AIAA-2002-1296

STRUCTURAL DESIGN, ANALYSIS, OPTIMIZATION, AND COST MODELING USING THE ADAPTIVE MODELING LANGUAGE

John Marino
Application Manager
TechnoSoft Inc.
Cincinnati OH
AIAA Member

Adel Chemaly
President
TechnoSoft Inc.
Cincinnati OH

Abstract

Vehicle frame structural engineering and optimization is discussed with focus on integrated design, analysis, and cost modeling within a single software environment. Unique to this environment is an innovative object oriented architecture that implements a unified part model, providing various levels of modeling and analysis fidelity while capturing conceptual, preliminary, and detailed engineering methodologies. It automates and manages data transfer and interaction among users, designs, analyses, and tools. It employs a feature-based design environment for structure design guiding the user through the various levels of modeling fidelity. It supports concurrent design, analysis, optimization, and cost modeling and links multiple users over a network of distributed heterogeneous workstations. The environment supports generative modeling enabling design trade studies for model design configuration and shape optimization.

The environment builds upon and leverages many successful programs already deployed and in use. Various vehicle design cases are illustrated, thus highlighting the modular functionality and the multidisciplinary knowledge domains supported. It significantly enhances the overall engineering process for evaluating potential vehicle concepts through detailed and fully integrated design, analyses, and cost modeling.

Introduction and Demonstration of Need

Product design is a collaborative effort among different engineering disciplines that typically evolves through three stages: conceptual, preliminary, and detailed design. The process steps in designing a product are: recognition of need, definition of the product's functions, and identification of a solution employing the various engineering theories and software packages needed [1]. Structural design is merely one critical facet

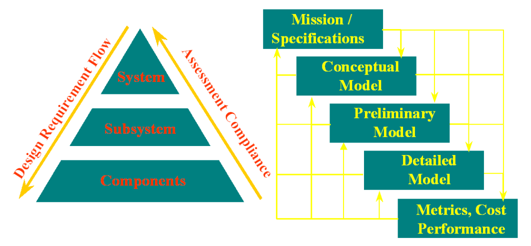


Figure 1: Product/Process Design

in this design process, as are mechanical design, structural analysis, manufacturing planning, and costing. The software industry has followed the concept of separating the engineering disciplines into specialized software packages, each focused on a single domain. Traditionally, cost analysis software products have adopted comparison methods that use databases to determine cost by weight using adjustments on similar previous design costs. Not only are such methods inadequate for accurate costing, but are often considered and treated as an afterthought.

The framework presented in this paper supports collaborative engineering integrating design, analyses, and cost into a common object-oriented unified model enabling cost analysis at the early stage in the design cycle. It also employs activity-based cost modeling methods that can compute the product cost through the management of the driving elements at the various steps of the product's production and assembly cycle.

In recent years, many research and development efforts have focused on automating the product design through the integration of the various engineering processes used in the design cycle. In reference to Air Vehicles, Blair notes, "A technologically advanced vehicle concept and a mission can be simultaneously designed in a single scenario based design environment. This allows the technology broker to aggressively market a product in terms which the warfighter can visualize and in the context of a vehicle concept" [2]. Adopting this integrated and unified model approach empowers the engineer. All parameters in the model can be linked to create dependencies among the disciplines that can affect part geometry, meshing, analysis, and cost. Most importantly, any change in these parameters will cause the dependencies to automatically propagate and update the model with little or no user interaction. This results in enormous time and resource savings especially during the performance of "what-if" trade studies.

Decisions made at the conceptual stage can have a significant impact on the cost of the final product. It is of vital importance to assess and accurately estimate the product cost at this early stage. To do so, the production costs reflecting the various processes and activities of the product design environment must be accounted for as early as possible in the design process. Cost estimation techniques must reliably and accurately represent production systems and processes while being tightly linked with the engineering design cycle. Therefore cost calculation methods should be an integrated part of an engineering framework that supports the concurrent design and analysis of a product. The product and process cost facets are integrated parts of the unified part model as described in this system and linked to product design through properties and attributes dependencies. Various facets of a product can be detailed through the unified model. The design strategies and related engineering and production processes are captured within a single part model, represented by a hierarchy of objects.

