Center for

SIMULATION AND DESIGN OPTIMIZATION

of Mechanical Systems

(D)ARPA Initiative In Concurrent Engineering Phase 4 and Phase 5

FINAL REPORT

College of Engineering
The University of Iowa
Iowa City, Iowa 52242

Center for Computer Aided Design
The University of Iowa

(D)ARPA Initiative In Concurrent Engineering Phase 4 and Phase 5

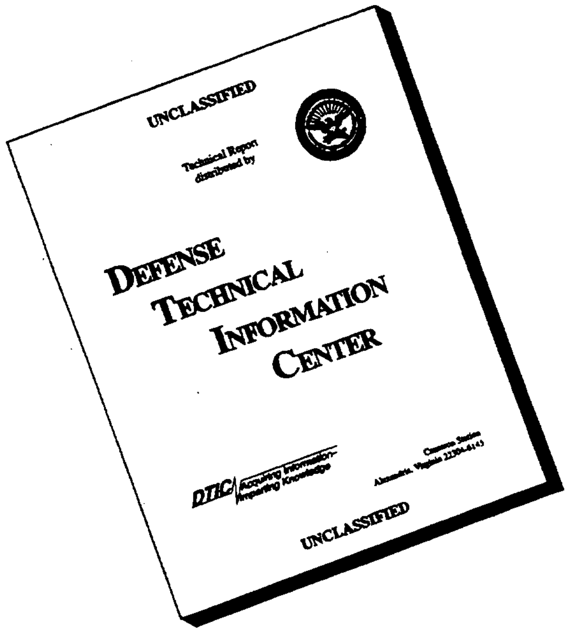
FINAL REPORT

Tracked Vehicle Concurrent Engineering Tool Development, Integration, and Validation

Collaboration Technologies for Large-Scale Mechanical System Concurrent Engineering

Sponsored by:
(Defense) Advanced Research Projects Agency
Defense Sciences Office
Grant No.: MDA972-91-J-1023
ARPA Order No.: 8363/8363.1

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

Center for Computer Aided Design
The University of Iowa
Iowa City, IA 52242-1000

(D)ARPA Initiative in Concurrent Engineering (DICE)

Phase 4:

Tracked Vehicle Concurrent Engineering Tool
Development, Integration, and Validation

Phase 5:

Collaboration Technologies for Large-Scale
Mechanical System Concurrent Engineering

FINAL REPORT

December, 1995

Sponsored by:

(Defense) Advanced Research Projects Agency
Defense Sciences Office
Grant No.: MDA972-91-J-1023
ARPA Order No.: 8363/8363.1

Table of Contents

Table of Contents (Con.)

List of Figures

Figure

List of Figures (Con.)

Figure

List of Tables

Table

Contributors

The principal contributors to this research effort were the following staff and affiliated faculty members of the Center for Computer Aided Design

STAFF

Dr. Kuang-Hua Chang
Mr. Joel Steele
Dr. Jun Tang
Mr. John Todtz
Dr. Chung-Shin Tsai
Mr. Tim VanFosson
Dr. Jia-Yi Wang

FACULTY

Dr. Kyung K. Choi
Dr. Edward J. Haug

Technical Writer: Joel Steele
Editor: Larry Long
Graphics: Benn Dunnington
Secretary: Kathy Snow

Center for Computer Aided Design
The University of Iowa
Iowa City, IA 52242-1000

Phone: (319) 335-3370
FAX: (319) 335-3380

Executive Summary

This document constitutes the Final Report for the (D)ARPA Initiative in Concurrent Engineering (DICE) Phase 4 and Phase 5 research projects, *Tracked Vehicle Concurrent Engineering Tool Development, Integration, and Validation* and *Collaboration Technologies for Large-Scale Mechanical System Concurrent Engineering*, respectively, carried out at the Center for Computer Aided Design (CCAD), The University of Iowa. Applying basic concepts developed under prior phases of the DICE program, these and related research efforts have defined and implemented a comprehensive, integrated suite of Computer Aided Engineering tools and design methodologies supporting multi-disciplinary product development for a broad base of military vehicle systems. This report describes the software tool, integration, and collaboration technologies developed under these efforts, as well as the underlying conceptual methodologies, research performed under parallel tool integration projects, a high level process for the utilization of these technologies in a multi-disciplinary design environment, and both internal and external example applications demonstrating the utility of these technologies for defense industrial use.

The DICE program was initiated in 1988 by the (Defense) Advanced Research Projects Agency ((D)ARPA) to define, develop, and transition to U.S. military and industrial organizations, technologies and methodologies that promote concurrent engineering of products. Concurrent Engineering (CE) is a systematic design development methodology that seeks to substantially reduce product development time by increasing the degree of simultaneity among design activities, while also providing for the development of more mature product design concepts through increased input early in the design stage from all elements of the product life cycle, e.g., requirements definition, design, evaluation, testing, manufacturing, support, etc.

In the context of the CCAD's DICE efforts, Concurrent Engineering exploits robust simulation-based design capabilities to support rapid development of product systems and components concepts through the application of a software integration environment architecture. In this environment, a number of structural and performance simulation tools, envisioned for use by diverse design perspectives throughout the product life cycle, are integrated by means of a common product model, sufficiently rich in data characteristics as to support the analysis requirements of each of the tools comprising the integrated tool suite. By means of powerful communication and data sharing functionalities, engineers can access the product model and obtain a product representation that meets the data requirements of their respective analysis tools. Once appropriate design representations have been obtained, each engineer can analyze the design representation with respect to his discipline, iteratively perturb and analyze the design representation to optimize performance, and suggest changes to the design of the product model which represent an improvement from his particular engineering perspective. Software technologies supportive of high level collaboration are embedded in the architecture to enable engineers to effectively engage in design trade-off to achieve an optimized product design that meets the requirements of the end user.

Under CCAD's DICE Phase 4 effort, a prototype integrated software environment was developed incorporating CAD and CAE modeling, multibody dynamics simulation for tracked vehicles, structural dynamic stress and fatigue life prediction, and structural design sensitivity analysis and optimization software capabilities. The integration architecture comprises commercial geometry modeling systems, finite element analysis codes, an object-oriented database and database server, a network communication channel, and a series of workspace wrappers. An example design exercise, based on modeling and analysis of an M1A1 Abrams main battle tank and a constituent suspension component, was performed to demonstrate functions and operations of the integrated design environment. In addition, three military vehicle developers were contracted to define and carry out design applications representative of an industrial level of vehicle design development, using the integrated tool environment. The results of these exercises, described herein, effectively validated the applicability of the integrated simulation-based design environment to address and solve realistic design problems using the concurrent methodology.

During the interim period between the conclusion of the DICE Phase 4 effort and the start-up of CCAD's DICE Phase 5 effort, a research effort sponsored by

the Defense Modeling and Simulation Office (DMSO) was undertaken to extend and refine the prototype integrated environment to include software capabilities supporting general multi-body vehicle dynamic analysis, reliability analysis, and maintenance and support simulation and analysis. This effort, *Simulation Based Design for Military System Supportability and Human Factors*, introduced CCAD's "view" concept for product modeling, whereby a common, CAD based design model is used to derive engineering analysis models for each analysis discipline. In this manner, a consistent schema for product representation and design change is maintained between the design (CAD) perspective and analysis perspectives, and subsequently between individual analysis perspectives, via the common CAD representation. As a result, a simplified global database model was obtained, substantially enhancing the "extendibility" of the integrated software environment to support additional design perspectives.

Under CCAD's DICE Phase 5 effort, technologies and methodologies designed to enhance collaboration among engineering users of the integrated simulation tool environment were defined and implemented. Phase 5 research targeted two areas for development, the first defining requirements and methods for parametric representation of mechanical system CAD and CAE models. The intent of the second research thrust was to formalize the process by which the parameterized view structure and analysis tools are employed in design evaluation and optimization, and implement process management methodologies and tools to promote focused, meaningful interaction among engineers within the framework of the multi-disciplinary design project. The introduction of the parametric methodology was achieved by the incorporation of a suitable parametric CAD modeler in the integrated environment and the identification of CAD and CAE model parameters appropriate for representation of mechanical system design. Management of the team of environment users has been enhanced through the extension of the integration architecture to include team organization modeling, process definition, project tracking, and communication tools.

The end result of the DICE and related efforts has been the establishment of the CCAD integrated environment software testbed. The CCAD testbed represents a near commercial quality level of implementation capable of supporting an industrial degree of design and analysis applications. The testbed comprises the entirety of CCAD's developments in dynamic, structural design sensitivity, durability and reliability, and maintainability analysis tool capabilities, as well as integration architecture utilities. The testbed is currently being upgraded for use over the Internet and is available for on-site installation at participating DICE program and CCAD member organizations.

While the software environment and methodologies developed under the CCAD's DICE efforts represent a significant achievement in the application of simulation-based design technologies to promote concurrent product development, considerable research and development remains to achieve seamless tool interoperability among diverse, distributed design and production enterprises. The CCAD is currently continuing research in this field under several industry and government sponsored programs. Two on-going project efforts include ARPA's *Integrated Product and Process Development (IPPD) Simulation* project and the NSF-sponsored *Information Integration for Simulation-Based Design and Management.* Under ARPA's IPPD Simulation program, CCAD is continuing development of the computational methodology for multi-disciplinary, parametric design trade-off in support of both concept and detailed product design. CCAD's NSF effort is exploring the application of STEP and other standardized data model formats to promote seamless exchange of parameterized CAD model data in supplier-manufacturer operations. The technologies and methodologies developed under CCAD's DICE effort provided a solid, realistic foundation for the application of simulation-based design technologies in these, and many other, cutting edge design technology initiatives.

I Introduction and Background

Recent evaluation of the product development process suggests that essentially all development activities prior to manufacturing and use of a product are in fact simulations.[1] In this broad sense, simulation can be either mathematical models or physical experiments which reproduce under controlled conditions the environment and circumstances under which the product is to be used. The previous decade, however, has seen a revolution in the development of computer aided technologies supporting design, testing, evaluation, and manufacturing of products, both in terms of computational sophistication and processing speed. As a result, an increasingly larger percentage of the product development life cycle is being conducted using high-fidelity computerized mathematical models, or simulations, displacing traditional methods of physical testing and evaluation. Beyond simple one-for-one displacement of physical analysis in the traditional design-build-test product life cycle, however, the advent of computer modeling and simulation has the potential to totally restructure the product development process, achieving a Concurrent Engineering approach to product development that can substantially reduce development time and costs.

