

A Web Centric Architecture for Deploying Multi-Disciplinary Engineering Design Processes

Scott Woyak*, Hongman Kim*, and James Mullins
Phoenix Integration, Inc., Blacksburg, VA, 24060

and

Jaroslav Sobieszczanski-Sobieski†
NASA Langley Research Center, Hampton, VA, 23681-2199

There are continuous needs for engineering organizations to improve their design process. Current state of the art techniques use computational simulations to predict design performance, and optimize it through advanced design methods. These tools have been used mostly by individual engineers. This paper presents an architecture for achieving results at an organization level beyond individual level. The next set of gains in process improvement will come from improving the effective use of computers and software within a whole organization, not just for an individual. The architecture takes advantage of state of the art capabilities to produce a Web based system to carry engineering design into the future. To illustrate deployment of the architecture, a case study for implementing advanced multidisciplinary design optimization processes such as Bi-Level Integrated System Synthesis is discussed. Another example for rolling-out a design process for Design for Six Sigma is also described. Each example explains how an organization can effectively infuse engineering practice with new design methods and retain the knowledge over time.

I. Introduction

The need to improve the engineering design process is a never ending challenge. Whether the goals are to improve the quality of a design, or to reduce the amount of time required to do design, the desire to get better always exists. In the 70's and 80's computers moved into the design process as a required engineering tool. Initial applications involved modeling physical phenomena to better predict the performance of a design and to reduce the need for costly prototype testing. It is now common place to simulate a design before building it.

The existence of many engineering analysis tools led to a first generation of process improvement: the application of integration and optimization technologies. Integration tools automated multiple engineering codes so that multidisciplinary impacts could be better understood. Optimization tools used sophisticated algorithms to help speed the design space search process to reveal potentially unintuitive designs.

Today, there are now multiple integration and optimization solutions commercially available¹. For example, ModelCenter provides a flexible framework to create an integrated design model and perform design optimization studies². When applied correctly, these tools have each demonstrated proven results. While these tools continue to evolve and improve, the second generation of process improvement will come from other sources. *The challenges today are about improving organizational efficiencies beyond improvement of individual efficiencies.* To this end, the areas where an organization can benefit the most are:

- *Knowledge Retention:* After a study is performed, results are often stored on an engineer's computer, or worse, distributed among multiple engineers. There is a need to learn from past experience and use that knowledge to jump start new efforts.
- *Better Utilization of Computing Resources:* With the increasing ease at which integrated models can be created, and the increasing desire to perform thousands of design iterations for techniques such as Monte

* AIAA member

† Senior research scientist, AIAA fellow

Carlo analysis, the limiting factor is the time required to perform all the runs. This problem can be addressed through effective use of massively concurrent processing with the aid of load balancing tools.

- Standardization of Design Methods: Many organizations are attempting to introduce standard design methods such as Design for Six Sigma (DFSS), but getting these methods out to users is proving difficult. The ability to quickly implement new methods in an organization practice is proving to be a differentiator.

This paper describes a solution architecture being developed to deliver the key functionalities to achieve the three goals stated above. Section II describes elements of a Web centric architecture for engineering design environment. Section III presents a couple of case studies for deployment of advanced design methods using the solution. Section IV discusses proof of concept test for load balanced trade study support. Lastly, section V summarized conclusions.

II. Solution Architecture

Most of existing design environments are built around the individual user. They are designed around the assumption that a single user will build and perform an engineering trade study. The next generation of solutions will be designed from the ground up to support the needs of an organization by coordinating the efforts of many individuals. There are many Web technologies available today to accomplish these goals. Standard Web based solutions are composed of common technologies: Web servers, databases, application servers, and browsers. These technologies have reached a level of maturity and are pervasive outside of the engineering community.

The architecture for engineering design environment described in this paper builds upon these technologies. Due to complexity of engineering design tasks, advanced software technologies are required to create a successful Web based design environment. Figure 1 shows concept of *Web portal system for engineering projects*. First, various project data is stored in a Web-enabled central database management system (DBMS). Users who log on to the Web portal will be able to view the projects they are working on and modify them as necessary. The Web-based database technology allows instant access to the project data from any computers on the network through Web browsers. Also the most up-to-date data will be instantly available for use to synchronize analysis and design activities of team members. At the same time, secure access and user authentication are critical to keep proprietary data safe and maintain its integrity.