Additionally, this framework employs unique functionality to enable the integration and control of "Components of the Shelf" software tools as well as linked libraries and software servers. The unified part model captures common domain knowledge through the inherent functionality available in these tools and automates the data management and execution control, thus minimizing unnecessary data transfer and tedious user interactions.

The engineering of a product is a multidisciplinary cycle that incorporates conflicting objectives as required by the various disciplines. For example, maximizing performance, minimizing production time and cost, while minimizing weight are often competing but complementary objectives. Also, the minimization of an air vehicle structure's weight conflicts with the requirement of maximizing its performance when competing cost against range. Material type, fabrication methods, loading conditions, boundary constraints, and initial conditions are all potential design variables of the different design facets as used in the multidisciplinary processes. Typically these variables are evaluated separately by executing different applications used by each discipline and by storing the results for use in the optimization methods. Several commercially available software packages have adopted this approach in support of their optimization algorithms. The problem lies in the large amount of time and computer resources taken to prepare the input for the optimization routine. Thus, an important requirement of a design and optimization environment is the seamless integration and control of the tools used in the product engineering process. Parameters may be altered and the response to these changes may be recomputed on demand as requested by the optimization methods thus supporting runtime "on the fly" design optimization.

Technical Approach

The following section documents an overview of the technology used in the environment presented in this paper. This environment is built upon the *Adaptive Modeling Language (AML)* from TechnoSoft Inc., and leverages work performed on several projects developed in collaboration with government labs and industrial partners.

System Framework Overview

At a high level, the framework enables the communication of design and implementation specifications through the effective use of captured, organized knowledge. By capturing engineering strategies and iterations as part of various scenarios used in product designs, the environment alleviates potential design conflicts and can suggest solutions that would otherwise not be found until much later in the design process.

The environment is based on the *AML* object-oriented software framework which is an Adaptive Modeling Language for knowledge-based engineering. It is a comprehensive modeling paradigm that supports concurrent engineering integrating design specifications, part geometry/features, manufacturing, inspection, and analysis processes. *AML* provides a web-enabled engineering framework that captures knowledge from the modeled domain to create parametric models with that knowledge. Classes may be defined and methods may be written against these classes providing user-defined behavior. Aside from its web-enabled platform independent JAVA interface, *AML* is supported natively on all Windows and UNIX platforms.

Hierarchical Tree Model

Models representing various aspects of engineering processes are composed of objects that make up a hierarchical tree as seen in Figure 2. In addition to a class hierarchy of super-class and sub-class relationships, the system also manages a model tree hierarchy, or part hierarchy, of object/sub-object relationships providing a real-time modeling environment. Objects are instantiated and properties are specified in real time, dynamically forming the hierarchy of the part's object. Objects and their properties can be added, edited, or deleted independent of the order of instantiation.

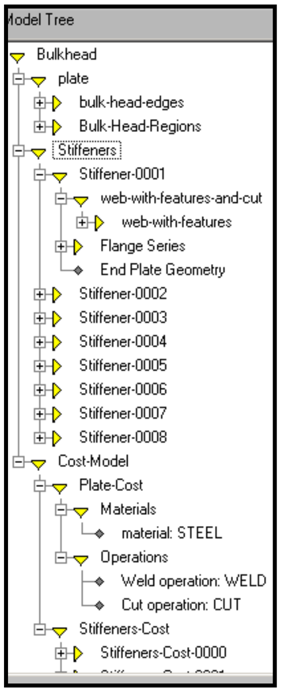


Figure 2: Model Tree

The modeling environment provides a grammar for traversing the object tree. This grammar allows any aspect of the object tree to be queried while also establishing a dependency. Objects' properties can be assigned simple values or formulas. The formulas can employ referencing to access other properties in the object tree thus creating a network of dependencies between objects and properties. Event-driven dependency management is provided to dynamically track when properties become undefined (smashed) due to their dependencies or need to be recalculated (demanded) when required by other properties.