Concurrent Engineering (CE) is a systematic design development methodology that seeks to substantially reduce product development time by increasing the degree of simultaneity among design activities, while also providing for the development of more mature product design concepts through increased input early in the design stage from all elements of the product life cycle, e.g., requirements definition, design, evaluation, testing, manufacturing, support, etc. Since before 1988, DoD-Industry Computer-aided Acquisition and Logistics System (CALS) Task Groups have worked to chart roadmaps for effective implementation of CE, suggesting the evolution of a "simulation-based design" approach to CE.[2] A fundamental difficulty in the application of computer simulation technologies in support of CE is, however, the sheer diversity and scope of modeling and simulation software capabilities available and employed in support of the various perspectives in the product development process. Each product development discipline, and subsequently each simulation application, exhibits model and data requirements and analysis output peculiar to that perspective/application with little or no cross-disciplinary commonality. A major roadblock exists, then, due to this lack of "interoperability" necessitating effective means to integrate both the operations of design disciplines as well as the simulation technologies which they employ.

DICE Concept

In 1988, the Defense Advanced Research Projects Agency (DARPA) launched the five-year DARPA Initiative in Concurrent Engineering (DICE) program to encourage and enable the application of CE methodologies and tools in U.S. military and industrial organizations. During the first three phases of the DICE program, General Electric and the Concurrent Engineering Research Center (CERC) at West Virginia University defined a basic integration framework for major CE software environment applications. This framework, depicted in Figure 1.1, establishes the use of a shared product database, "wrapped" workspace tools, and a network communications channel to attain the requisite tool interoperability for data exchange and design development in the CE context.

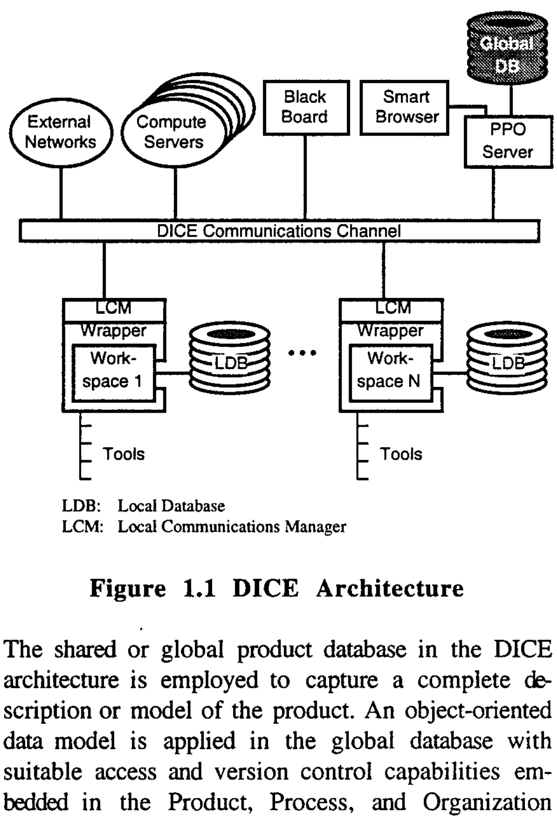


Figure 1.1 DICE Architecture

The shared or global product database in the DICE architecture is employed to capture a complete description or model of the product. An object-oriented data model is applied in the global database with suitable access and version control capabilities embedded in the Product, Process, and Organization

(PPO) server to manage the evolving product design. A high-speed, high-capacity network communication channel serves as the backbone for the DICE architecture, enabling distributed access and communication in the integrated environment, promoting the establishment of the virtual enterprise. Wrappers are the software entities that enable the connection between the individual tool capabilities and the integrated architecture. Wrappers provide the engineer with the front end interface to the product design, functioning as data extraction and translation mechanisms; selecting product model information from the global database and formatting it according to the requirements of the particular workspace application which it serves. The wrapper concept provides the integrated simulation environment with its extendibility, whereby additional workspace tools are connected to the environment through tailoring of the wrapper, rather than the workspace itself.

CCAD/CERC Pilot Project

In 1990, the CCAD, under DICE Phase 3 subcontract funding from the CERC, initiated the *Tool Integration for Concurrent Engineering* (TICE) pilot project. Based on the conclusions of the CALS R&M Mechanical Design Study[3], this effort was undertaken to demonstrate the feasibility of implementing a Concurrent Engineering tool environment and concomitant database schema to support combat vehicle engineering. The CALS Task Subgroup concluded that the primary technical challenges to be addressed in the design and engineering of combat vehicles involved vehicle system dynamic and structural performance, reliability, and soldier-system interaction. Building on research in multi-body dynamics and structural Design Sensitivity Analysis (DSA) ongoing since 1987, the CCAD developed integration methods and technologies at both the workspace and environment level to create the pilot project environment illustrated in Figure 1.2, using the DICE architecture as a basis. The principal accomplishments under the joint CCAD/CERC effort, then, consisted of the definition of database model schema at both the global and local levels supporting mechanical system design and analysis, and definition and implementation of workspace wrapper software utilities.

Development of local database model schema constitutes the fundamental aspect for tool integration at the workspace level. The database model and file formats are defined specifically to correspond to the requirements of the analysis and modeling tools embedded in

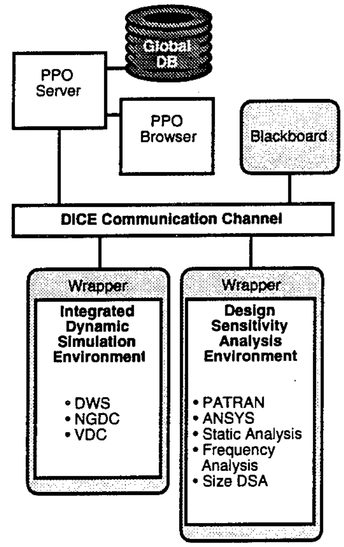
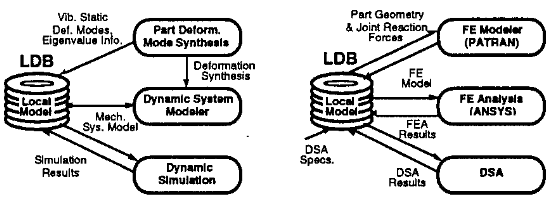


Figure 1.2 CCAD/CERC Pilot Project Environment Architecture

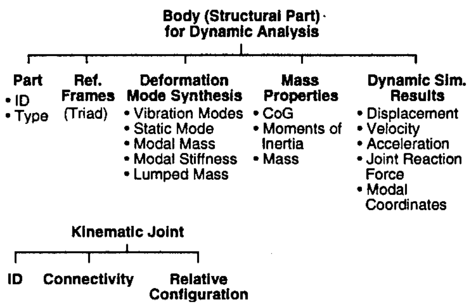
the workspace. Data retrieval within the workspace consists of simple call functions appended to the embedded analysis tools, with no data translation mechanisms required. It then follows that the local data model definition corresponds to the analysis data requirements of the embedded workspace tools. Figures 1.3 and 1.4 respectively illustrate the workspace configurations and local database model hierarchies defined for the dynamic simulation and structural DSA capabilities developed under the CCAD/CERC pilot project.[4]

Whereas data models for individual workspaces are defined in accordance with the application's narrow view of a mechanical system, in the integrated multidisciplinary environment the global data model requires fundamental and shared characteristics supporting all applications. The mechanical system global data model describes the system in terms of a high-level, physical design configuration, in such a manner that system characteristics can be derived for specific applications. In the global data model (see Figure 1.5(a)) a generalized mechanical system is composed of bodies, connectors, and subsystems. Given the application of dynamic and structural design sensitivity tools in the environment, a structural focus must be supported in the mechanical system data model as represented in Figure 1.5(b). In the global data model supporting these workspace applications, a structural part is defined as a solid entity having specific geometry, material properties, loading, and geometric boun-

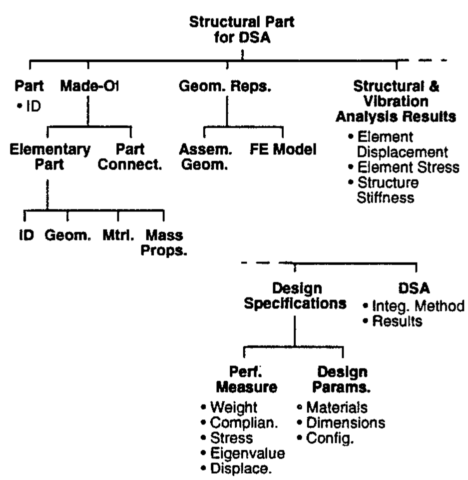


(a) Dynamic Workspace (b) DSA Workspace

Figure 1.3 Dynamic Simulation & Structural DSA Workspace Configurations



(a) Dynamics Local Data Model

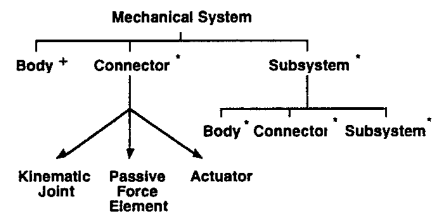


(b) Structural DSA Local Data Model

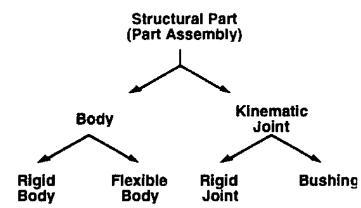
Figure 1.4 Dynamic Simulation & Structural DSA Local Data Models

dary conditions. The subset of the global data model for a structural part supporting the dynamic and structural DSA tools in this environment would then be as represented in Figure 1.5(c).