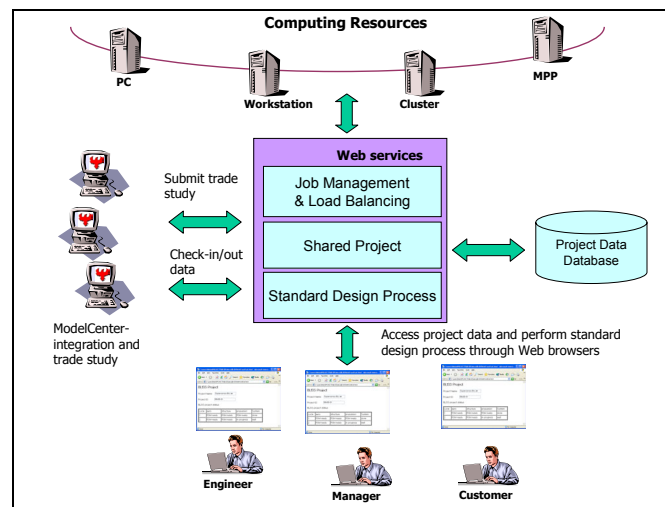


Figure 1: Web Centric architecture of the engineering design solutions.

Second, computational resources, engineering simulations, and trade studies can be managed through Web interfaces. For example, ModelCenter can submit trade studies to Web services, which will send off the required simulation runs to available computing resources. The status of the trade studies can be monitored remotely using Web browsers. The Web portal system will provide administration capability to organize available computing resources and assign them for use of selected project groups or users.

Third, the Web portal will provide an effective vehicle to deploy standard engineering design process to a wide range of people in the organization. For example, an advanced collaborative optimization technique can be presented as Web interfaces to perform standard design optimization as a daily routine. The Web portal service will need to

implement the business logic of the engineering process. The Web portal system lies at the core of this project as a glue to bring together project data, engineers, computing resources, and the engineering design process.

The Web centric architecture consists of several key functional modules described below.

A. Shared Project

Shared Project is a Web-enabled database to serve as a single entry point to access all design data related to a project, including computational models for simulation, key design parameters, and trade study results. To retain knowledge after a design study is completed, it is imperative that information be organized into a central location where it can be searched for future reference, and potentially reused. The primary reason many organizations fail in this goal is that archiving and storing data is an afterthought of normal day-to-day activities. Projects are backed up and stored after the project completes.

To address this problem, data storage and sharing must be an action that happens as part of the normal day-to-day activities. The process of performing work must result in the accumulation of a central repository of knowledge that can later be mined for information. Shared Project consists of a few libraries storing each type of project data: Analysis Library, Trade Study Library, and Variable Library.

1. Analysis Library

The Analysis Library is a central place where computational models for simulation are stored and version controlled. These components can be any of several things:

- Computational model of a disciplinary analysis, which may include input files to the disciplinary analysis program, definition of key design variables and scripts to run the analysis program.
- Computational model of multidisciplinary analysis, which involves many disciplinary analysis modules. For example, this can be a model integrated using the ModelCenter process integration environment.
- Approximation models that were created to replace actual computational simulations. Approximation Model Toolkit described below can be used to generate them.

Analysis Library contains a version controlled listing of computational models that can be run within an engineering workgroup. A key feature of the Analysis Library is that once an item is stored in the Analysis Library, it may be accessed in multiple ways:

- Directly from the Web
- From integration environments like ModelCenter
- From published specifications of functional interface of Analysis Library, or API (application programming interface)

Consider a typical design process for submarine design. Analysis tools exist for performance disciplines such as structures, propulsion, and power supply as well as other disciplines such as cost, operations, and reliability. Traditionally, a group of specialists operates each of these disciplines. At the end of a design study, the pieces that composed the project are left scattered across an organization. Additionally, the only people that can reproduce results are the discipline experts themselves. By storing computational models of single or multidisciplinary analysis in the Analysis Library, a central, coordinated system is created. The computational models used for a study can then be reused years after the study was completed.