Levels of Model Fidelity

In traversing the conceptual, preliminary, and detailed stages of design, various engineering process requirements dictate different representations of the same part. The part's design features could be different from the part's manufacturing features or analysis features. Therefore, a part model could include a number of object hierarchies representing various discipline requirements at different stages of the design.

Different hierarchies of objects representing the domains of design, manufacturing, analysis, costing, etc., could make up one single part model at one particular instance in the design process. These hierarchies can be cross-referenced throughout different stages of the design process, thus making a property within one hierarchy dependent on a number of properties within different hierarchies. Additionally, various models could be cross-referenced. This occurs as a design evolves, deriving some properties from past model instantiations of the same properties.

Distributed Collaborative Design

A part model is created by instantiating various classes to form the hierarchy that describes/captures the design intent. The objects making up the part model may be part of several modules' classes and may be distributed over a network of machines. The framework's "model manager" manages the communication among the objects' properties. Through the integrated web-based environment functionality, a single part model can be distributed across the network, with multiple applications running on different machines, supporting various operating systems, going through firewalls, and is interconnected via standard Internet connections.

Distributed models can be classified in different categories due to the fact that: 1) the model is made up of various objects that are distributed over the network, 2) the model is made up of objects residing over a single station, but linked to application servers distributed over the network, and 3) multiple models independently initiated on several machines may establish a dependency on each others objects after being initiated. Various mechanisms are supported in providing multi-user access control and security.

Activity Based Cost Modeling

Activity Based Costing (ABC) accurately models the cost of a product as reflected by the various activities involved in its design-to-production cycle. ABC can be applied equally well to both engineering processes and management systems. Similarly, ABC can be used in the modeling and prediction of cost associated with the sustainability of the product [3]. The approach combines the important cost elements of a system with feedback loops representing the complex inter-relationships between the various cost elements. These feedback loops link the decision-making process with the costs and performance of a system. These techniques allow engineers and managers to test alternative decisions and explore a wide range of “what-if” scenarios, thus saving time and money, reducing risk, while also improving performance.

The ABC model of a product or service is created by capturing and representing the various activities and triggers for events that change the model entities. These triggers occur due to the changes in the related production resources, processes, equipment, etc. The cost of performing an activity is measured by using cost drivers that reflect the relationship between an activity and the resources consumed by it. Similarly, the cost of a product is measured by using cost drivers that reflect the relationship between the product and the activities contributing to it.

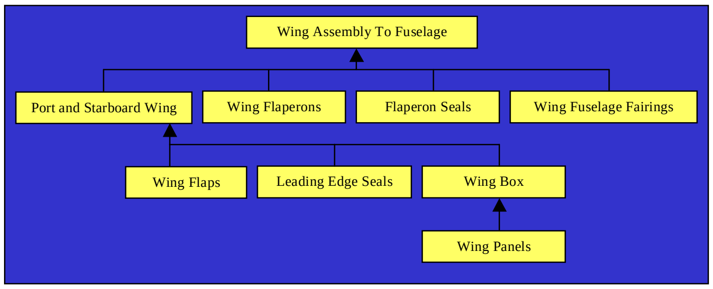


Figure 3: Wing/Fuselage Assembly Hierarchy

Manufacturing systems are one example where the reliability of cost estimates is required if the cost of the product is to be managed. These cost models need to be explored at the conceptual stage of the design process when manufacturing and design options are performed in order to determine the feasibility of the product. Such models need to evolve as the product design moves into the advanced stages, and more details become available about the activities involved in the process.

Figure 3 shows a simplified manufacturing and assembly process for an aircraft structure as an example of a system where accurate cost modeling is required. Typically, such a system is composed of several main assembly stages. These stages focus on the production of the major components that are part of the final product assembly.

At the various engineering design cycle stages, the environment provides a framework for modeling and simulating complex systems, accounting for their cost, design, production, and their sustainability, while also keeping in mind performance issues. Various design, manufacturing, and inspection processes (and their activities) are easily modeled, and can be dynamically linked to various engineering activities to effectively simulate and forecast a vehicle/system cost. Similarly, the various elements of the operation, and maintenance of a product could be included to forecast a model’s sustainability cost.

