visits per problem type

		Modal	Linear Struct.	Non-Lin Struct.	Heat Transfer	Fluid Flow	Rad. Trans.	Comm. Model
A	Design Solid Model Creation and/or Edit	2.89	0.79	13.45	2.01	1.00	2.63	7.40
B	Analysis Solid Model Creation and/or Edit	5.34	2.22	21.21	2.12	1.20	4.30	11.78
C	Geometry Decomposition	10.63	2.79	20.94	1.49	1.20	4.35	12.47
D	Meshing	12.46	2.52	19.58	2.36	1.20	3.78	12.17
E	Mesh Manipulation	10.91	2.15	11.86	1.55	1.20	4.48	7.99
F	Model Parameter Assignment	9.79	2.69	7.33	3.63	1.70	4.48	6.10
G	Simulation Model Assembly	10.86	2.01	9.59	2.38	1.70	3.45	6.94
H	Run Simulation	6.47	2.01	9.34	3.99	1.70	3.45	6.53
I	Results Post-Processing	4.32	2.01	8.81	3.45	1.70	1.62	5.94
J	Archiving of Analysis Artifacts	1.12	0.79	3.83	1.00	1.00	1.00	2.37

Sensitivity calculations associated with the initial time required for an analyst to complete a process step and the fraction of rework required for subsequent visits show that changes in the geometry manipulation and meshing steps produce the greatest changes in process time for the overall community, at least in the neighborhood of the parameter selection defined by the community model.

Practical considerations described in the body of the report prohibit a rigorous evaluation of sensitivities related to process step transition probabilities, which are directly related to process iteration loops. Nevertheless, considerable insight is gained from a less rigorous mathematical evaluation of these sensitivities. The study identifies a general trend of increasing sensitivity for transition probabilities associated with later process steps, which implies that "outer" iteration loops cost more than "inner" iteration loops, an expected result. Two factors are important when considering iteration loops initiated by process problems: the step at which a problem was created and the step at which that problem was discovered. With this recognition, the value of identifying and correcting problems as early in the process as possible becomes evident. In addition, the study shows that the sensitivities associated with geometry manipulation and meshing are greater than would be indicated simply by their position in the process flow, reinforcing the conclusion that they represent areas for improvement. In general, the potential impacts one might expect to realize from similar relative improvements in step times and transition probabilities are equal. One might therefore equitably allocate resources to efforts designed to reduce the time required to complete a pass through an important process step and to efforts to minimize process iteration, presumably by identifying and correcting mistakes as early in the process flow as possible.

Reducing the rework fraction, the portion of original work that must be repeated during subsequent step visits, also offers significant opportunity to reduce overall process time. This study shows that the impact of reducing rework fraction is approximately 40% to 80% of that for reducing the "once-through" time or the iteration probability. Tools that preserve information generated during previous process step visits and that make it readily available during subsequent visits offer the promise of reducing rework fractions. With comparable sensitivities, all three major process parameter sets ("once-through" times, rework fractions, and iteration probabilities) should be considered important opportunities for process improvement.

Introduction

The Design-through-Analysis Realization Team (DART) initiative's primary goal is to significantly reduce the design-through-analysis (DTA) process time for the Sandia analyst community. Measuring such time, however, is a formidable task because the processes employed by analysts vary significantly and because analysts do not rigorously track the time they spend working individual process steps. Without understanding the nature of the DTA process as practiced at Sandia, at least in a generally quantitative way, it is difficult for DART leadership to best allocate limited resources and to predict the impact of expected process improvements.

The goal of the system analysis effort is to produce a DTA process model that generates quantitative estimates of important process metrics, such as flow times associated with specific process steps. With such information, we can characterize the current process and evaluate the potential impact of proposed improvements, and then allocate resources to those tasks likely to produce the most significant improvements. The system analysis described here focuses on evaluating the primary DTA goal of reducing overall process time. To bound the analysis effort, this report does not directly address other goals such as improving the quality of analysis results.

The analysis is based on a generic, parameterized DTA process model, populated with data provided by Sandia analysts. Drawing on their own experience and using a data collection tool specifically developed for this study, the analysts enter data sets representing a variety of simulation problem types and complexities. The analyst data is then used to parameterize the process model that in turn provides information that DART leadership can use for planning.

This page intentionally left blank.

The Generic DTA Process Model

The analysis community is heterogeneous, so no single analysis process description readily characterizes the depth and breadth of processes collectively used by analysts. Factors that affect the nature of an analysis project are many and include the problem type (e.g., structural or thermal), the selected solver code, the number of parts in an assembly, the number of model elements, the required level of model and results fidelity, and the experience of the analyst. One may consider these factors as characterizing different sectors of the DTA community. This analysis captures information sufficient to characterize a data set only by problem type and complexity, as measured by the number of parts and elements in the model.

Despite the heterogeneous nature of the DTA process, this analysis is based upon a single generic process model. The level of detail in the generic process model is balanced to capture the desired process details without creating an overly complex[2] description. Specifically, the generic model includes the following steps (see Figure 1).

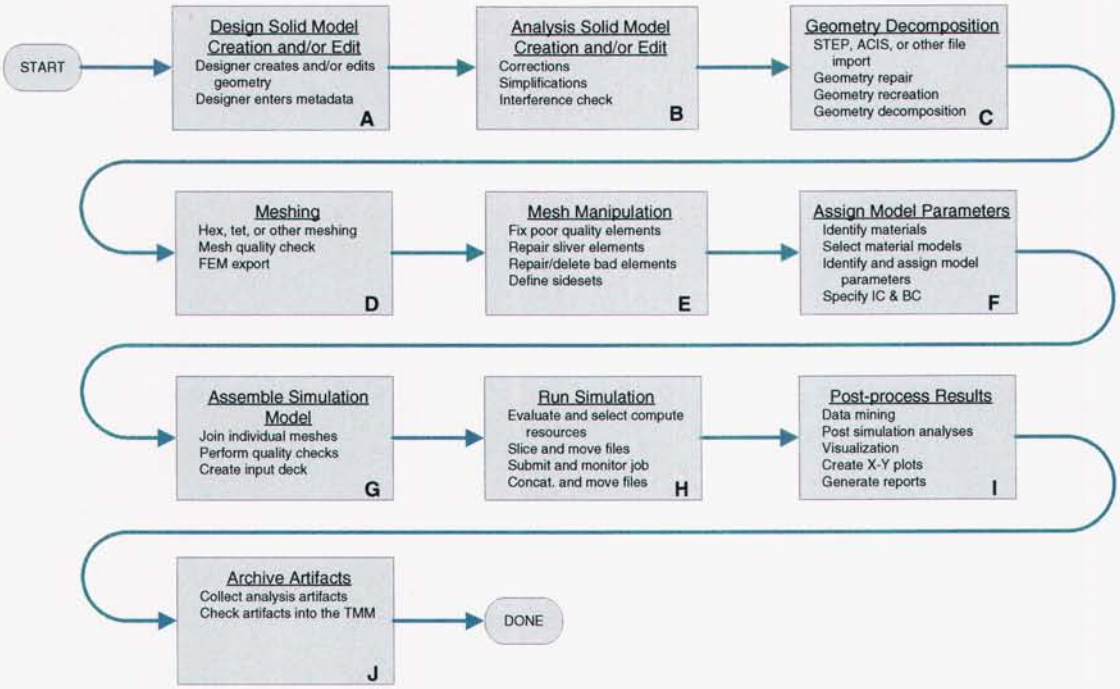


Figure 1. Generic DTA Process Model

[2] In this context, "overly complex" means containing more detail than the measured data can support.

A. Design Solid Model (DSM) Create/Edit

As currently defined at Sandia, the DSM is a 3D solid model of a part or assembly with information sufficient for manufacturing. These models contain all features that a manufactured part is expected to have. While they are most often generated by a designer whose primary job responsibility is to create such models, at times they are created and edited by an analyst. Tools used to create DSMs include *Pro/Engineer*[3] and *SolidWorks*[4].