2. Trade Study Library

When trade studies such as design of experiments (DOE) and optimization are completed using the models from Analysis Library, the trade study results (e.g., optimization design history or DOE run table) will be automatically stored in Trade Study Library. This, combined with the Analysis Library, results in a system where the key data associated with a project is stored in a central location where it is accessible using a Web browser.

Trade Study Library offers capability to upload and download trade study results from ModelCenter. Massive data generated from the architecture's parallel trade study support will be automatically stored into Trade Study Library. The trade study data is stored into the backend database. Due to data persistency of database, the trade study results are protected from potential system failure so that the completed portion of trade studies can be

restored. The trade study results can be examined using a table view or plots through Web browsers, allowing project participants can shared the results effectively.

3. *Variable Library*

Variable Library serves as a common data model of key design parameters of a complex system being designed. Consider a situation where geometry information needs to be shared between aerodynamics, structures, and systems group for a design of an aerospace system. It is important that all groups are using the same geometry data for their work and it can be difficult because the geometry may be changing as the design progresses. To solve this problem effectively, a common data model of the shared design parameters can be created and shared through the Variable Library. A hierarchical data model can be created for a complex system and the data will be under version control.

The Variable Library offers an effective means for multiple ModelCenter users to exchange design parameters. For example, a user first downloads current values of the design parameters from Variable Library and performs his analysis. Once it is done, he uploads key output parameters of his analysis to Variable Library. The second user now downloads the shared design parameters and the values the first user has uploaded and performs his part of analysis. In this collaboration scenario, Workflow tool explained later will coordinate task of each of the ModelCenter users.

B. Approximation Model Toolkit

An approximation model is a surrogate of higher fidelity analysis that is generally accurate within a given trust region of the design space. There are two main reasons to adopt approximation models into the design process:

- The desire to introduce high fidelity methods earlier in the design process.
- The need to perform many runs for studies such as Analysis of Alternatives or Design for Six Sigma.

While design cycle time can be reduced by distributing computer simulation cases to available computers, it is still usually not practical to perform thousands of runs on a high fidelity analysis. The solution is to perform a smaller number of runs of the high fidelity application and then build an approximation model acting as a surrogate for that application. Common techniques include polynomial response surface model and Kriging models. Once constructed, it becomes much more practical to perform brute force design techniques such as Monte Carlo methods using the fast running approximation model. An approximation model toolkit will be used to create polynomial or Kriging model based on simulation results generated using ModelCenter. The approximation models can be uploaded to Analysis Library so that they are accessible through Web interface.

Example applications include computational fluid dynamics (CFD) models where a single analysis run may require 24 hours to perform. Trying to perform a Monte Carlo study with a thousand runs would take years. By creating an approximation model with potentially as few as 16 runs, it is possible to produce similar results in a fraction of the time. The key practical advantage of this approach lies in the ability to use the high fidelity codes unchanged, only replicated over arrays of processors, to generate surrogate models.

C. Web-Based Trade Study Wizards

A standard design method can be presented as a step by step process guided through Web-Based Trade Study Wizards. Introducing programs like Design for Six Sigma requires deploying a standard process so that many people can easily follow it. Problems often arise due to the complexity of processes and the lack of support for coordinating the efforts of multiple people. The need to constantly tweak and evolve processes also results in version control problems. End users quickly become confused and revert to the old way of doing things. The solution to these problems is two-fold:

- *Deploy design processes using a step-by-step (i.e. Wizard-based) approach that guides users through the process.* For this purpose, a Wizard must combine on-line help with a step-by-step recipe for implementing a particular method. This is crucial to assuring that advanced methodologies are performed correctly and that new users can educate themselves along the way. Note that for experienced users, there must also be a short-cut approach that allows data to be entered quickly without unnecessarily burdening the user with tutorial steps.
- *Host the Wizards on a central Web server and deploy the methods through a Web browser.* Trying to deploy evolving software throughout an engineering organization can quickly lead to problems. Multiple versions of software emerge and installation problems tend to overshadow program benefits. For these reasons, it is better to host processes on a server where one installation can be used by an entire organization.