<u>Optimization</u>

Engineers often perform interdependent trade studies or design optimizations on design parameters, spread across several disciplines, each using a different tool. However, the optimum solution may reside outside the search domain, rendering the internally focused system incapable of finding it. Therefore, the use of many software tools related to optimization often focuses on a local optimum within a set domain. In general, the design-parameter search domain is defined either by establishing a set of upper and lower bounds, and/or a set of constraints. The “evaluated” parameters depend on design parameters as the optimization process executes new design parameters. These are examined to investigate which “evaluated” parameters are required. Initially, multiple iterations are conducted, establishing variation trends, or sensitivity, to the design parameters. Current optimization software employs various methodologies to iterate over the design variables’ domain (input), deriving the sensitivity (dependencies) of the input over the output (dependent variables) of the requested optimum value. Once the sensitivity is identified, the search focuses on converging to the appropriate local optimum. When set, the user is required to provide the optimization problem with most of the search data, and may also be required to interactively provide data as the search progresses.

To initiate the optimization process, a search range and a general guidance direction must be set. For each new set of design parameters the optimization tool requests, the value of the “dependent” variable must be provided or computed. Often preparation of the data required for optimization input code is many times more computationally intensive than the optimization algorithm demands. Another issue is associated with the effort necessary to establish relations among parameters being optimized, and independent variables targeted to compute the sensitivity among interdependent input parameters. This effort can be very time-consuming, as these input/output parameters are often the result of different nonintegrated procedures and tool inputs/outputs; thus requiring user interaction to manage the process.

Consider an example of a trade study, performed on a wing substructure, targeted to identify an optimum number of spars and ribs. In addition to identifying the component's location for minimum wing weight, this study requires the user to act as an intermediary among the optimization codes and various design/analysis tools. The design tools, such as a CAD system, provide the geometric modeling and the weight computations. An analysis tool, such as a finite element based solver, provides the detailed structure analysis to assess load and stress distributions for a particular design. Another finite element modeling tool is also required for pre/post-processing. This tool aids in both generation of the mesh, and also in the interpretation and visualization of the results. For each set of design parameters, these tools are used to provide a set of "evaluated" parameters, fed to the optimization tools for correlation. These tools also recommend a new set of design parameters to be used for the second iteration, and so on. This cycle could require a large number of iterations before the optimization converges to an optimum set.

To assist in setting the various tools as part of the optimization process, some efforts found in standard literature has focused on the "black box" interface concept. A major shortfall of many design study software packages implementing this approach is that they are black box in nature [5]. The "black box" interfacing environment focuses on the use of a wrapping technique to link the processes within a loop for optimization. This approach offers little or no capability to integrate the overall design-to-production process within a framework, allowing the identification and tracking of various dependencies among different models (analysis, design, manufacturing, cost, etc.).

The key goal is to seamlessly integrate the optimization with various engineering tools into one unified part model. The framework automates the data information flow in and out of optimization tools, guiding the search for an optimum design or set of design optimums as is the case with multi-objective or sensitivities studies. This approach presents major time savings in: 1) setting the optimization process, 2) computing dependency trends prior to initiating the optimization process thus presenting major advantages in speed of computation, and 3) integration of several tools and models. This automates much of the tedious user interactions between various tools/processes and computes required data for the executed optimization iteration routines.

Model Documentation and Presentation

The documentation of product evolution and version control of product releases are critical to any successful design. Additionally, the presentation of the results is vital to any successful collaboration and knowledge exchange. Engineers and product managers often use tools that support advanced visualization methods for the manipulation and display of the part geometry and associated analysis results. Bar, pie, and curve fitting charts are also used to present and depict costs, production time, and other data to quantify many of the non-geometric aspects of a design and its related processes' results.