B. Analysis Solid Model (ASM) Create/Edit

Although an ASM is generally created from a design solid model, analysts sometimes directly create ASMs on their own. In the former case, an analyst and CAD designer partner to remove model features (a process called defeaturing) that are not important to the planned analysis but otherwise add model complexity. Typically removed features include fillets, chamfers, rounds, and holes.

C. Geometry Decomposition

For some classes of problems the first step in generating a mesh is to decompose complex geometry into a collection of more basic shapes from which a mesh can be more easily created. Hexahedral meshes often require geometry to be decomposed while tetrahedral meshes can often be generated with little or no geometry decomposition.

D. Meshing

Meshing is the process of instantiating finite elements to represent the geometry being modeled. Many different mesh-building tools are employed by analysts, including *CUBIT*[5], *Patran*[6], and *CosmosWorks*[7]. The analyst often performs a mesh quality check on the final mesh using a tool such as *Verde*[8]. Finally, the mesh is usually exported for use in other tools by converting it to *Exodus*[9] or other data format.

---

[3] http://www.ptc.com
[4] http://www.solidworks.com
[5] http://cubit.sandia.gov
[6] http://www.mscsoftware.com/products/products_detail.cfm?PI=6
[7] http://www.solidworks.com/pages/products/cosmos/cosmosworks.html
[8] http://cubit.sandia.gov/verde
[9] http://endo.sandia.gov/SEACAS/Documentation/exodusII.pdf

E. Mesh Manipulation

A next step toward building the desired model is to manipulate the mesh created in the previous step. Such manipulation may include repairing or removing poor quality elements, nudging node locations, and creating side sets.

F. Model Parameter Assignment[10]

After creating the mesh, the analyst must establish values for various parameters in order to completely define the model. Needed values typically include parameters for material constitutive models, boundary conditions, initial conditions, and loads.

G. Simulation Model Assembly

If more than one analyst creates a single model or if a single analyst decomposes the meshing work into separate tasks, it is likely that individual part or subassembly meshes must be joined to create the overall system mesh. Further, some analyses, such as a weapon drop test simulation, require specific mesh positioning in space. The work to assemble and position individual meshes is referred to in this analysis as simulation model assembly. Also included in this step are final mesh quality checks and creation of the input files required by the chosen solvers.

H. Simulation Run

Once the finite element model is completely defined and the appropriate input deck generated, the analyst submits the job to a compute system. Job submission may require several steps, including partitioning the problem for parallel processing, moving the partitioned files to the appropriate file system, entering job parameter information, monitoring ongoing jobs, and retrieving and concatenating results files. Because Sandia compute systems are heavily used, jobs often wait in a queue for processing. Queue time is included in simulation run time, but has no impact on engineering time.

[10] While the analysis methodology was being developed, analysts presented the system analysis team with a request for a slightly modified process to be available as an alternative to the flow just described. This alternative flow defines step F, *Assign Model Parameters*, as occurring in parallel with steps A through E. The intent is enable analysts who begin defining material properties early in the overall process to better represent their DTA process. Although this option was made available, no analysts who provided data for this study chose to use it.

I. Results Post-processing

After retrieving the results files from a successful simulation run, the analyst reviews the data to obtain information of interest. The review may be as simple as identifying standard solver output in critical mesh regions or may require a substantially more complex post-simulation analysis. Tools in common use at Sandia for visualizing results and performing additional analyses include *Paraview*[11] and *EnSight*[12].

J. Simulation Artifact Archiving

At the conclusion of a simulation or set of simulations, the analyst archives the analysis artifacts, which may be numerous. At present, many analysts simply copy the artifacts to a specific storage location. At some point a more sophisticated archiving system may be available. Documentation generation is not considered an archiving task.

It is worth emphasizing that no single process description can perfectly represent the actual analysis processes employed by analysts. This ten-step representation is intended to provide a balance between the complexity required to obtain meaningful information and the simplicity needed to perform a reasonable analysis.

Clearly, no analyst actually works progressively through the ten generic analysis steps without encountering problems and having to repeat previous efforts. One similarity among all analysts is the fact that they iterate within the process flow. While not explicitly illustrated in the generic process diagram, the system analysis methodology allows flow from any process step directly to any other step and can account for any and all backward or forward iterations an analyst might wish to represent.

[11] http://www.paraview.org/HTML/Index.html
[12] http://www.mscsoftware.com.au/products/software/cei/ensight

The Analysis Methodology

To perform the analysis, the system analysis team collected data to parameterize the generic process model so that it represents the actual processes employed by analysts. The model encapsulates several important parameters as described below.

1. **Initial step times** represent the amount of time required for an analyst to complete each step in the generic process the first time it is encountered for the problem being considered.
2. **Rework fractions** represent the portion of the original work that is typically repeated during the second and subsequent visits to a process step. This parameter accounts for analyst learning and typically has a value less than unity.
3. **Initial transition probabilities** represent the probability of branching from the output of one process step to the input of another step the first time a step is encountered. They comprise an initial transition probability matrix and enable the analysis to model process iteration loops. The probabilities are decimal values between zero and one.
4. **Subsequent transition probabilities** are similar to their initial counterparts, but are used for the second and subsequent visits to a process step. They are intended to account for differences in iteration probability due to analyst learning.

Appendix 1, which describes the data collection tool, provides additional information about these and other parameters. Once the initial step time, rework fraction, and transition probability data are specified, the process model is run to provide information about the expected number of visits to each process step, the expected time in each step, and the total process time. This information is presented to the analysts as they enter the data to provide real-time validation of the input. The analysis algorithm is described in detail in Appendix 2.

The system analysis team also collected information about the distribution of various problem types (modal, linear structural, etc) and complexities across the Sandia analyst community and used it to develop a *community model*, which is an appropriately weighted average of the analysis outputs from all of the collected data sets. In addition to these results, the team performed sensitivity studies to develop insight about the relative impact of individual and collective parameter changes, which would be an indicator of potential high impact investment areas.

This page intentionally left blank.

Data Collection and Characterization

The single most difficult part of this analysis was the effort to collect meaningful data from analysts. An initial attempt to do so used a paper data collection form that analysts were requested to complete with help from a system analysis team member. In terms of collecting meaningful data, this first attempt was a failure. However, the analysis team learned much about the collection process and subsequently produced a web-based tool to enable acquisition of higher quality data. A description of this tool is included in Appendix 1.

During the initial data collection attempt, analysts did not have immediate access to the process model results calculated from their input. Since the relationship between analysts' input data and the overall process characteristics is not intuitive, analysts found it difficult to confirm that their input was appropriate. The web-based tool not only provided a convenient mechanism for data collection, but also calculated the resulting process model statistics and presented them to the analyst. Analysts iteratively entered data and calculated the results until they were satisfied that both the inputs and the outputs described the process as they intended to represent it. This real-time-feedback approach was very effective.

The data collection tool was not meant to be completely self-explanatory. Rather, the system analysis team intends it to be used with their guidance, at least the first time an analyst exercises it. To help the analyst community provide data, the system analysis team held four data collection sessions, one in California on December 9, 2004 and three in New Mexico on Dec 16, 2004. During these sessions, analysts received an introduction to the collection process and entered data with help from the team members. Twenty-three analysts participated in these sessions and collectively submitted 34 useful data sets.

The data sets spanned problem complexities (12 simple, 14 medium, and 8 complex) and types (7 modal, 4 linear structural, 12 non-linear structural, 5 heat transfer, 1 fluid flow, and 5 radiation transport). The distribution is shown in Table 3. While the tool continues to be available and in use, the analysis presented in this report uses only the information acquired during the cited data collection sessions.

Table 3. Data set problem type and complexity distribution

	Modal	Linear Structural	Non-linear Structural	Heat Transfer	Fluid Flow	Rad Transport	Total
Simple	4	1	3	2	0	2	12
Medium	2	3	5	1	1	2	14
Complex	1	0	4	2	0	1	8
Total	7	4	12	5	1	5	34

The data was first evaluated to understand the variation in the information provided by the analysts. A simple, albeit crude, method for measuring that variation is to use the ratio of standard deviation to average for input quantities within various community sectors. While the data sampling is too small to draw strong statistical conclusions, the measure does provide a feel for the variability with which different analysts employ their DTA process as well as the variability in the difficulty of problems they encounter. When evaluating this information, one must consider the uncertainty associated with differing degrees of data entry fidelity by the analysts. Although they were instructed to iterate with the data collection tool until comfortable with both the input and results, the system analysis team expects that some analysts were more rigorous than others.