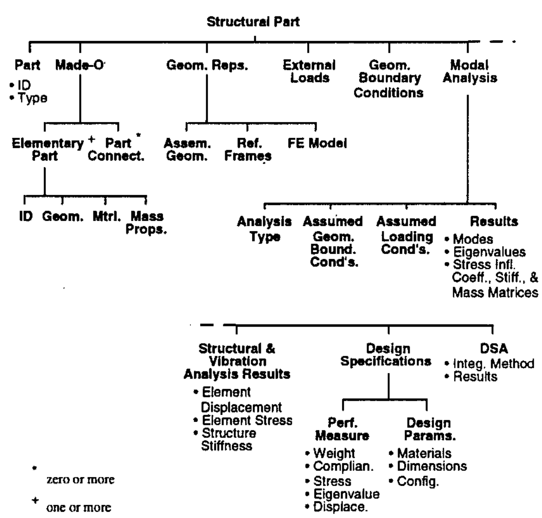
The data model presented in Figure 1.5 attempts to employ a hierarchy which encompasses product model schema typical of those found in the industrial engineering community. The model hierarchy provides for decomposition of the mechanical system from the system level to subsystem to part assembly to elementary part. However, as the model hierarchy is structured to support specific multibody dynamic and structural engineering analysis representations, at this stage of development, product decomposition tends toward grouping of subsystems, components, etc. by special mechanical function rather than by manufacturing-oriented schema.



(a) General Mechanical System Data Model



(b) Structural Part Subset



(c) Structural Part Global Data Model

Figure 1.5 Mechanical System Data Model

Having defined the structure for the global database for mechanical systems sufficient to support the data requirements of the engineering workspace applications, the means by which these applications access the database and achieve interoperability in an integrated context is afforded by the use of the "wrapper" concept. Wrappers are the set of software functionalities which enable multi-disciplinary simulation-based engineering analysis to be used to contribute design information to achieve global product optimization. [5] The principal function of wrappers is to provide bi-directional data access capabilities between any engineering application and the global database. Basic functionalities required for a nominal wrapper capability include (1) browse, preview, and select objects (data) of interest stored in the global database, (2) select objects to be transferred from the engineering analysis application to the global database, (3) translate data between the global database and the engineering application, (4) transfer data to specific locations within the application or the global database, and (5) invoke the engineering applications.

Figure 1.6 illustrates a simplified wrapped application configuration, adapted from CERC-developed information sharing and management concepts, [6] and applied to the integrated environment depicted in Figure 1.2. The functional wrapper modules are the PPO Access Utilities, the client side Communication Manager, and the User Interface. Together with the PPO server modules, this configuration satisfies all requirements defined in the preceding.

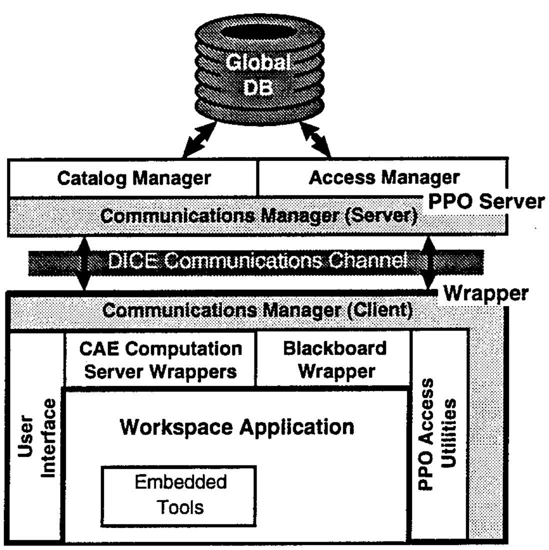


Figure 1.6 CCAD/CERC Pilot Project Wrapper/PPO Server Architecture

Wrapper PPO Access Utilities comprise the data translation and transfer tools and enable the workspace user to input, in a suitable format, design model data selected from the global database. These utilities are structured to present only that data meaningful to a specific application. The client side Communication Manager is the agent by which data transfer is effected. Finally, the User Interface provides the capability by which global database information is browsed and selected and by which the application is invoked. It should be noted that for specific wrapper modules, corresponding modules are defined for the PPO server. For example, PPO Access Utilities are defined for both client and server. Communications Managers are defined in this manner as well - using Remote Call Procedures, the client and server Communication Managers together form the functional basis for the DICE Communication Channel. In effect, the wrapper concept requires a well defined client-server architecture to support versionable design development; assuring the workspace application access to only the data it needs, and assuring design input to the global data model is controlled and appropriate for each application.

It should also be noted that the wrapper configuration in Figure 1.6 embeds CAE computational server and black board wrappers within the architecture of the workspace application wrapper. This configuration reflects more the limited engineering analysis applications employed to define the environment developed in this pilot project, where dynamic and FE analysis computational capabilities are uniquely employed in each workspace. In this environment, these computation server and blackboard wrappers serve more simply as function calls than a true wrapper configuration. As will be seen in the remainder of this report, as the design environment has evolved, wrapped computational servers will be defined as elements of the high level integration architecture, serving a number of workspace applications.

The global database modeling and wrappers concepts presented here provide the conceptual foundation for CCAD's DICE Phase 4, as well as additional CCAD efforts sponsored by other DoD and federal agencies. The remainder of this report presents how these and complementary collaboration concepts have been employed to develop a broad-based engineering environment supporting tracked and wheeled vehicle design.

II Objectives and Approach

Since its inception under the NSF Industry/University Cooperative Research program in 1987, the Center for Computer Aided Design (CCAD), The University of Iowa, has been at the forefront of multi-body dynamic simulation and structural analysis research. Center achievements in these fields include a number of stand-alone, computationally intensive software applications, supporting a broad base of engineering disciplines, including vehicle dynamic simulation and analysis, structural fatigue life prediction and reliability, mechanical system maintainability, and structural design sensitivity. In addition to workstation-based computational analysis capabilities, the Center has developed the Iowa Driving Simulator (IDS) to support operator evaluation of vehicle performance far in advance of construction of physical prototype hardware. The IDS, arguably the most advanced ground vehicle simulator in the world, provides a means by which critical engineering evaluation of dynamic and structural performance can be obtained from the end user as design input throughout the life cycle of the vehicle system. Under the Center's DICE and subsequent programs, the fundamental objective has been to bring these capabilities together to construct a networked, highly interactive tool environment, employing simulation technologies and Concurrent Engineering methodologies that are capable of supporting a distributed team of vehicle designers at an industrial level of application. Specific goals under each phase of the DICE program conducted at the Center since 1991 are outlined as follows.

DICE Phase 4

Having demonstrated the feasibility of developing a software architecture that is capable of supporting concurrent application of design and engineering analysis tools under the CERC/CCAD pilot project, the goal of the Center's DICE Phase 4 project was to develop, implement, and validate these technologies for an integrated environment specifically targeting military tracked vehicle engineering and analysis. Four basic objectives were addressed under this effort, as follows:

(1) Use DARPA DICE CE tool environment and database [technologies] to integrate and harden simulation based design software and operator-in-the-loop simulation tools for use in tracked combat vehicle CE.

(2) Implement a computer network to support industrial use and evaluation of the CE environment implemented at participating industrial sites and at The University of Iowa.

(3) Validate tools and methods developed and implemented with industrial applications.

(4) Deliver tested software and CAE tools to the... industrial user sites, TACOM, and CERC to serve as the foundation for continued refinement and application of CAE tools for tracked combat vehicle system development.

To accomplish the basic objective for development of an integrated software environment under this effort, the software design approach first postulated an elementary Concurrent Engineering scenario (see Figure 2.1) representative of a nominal tracked vehicle design and engineering analysis concept of operation. The scenario comprises the fundamentals of the design process and assumes the application of the Center's simulation based design capabilities to perform these activities. A degree of concurrency is achieved in this process in that engineering analysis activities can be performed more or less simultaneously with each engineering analysis contributing design information to the evolving vehicle design.

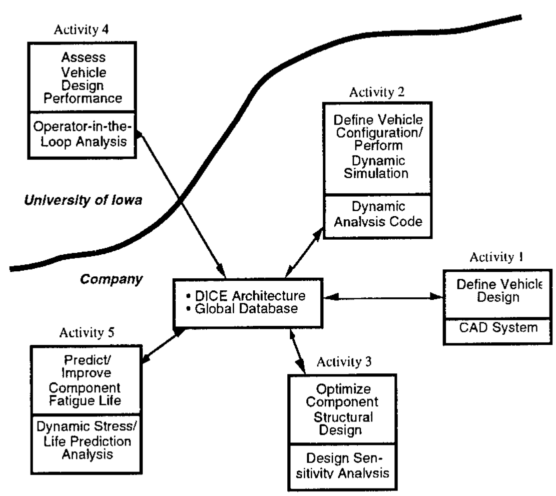
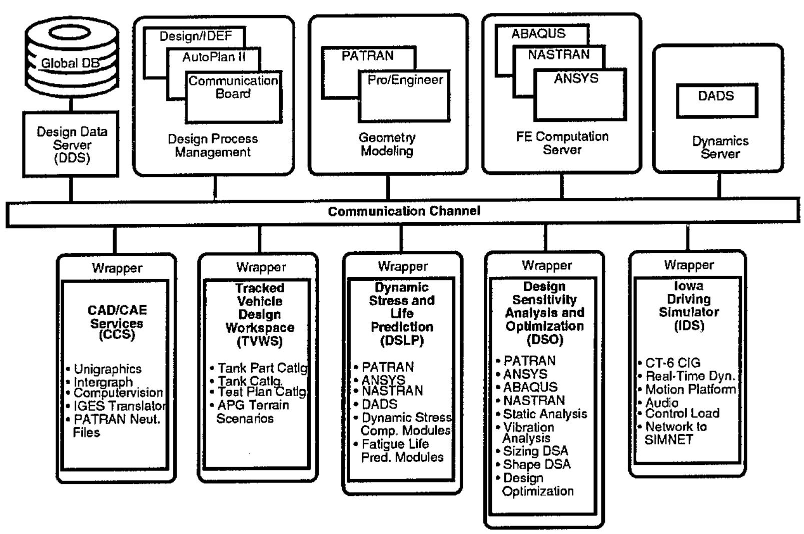


Figure 2.1 Tracked Vehicle Concurrent Engineering Scenario

In this scenario, a CAD system is used to specify the design of vehicle parts and assemblies (Activity 1). Geometry, assembly information, and mass property data, including mass moments of inertia, and center of gravity, are transferred to the global database. Dynamic simulation capabilities are used to generate tracked vehicle dynamic analysis models (Activity 2) and to launch standard dynamic simulation tests using the Dynamic Analysis and Design System (DADS). Dynamic load data is stored in the global database for use in downstream structural analysis. Dynamic analysis is also employed to determine a more effective vehicle configuration from a mobility perspective. Activities 3, 4, and 5 are essentially concurrent once Activities 1 and 2 have been completed. Vibrational and stress Design Sensitivity Analysis is performed and parametric component models are perturbed to determine optimal component design from a structural perspective (Activity 3). Activity 4 allows the vehicle operator to assess total vehicle performance as design change is implemented in the evolving design model. Component fatigue life prediction (Activity 5) is performed to predict structural failure of vehicle components based on how the operator drives the vehicle in Activity 4 and to verify design improvements from Activity 3. Fundamental interactions will occur among all the tool capabilities in the integrated vehicle system CE. For example, driver evaluation using the operator-in-the-loop simulation capability will establish bounds on speed and mobility of the combat vehicle system due to the soldier's interaction with the vehicle, from a mobility/maneuverability perspective. Availability of realistic vehicle operational data enhances the definition of realistic conditions used in off-line analysis carried out in the dynamics, design sensitivity, and fatigue life prediction capabilities. A high level of interaction is then enabled whereby each analysis discipline employs information generated in other disciplines in a continual process of evaluation, improvement, and verification of the evolving vehicle design.