Model dependencies can be viewed as hierarchical trees. These dependencies can be cross-correlated to determine whether common dependencies exist among the various parameters being optimized. Common dependencies may indicate the parameters best suited to be used as the design variables for optimization. Providing the ability to set bounds or constraints on these variables and to associate these constraints with other dependencies is a critical functionality of the environment.

In general, collaboration is classified either as an asynchronous or synchronous process. When multiple users are collaborating in real time in the engineering of a product, this is referred to as synchronous collaboration and must be a capability supported through the framework environment. Conversely, the collaboration is performed often as part of a distinct process, but yet parametrically connected and interdependent. This is referred to as asynchronous (off-line) collaboration and usually is supported by a product data manger (PDM).

Framework Architecture and Functionality

The framework environment is modular. Each module focuses on a single domain knowledge. The overall system seamlessly integrates all modules through the unified part model in which all data (input) and all information (output) are stored. These modules are organized as follows:

1. Design Module
2. Analysis Module
3. Optimization Module
4. Model Documentation & Presentation Module

The Design Module supports a feature-based environment with an integrated solid/surface modeler. It enables the configuration and layout of the frame structural members, and definition of their dimensions, topological parameters, and material properties. At the concept stage, the members are simple in representation. Their detailed properties will be automatically derived as the design process evolves through the use of the integrated analysis methods for structural sizing as well as the manufacturing methods for shape refinements. Therefore the environment supports efficient and effective design evolution. The strength of such an environment lies in the ability of the user to create conceptual representation of the part(s), while preliminary and detailed representations coexists and can be traced throughout the design evolution. The level of fidelity required at each stage is captured and represented in the part model's objects. Therefore, no need exists to "suppress" or exclude design features in a design for the purpose of analysis. The features are simply included, or dynamically selected, in an analysis when needed.

The Analysis Module integrates custom methods supporting parametric, rule-based, manifold, and non-manifold mesh generation. Attributes for load, boundary, and initial conditions can be directly set on the part geometry and automatically associated with the generated mesh groups. Analysis features can be directly applied to the mesh or the geometry. For example, loads and constraints may be applied directly to the solid model geometry resulting in automatic regeneration of mesh parameters and constraints if the geometry is altered. Mesh element size can be parametrically associated with the part model feature dimensions and element distribution could also be associated to geometry topology type. Unique to this environment is "attribute tagging and propagation". The attribute tagging process traces the enhancement of model topology during the geometry creation process and as it changes through the Boolean operations. The final geometry attributes required for meshing are derived using the model tagging and propagation methods, which compute the final tags' properties from the initial feature geometry tags. Automated 2D, 3D, structured, and unstructured meshes are supported.

The Analysis Module links with several structural analysis solvers including *MSC.Nastran*, *MSC.Marc*, and *ANSYS* supporting several types of analysis. Other analyses packages, solvers, and analysis types may be easily integrated. Identical analysis studies can be performed using different solvers from the same model in a "plug and play" fashion. Nodal coordinates, connectivity, elements and material properties, loading conditions, bounding constraints, and solution control parameters are specified through the analysis model objects. The node sets, element sets, load sets, constraint sets, are specified through mesh queries which are dynamic and parametric subsets of the integrated mesh. These parameters/controls are linked parametrically to the mission specifications or any other model parameter thus providing complete control of the model within the environment.