Within each problem type and complexity category, the ratio of standard deviation to average was usually greater than 0.5 and in many cases the standard deviation exceeded the average. This substantial input data variation indicates that analysts do not employ a common DTA process and/or that problem difficulty is varied. While the general steps that analysts traverse may be similar, the nature of how they develop a simulation varies significantly. This conclusion is based on the assertion that the analyst data is valid, which the system analysis team accepts as true. An awareness of the heterogeneous nature of analysis processes used at Sandia should guide conclusions drawn from this system analysis.

Even though significant variation in the input data exists, one may gain insight by examining the average values associated with the overall analyst community and within various sectors. Average engineering time from the *step times data collection* sheets is shown in Table 4 for the overall community and for simple, medium, and complex problems. Table 5 shows average engineering time for each problem type.

Table 4. Step times input data averages per problem complexity (engineering time in hours).

		Overall	Simple	Medium	Complex
A	Design Solid Model Creation and/or Edit	12.41	2.15	21.43	12.00
B	Analysis Solid Model Creation and/or Edit	97.77	12.33	57.31	296.71
C	Geometry Decomposition	57.74	15.36	16.00	194.36
D	Meshing	43.99	5.86	38.29	111.18
E	Mesh Manipulation	24.88	5.20	12.43	76.18
F	Model Parameter Assignment	23.13	6.17	8.91	73.45
G	Simulation Model Assembly	25.70	3.13	8.61	89.45
H	Run Simulation	22.31	4.90	36.02	24.42
I	Results Post-Processing	34.83	10.30	23.50	91.47
J	Archiving of Analysis Artifacts	7.47	2.07	7.46	15.60

Table 5. Step times input data averages per problem type (engineering time in hours)

		Modal	Linear Struct.	Non-Lin Struct.	Heat Transfer	Fluid Flow	Rad. Trans.
A	Design Solid Model Creation and/or Edit	2.97	0.00	9.33	24.60	0.00	33.20
B	Analysis Solid Model Creation and/or Edit	312.22	54.40	43.00	48.80	120.00	8.20
C	Geometry Decomposition	154.55	38.00	42.73	36.80	8.00	4.90
D	Meshing	84.00	33.00	38.80	19.62	32.00	36.00
E	Mesh Manipulation	78.19	18.20	7.10	11.30	4.00	16.00
F	Model Parameter Assignment	54.53	8.20	16.20	19.40	8.00	14.50
G	Simulation Model Assembly	56.25	6.60	13.72	55.40	4.00	1.60
H	Run Simulation	14.48	31.70	18.75	6.56	8.00	52.90
I	Results Post-Processing	7.13	29.00	24.13	123.40	48.00	12.77
J	Archiving of Analysis Artifacts	4.80	2.00	10.43	4.80	2.00	12.26

This page intentionally left blank.

The Analyst Community Model and Results

DART spans a variety of problem types and complexities, and is developing solutions for the entire analyst community. It is therefore necessary to understand the distribution of the various problem categories throughout that community, and to analyze the system as an integrated whole. Thus we examined the problem category distributions and developed an integrated community model of the DTA process.

The problem category (type/complexity) weighting results are presented in Table 6. This is a baseline result given our measure of the current FTE work loads assigned to each problem category, the data for which was provided by the analysis department managers and based on nearly 70 FTEs worth of ongoing workload. A compilation of the weighting data is available in a spreadsheet developed for this exercise and stored in Web Fileshare (Model ID: WFS WFS253069). It is important to note that these weights are estimates, and furthermore, they are certain to change over time as the workload configuration changes. Hence, examining the sensitivities is useful if not necessary to better understand the results. A thorough weighting sensitivity analysis is outside the scope of this report, but such studies are ongoing and are expected to be published. However, in the interim, we do provide some revealing insights into the sensitivities based on a few parametric variations of the weights. Before considering those conclusions, we first describe the community model.

Table 6. Baseline problem category weightings according to FTE loads

	Modal	Linear Structural	Non-linear Structural	Heat Transfer	Rad Transport	Fluid Flow	Total
Simple	0.014	0.028	0.102	0.026	0.000	0.090	0.260
Medium	0.027	0.045	0.089	0.044	0.015	0.076	0.296
Complex	0.060	0.045	0.168	0.059	0.001	0.111	0.444
Total	0.101	0.118	0.359	0.129	0.016	0.277	1.000

In order to understand the overall effects and sensitivities it is necessary to look at a 'community model' that integrates the responses of the collection of problem types and complexities relevant to DART. Specifically, this community model computes the weighted sum of the model results using multiple data sets, where the weights are estimates of the fraction of the workload that problem categories represent at Sandia. It is essentially a linear combination of the individual model results, where each weighting coefficient is an estimate of the fraction of overall work that

each data set (according to problem type and complexity) represents within DART's problem space. So, for our analysis where there are 34 data sets spanning 18 different problem types and complexities, the individual model is run 34 times, and a combined result is determined by weighting all the outputs such that the 34 weights sum to 1.0, and any redundant (i.e., multiple samples for a given problem type/complexity) data sets are further normalized within their category so as to not overweight that category (e.g., if the weight for category thermal/simple was 0.3 and there were three data set samples for that category, then each of those samples would be weighted as 0.1).

To that end, we developed a computer program that calculates a weighted solution for any subset of the data samples collected, given a user-supplied weighting. This code and the baseline weighting are available in Web Fileshare (Model ID: WFS253067). The code contains documentation explaining how to run subsets of the problem space, for those interested in more detailed analysis using the model we developed. Further, this community model code can be easily extended to include new data sets, or to accommodate studies of different weightings, etc.

On January 12, 2005, the DART Oversight Committee, which represents the analyst community, reviewed the system analysis data and concluded that the observed input data variability indeed represents the varied uses of the DTA process and that the individual output values reasonably represent typical DTA process use. Further, they validated the problem type and complexity distribution information collected from the analysis department managers. With that validation, the system analysis team proceeded with the community model and developed results presented in Tables 7, 8, and 9.

Table 7. Community output engineering time per problem type (engineering time in hours).

		Modal	Linear Struct.	Non-Lin Struct.	Heat Trans.	Fluid Flow	Rad. Trans.	Comm. Model
A	Design Solid Model Creation and/or Edit	8.8	0.0	109.0	84.0	0.0	63.9	66.8
B	Analysis Solid Model Creation and/or Edit	1865.8	82.4	193.2	81.1	124.8	21.0	380.1
C	Geometry Decomposition	1190.9	106.2	821.7	45.8	8.2	11.3	581.8
D	Meshing	626.4	35.8	322.0	41.8	33.9	106.8	249.9
E	Mesh Manipulation	389.8	16.0	105.8	19.0	4.1	33.2	105.5
F	Model Parameter Assignment	325.1	11.9	116.7	25.7	10.8	23.0	106.2
G	Simulation Model Assembly	513.9	7.3	104.7	93.4	4.3	9.1	135.7
H	Run Simulation	105.5	27.3	108.0	16.7	8.6	113.3	68.8
I	Results Post-Processing	27.2	28.2	72.8	245.3	56.4	31.9	88.2
J	Archiving of Analysis Artifacts	5.5	1.6	29.5	6.4	2.0	19.0	15.2
	TOTAL	5058.8	316.7	1983.5	659.2	253.0	432.3	1798.1

Table 8. Community output engineering time per problem type (normalized)