Figure 2.2 illustrates the Tracked Vehicle Concurrent Engineering (TVCE) Environment developed to support the elementary tracked vehicle design scenario. The functional workspace elements of this environment comprise a suite of modeling, dynamic simulation and structural analysis software tools (see Section III for detailed descriptions) developed at the Center. The integration architecture for this environment follows the basic design of the DICE Architecture illustrated in Figure 1.1, with geometry modeling and FE and dynamics computation servers supporting more than one workspace relegated to the integration architecture. Simulation data sharing in this environment is supported through the application of the ROSE database system and a customized Design Data Server (DDS) that controls access between engineering workspaces and a versionable tracked vehicle data model.



- Developed/Implemented Under DICE Phase 4
- Proposed Under DICE Phase 4
- Developed/Implemented Under DICE Phase 5

Figure 2.2 Tracked Vehicle Concurrent Engineering Environment

Center DICE Phase 4 activities leading to the achievement of the integrated environment illustrated in Figure 2.2 centered predominantly on the development and implementation of database models and data sharing functionalities at both the workspace and environment levels, development and implementation of wrapper configurations and utilities for each workspace, and the development and refinement of network connection and communication functionalities supporting environment infrastructure requirements. Refinement of workspace computational analysis capabilities was performed as required to support industrial level design and analysis.

Validation of the utility and functionality of the integrated software environment depicted in Figure 2.2 in a realistic setting was of primary concern under this effort. As proposed, a computer network was implemented to enable distributed access to the software environment by industrial participants for the purpose of evaluating the software tools. Local area network connections were implemented at industrial sites to take advantage of existing hardware and software resources, minimizing the cost to contractors participating in the validation effort. Long-haul computer network connections were established between the Center and the industrial sites (see Figure 2.3) to enable portions of the environment that were unavailable at partner sites to be centrally hosted at The University of Iowa.

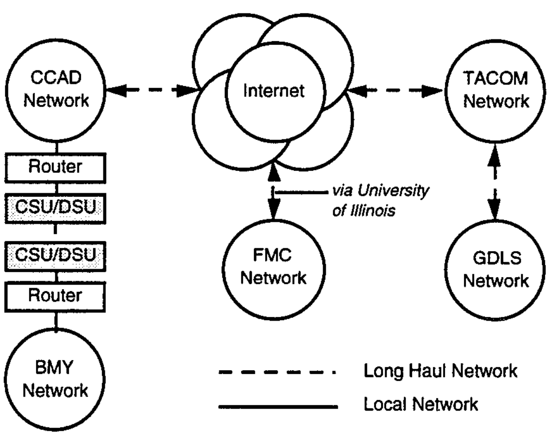


Figure 2.3 DICE Network Topology

The initial objective of establishing a network link between the University of Iowa and participating industrial sites had been originally intended as a means to expedite validation of the integrated environment software and to support industry personnel in the utilization of workspace capabilities. Implementation of a computer network under the DICE Phase 4 effort has served another important purpose, however. With the development and application of networking capabilities and methodologies under this effort, in terms of both distributed communications and software integration, conceptualization of the infrastructure requirements necessary for the formation of a "virtual enterprise" has in some measure been achieved. "Virtual enterprise" is a relatively recent concept proffered in the defense and industrial communities where teams of geographically distributed organizations or individuals can be virtually co-located to bring distinct areas of expertise together in product development efforts, in an organized, focused manner. As will be discussed in the conclusions to this report (Section VIII) the implementation of networked, integrated CE environments such as the one developed under this DICE efforts provides a solid foundation for effecting the realization of virtual enterprises.

Validation of the DICE Phase 4 integrated environment and engineering analysis workspace capabilities was achieved using realistic prototype examples representative of an industrial degree of problem solving. Both Center in-house and contractor specific test applications were developed for the validation effort. A generic test application, characteristic of an M1A1 main battle tank tracked vehicle system was employed in the Center's in-house validation exercise (see Section III) to develop a comprehensive reference for tool utilization for all three of the contractors participating in this effort. This generic application targeted the design definition of the M1 tank at the system level to support rigid-body off-line dynamic simulation. Dynamic load duty cycle data was generated for a selected track suspension component with structural fatigue life prediction and design sensitivity analyses performed for that component, culminating in a significantly improved component design.

Each of the three companies participating in this project, the former BMY Combat Systems (now United Defense LP Combat Systems Division), the former FMC Ground Systems Division (now United Defense LP Ground Systems Division), and General Dynamics Land Systems Division, has carried out a specific design and engineering analysis application using the integrated TVCE Environment as installed at their sites. Each of these applications is associated with an on-going tracked vehicle program/project. A non-proprietary discussion of the results of these application is given in Section III.

Interim to the development of the integrated tool environment and methodologies under DICE Phase 4 and the extension of integration technologies under DICE Phase 5 to incorporate techniques enhancing collaboration among users of the environment, additional workspace capabilities and a substantial restructuring of integration concepts were incorporated under a Defense Modeling and Simulation Office (DMSO) project conducted at the Center. The achievements under this effort, *Simulation Based Design for Military System Supportability and Human Factors*, extend the environment developed under DICE Phase 4 for simulation-based supportability analysis of wheeled vehicles and material handling equipment as well as tracked vehicles. A wrapped workspace module that supports maintainability simulation and analysis at the vehicle system level, maintenance task sequencing, and maintenance personnel and tool selection was added to the environment. Structural analysis tools in the TVCE environment were extended to support component reliability simulation and analysis, and operator-in-the-loop simulation integration was extended to support operability analysis of the vehicle system. As a result of the increasing complexity of the global database model schema, due to the introduction of new and extended workspace capabilities, a simplified global model and local product views were defined and taken into consideration in the development of collaboration methodologies and tools under DICE Phase 5. A detailed description of interim refinements to the integrated tool environment and model schema is presented in Section IV.

DICE Phase 5

The research effort carried out under the Center's DICE Phase 5 project proposed to enhance the integrated tool environment developed under the DICE Phase 4 and subsequent efforts to promote a higher degree of collaboration among users of the environment. Although the achievements under the DICE Phase 4 effort represent significant gains in focusing product development and decreasing the duration of the design process, the increased potential for exploration of a substantially larger number of design alternatives presents a real need for effective collaboration among designers and analysts employing simulation-based design technologies. Also, the experience gained by the Center team during the course of its environment development programs clearly indicates that the complexity of the tools, scale of data management, lack of interoperability of simulation and design tools, and lack of experience in the management of the large scale mechanical system concurrent product development process represent fundamental challenges to achieving DICE goals for broad classes of defense systems. To address these issues, the Center's DICE Phase 5 identified the following specific objectives:

(1) Implement, test, and refine collaboration methods in a large-scale mechanical system CE environment using DICE concepts and tools,..., suitable for a broad range of industrial applications.

(2) Define CE process and product models implemented by the Iowa team using DICE and related tools.

(3) Establish metrics quantifying attributes of concurrency in a large-scale, multi-disciplinary mechanical system environment.

(4) Define and implement formulation methodologies for design and analysis constraints applicable to the CE environment.

(5) Create a large-scale mechanical system CE testbed.

Collaboration, in the most basic interpretation of the concept, is the process of working together in a joint effort. Collaboration in the Concurrent Engineering context can be construed as product disciplines working together in a joint development effort to produce focused design concepts in the minimum time possible, with the minimum cost. Intrinsic to the requirement to work together is the need for each perspective in the product development effort to be aware of the activities to be performed, results to be communicated, the interaction (requirements of and obligations to other design perspectives) between activities, and to be assured that the concerns and requirements of that design perspective are being addressed in the development of the product design.

In the operation of the integrated simulation-based design environment, collaboration is a function of engineers, i.e. tool users, communicating and interacting in two forms, (1) quantitatively, through product model representations, and (2) qualitatively, through direct textual, graphical, verbal, and visual communications (see Figure 2.4). The terms quantitative and qualitative collaboration, in this context, loosely refer to the fundamental content expressed during these forms of communication. *Quantitative* collaboration refers to the information used to update the parameterized product model, typically a change in the numerical value of any design parameter. In addition, use of a parametric methodology facilitates formulation and employment of well-defined constraints expressed as equational relationships calculated using design parameter values. As such, a level of communication is identified that is highly quantitative in nature. Collaboration through product model representations supports the ICEE environment users in this quantitative aspect of product development, ultimately resulting in the specification of the optimized product design expressed in CAD form.