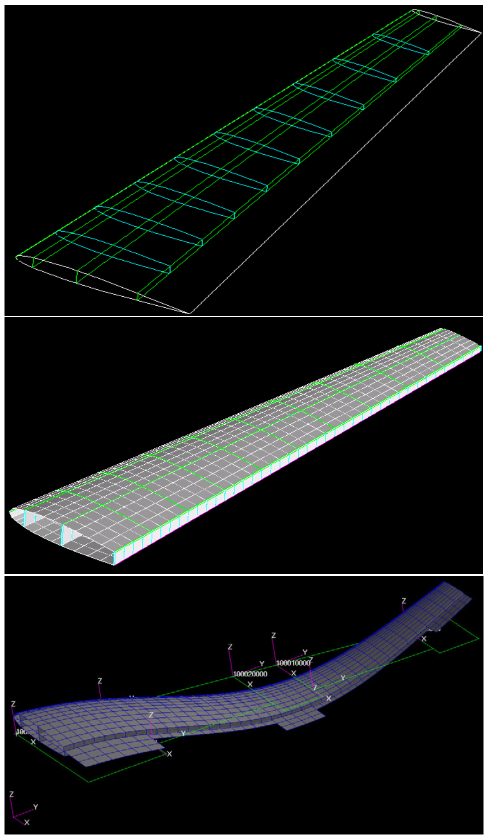
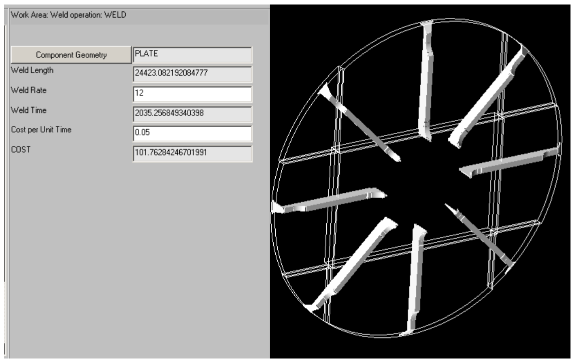


Figure 4: The design and configuration of an air vehicle wing is illustrated. It couples the design with aero and structure analyses, and integrates the production model for affordability assessments.

The Optimization Module supports and manages exploration methods and trade studies for optimizations. It captures interactions among disciplines and, therefore, quantifies interdisciplinary relationships. This enables system level trades such as cost versus performance, thus allowing the real-time creation of trade studies supporting design guidance and exploration. It provides a unique visual environment that bridges the gap between model design and optimization methods. It supports numerous algorithms and methods enabling the traversal of an entire design space linking design variables, constraints, multiple and single objective exploration methods, and various solution analyses through the automated generation of data and responses in real-time or in batch mode.

The Optimization Module provides a suite of graphical tools enabling rapid setup, performance of trade studies, and exploration of parametric designs and analyses. It also enables multiple users to collaborate in the real-time setting of exploration studies. It supports the following design exploration methods: design of experiments, enumeration (nested iteration), a multi-objective genetic algorithm, Nelder-Mead simplex method, Powell method, and user-defined methods via linked libraries or integrated stand-alone executables.

The elements of a trade study can include variables from the different models' facets. For structural analysis, the system has specialized algorithms automating structural layout configuration and sizing. As an example, the design variables may include non-geometric parameters such as material properties, loading conditions, displacement constraints, and environmental (ambient) conditions. For analysis sensitivity response, element mesh size, element type, and element aspect ratio may also be varied/analyzed to inspect their effect on the design exploration. Unique to this environment, these variables may be defined as functions of other parameters in the model


Work Area: Weld operation: WELD

Component Geometry	PLATE
Weld Length	24423.082192084777
Weld Rate	12
Weld Time	2035.256849340398
Cost per Unit Time	0.05
COST	101.76284246701991

Figure 5: A submersible pressurized vessel bulkhead design is illustrated. The model configuration, finite element analysis for structural sizing, and activity-based costing are performed.

such as mission specifications, geometric parameters, loads, constraints, input tables, and cost.

The optimization methods, variables, and constraints can be altered at runtime to suit the user's discretion. Any libraries, tools, as well as applications linked through the unified-part model are directly controlled through the system's data manager as values are assigned to variables which are dictated by the optimization algorithms.

The Model Documentation and Presentation Module manages releases and versions of the model, trade studies, and data supporting results/outputs visualization. It supports the model and process data management by storing and organizing their configuration parameters and results. When conducting trade studies during a design optimization, the configuration settings applied to the selected design variables, and the objectives results are automatically traced and stored. When multiple users conduct manual "what-if" trade studies by interacting in the off-line collaborative design of a product, the system manages and traces the user interactions. Requests for changes in the design parameters are documented and tracked by this module including sending notifications to the appropriate users and tools.