The architecture must be extensible so that organizations can implement and deploy their own Wizards to the system. Each organization has its own needs and no single approach will meet all needs. Thus, it is critical that the services provided by the architecture are implemented in an open fashion that permits custom Wizards to be created.

The recent trend towards implementing programs such as Design For Six Sigma illustrates the need for Web-based trade study wizards. The DFSS process is an involved, multi-step process that requires multiple people to coordinate efforts and produce results in common formats. An example deployment of a DFSS process is described later in this paper.

D. Workflow

Design activities of team members can be coordinated via a Web-based workflow tool. Management of a business process can be tedious, time consuming, and inefficient. Managers must keep close watch on the status of tasks, and initiate subsequent tasks that depend on the former tasks once completed. This method of business process management is prone to inefficiency.

To address this problem, the concept of a workflow was developed. Basically, workflow is a tool that automates the business process by assigning tasks, tracking progress, and then assigning subsequent dependent tasks all in an automated fashion. The workflow methodology improves business process efficiency and control of the business process by virtually eliminating the time between the end of task A and the start of task B, and by allowing managers to focus more important issues than the assignment and tracking of tasks.

In the architecture presented, this is accomplished by allowing users to create workflows by defining a design process of tasks, with the tasks being analyses from the Analysis Library. Once the design process is defined, the manager starts the process, and then the process assigns tasks to perform specific analyses to specific engineers by sending them an e-mail. Once the engineer has completed the task, he or she uploads their results via a Web browser, and the workflow automatically identifies and assigns the next appropriate task in the workflow, and repeats this until the design process is complete.

E. Load Balancing

When items in the Analysis Library are executed, the runs are distributed evenly among a cluster of computers to minimize execution time and maximize resource utilization. Once an item is stored in the Analysis Library, an API exists that allows applications to invoke that analysis without needing to parse files or create script applications. Crucial to this capability is the ability to load balance out a set of requests to available resources.

When a request to run an analysis is passed to the system, the requirements for the analysis are first examined. For example, a structural analysis task may require a particular version of a commercial finite element analysis tool on a particular UNIX operating system. The system then identifies computers that support the requirements and then ships jobs to computers where they are run.

The trend towards probabilistic design methods illustrates the need for load balancing capabilities. Brute force probabilistic techniques such as Monte Carlo methods can require thousands of code evaluations to be performed. For most design problems, there simply isn't enough time to perform these jobs one at a time. The jobs must be performed in parallel across available resources.

III. Case Studies

Two case studies are discussed to show how these software modules can be combined to create advanced design processes through the Web centric architecture. The first scenario is an advanced multidisciplinary design optimization (MDO) process and the second is a robust design process.

A. Deploying BLISS Methodology

Bi-Level Integrated System Synthesis (BLISS) is a method that allows design problems to be naturally decomposed into subsystem optimizations and system optimization³. BLISS methodology has multiple advantages that make it very well suited to the design of complex systems: First, a complex optimization problem is divided into smaller problems that can be solved more quickly and easily. Second, it allows sub-system optimizations to be performed by disciplinary experts autonomously. Third, approximate surrogate models are effectively used to enable a system-level optimization that accounts for inter-disciplinary couplings at an acceptable total computational cost. And lastly, the decoupled subsystem optimization problems can be computed concurrently fully utilizing available computing resources for compression of the total elapsed project time. The authors have performed an initial study using ModelCenter to establish requirements of deployment of the BLISS methodology⁴.