		Modal	Linear Struct.	Non-Lin Struct.	Heat Trans.	Fluid Flow	Rad. Trans.	Comm. Model
A	Design Solid Model Creation and/or Edit	0.00	0.00	0.05	0.13	0.00	0.15	0.04
B	Analysis Solid Model Creation and/or Edit	0.37	0.26	0.10	0.12	0.49	0.05	0.21
C	Geometry Decomposition	0.24	0.34	0.41	0.07	0.03	0.03	0.32
D	Meshing	0.12	0.11	0.16	0.06	0.13	0.25	0.14
E	Mesh Manipulation	0.08	0.05	0.05	0.03	0.02	0.08	0.06
F	Model Parameter Assignment	0.06	0.04	0.06	0.04	0.04	0.05	0.06
G	Simulation Model Assembly	0.10	0.02	0.05	0.14	0.02	0.02	0.08
H	Run Simulation	0.02	0.09	0.05	0.03	0.03	0.26	0.04
I	Results Post-Processing	0.01	0.09	0.04	0.37	0.22	0.07	0.05
J	Archiving of Analysis Artifacts	0.00	0.01	0.01	0.01	0.01	0.04	0.01

Table 9. Community output expected number of visits per problem type

		Modal	Linear Struct.	Non-Lin Struct.	Heat Trans.	Fluid Flow	Rad. Trans.	Comm. Model
A	Design Solid Model Creation and/or Edit	2.89	0.79	13.45	2.01	1.00	2.63	7.40
B	Analysis Solid Model Creation and/or Edit	5.34	2.22	21.21	2.12	1.20	4.30	11.78
C	Geometry Decomposition	10.63	2.79	20.94	1.49	1.20	4.35	12.47
D	Meshing	12.46	2.52	19.58	2.36	1.20	3.78	12.17
E	Mesh Manipulation	10.91	2.15	11.86	1.55	1.20	4.48	7.99
F	Model Parameter Assignment	9.79	2.69	7.33	3.63	1.70	4.48	6.10
G	Simulation Model Assembly	10.86	2.01	9.59	2.38	1.70	3.45	6.94
H	Run Simulation	6.47	2.01	9.34	3.99	1.70	3.45	6.53
I	Results Post-Processing	4.32	2.01	8.81	3.45	1.70	1.62	5.94
J	Archiving of Analysis Artifacts	1.12	0.79	3.83	1.00	1.00	1.00	2.37

Before discussing observations and conclusions, it is necessary to point out that considerable uncertainty in the results exists. Data, especially in certain combinations of problem type and complexity, is limited and the varied nature of the problems encountered and processes employed by analysts is significant. In addition, no strict definition was provided for the three complexity categories and analysts demonstrated varying opinions about what constitutes simple, medium, and complex analyses, especially for different problem types. Table 7 shows an apparent inconsistency in the modal problem type results as compared to other problem types. In this case the large engineering time values were driven by two anomalous data sets, one where the analyst created the entire (complex) solid model on his own and the other which used a larger than typical set of transition probabilities. While an extension of this work might reevaluate data collection effectiveness and might strive to collect additional input data, this study is limited by the process and data previously described. In spite of these shortcomings, the results provide useful insight if those reviewing them keep in mind the methods used to collect and analyze the raw data.

From these tables one immediately recognizes that for structural problems geometric manipulation (steps B and C) and meshing represents a large fraction of overall process time. For non-linear problems, where the absolute overall time is greatest, the fraction of work devoted to geometry manipulation is 53%. For non-structural problems, geometry decomposition is less significant while geometry creation and meshing remain important. While the uncertainty associated with these results may be significant, the conclusion that geometry manipulation and meshing consumes a significant portion of the average analyst's time is clear. The notable problem type exceptions are heat transfer and radiation transport. In the case of heat transfer, analysts typically are able to use tetrahedral elements which automesh much easier than the hexahedral elements typically used for structural analyses. Thermal analysts therefore spend less time on geometry and meshing. The radiation transport problems do not require extensive geometry manipulation once a basic model is created.

Table 10 shows the results of a number of runs of the community model in terms of time sink breakdown (percent scale) for all problem categories considered in this study. It attempts to provide some insight about weighting sensitivity. The columns labeled according to type of problem (e.g., linear structural) are the values computed by the community model for the case where the weight for that type of problem is increased by 20% over baseline (all complexities increase 20%). For each of these columns, the other weights were decreased to normalize the overall weights to 1.0. While this procedure does not determine the actual sensitivity, it does give an indication of how sensitive the results are to the weights. From these 120% scenario results, one can readily see that the fundamental spread of time sinks for the DTA process model is not particularly sensitive to the problem category weights, at least not at the 20% deviation level. While further analysis and sensitivity studies of the model parameters (vs. weighting) are required to understand the overall model behavior more fully, this result does provide some confidence in the results and conclusions drawn from this weighting.

Table 10. Community model results showing each category weight increased by 20%

Case: 120%		Modal	Linear Struct.	Non-Lin. Struct	Heat Transfer	Fluid Flow	Rad. Trans.	Baseline Weights
A	Design Solid Model Creation and/or Edit	3.48	3.72	3.85	3.84	3.70	3.73	3.70
B	Analysis Solid Model Creation and/or Edit	22.19	21.07	20.15	20.95	21.19	21.09	21.20
C	Geometry Decomposition	31.74	32.39	33.34	32.10	32.27	32.34	32.32
D	Meshing	13.80	13.89	14.06	13.82	13.92	13.91	13.89
E	Mesh Manipulation	5.99	5.85	5.79	5.82	5.86	5.87	5.87
F	Model Parameter Assignment	5.94	5.90	5.91	5.88	5.90	5.90	5.91
G	Simulation Model Assembly	7.73	7.52	7.33	7.60	7.53	7.54	7.55
H	Run Simulation	3.72	3.84	3.90	3.82	3.84	3.86	3.82
I	Results Post-Processing	4.62	4.97	4.79	5.31	4.94	4.91	4.90
J	Archiving of Analysis Artifacts	0.80	0.85	0.89	0.85	0.85	0.85	0.85

Community Model Parameter Sensitivities

Understanding the effect on community model process time of changes in individual model parameters provides additional insight for determining where to best allocate DART resources. To calculate sensitivities for the initial step times and rework fractions, each parameter is perturbed by a small amount for every input data set and the community model is run, using the baseline weightings. Table 11 presents the resulting sensitivities for relative perturbations.

Table 11. Initial step time and rework fraction sensitivity estimates as calculated using the community model with baseline weights.

		Initial Step Time	Rework Fraction
		output time change (%) / input time change (%)	output time change (%) / input fraction change (%)
A	DSM Creation/Edit	0.037	0.031
B	ASM Creation/Edit	0.212	0.096
C	Geometry Decomposition	0.324	0.254
D	Meshing	0.139	0.097
E	Mesh Manipulation	0.059	0.031
F	Assign Model Parameters	0.059	0.034
G	Assemble Simulation Model	0.076	0.048
H	Run Simulation	0.038	0.027
I	Post-process results	0.049	0.026
J	Archive Artifacts	0.009	0003

Since the initial step time is a direct factor in the subsequent visit time calculations, the sensitivity values in the initial step time column are essentially identical to the community model values in Table 10. The initial step time and rework fraction sensitivities reinforce what was learned from the community model output, that geometry manipulation and meshing represent opportunities for significant improvement. Rework fraction sensitivities, which are from 40-80% of the initial step time sensitivities, offer additional significant opportunity for improvement.

While community sensitivity calculations for the initial step times and rework fractions are straight-forward, similar calculations for the transition probabilities are more problematic. Because the model is highly non-linear (a validated premise), a single individual input parameter set cannot be used to represent the community model outputs. The preferred procedure to develop transition probability sensitivities would be to alter the same parameter in each of the 33 input sets, run the community model, calculate the weighted average of the results, and evaluate the

change from baseline values. Complicating this finite difference problem, however, is the fact that the probabilities in any row of the transition probability matrix must sum to one and increasing a single probability forces a reduction in one or more other probabilities.

In spite of the above difficulties, one may gain qualitative sensitivity insight with a scheme that focuses on improving the probability of exiting a process step without entering a backward iteration loop. The scheme begins by selecting a row in the transition probability matrix[13] (which corresponds to all possible exits from a given process step) and perturbing all forward exiting probabilities by a given relative value. The entire row is then normalized so that its entries sum to unity. Again, we recognize that this process does not yield a mathematically rigorous evaluation, but expect it to provide qualitative insight.