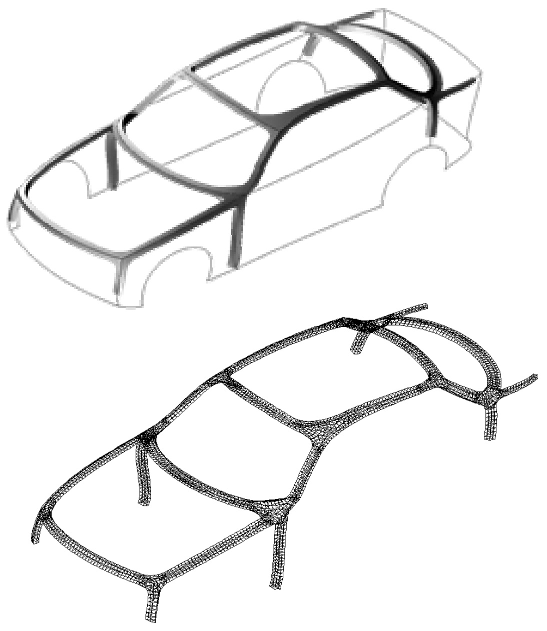


Figure 6: A vehicle space frame configuration is illustrated as used in the system with focus of on structure layout, body stiffness optimization, crash assessment, and production planning.

Additionally, this module supports a web-enabled environment for defining, managing, and publishing all aspects of an engineering project. Most of this module's functionality builds upon *Views*, a product from TechnoSoft. It consists of three integral sub-modules for publishing, managing, and distributing, complete product engineering data models. Geometric as well as non-geometric data relevant to product design details and associated processes can be released and distributed for viewing and inspection through a controlled-access environment.

This module facilitates collaboration among the participants involved in a product engineering design by incorporating a web application software server that provides project and user management functions. It controls the data flow among globally dispersed team members and application tools, manages the releases of product and process data models, and provides key functionality for project data warehouse access control, version tracking, and data processing.

Product models, process data, and analysis/simulation results can be viewed and annotated, and evaluation of alternative parameters can be requested. It provides a facility to capture and publish trade studies and simulation results. Given that the model hierarchy is captured and documented, the product models can also be used as "live" engineering reports showing engineering drawings, analysis results, and process documentation.

Conclusions

To date, the software industry has followed the concept of developing software application tools focused on specialized disciplines or single engineering domains; each is developed with little or no interoperability. This paper discussed an object-oriented modeling framework incorporating a unified part model that supports the modeling of domain knowledge and integrates software tools and engineering processes as needed in the support of a collaborative design.

This framework significantly enhances the overall engineering process supporting the evaluation of potential vehicle concepts through detailed integrated designs and analyses. Detailed studies are fully automated to reduce engineering time and cost while expanding the exploration of the design space; integrated optimization capabilities enhance this exploration. By enabling the thorough investigation of alternative designs, it facilitates informed design decisions.

TechnoSoft is collaborating with DoD labs and industry partners for refinement of the framework architecture and its deployment in various industries. Numerous applications have adopted this framework and are presently deployed successfully.

REFERENCES

[1] Groover, M. P., "Automation, Production Systems, and Computer Integrated Manufacturing", Prentice-Hall, Inc, NJ, 1987, Pages 710-712.

[2] Blair M., "ENABLING CONCEPTUAL DESIGN IN A TECHNOLOGY-DRIVEN ENVIRONMENT", AIAA Paper 98-4741

[3] Blair M., Hartong A., "MULTIDISCIPLINARY DESIGN TOOLS FOR AFFORDABILITY", AIAA paper 2000-1378

[4] Bharatram G., Blair M., Hartong A., Kamhawi H., Zweber J., "STRUCTURAL AND MANUFACTURING ANALYSIS OF A WING USING THE ADAPTIVE MODELING LANGUAGE", AIAA paper 98-1758

[5] Engineous Software Inc. Website: http://www.engineous.com/matrixdesc_popup.htm#solution

[6] TechnoSoft Inc. Website: http://www.technosoft.com/docs/Product-Sheet-Views-v1.pdf