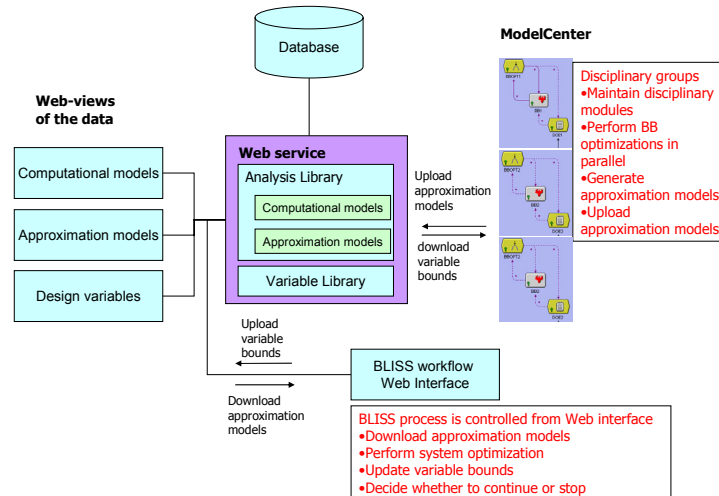


Figure 2: Collaboration solution for BLISS MDO.

The engineering design environment discussed in the work was designed to realize the BLISS methodology in a collaborative and parallel environment. The BLISS methodology will be implemented around the Web-centric architecture, allowing each engineering group to effectively contribute to the design work without caring about details of the sophisticated BLISS algorithm. The solution architecture in Figure 2 illustrates deployment of advanced MDO algorithms to multiple engineering groups involved in a design project.

1. *Performing Subsystem Optimization*

It is assumed that each disciplinary group creates and maintains disciplinary simulation models. The BLISS methodology requires creating approximation models for each sub-system optimizations, or black box (BB) optimizations. DOE technique is typically used to generate a set of design points in the system variable design space. For those DOE points, BB optimizations are performed to obtain optimized subsystem responses. For example, ModelCenter can be used to create simulation models and perform DOE on the simulation model. The DOE tool in ModelCenter sends off hundreds or thousands of BB optimizations to available computing resources using the load balancing service of the solution architecture. It is important that each disciplinary group can perform BB optimizations in parallel from ModelCenter in a transparent manner.

2. *Generating Approximation Models*

Each disciplinary group creates approximation models of its own BB module. The BB optimization results from step 1 are used to fit appropriate models such as polynomial or Kriging. For each of the response variables, an approximation model is created and the set of approximation models generated by each group will be used by the system optimization step of BLISS.

3. *Sharing Approximation Models*

Each disciplinary group uploads the set of approximation models to the central repository. For example, the upload can be done from ModelCenter by saving the approximation models into the Analysis Library. Now the approximation models become available for use of other ModelCenter users or other Web-based trade study clients. The approximation models will be version controlled with appropriate description meta data so that progress can be tracked during the BLISS process.

4. *Performing System Optimization*

System optimization of BLISS uses the approximation models from BB modules to optimize the system objective function and to satisfy compatibility constraints of coupled variables. The system optimization is isolated from the BB modules and typically can be solved quickly because only approximation models are used. A BLISS-specific Web page will be developed for the system optimization. The system optimization problem will be defined through the Web page by selecting appropriate approximation models stored in Analysis Library. The BLISS Webpage is built upon Web services used to check availability of data in the central repository and to perform

calculations using the approximation models. The system optimization will produce a new current design point and variable bounds will be updated. The variable information will then be uploaded into the Variable Library which can be accessed from the disciplinary teams for use in BB optimizations in the next BLISS cycle.

B. Deploying a Design for Six Sigma Process

To further illustrate how the architecture can be deployed, the implementation of a Design for Six Sigma (DFSS) program is described. This is typical of an effort that starts out with a high-level initiative to improve the design process through wide spread adoption of new design methodologies. While this example pertains to a DFSS process, it would be similar in application to other programs such as Robust and Reliability Based design, and initiatives such as Cost as an Independent Variable (CAIV) and Design to Cost (DTC). By implementing design processing using the Web based architecture, the design methods are hosted on a central server. Introduction of the processes to all users occurs simultaneously. Modifications to the design process are also updated at the same time and there is no need to update software on end users computers.

To present a design process to a group of engineers, it is important that tools and processes are standardized. In the described architecture, this would involve creating a set of Web pages that step the user through the DFSS process of performing a DOE, creating approximation models, and then performing a Monte Carlo analysis on the approximation models.