Figure 2 shows the results of the qualitative transition probability evaluation. In this case, both the initial and subsequent probabilities were perturbed as described above. The relative perturbation value (pre-normalization) is plotted on the x-axis while the resulting total process time as calculated by the community model using the baseline weighting is plotted on the y-axis. In addition to the individual probability sensitivities associated with specific process steps, the plot also includes a curve representing the sensitivity associated with the collective process step transition probabilities.

While the large probability fraction increases indicated on the x-axis may seem surprising, one must recall that the selected row in the probability transition matrix is subsequently normalized. Thus, as the pre-normalized forward exit probability value becomes large, the sum of all forward probabilities approaches unity while all backward exit probabilities approach zero, which eliminates all backward iteration loops. With that stated, one should recognize that some individual data sets contain transition probability rows with no forward exit probability and as a result the ideal limit of no backward iteration loops is not realized. Nevertheless, the effect of these anomalous data sets is not large and does not change the main conclusions drawn from the sensitivity analysis.

[13] See Figure 5 for an example of a transition probability matrix.

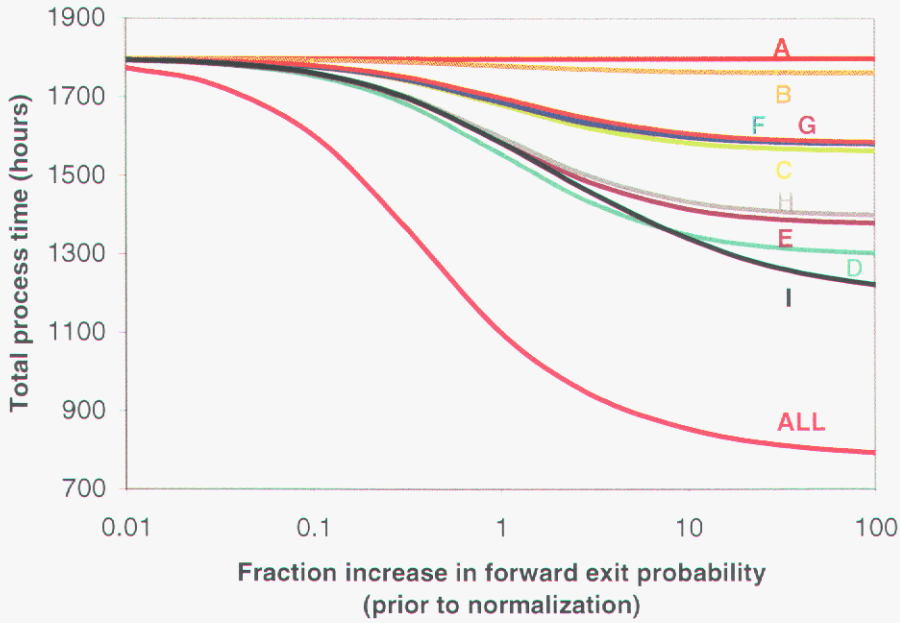


Figure 2. Qualitative transition probability evaluation results.

One can draw several conclusions by inspection of Figure 2. First, improving the forward exit probability from the geometry and meshing steps promises significant overall process improvement. The large number of visits to these steps clearly calls for attention. Understanding and compensating for this effect, one also notices that later process steps generally have a greater impact than earlier process steps. The reason for this effect becomes clear when one recognizes that backward iteration loops originating from later process steps force the repeat of earlier iteration loops, similar to the programmer's nested control loop effect. Thus, reducing the need for late process iterations offers a significant opportunity for process improvement.

One should note that a backward iteration loop originating from a given process step does not imply that a problem was created within that step, only that a problem was identified there. There appears to be significant benefit to not only preventing problems from occurring late in the overall process, but also in identifying problems as early in the process as possible. To this end, tools that help an analyst identify potential problems before much subsequent effort is expended promise to be of value. With such potential in mind, a follow-on study effort should identify the primary causes of problems resulting in backward iterations throughout the analyst community, and suggest ways to minimize these iterations.

In contrast to the finite difference implementation, Boggs developed an analytical solution for the derivative of the overall process time with respect to the transition probabilities. Because evaluation of the derivatives requires identification of a specific single input parameter set, the method cannot be readily used in a community model sense where no single representative input parameter set exists. Nevertheless, the derivatives calculated for the analyst supplied data sets are informative and corroborated the conclusions drawn from the finite difference method.

One might attempt to qualitatively compare the relative impact of step times and transition probabilities in order to determine where the greatest opportunity for process improvement lies. Halving the initial step times associated with all of the process steps results in a 50% reduction in overall process time. Figure 2 indicates that doubling the pre-normalized forward exit probabilities for all process steps reduces the overall process time by 40%. Accounting for the normalization, the actual value may be closer to 50%. In any case, the relative benefit of focusing effort on initial step time reduction appears to be similar to that for focusing on reducing process iteration loops.

Reducing the rework fraction, the portion of original work that must be repeated during subsequent step visits, also offers significant opportunity to reduce overall process time. Table 15 in the body of this report shows that the impact of reducing rework fraction is approximately 40% to 80% of that for reducing the "once-through" time or the iteration probability. Tools that preserve information generated during previous process steps visits and make it readily available during subsequent visits offer the promise of reducing rework fractions. With comparable sensitivities, all three major process parameter sets ("once-through" times, rework fractions, and iteration probabilities) should be considered important opportunities for overall process improvement.

Summary and Recommendations

Summary

The DART system analysis team studied the design-through-analysis processes employed by analysts in Centers 9100 and 8700 at Sandia to identify opportunities for reducing overall process time. The study created and implemented a rigorous analysis methodology based on a generic process flow model parameterized by information obtained from analysts. The analysts provided data during collection sessions where study team members helped them use a computer-based data collection tool developed to maximize the validity of the data. These input data indicate that the processes used by Sandia analysts vary significantly, as do the type and complexity of the problems themselves.

The study team also collected data from analysis department managers to quantify the problem type and complexity distribution throughout the analyst community. A *community model* was developed from these data, which enabled a simple characterization of processes that span the community. The community model yielded information about how an analyst's time is distributed throughout the generic process steps. It also provided insight about sensitivities associated with model parameters such as the step times, rework fraction, and the transition (iteration) probabilities.

The analysis indicates that equal opportunity for reducing overall process time for the community is available both by reducing the "once-through" time required to complete a process step and by reducing the probability of backward iteration. Moreover, reducing rework fractions offers additional benefit, about 40–80% of the impact of reducing the process initial step time or backwards iterations. Geometry manipulation and meshing consume a large portion of an analyst's time, especially for structural problems, and offer significant opportunity for process time reduction. Iteration loops initiated late in the process are more costly than others because they cause "inner loop" iterations to be repeated. Identifying and correcting problems as early as possible in the process offers significant opportunity for time savings.

Recommendations

This study identifies several general investment areas that provide opportunity for the largest improvement in process time. These include geometry and meshing time reduction, rework minimization, and the identification and correction of problems early in the modeling process (elimination of problems that lead to subsequent iteration is the larger goal). The study does not, however, provide detailed information about the specific problems that lead to longer than desired individual step times, significant rework, or frequent iteration. As follow-on to this study, we recommend that data about these problems be collected and analyzed. For the iteration problem, the follow-on should determine what problems cause costly iteration loops, where those problems are created in the process and typically where they are discovered, and should provide suggestions for improvements. While this study indicates that we *should* invest in geometry and meshing as well as model verification, the additional data will shed light on specifically *how* to invest within those areas. Further, we need to better understand how to reduce rework time, since this study indicates that this is a significant factor for improving process efficiency.

We recommend that DART leadership immediately begin investing in the important process areas identified in this study, refining those investments as additional analysis provides more information. We recommend identifying program elements that reduce rework time and increasing investment in those areas according to the high return suggested by this study. To free funding for the high-impact work, the project will be forced to reduce funding in other areas. When selecting the lower impact areas for divestment, DART leadership should realize that while some steps may offer less overall improvement impact, they remain wholly necessary for the complete process to work. DART leadership should also understand the limited scope of this study and draw conclusions appropriately. For example, while the *Run Simulation* step represents only 4% of overall process time, we did not evaluate the effect of parameterized runs (i.e., those run after the first simulation with only a small amount of additional analyst time), primarily because we were concerned about the fidelity of our parameterized run data. Nor did we evaluate the impact of Run Simulation efforts (e.g., automated work flows in ASETS) on reducing rework times.