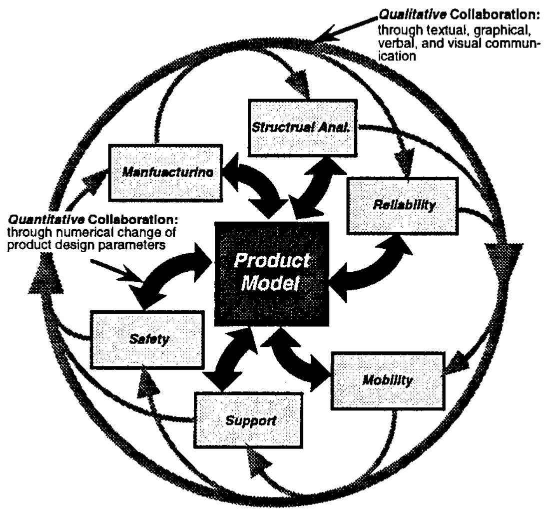


Figure 2.4 Collaboration in a Concurrent Engineering Environment

The *qualitative* form of design interaction, while potentially composed of a large amount of numerical data, represents a higher level of abstraction than update of the product model through changes in parameter values. This qualitative communication consists of the give-and-take interaction of the engineering team engaged in a discussion format, where ideas, suggestions, and possibilities are exchanged among the design disciplines. Users of the integrated tool environment require capabilities that support communication of design development rationale and effective design process management in order to focus the product design effort, implement the parametric product modeling methodology, and achieve maximum collaboration in the utilization of the integrated tool environment. These capabilities comprise textual, graphical, verbal, and visual communication utilities that aid the environment users in the capture and expression of product design objectives and development rationale. Communication capabilities supporting enhanced collaboration in this context currently exist in many forms, including teleconferencing, videoconferencing, e-mail, shared computer graphics, computer-aided design process planning, etc.

Given the above working concept of collaboration, the Center defined a two phased approach for achieving its DICE Phase 5 objectives. The first phase focused on the development and application of a parametric modeling and design change methodology supporting enhanced collaboration; extending the Integrated Concurrent Engineering Environment (ICEE) mechanical system product model to include appropriate mechanical system parameters. The approach taken also defined the extension of the Design Data Server (DDS) functions and ICEE data sharing capabilities to enable application of this parametric modeling scheme in the Concurrent Engineering design process and software environment. A parameterized mechanical system CAD product model was developed and implemented to provide a base product definition from which CAE analysis models are to be derived. In this manner, design changes suggested by CAE analysis are made in accordance with "allowable" design change as represented by the CAD product model parameters, thereby providing environment users with a structured method for product design trade-off and optimization.

With the establishment of the CAD-based parametric modeling scheme described in the preceding, the potential for application of feature based modeling standards, in particular, selected STEP standards, was also explored during this phase. Investigation of the STEP product specifications supported the determination of appropriate and meaningful data model requirements, given the current level of technology for standards-based design representation, for application of the parametric design change methodology in a distributed design enterprise employing diverse CAD modeling capabilities.

The second phase of the Center's DICE Phase 5 effort centered on the application of design process definition, dissemination, management and communication methods and capabilities to promote collaboration. This phase included the specification of a formalized CE process that employs the tools comprising the ICEE, the development of a software infrastructure that provides a capability to model, analyze for concurrency, disseminate to the ICEE users, and manage the CE process, and provide a means for the environment users performing the concurrent simulation and design process activities to communicate in a manner commensurate with achieving enhanced interaction and collaboration.

With the application of the parametric modeling methodology in the ICEE, formal definition of the design process becomes necessary to enable tool users to construct consistent, meaningful global-local vehicle model representations and employ the concurrent simulation-based design methodology and environment capabilities to their fullest extent. Based on the concurrent design scenario illustrated in Figure 2.1, the formalized process captures the utilization of the ICEE to effect Design Evaluation and Optimization at the detailed part/component level for existing large scale mechanical systems design. The generic process model developed under this effort captures the design project operations from vehicle requirements definition through design modeling, engineering analysis and design trade-off, for the tool environment developed at the Center and similar capabilities.

While capture of the simulation-based design process is an important step leading to enhanced collaboration, effective product development requires that designers and analysts, i.e., environment tool users, maintain awareness of their activities, responsibilities and their relationships with each other, in terms of data requirements and design change/suggestion information. It is also fundamental to effective product development that the process be managed in a manner that focuses the activities of the users to address the specific design issues at hand. The DICE Phase 5 effort therefore examined the operation of the ICEE capability as a team endeavor, and assumed the participation of a team leader or project manager responsible for defining design project objectives and coordinating team member activities. The basic approach was to define and implement tool capabilities supporting the team leader as an ICEE user. As a result, a variety of existing process modeling, project tracking and communication software tools were explored for application in the ICEE. Based on the Blackboard or Project Coordination Board functionalities conceived under the initial DICE architecture (Figure 1.1), an appropriate suite of software technologies was selected and incorporated into the Center's integrated environment (see Figure 2.2).

Section V details the parametric modeling, process, management methodologies, and tool capabilities developed under DICE Phase 5. A full-scale simulation-based design evaluation and optimization example application, using a US Army High Mobility Multi-Purpose Wheeled Vehicle (HMMWV), is also described, to demonstrate process flow and parametric modeling techniques. In fulfillment of testbed objectives for both DICE Phase 4 and Phase 5, Section VI provides a complete description of the hardware and software environment comprising the Center's ICEE testbed. The testbed incorporates all engineering analysis workspace and integration architecture capabilities developed to date at the Center under DICE and other programs.

Finally, although the Center's efforts to date represent a significant achievement in the field of concurrent simulation-based design, much work remains to be accomplished to attain seamless operation of integrated tool capabilities in the distributed product development enterprise. Center personnel are continuing research in multi-disciplinary computational design trade-off methodologies and the application of design and analysis model standards and data transfer schema. Section VII presents brief descriptions of two ongoing research projects in these areas.

III DICE Phase 4: Tracked Vehicle Tool Development, Integration, and Validation

The Tracked Vehicle Concurrent Engineering (TVCE) environment developed under the DICE Phase 4 effort consists of a series of four modeling and computational workspace tools, integrated by means of a high level architecture comprising the ROSE object-oriented database system, a Center developed database server/browser, the DICE communications channel, and workspace wrappers (see Figure 2.2). A suite of commercially available geometry modeling and computation servers are also interfaced with the integration architecture using basic wrapper constructs. Of the four workspace applications implemented in the TVCE environment, three provide functional engineering level simulation capabilities in dynamics and structural performance analysis and design. These are the Tracked Vehicle Workspace (TVWS), developed in partnership with the US Army Tank Automotive Command to enable rapid tracked vehicle system design configuration definition and dynamic analysis at the journeyman level, the Dynamic Stress and Life Prediction (DSLP) workspace, a flexible, general purpose, integrated CAE environment for structural fatigue life estimation of complex mechanical subsystems and components, and the Design Sensitivity Analysis and Optimization (DSO) workspace, a continuum-based sensitivity analysis capability that enables engineers to determine the best direction for design change from a structural stress distribution perspective. A fourth workspace capability, the CAD/CAE Services (CCS), was developed during the course of the DICE Phase 4 effort to provide modeling and model translation support between the multiple CAD modeling systems implemented in the TVCE environment and the CAE simulation-based design tools. Also proposed under the DICE Phase 4 effort, initial real-time and off-line dynamics model transformation have been developed in support of the incorporation of the Iowa Driving Simulator in anticipation of long term virtual prototype capability development.

This section provides a brief conceptual overview of each of the simulation and model support workspace capabilities employed in the TVCE environment, and continues with a general discussion of the initial real-time/off-line dynamic model translation developments that anticipate virtual prototyping using the Iowa Driving Simulator. An in-depth discussion of the tool integration architecture including database modeling, database server functionalities, and workspace wrapper utilities is provided. DICE Phase

4 development discussion in this section concludes with detailed analysis of the results of validation exercises performed under this effort, including both the generic M1A1 Abrams application performed at the Center and the three contractor applications.

Computer Aided Engineering Workspace Capabilities

CAD/CAE Services/IGES Translator

Although its development was not envisioned at the outset of the Center's DICE effort, the need for a capability to provide TVCE environment users with modeling and model translation support was quickly recognized during this project. The DICE methodology as implemented in the TVCE environment is based on the application of simulation-based analysis to CAD design models, under the assumption that CAD provides the fundamental design model representation in any product development enterprise. As such, the Center's DICE paradigm targets the derivation of CAE analysis models from CAD solid geometry representations in an effort to maintain consistency and correlation between design and analysis operations. This method is in direct response to a heretofore common problem in the application of CAE analysis tools in a multi-disciplinary design environment - each discipline views the system under development in a manner peculiar to that discipline, resulting in as many models as there are perspectives, with little or no commonality or correlation with the basic CAD design. The use of a common CAD global design representation exhibits significant potential to counter this problem, particularly with respect to feature-based parametric design and design change propagation. Additional design information, such as mechanical system body connections must be appended to the CAD model to satisfy the data requirements of the CAE analysis tools, however. Mechanisms supporting model translation between multiple CAD systems, such as IGES formatting, and between CAD and CAE, such as PATRAN neutral files, are also required for effective derivation of consistent model representations between analysis disciplines. CAD/CAE Services has been developed to perform these functions.

The Computer Aided Design/Computer Aided Engineering Services (CCS) workspace provides an environment for a multi-disciplinary design team to process CAD data for CAE applications. Within this environment, CAD data is entered, translated, or supplemented with essential information to create CAE models for downstream CAE activities. Based on the designers' CAD model, a CAD system can be launched within CCS to calculate mass properties and translate the CAD geometry into IGES or PATRAN neutral files. Engineering analysts can also employ this environment to create CAE models, ranging in scale from a complete mechanical system, to individual parts. For example, a structural engineer can create a finite element model of a fundamental part, and a dynamic simulation analyst can define all elements of the mechanical system, including bodies, joint/force connections, and assembly information.

Using CCS, design team members can:

- import existing or create new CAD design models.
- translate CAD geometry output files so that PATRAN can use then to reliably generate finite element models.
- translate CAD geometry output files to an animation format for dynamic simulation.
- specify information essential to dynamic analysis, such as mass properties, connection types, and assembly information.
- export CCS-generated data files to the global database via the Design Data Server (DDS).

Figure 3.1 illustrates the basic operation and information flow for the CCS workspace. CCS has four primary functions: (1) manage a local database using an embedded model catalog, (2) specify essential information to carry out dynamic and structural analysis, (3) launch commercial CAD packages, a finite element modeler, and a Center-developed IGES translator, and (4) communicate with the global design data server.

The model catalog provides several options for managing model information. Engineers can use it to create a new model under a specified model catalog name; they can also open, modify, rename, close, and delete an existing model. The catalog also displays a model hierarchy for composite models.