1. Preparing Simulation Models

Once a set of design wizards has been created, the system is ready for use. End users will first host their analysis tools in the Analysis Library. This will happen by creating and uploading a wrapper for the analysis tool, or in the case of frequently used tools such as Microsoft Excel, simply uploading the Excel file to the Analysis Library.

Once hosted in the Analysis Library, the user starts a design process by stepping through the DFSS design wizards. These wizards ask the user to select an analysis from the Analysis Library and then guide them through the steps to define key design parameters and performance parameters for robustness and reliability assessment. Problem formulation and trade study results are each managed and stored in the Shared Project.

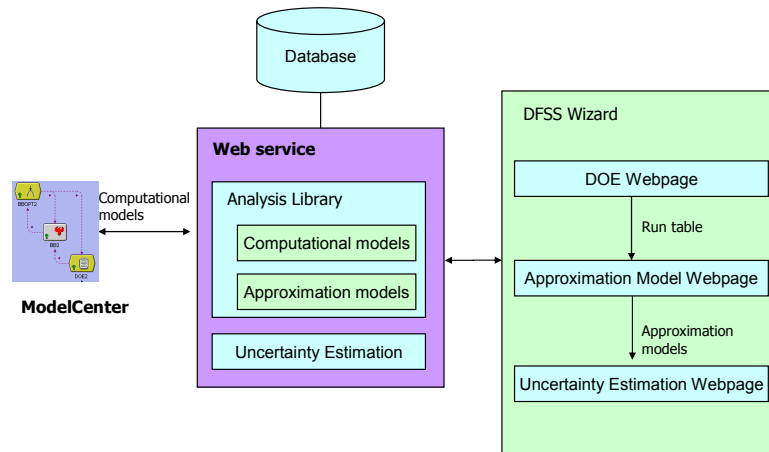


Figure 3: DFSS process presented through Web interfaces.

2. Creating Approximation Models

DFSS process requires capability to estimate robustness and reliability of the product, which involves many function evaluations of the model responses. To reduce computational cost, a common practice is to use approximate surrogate models of the computer simulation model. For each of the response variables, DOE techniques are used to select sample points, where computational simulations are performed. The set of approximation models are stored in Analysis Library. This step is similar to step 2 of the BLISS example, except that here BB analysis runs are performed instead of BB optimization runs. Using the fast running approximation models, variability of responses can be quickly evaluated.

3. Performing Uncertainty Estimation

A common approach is to perform Monte Carlo studies where probability density functions (PDFs) are assigned to input variables and the approximation models are run many times to predict the PDFs for key output parameters. Using the approximation models stored in the Analysis Library, this task can be performed in a short amount of time. The uncertainty estimation report will be presented to users as textual and graphical format so that users can adjust the design parameters to meet the design requirements.

IV. Development of Proof of Concept for Load Balancing

The initial focus of the architecture development is on the parallel trade study support. As a requirement gathering step of the software development cycle, a proof of concept of the parallel trade study solution has been developed. There are four elements in the proof of concept architecture.

- Trade Study Management Service (TSMS) is a Web service that maintains trade study requests from ModelCenter and stores the input and output variables of the runs.
- ModelCenter plug-in defines a trade study and submits it to the TSMS Web service. A parametric study plug-in and a genetic optimization plug-in have been implemented for this proof of concept.
- Web interface of TSMS provides user authentication mechanism and progress monitoring via Web-browsers.
- Connectors to load balancing service provide capabilities to submit runs to load balancing tools such as LSF⁵ or custom load balancing services being developed for ModelCenter.

A performance testing was performed to demonstrate load balancing capability of the proof of concept implementation. A small size PC cluster was setup consisting of four Windows nodes with LSF installed. A parallel parametric study was performed using the TSMS architecture. Figure 4 shows a preliminary performance testing results. The analysis model is a dummy component in ModelCenter whose run time can be changed as desired. Parametric studies with 12 runs were performed with varying length of run time, from zero to 300 seconds. It is expected that each node will perform three runs in average. Because of some overhead of LSF, there is no speedup gain until the single run time is over 10 seconds. The results show that the speed up converges to about 3.8, which is slightly less than the ideal speed-up of four.