An economic extension of the information provided in this report would provide a useful means of comparing existing or proposed work packages. We recommend that the principal investigators of all DART (and perhaps DART-related) projects estimate the impact of their work (in approximately $250K elements) in terms of reducing individual process step times and rework

fractions as well as in lowering the probability of specific iteration loops. Using the model to estimate process time reduction, DART leadership can develop an economic metric (dollars per unit of relative process improvement) and routinely use it as an investment guide.

We also recommend that any DART strategy developed with the assistance of this study be reviewed by a group independent from but knowledgeable of the DART project, to provide an independent evaluation before resources are allocated.

Finally, the study suggests that certain research efforts outside the scope of DART, once they are practically implemented, can result in large process improvements. Such efforts include the development of a structural tetrahedral element that produces accurate analysis results while automeshing well and automatic meshing techniques for hexahedral elements.

This page intentionally left blank.

Appendix 1: The Data Collection Tool

The data collection tool is a web-based application found at http://sid-srn/Dart/DartSystemAnalysis/dart.html. The tool is structured in two parts, a Java applet that presents a graphical user interface (GUI) and a Java servlet that performs the necessary model calculations. To use this tool, the local computer requires a Java-compatible browser as well as an installed Java runtime system. *Internet Explorer* and *Mozilla Firefox* are examples of Java-compatible browsers. The Java runtime system is available as a free download from Sun Microsystems and can be found at http://java.sun.com.

As shown in Figure 3, the GUI contains six tabbed sheets and a row of buttons. Each of the six sheets presents opportunities for the analysts to input data describing their understanding of a class of analysis problems. The buttons provide a mechanism for the analyst to initiate actions upon that data. The following subsections describe each tabbed sheet and the buttons.

Metadata Sheet

The *metadata* sheet, shown in Figure 3, enables entry of information to categorize the data set. Entries include the name of the analyst entering the data set, the date (automatically generated), the process flow (serial or parallel as described in the *Generic DTA Process* section), the analyst's perception of problem complexity and the typical number of elements and parts for the selected complexity, the name of the FEA solver, and a number of parametric runs. The last item, number of parametric runs, deserves additional comment. The intent of including this item is to account for analysts who spend significant effort to set up an initial simulation run, but then are able to quickly run subsequent simulations where only minimal engineering time is required. For example, an analyst may wish to perform a sensitivity analysis where subsequent runs systematically alter a few model parameters. The process model ignores engineering time associated with these additional runs, but does account for the additional compute time.

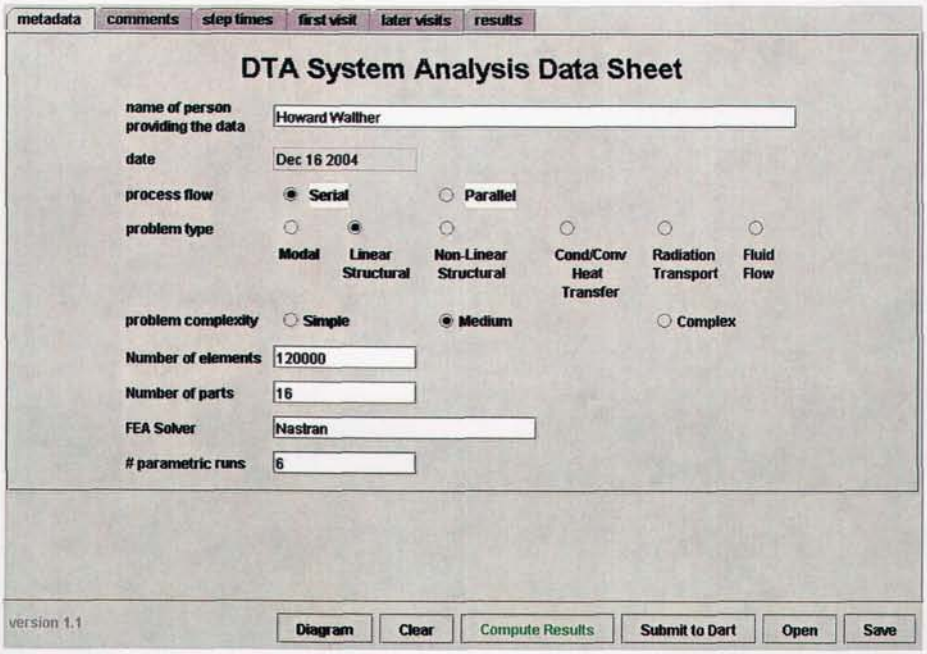


Figure 3. Metadata Input Sheet

Comments Sheet

The *comments* sheet, not shown, is simply a single area in which an analyst can enter text. There are often many conditions associated with entered data and this sheet provides a means of recording these subtleties. Analysts are encouraged to include as many comments as possible so that the system analysis team can best understand the intent of their data.

Step Times Sheet

The *step times* sheet, shown in Figure 4, enables an analyst to enter information about the time required to progress through each of the generic process steps, both the first time the step is encountered and for subsequent visits. By distinguishing between the first visit and subsequent visits, the tool accounts for potential learning or reduced work requirements during process iteration. For example, an analyst who spends five days initially decomposing geometry may require only one day the next time the step is encountered in an iteration loop, either because the analyst is more familiar with the task or because less work is required. Note that the tool does not distinguish between the second and subsequent visits to a step.

metadata | comments | step times | first visit | later visits | results

Step Times

	Process Step	Engineering Time	Compute Time	Iteration Rework	Tools
		days	days	fraction(0-1)	
A	Design Solid Model Creation and/or Edit	0			Proe
B	Analysis Solid Model Creation and/or Edit	10		0.5	Patran
C	Geometry Decomposition	5		0.5	Patran
D	Meshing	5		0.5	Patran
E	Mesh Manipulation	1		0.25	Patran
F	Model Parameter Assignment	1		0.5	Patran
G	Simulation Model Assembly	0.5		1.0	Patran
H	Run Simulation	0.5	0.1	0.5	Nastran
I	Results Post-Processing	1			Patran
J	Archiving of Analysis Artifacts	0			

version 1.1 | Diagram | Clear | Compute Results | Submit to Dart | Open | Save

Figure 4. Step Times Input Sheet

The first data entry column in the *step times* sheet is the e*ngineering time* associated with the first visit to each generic process step. The analyst can select the desired time units in the drop-down box at the top of the column. The second data entry column is for *compute time*, where all entry areas except for *run simulation* are non-editable. The intent of this column is to separately capture the compute time associated with the data set, as distinguished from the analyst's engineering time.

In order to capture the time required for subsequent visits to each step, the tool does not request an absolute time value. Rather, it requests a rework fraction, specified as a decimal between zero and one. For the example of the analyst whose first visit to *geometry decomposition* required five days while the second required only one, the corresponding rework column cell would contain 0.2.