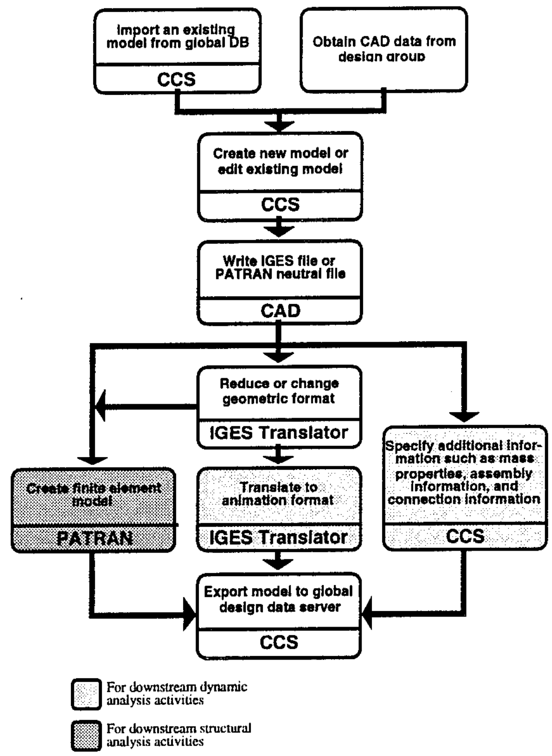


Figure 3.1 CCS Data Flow

Information essential to the mechanical system CAE models that can be specified using CCS includes:

- **Mass properties** - mass, moments of inertia, and center of gravity for parts and assemblies (bodies).
- **Assembly information** - assembly information among member models of a composite model can be given by specifying a set of PQR coordinate systems associated with each member model.
- **Connection information** - connection information among member models of a mechanical system or subsystem can be given by specifying the connection type (revolute joint, translation joint, etc.) and a set of PQR coordinate systems associated with each member model.

A number of utilities have been implemented in the CCS workspace to process CAD data. These include:

- **Mass property calculation** - This option launches the CAD system to read an existing geometry data file; functions provided by the CAD system are used to calculate mass, moments of inertia, and center of gravity for part and assembly models for a given density value.

- **CAD/IGES geometry data translation** - This option launches the CAD system to read an existing geometry data file and convert CAD geometry into an IGES format file.

- **CAD/PATRAN neutral file translation** - Depending on the CAD system used, this option launches the CAD system to read an existing geometry data file and convert the CAD geometry to a PATRAN neutral file.

- **Updated IGES file generation** - The CCAD-developed IGES translator reads IGES format geometry and converts the representation into a format that can be interpreted correctly by the PATRAN modeling software.

- **Animation format geometry translation** - The IGES translator reads IGES format geometry files and converts them to .mod format files for use in animation capabilities.

- **PATRAN finite element model generation** - This option allows the user to create a finite element model from an updated IGES file or from the neutral file translated from the CAD system. This FE model contains the solid model, finite element mesh, boundary conditions, and material property data for all structural analyses.

The fourth function performed by CCS provides the engineer with data export and import capabilities to and from the global database. Together this series of functionalities enables CAD and CAE data to be defined, created, and stored in a consistent and organized manner, providing a hierarchical model structure for all data and files. The CCS workspace benefits engineers employing subsequent simulation workspace capabilities by defining what data is required for a particular application, by acting as an interface with other software packages, by controlling the working directories whenever a software tool is invoked, and by providing a communication channel to the global database.

A significant drawback to CCS operation in the CE context exists, however, in that a sole user, acting as a "database populist," is assumed in the design team organization. This team member would need considerable knowledge of both CAD and CAE modeling techniques, sufficient to support downstream analysis operations, to effectively employ the CCS capability. The need for such a broadly experienced user may be awkward in the concurrent context, where the intent is to enable designers and engineers to interact directly without the need for intermediate support. As will be seen in Section IV, the application of CCAD's model view concept resolves this drawback by enabling transparent CAD to CAE model derivation by simulation and analysis engineers, with CCS functionalities absorbed into DDS and wrapper utilities.

IGES Translator

As can be perceived in the preceding overview of CCS capabilities, the Center's IGES translator serves an important role in facilitating CAE model derivation from a CAD design. While strictly an embedded CCS utility, an overview of this capability is discussed separately here, to address both the development of this capability as a Center accomplishment under the DICE effort, and establish a precedence for the implementation of CAD data exchange (STEP) standards in the integrated simulation-based design environment to be discussed in Section VII of this report.

Many graphical design systems, such as Unigraphics, ComputerVision, etc., create output files in the well-defined Initial Graphics Exchange Specification (IGES) format. These output files often contain information that is either unnecessary or incompatible with software packages that are used for subsequent design simulation and analysis. This information may be in the form of notations such as dimension lines or occur as unsuitable geometrical representations. The IGES Translator provides engineers with a means of converting graphics files developed with IGES-based CAD systems into files suitable for use with CAE simulation and analysis software packages. Translation into animation format types is also possible.

Basic IGES Translator capabilities include the following:

- **Format Translation** - Users can translate IGES formatted files into files written in different format types for use in animation and other design analysis work. Formats than can be generated include, ".mod", Movie.BYU, and Psurf.

- **Geometry Verification** - During the translation procedure, the geometry of an object can be viewed on-screen from any location on a "viewing sphere" which the user may place completely around the object or localize at a particular feature of the object.

- **Representational Changes** - If the representation of a geometrical entity used in an IGES file is incompatible as input for a simulation or analysis package, this representation can be converted into one more suitable. For example, the requirements of some finite element modelers dictate the translation of parameterized geometric entities into discrete polygons before the file can be used as input, otherwise inaccurate results may be generated.

- **IGES File Reduction** - This capability allows the interactive removal of notations, dimension lines, and other non-essential information from existing IGES formatted graphics files. This enables the engineer to clean up the on-screen image of a design, making it easier to perform other Translator procedures.

The IGES Translator recognizes all commonly used IGES entities. Non-supported entities are ignored by the Translator. Table 3.1 provides a complete list of the entities supported by the IGES Translator.

Table 3.1 IGES Translator Entities

Circular Arc	Transformation
Composite Curve	NURBS Curve
Conic Arc	NURBS Surface
Copious Data	Nodal Point
Plane	Finite Element
Line	Curve on Parametric Surf
Parametric Spline Curve	Trimmed Parametric Surf
Parametric Spline Surf	Subfigure Definition
Point	Associativity
Ruled Surface	Single Instance
Surface of Revolution	Rectangular Array
Tabulated Cylinder	Circular Array

Tracked Vehicle Workspace

The Tracked Vehicle Workspace (TVWS) is the CAE analysis capability supporting dynamic simulation and tracked vehicle configuration design in the TVCE. Developed under sponsorship from the US Army Tank Automotive Command (TACOM), the TVWS is designed to support system definition and performance assessment at the journeyman engineer level. [7,8] The TVWS capability employs CAD design data imported form the global database to quickly generate dynamic simulation models. Simulation model definition in the TVWS has been constructed around the use of subsystem modules, or templates, that enable the user to develop high-fidelity models without the need for arduous computational formulation. Simulation and analysis of these models is accomplished using a library of test scenarios and road profile input data to computational dynamics servers, such as the Dynamic Analysis and Design System (DADS) and the NATO Reference Mobility Model (NRMM). The TVWS capability supports simulation engineers responsible for vehicle evaluation from dynamics, mobility, and maneuverability perspectives. TVWS analysis output is also the primary source for system, subsystem, and component load, position, velocity, and acceleration data required by the structural engineering analysis disciplines in the TVCE environment.

A conceptual diagram of the TVWS system is given in Figure 3.2. The TVWS consists of three basic modules, the communications link, the local database and object manager, and the vehicle performance evaluation control, supported by a number of utilities. The principal functions of the TVWS capability are:

- Database storage of mechanical system and related data.

- Graphical database viewing, database table editing, ASCII text editing, 2D plotting, and 3D animation.

- Mechanical system model assembly using templates.

- Simulation parameter and conditions selection.

- Remote dynamics analysis run launch.

- Retrieval and storage of dynamic analysis results.

TVWS communications links, i.e. wrapper functions, are used to import mechanical system data, generated by CAD/CAE Services, from the global database. Geometry dimension, mass property, and moment of inertia data are transferred to the local database for development of the dynamic system model.

The TVWS system database consists of two major elements, a commercial relational database (Informix), and a large-object storage system (DAFS). These database elements store dynamic analysis related vehicle data, specifically CAD file information, simulation inputs and results, organizational structures, and var-

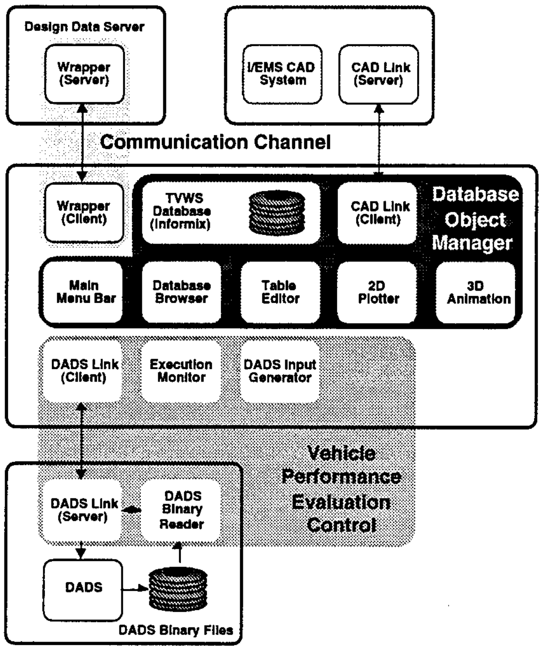


Figure 3.2 TVWS System Diagram

ious other types of information. The TVWS database features catalogs of customized tracked vehicle parts, test, and test scenarios, as well as part templates, which can be used to import new part design into the TVWS database for incorporation into appropriate evaluation simulations. The TVWS database also enables relationships to be created between objects in the database; a capability employed to define a complete mechanical system simulation in terms of vehicle model, terrain profile, initial conditions, etc.

The TVWS is used to assemble vehicle models and test plans for dynamic simulation, to remotely launch dynamic simulations using the DADS computational server, retrieve DADS results, and view results using 2D plot and 3D animation utilities. Dynamic model assembly is accomplished by importing vehicle component information from the global DDS and selecting a test plan, test scenario, road profile, and suspension system from the TVWS database. Once the tracked vehicle system is assembled and the desired test plan is defined, the simulation engineer can select bodies of interest for dynamic simulation. Dynamic analysis incorporates commercial and TACOM-developed DADS super-element modifications for steering and other vehicle motion control.