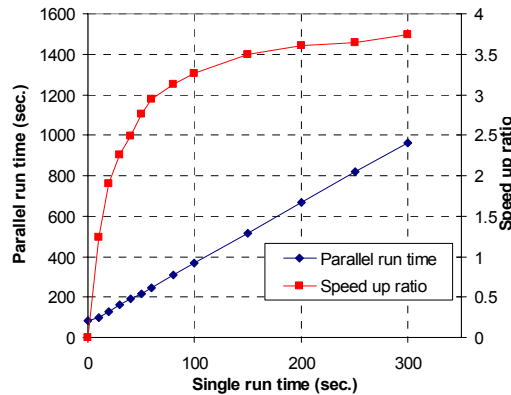


Figure 4: Performance testing of the load balancing proof of concept.

In summary, the proof of concept verified the capabilities of the Web-based trade study management system and demonstrated submitting runs from ModelCenter to available load balancing tools via the Web service.

V. Conclusions

This paper discussed the challenges facing engineering organizations today. There is need to produce designs more quickly, with better results, and of higher quality. To accomplish these goals, process improvements must occur at an organizational level beyond individual levels. The features of the described architecture each deliver

value in the engineering design process. Combined, they deliver on the areas identified as key to achieving the next level of process efficiency:

- *Knowledge retention:* The use of Shared Project results in an accumulation of knowledge during a project. This accumulation is not an extra step that must be performed, but rather integral to the day-to-day process. At the end of a project, key information is automatically stored in a central repository where it can be accessed using only a Web browser.
- *Better utilization of computing resources:* The use of massively concurrent data processing technology, aided by load balancing that evenly distributes jobs across available resources enables computing at a scale that has never been practical before. The use of a centralized Analysis Library combined with Web hosted design wizards makes the use of load balancing techniques transparent. The result is an efficient use of resources with minimal overhead required to configure and setup jobs, and more extensive exploration of the space of design alternatives within given time and budget limits.
- *Standardization of design methods:* Using the Web as a means to distribute design methodologies provides an effective mechanism to deliver company approved and well-defined methods to many end users.

The architecture described in this paper delivers on these needs. A Web portal based system was described that allows teams of engineers to coordinate and manage their efforts. The key components of this system are described. To illustrate how this architecture can benefit organizations, two examples have been described. The first example describes how to deploy the BLISS methodology. The second example describes how to deploy a DFSS process. Both examples are sophisticated methods that require effective mechanisms to realize a return on investment. Lastly, a proof of concept of the load balancing concept through Web service was demonstrated. It was shown that ModelCenter can submit a trade study to the Web service which distributes the runs to back-end load balancing tools.

Acknowledgments

This work done by Phoenix Integration, Inc., was partially funded by a NASA contract, NNL04AA10C. The authors would like to thank the many customers of Phoenix Integration who have contributed suggestions for improving the engineering design process.

References

1. Scott, A., "An Evaluation of Three Commercially Available Integrated Design Framework Packages for Use in the Space Systems Design Lab," Georgia Tech, 2001. (See http://www.ssd.gatech.edu/main/ssdl_pres_archive/SSDL_FrameworkCompare.pdf)
2. Phoenix Integration, Inc., "Improving the Engineering Process with Software Integration," a white paper, 2002, see also www.phoenix-int.com.
3. Sobieszczanski-Sobieski, J., Altus, T. D., Phillips, M., and Sandusky, R., "Bi-Level Integrated System Synthesis for Concurrent and Distributed Processing," *AIAA Journal*, Vol. 41, No. 10, 2003, pp. 1996-2002.
4. Kim, H., Malone, B., and Sobieszczanski-Sobieski, J., "A Distributed, Parallel, and Collaborative Environment for Design of Complex Systems," AIAA Paper 2004-1848.
5. Platform Computing Corporation, *Using Platform LSF on Windows*, 2004. (see also <http://www.platform.com>)