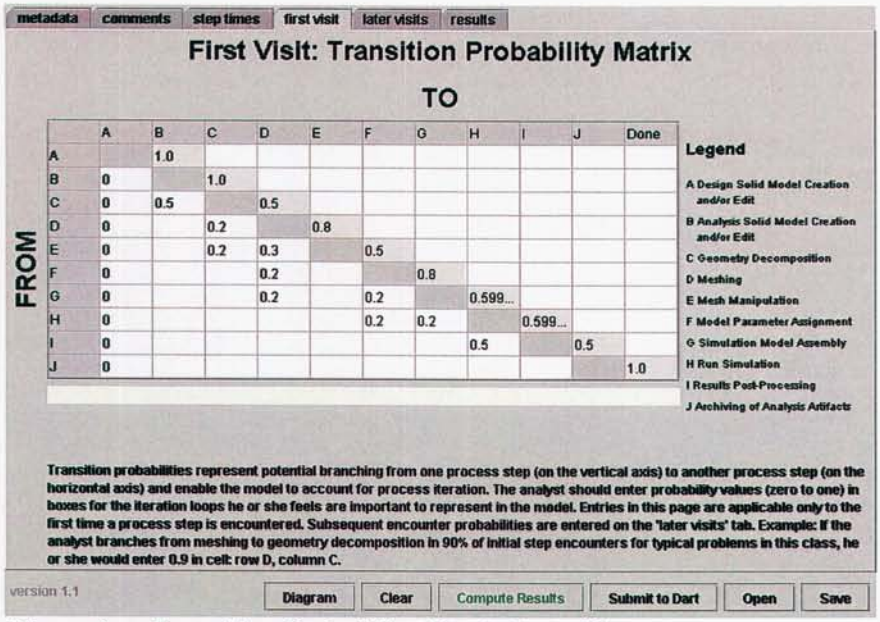
Note that the tool does not distinguish between the second and subsequent visits to a step. The system analysis team concluded that the additional information gained by distinguishing between these visits was not worth the added tool complexity. When determining the appropriate rework value for any step, the analyst should consider all expected subsequent visits and average the rework times associated with each.

First Visit and Later Visits Sheets

The next two sheets, *first visit* and *later visits*, offer the opportunity to enter data about iteration loops or other process flow transitions an analyst feels are important to represent. Flow transitions are jumps from one step in the generic process to any other step and can be represented in matrix form as shown in the *first visit* sheet, Figure 5, where the matrix is titled *First Visit: Transition Probability Matrix.* In this representation, the vertical axis (left) shows the "from" steps while the horizontal (top) axis shows the "to" steps.

The tool requires an analyst to enter a probability value, a decimal from zero to one, for the important flow branches. For example, if an analyst expects for the chosen class of problem to branch from *Run Simulation (H)* to *Meshing (D)* on 70 percent of first visits, then the *from H to D* cell in the *first visit* sheet would contain 0.7. The *later visits* sheet would contain a corresponding value for the second and subsequent visits. In general, one might expect the matrices to be sparsely populated and that corresponding cells in each matrix will contain non-zero values. As with *step times* data entry, transition probabilities for all subsequent visits to a step should be averaged and entered in the *later visits* sheet.

Iterations within a single step are not explicitly represented; rather, they are accounted for solely in the *step times* sheet. As a result, the matrix diagonal requires no input and is grayed out. For example, if in the process being modeled the analyst spends two days meshing and immediately discovers problems that require one day of remeshing, this time would be recorded as three days of step time with no iteration. If, however, the analyst progressed to the *mesh manipulation* step and then returned to *meshing*, a non-zero value would be entered in the "from E to D" cell in the Transition Probability Matrix. Further, the cell to the immediate right of the diagonal is the probability of progressing normally from one generic process step to the next. Since the sum of the probabilities in any row must be equal to one, the values in these "normal exit" cells are automatically updated to meet this requirement.



metadata | comments | step times | first visit | later visits | results

First Visit: Transition Probability Matrix

TO

FROM	A	B	C	D	E	F	G	H	I	J	Done
A		1.0									
B	0		1.0								
C	0	0.5		0.5							
D	0		0.2		0.8						
E	0		0.2	0.3		0.5					
F	0			0.2			0.8				
G	0			0.2		0.2		0.599...			
H	0					0.2	0.2		0.599...		
I	0							0.5		0.5	
J	0										1.0

Legend

A Design Solid Model Creation and/or Edit
B Analysis Solid Model Creation and/or Edit
C Geometry Decomposition
D Meshing
E Mesh Manipulation
F Model Parameter Assignment
G Simulation Model Assembly
H Run Simulation
I Results Post-Processing
J Archiving of Analysis Artifacts

Transition probabilities represent potential branching from one process step (on the vertical axis) to another process step (on the horizontal axis) and enable the model to account for process iteration. The analyst should enter probability values (zero to one) in boxes for the iteration loops he or she feels are important to represent in the model. Entries in this page are applicable only to the first time a process step is encountered. Subsequent encounter probabilities are entered on the 'later visits' tab. Example: If the analyst branches from meshing to geometry decomposition in 90% of initial step encounters for typical problems in this class, he or she would enter 0.9 in cell: row D, column C.

version 1.1 | Diagram | Clear | Compute Results | Submit to Dart | Open | Save

Figure 5. Transition Probability Matrix Input Sheet

Results Sheet

The *results* sheet contains the process statistics calculated from the input on the previous tabs and is populated after the analyst clicks the *compute results* button. As shown in Figure 6, the *results* tab displays four columns of values. The first and second columns show the expected number of visits to each generic process step while the third column displays the engineering time associated with those visits. Finally, the fourth column, titled *wall clock time*, shows the sum of engineering and compute time. Since the only step with a compute time is *run simulation*, it is the only one with a value different than the adjacent cell in column three.

As previously described, the analyst entering data is expected to review these results and iterate with appropriate input values until he or she is comfortable that both the input and output represent the process as they understand it. Since it is expected that input data will be modified after an analyst has computed results, the tool will at times be in an inconsistent state, with the results not representative of the current input. At such times the *results* tab will include a message displayed at the top of the sheet in a large red font reading "*****RESULTS ARE STALE*****." Once the analyst computes new results, this message is removed. Additionally, when the tool is in an inconsistent state, the *Compute Results* button text is displayed in green.

metadata | comments | step times | first visit | later visits | results

*****RESULTS ARE STALE*****

	Process Step	# initial visits	# revists	Eng. Time (days)	Wall Clock Time (days)
A	Design Solid Model Creation and/or Edit	1.00	0.00	0.00	0.00
B	Analysis Solid Model Creation and/or Edit	1.00	0.50	12.50	12.50
C	Geometry Decomposition	1.00	0.90	7.25	7.25
D	Meshing	1.00	1.10	7.75	7.75
E	Mesh Manipulation	1.00	0.90	1.22	1.22
F	Model Parameter Assignment	1.00	0.80	1.40	1.40
G	Simulation Model Assembly	1.00	0.80	0.90	0.90
H	Run Simulation	1.00	0.90	0.72	1.52
I	Results Post-Processing	1.00	0.50	1.00	1.00
J	Archiving of Analysis Artifacts	1.00	0.00	0.00	0.00
	Total Time			32.75	33.54

version 1.1 | Diagram | Clear | Compute Results | Submit to Dart | Open | Save

Figure 6. Results Sheet

Button Bar

The button bar, shown in Figure 7 and located at the bottom of the tool GUI, displays six buttons an analyst can use to perform specific actions as follows:

Diagram | Clear | Compute Results | Submit to Dart | Open | Save

Figure 7. Button Bar

Diagram

The *Diagram* button opens a new browser window and displays a graphical representation of the selected process flow diagram. The analyst may wish to reference this diagram when considering step times and transition probabilities. The browser's *print* button will print the diagram.

Clear

The *Clear* button clears all data entries from the currently displayed tab sheet. It does not clear entries from other sheets.

Compute Results

The *Compute Results* button sends the input data from the first five sheets to the servlet, which performs the necessary calculations. The user receives a message stating either that the calculations were successful or that a problem was encountered. If notified of a problem, the user should review the input data to identify any obvious problems, negative probabilities for example. The button text is displayed in green while the results are stale.

Submit to DART

When comfortable that both the input values and results are as intended, the analyst can enter the data set into the DART database by clicking the *Submit to DART* button. After clicking this button the user is presented with a confirmation dialog, where clicking the OK button submits the data.

Save and Open

An analyst may save a data set to the local computer for later use. Clicking the *Save* button opens a file browser dialog and allows the analyst to select a save location and filename. The *Open* button presents a file browser dialog through which the analyst can locate and open a previously saved data set.

This page intentionally left blank.

Appendix 2: The Analysis Algorithm

The algorithm to compute the process statistics derives from the analysis of a so-called "Markov process." Basically, a Markov process consists of a set of states with the property that each state has a fixed set of transition probabilities to other states. Here, we use a special case of a Markov process known as a "Markov process with an absorbing state." An absorbing state is any state that only has a probability of one of returning to itself, i.e., once it is entered, it cannot be exited. In our case, the absorbing state is the artificially added "Done"' state seen as the last column in Figure 1.