TVWS employs the DADS input generator to produce DADS pre- and post-processor command files. Pre-processor command file generation functionalities automatically update the vehicle dynamics model to incorporate user-defined changes in template files. DADS post-processor and DADS link functionalities enable the engineer to select specific body dynamic analysis results to transfer to the TVWS database, including load, position, velocity, and acceleration for that body. Simulation results are viewed using 2D plot and 3D animation utilities to verify accurate dynamic performance. Load history and related dynamic results are exported to the global database for use in downstream structural analysis and design.

Dynamic Stress and Life Prediction

The Dynamic Stress and Life Prediction (DSLP) workspace is a flexible, general purpose, integrated CAE tool for the prediction of the fatigue life of complex mechanical systems subject to dynamic stress, as a function a fatigue crack initiation and propagation. DSLP employs a number of CAE analysis tools to aid engineers in the analysis of components in multi-body mechanical system design. Comparable with the DICE architecture, the DSLP system incorporates a layer of network computational services for reliable remote computations and data transfers between an engineering workstation and a computation server.

Fatigue is a complex process that causes premature failure or damage to mechanical components that are subjected to repeated loading. The definition of a *fatigue failure* is dictated by the design philosophy employed or functional requirements of mechanical components. To formulate this definition it is necessary to determine the significance of a crack in the component both in terms of safety and reliability. The goal of designing a mechanical system is to adjust the structure so that all components of the system can withstand a predetermined period of dynamic loading. Dynamic stress and fatigue life (the length of time before failure occurs) in components must therefore be predicted in the design process in order to satisfy performance and reliability requirements and guarantee that loads imposed on the system are safely supported.

DSLP computational methodologies consider both rigid and flexible body dynamic analysis. For a mechanical system in which all components are considered to be rigid bodies, no deformation modes are used for simulation. The rigid body simulation is performed to solve the equations of motion using DADS to execute dynamic analysis for an input file provided directly by the user. For a mechanical system wherein elastic deformation is considered in at least one component, DSLP employs a flexible body dynamic simulation capability based on *modal* synthesis for component analysis. In contrast to nodal coordinates, which uniquely locate every point in the system, component modal synthesis represents a complex mechanical system wherein the motion of each component is represented by a set of component modes. The use of modal synthesis results in a reduced number of degrees of freedom from the use of nodal coordinates, enabling the achievement of computational efficiency for realistic problems. Component modes constrain the assemblage of components to act together; vibration and static modes may be introduced to insure that various components act as part of the system rather than independently. Three types of component modes, vibration normal, constraint, and attachment, can be used to represent deformation in the flexible mechanical system. In the DSLP computational methodology, vibration normal modes and static deformation modes are combined to best represent deformation of flexible mechanical systems. The DADS system provides an intermediate processor to employ this combination of modes in the computation of dynamic stress for flexible systems. [9]

Based on a hybrid quasi-static method, the DSLP dynamic stress computational approach employs a new algorithm using the stress field due to inertial and joint reaction forces applied distributively to each node of a finite element model to calculate dynamic stress time histories. [10] A finite element analysis numerically produces stress fields of the component with stress influence matrices using the space-dependent portion of inertia forces and joint reaction force and/or externally concentrated force unit load. Dynamic stress time histories are then calculated by elastic superposition using stress influence matrices, the time-dependent portion of the inertia force, and the real dynamic loads, obtained from the results of TVWS dynamic simulation. Since the number of variables that are involved in the nodal acceleration expression is much less that the number of nodes in the finite element model, the number of stress fields needed to be computed can be significantly reduced. Furthermore, the number of superpositions at every time step can also be substantially reduced.

Currently, there are three main divisions in the field of fatigue life prediction, each focusing on the type of behavior one is interested in predicting: crack initiation, crack propagation, and total life. The crack initiation prediction models are used to predict the onset of a visible crack, while the crack propagation models are used to predict the growth of a crack to a certain size. Total life prediction models are used to predict the life of the component to final fracture.

The local strain-life technique [11] is used to predict fatigue crack initiation, and failure is said to occur when a crack has grown to approximately 2 mm, under repeated application of the load block (stress time history). Forman's equation [12] is used to predict the fatigue crack propagation life, resulting in crack growth rate and the crack length. The basic computational procedure for fatigue failure prediction can be broken down into four components: (1) peak and valley screening of the elastic principal stress histories, (2) true stress and true strain computation, (3) rainflow counting technique, and (4) life prediction - crack initiation and/or propagation.

The principal input data for fatigue failure estimation is in the form of the nodal stress tensors obtained from dynamic stress computation for the component. The first and second elastic principal stress histories are calculated from these stress tensors. Both stress histories are screened to capture the peak and valley values. A peak value is defined as a local maximum value in the neighborhood of one position to either side of the current location in the history, whereas a valley is defined as a local minimum. A rainflow counting algorithm is used in this segment of the computational method and assumes an even number of reversals in a block of true strain data to insure a continuous computational loop.

Using the stress magnitudes in both principal stresses, a Hoffman-Seeger biaxial factor is computed for each pair. The local stress and strain magnitudes are computed by simultaneously solving Neuber's rule and the nonlinear stress-strain relationship using the Newton-Raphson numerical method. During the computation, necessary material properties are automatically called up from the material property data library.

The constant amplitude fatigue damage curve represents a set of tests on standard specimens under controlled conditions. Considering the Palmgren-Miners linear damage rule, operation at a certain amplitude will result in failure in say N cycles. Operation at the same amplitude for a number of cycles less than N will result in a smaller fraction of damage which is often referred to as a partial damage. Operation over a spectrum of different levels, results in a partial damage contribution D from each cycle. Failure is then predicted when the sum of these partial damage fractions reaches unity.

For fatigue crack propagation, linear elastic fracture mechanics [13] has been used to predict the propagation of a crack through a component to the point where rapid fracture causes ultimate failure. This is known as the crack growth approach and is used to deal with a situation where cracks are known to exist, but may be tolerated. The fatigue crack propagation analysis uses the FLAGRO software developed by NASA. [14] The iterative technique is used for solving several equations in crack propagation prediction, such as the modified Forman equation constants, C, n, from crack growth rate data.

The DSLP tool environment structure corresponds to the architectural configuration of the TVCE environment as a whole, comprising CAE tools, a data server and local database, and subworkspaces for executing high-level engineering activities (see Figure 3.3). Most DSLP workspace CAE tools, such as PATRAN, ANSYS, NASTRAN, and DADS are employed in other workspace capabilities, and are relegated to the TVCE integration architecture.

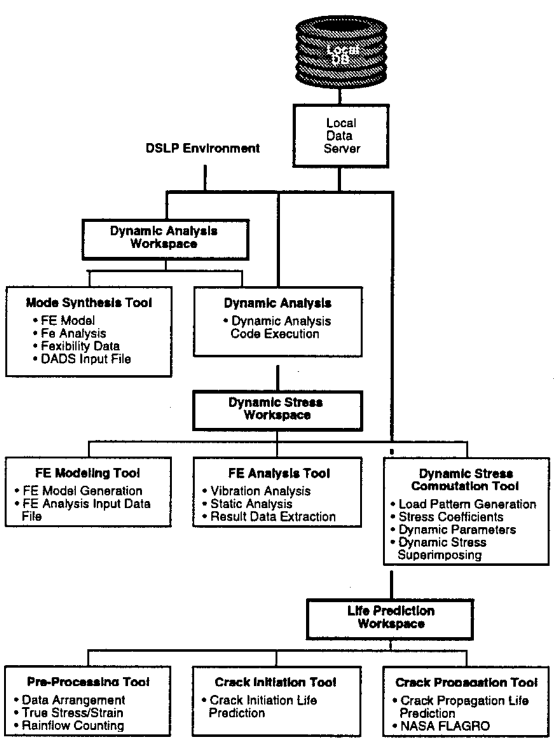


Figure 3.3 DSLP Architecture

From the structural mechanical engineering perspective, the engineering modules are organized into three subworkspaces in DSLP. These are the dynamic analysis subworkspace, the stress computation subworkspace, and the life prediction subworkspace. The dynamic analysis subworkspace executes the dynamic simulation for the mechanical system. The stress computation subworkspace calculates the dynamic stress histories of critical (or interesting) points on a structural component. The life prediction subworkspace predicts fatigue crack initiation and propagation life. Each subworkspace groups CAE tools together to perform different engineering activities.

The DSLP analysis process consists of four principal segments: (1) finite element analysis, (2) dynamic analysis, (3) dynamic stress computation, and (4) fatigue life prediction. Correlated to the system architecture depicted in Figure 3.3, these four analysis segments implement the three subworkspaces as illustrated in the detailed process model in Figure 3.4.