A Markov process can be described by a transition probability matrix similar to what has been previously discussed. In particular, the rows represent the transitions from a state and the columns represent the transition to a state. Thus, if we denote the transition matrix by T, the entry $T_{i,j}$ is the transition probability from state i to state j. For a Markov process with one absorbing state, the matrix T can be partitioned as

$$T = \begin{bmatrix} S & R \\ 0 & 1 \end{bmatrix} \tag{1.1}$$

where S is the matrix of transitions among the nonabsorbing states, R is the matrix of transitions from the nonabsorbing states to the absorbing state, and the one in the lower right indicates that this state can only transition to itself.

Using the classical analysis of Markov processes, one can compute the expected number of visits to each state. In particular, define the matrix

$$F = (I - S)^{\mathsf{T}}$$

where T represents the transpose operation. Then the (i,j) element of F^{-1} gives the expected number of visits to state j given that the process was begun in state i. In our case, we always start in the first state, so we only need the first row of F, i.e., we need to compute

$$F^{-1} e_1$$

where e_1 is the vector with a 1 in the first position and zeros elsewhere.

Although this result provides a fast way to compute the values that we need, we don't yet have a Markov process. Recall that in our previous description, the transition probabilities are not fixed, but change after the first pass through a state. Thus we need a way to create a valid Markov process before we can use the above formulas.

Our approach is to replicate the states as necessary to keep track of how many times each state was entered. A state in the expanded Markov process is, therefore, a combination of the state, whether it is a first visit or a later visit, and a count of the number of time each state has been visited. The count here needs only to be a 0 or 1 to indicate if a transition to that state will be for a first visit or for a later visit. To generate these states, we do the following. Create the first state, which is state 1 in first visit with all other states unvisited (i.e., with counts of 0) and put this state in a list. Then perform the following loop.

Loop: For each state in the list do:

Change the count for this state to 1;

Loop: For each nonzero in the transition table for this state and this pass do:

If the count for the corresponding state, say j, is 0, create a new state j with first visit and the current counts.

If the count for the corresponding state, say j, is 1, create a new state j with later visit and the current counts.

Add this state to the list if it is not already there.

Upon completion of this loop, we will have a list of all of the expanded states that could possibly arise in the given problem. From this list, it is now relatively easy to generate the corresponding transition matrix T. We can then form F and do the computations noted above. Specifically, when the "Compute Results" button is pressed, the code reads the data that were entered into "First Visit: Transition Probability Matrix" and the "Later Visits: Transition Probability Matrix" and creates the matrix S from (1.1) from the above procedure. Given S, the matrix F is computed and factored. Finally the factorization is used to compute $F^{-1}e_1$, the expected number of visits to each state in our enlarged process. These results are then post-processed to give the expected number of visits to each of the original ten states in each pass. From this information, the expected time in each state and the total expected time for the entire process are computed.

Another strategy for generating an expanded problem was used in earlier versions of the code, but that strategy did not allow any states to be skipped It also was not easily generalized to the case of more than two passes. That is, it is not easy to see how we could allow a case in which we had transitions for the first visit, different transitions for the second visit and different transitions for the third and subsequent visits. The above approach easily handles all of these cases. In fact, the underlying algorithm as coded is completely general; it handles arbitrary transitions and any number of visits.

The total expected time could also be computed in terms of the expanded problem. To do this we let n be the number of states in our expanded problem and let t be an n-dimensional vector of the step times (obtained from the *Step Times* sheet) associated with each state in our expanded problem. Then the total expected time can be written as

$$E_{time} = (F^{-1}e_1)^{\mathrm{T}} t$$

This form will be useful in what follows.

One final topic is covered in this section. It is of interest in the overall analysis to understand the sensitivity of the final total expected time with respect to changes, or perturbations, in the transition probabilities. Such information can be used to help determine investment strategies by understanding what probabilities have the most leverage on the final result. In doing this, however, it does not make any sense to change just one of the transition probability entries, since the resulting row in the transition matrix would not be such that its elements summed to one. The allowable perturbations, therefore, are those such that the resulting matrix T is still a transition probability matrix, i.e., the row sums must all be 1, and, in our case, the absorbing structure of T must be maintained, which means that the last row cannot be changed. Formally, let P be a perturbation matrix with the same dimensions as T. Then P is a valid perturbation for our purposes if the row sums of P are all zero and the last row is zero. It can be shown that the derivative of E_{time} with respect to a perturbation P, is given by

$$E'_{time}(S;P) = -e_1^{\mathrm{T}}(F^{-\mathrm{T}}PF^{-1})t \qquad (1.2)$$

where the argument S implies that it is evaluated at the given transition probabilities. Note that since $F^{-1}e_1$ is already computed, the application of (1.2) can be done with just one more use of the factorization of F that is already available. Finally, the interpretation of P is that it is a

direction of change, which means that the sizes of the elements in P don't matter as long as the row sums are zero. To make meaningful comparisons between two perturbations, however, their sizes should be normalized.

Picking interesting perturbations *P* can be done by users of the tool. For example, suppose that the "First Visit" probabilities for the row corresponding to step *E* have the values

$$(0, .2, 0, .3, 0, .5, 0, 0, 0, 0)$$

implying that there is a 20% probability of going back to state *B*, a 30% probability of going back to state *D*, and 50% probability of going on to state *F*. Thus a possible perturbation would be a corresponding row of the form

$$(0, -.4, 0, -.6, 0, 1, 0, 0, 0, 0).$$

Here the largest perturbation element has been normalized to be one and the negative values have been allocated proportionately to the original values. This can be interpreted as asking the question, "What if we invest in process *E* so that an analyst is more likely to complete step *E* successfully the first time through?" Doing a similar calculation for each row would then allow the DART team to examine the trade-offs of investing in each process.

Other possible perturbation schemes could be worked out; the details of using this are still being considered.

Distribution

MS0139	Current, M. F.	(09905)	MS0834	Lash, J. S.	(09114)
MS0139	Hale, A. L.	(09900)	MS0836	Dobranich, D.	(09116)
MS0139	Thomas, R. K.	(09903)	MS0836	Griffith, R. O.	(09117)
MS0139	Yarrington, P.	(09902)	MS0836	Hertel, E. S.	(09116)
MS0140	Leland, R. W.	(09300)	MS0847	Redmond, J. M.	(09124)
MS0316	Dosanjh, S. S.	(09220)	MS0847	Wilson, P. J.	(09120)
MS0321	Camp, W. J.	(09200)	MS0893	Pott, J.	(09123)
MS0372	Jung, J.	(09127)	MS1113	May, R. A.	(09126)
MS0376	Adams, C. R.	(09125)	MS1310	Kempka, S. N.	(09113)
MS0376	Blacker, T. D.	(09226)	MS9042	Kistler, B. L.	(08774)
MS0382	Sjaardema, G. D.	(09143)	MS9042	Kwon, D. M.	(08774)
MS0382	Stewart, J. R.	(09143)	MS9042	Revelli, V. D.	(08774)
MS0384	Morgan, H. S.	(09140)	MS9151	Handrock, J. L.	(08900)
MS0384	Ratzel, A. C.	(09100)	MS9151	Oien C. T.	(08940)
MS0469	Corbett, D. W.	(02900)	MS9158	Sukalski, M. W.	(08961)
MS0555	Garrett, M. S.	(09122)	MS9159	Clay, R. L.	(08964)
MS0557	Baca, T. J.	(09125)	MS9159	Hardwick, M. F. (5)	(08960)
MS0822	White, D. R.	(09227)	MS9159	Hoffman, E. L.	(09511)
MS0825	Hassan, B.	(09115)	MS9405	Wilson, K. L.	(08700)
MS0825	Hermina, W. L.	(09110)	MS9409	Moen, C. D.	(08775)
MS0834	Johannes, J. E.	(09112)			

3	MS 9018	Central Technical Files, 8945-1
1	MS 0899	Technical Library, 9616
1	MS 9021	Classification Office, 8511 for Technical Library, MS 0899, 9616 DOE/OSTI via URL