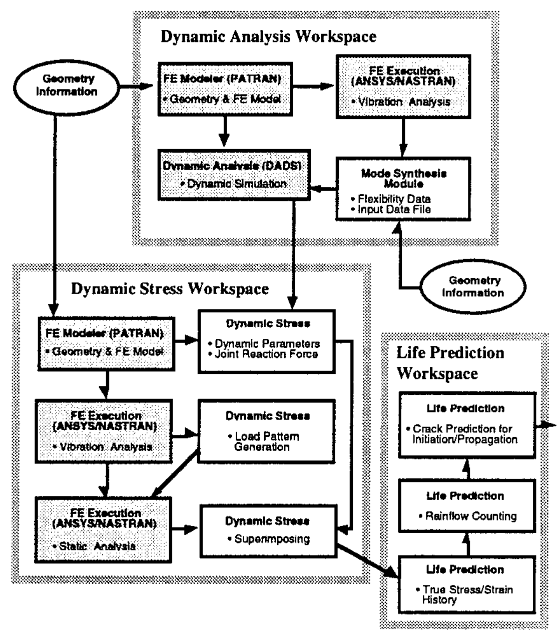
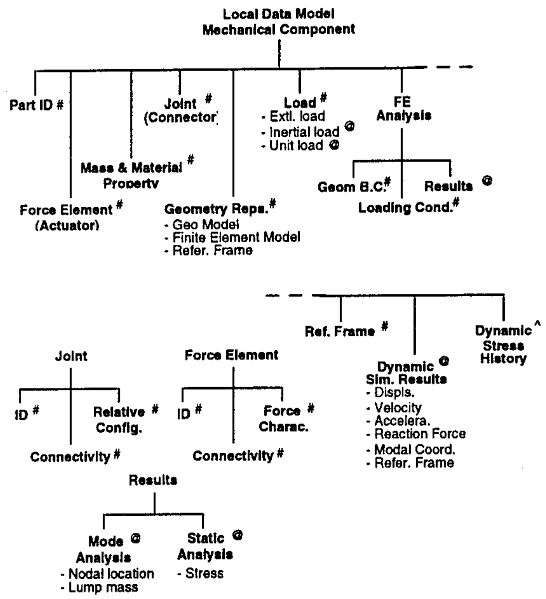
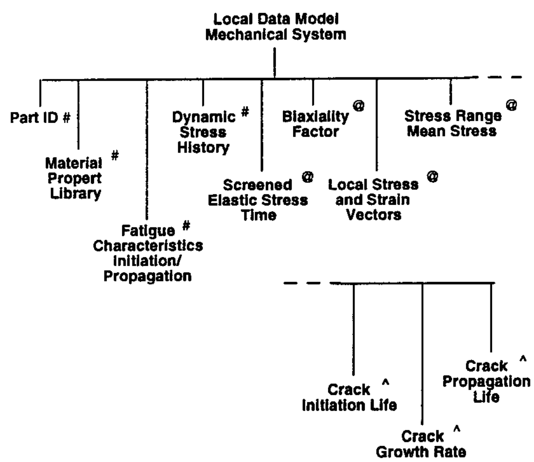


Figure 3.4 DSLP Analysis Process

Due to the amount and types of information, DSLP make use of a host file system. Local data management within the DSLP workspace is supported through the application of appropriate data model schema to the local database for component dynamic stress and component fatigue life. The hierarchy for these data models is presented in Figure 3.5.



(a) Component Dynamic Stress Local Data Model



(b) Component Fatigue Life Local Data Model

Figure 3.5 DSLP Local Data Models

Design Sensitivity Analysis and Optimization

In the structural design community at large, the design optimization process typically tends to be treated as a black box. Current methodologies applying available optimization tools for structural design require the designer to create a finite element model based on the physical model, prepare an ASCII input data file for finite element analysis (FEA), parameterize the FE model, and send the parameterized FE model to the optimization tools to perform design optimization (OPT). The current methods by which design parameters are defined in the finite element model are counterintuitive, and may lead to unacceptable design, especially for shape design applications. [15] Moreover, the designer has few opportunities to interact with the optimization process. As well, the optimization process is subject to numerous failures due to problems ranging from errors in input files to design convergence to a poor local minimum. Thus, only experienced design engineers are able to achieve optimum design in a cost effective manner.

The Design Sensitivity Analysis and Optimization (DSO) capability provides the journeyman design engineer with an interactive, geometry-based sensitivity analysis and optimization tool for general sizing and shape design applications that:

- Employs a systematic design procedure for general design applications.
- Utilizes a computer-aided design (CAD) modeler to generate geometric and finite element models.
- Parameterizes the geometric model by assigning geometric quantities as design parameters.
- Integrates dedicated commercial codes to model, analyze, and improve designs.
- Combines effective interactive design methods with visualization of important design information to efficiently obtain improved or optimum designs.
- Maximizes the computer's computational and graphical power to achieve fast turnaround to support interactive design processes in a graphical design environment.
- Provides a data management system for efficient and unified data access.
- Implements a menu-driven user interface to provide a convenient and easy-to-use design tool.

The sensitivity analysis and optimization methodology employed using the DSO capability is carried out in three stages: pre-processing, design sensitivity computation, and post processing.

Pre-Processing

The major objective in the pre-processing design stage is to formulate the design problem by creating a design model (geometry and finite element), parameterize the model, carry out finite element analysis, error analysis, and mesh adaptation, and define performance measures. The PDA Engineering software package, PATRAN, is used for geometric modeling and FE mesh development. Line, patch, and hyperpatch geometric entities are used for modeling line, surface, and solid design entities, respectively. Line elements, such as truss and beam, surface elements, such as membrane and plate, and solid elements are employed for FEA. Material properties as well as boundary conditions are also defined using PATRAN.

Design parameterization is a key step in the structural design optimization process. The purpose of design parameterization is to define parameters to characterize the section properties of the geometric entities for sizing design applications or to characterize the movements of geometric control points that govern the shape of the structural boundary. The designer collects a subset of these parameters as design parameters that he allows to vary in the design process in order to improve structural performance. Design parameters must be defined based on both design and manufacturing considerations. For sizing design applications, the geometric and finite element modeling cannot be completed until the design parameters are defined, since element section properties in the analysis model must be consistent with the design model.

Sizing Design Parameterization

The DSO supports constant and linear design parameterizations, as shown in Figure 3.6. Geometric parameters are defined at end grid points of a line or at corner points of a patch. Bilinear thickness distribution can be used to characterize a surface design entity, as shown in Figure 3.6. Note that each dimension that defines the cross-sectional shape in Figure 3.6, such as width or height, could be treated as a design parameter, and be allowed to vary in the same amount as the corresponding parameter at the other end (constant parameterization), or in different amounts (linear parameterization). Moreover, through design parameter linking, design parameters can vary independently of or proportionally to certain parameters across design entities, to maintain design continuity for symmetric design, or to reduce the number of design parameters.

Shape Design Parameterization

The shape design parameterization method developed in the DSO parameterizes geometric features. A geo-

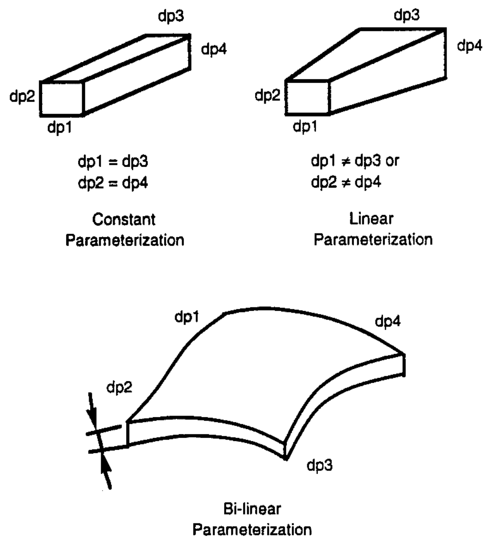


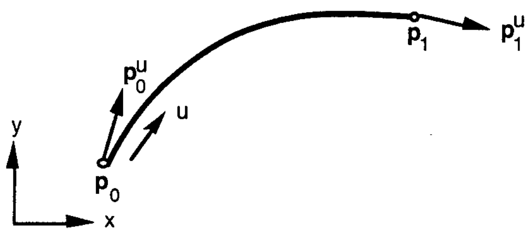
Figure 3.6 Line and Surface Parameterization

metric feature is a subset of the geometric boundaries of a structural component. For example, a fillet or a circular hole is a geometric feature that has certain characteristics associated with it and may be chosen to perturb the design. A geometric feature with design parameters defined is a parameterized geometric feature and is treated as a single entity in the shape design process. For example, a circular hole, with the radius and location of its center defined as design parameters, is a parameterized geometric feature. In accordance with design changes, the parameterized circular hole can be moved around in the structure, and its size can be varied. However, the shape of the circular hole is retained.

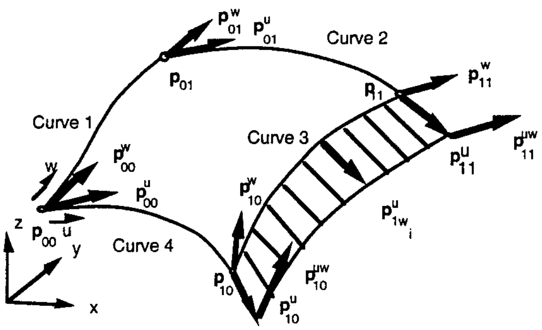
A three-step shape design parameterization procedure has been developed in the DSO. The first step is to create a geometric feature by grouping a number of inter-connected geometric entities and defining the type of the geometric feature. The second step is to define design parameters within each geometric feature. To generate a parameterized geometric feature, the designer can use the design parameter definition within the geometric entities and link design parameters across the entities. The third step is to link design parameters across parameterized geometric features, if necessary.

Shape design parameterization is defined within geometric entities, and parameterized geometric features are created using the geometric entities. The shape design parameterization method developed for the

DSO uses the geometry representation defined in PATRAN. In PATRAN, all geometric entities are represented using parametric cubic (PC) lines, patches (surfaces), and hyperpatches (solids). For 2-D structural shape design, the design boundaries are planar curves (see Figure 3.7(a)); the DSO supports parameterization for three curves: geometric, four-point, and Bezier. For 3-D structural shape design, the design boundaries are surfaces in space (Figure 3.7(b)); in the DSO, geometric, 16-point, and Bezier surfaces are supported for 3-D shape design parameterization. [16]



(a) Geometric Curve



(b) Geometric Surface

Figure 3.7 Curve and Surface Parameterization in DSO

Three major commercial FEA codes, ANSYS, MSC/NASTRAN, and ABAQUS, are integrated into the DSO to analyze the design model. The integration uses the DSO database, the PATRAN-FE interface, a unified design parameterization method, a finite element model update method, and finite element interfaces. The data source in the DSO is a PATRAN neutral file, which contains geometric and finite element model definitions of the structure being designed. Using one of the PATRAN-FE interface programs provided by PATRAN, [17-19] the PATRAN model can be translated to an analysis model. For sizing design, finite element section properties are computed, based on the design parameter values defined during the design parameterization. These section properties are stored in the DSO data tables and utilized to update the analysis input data files for each analysis code. After the design model is analyzed, the finite element interface programs developed in the DSO are executed to retrieve node responses, displacements, and stresses from the database of the analysis codes. Analysis responses at integration points are then interpolated using node responses to support numerical integration for sensitivity computation. [20]

To assure accuracy of the analysis model during shape design applications, DSO incorporates finite element error analysis and mesh adaptation. The Simple Error estimator developed by Zienkiewicz [21-22] is used for finite element error analysis. An interactive mesh adaptation algorithm [23] uses error information as a criterion to adjust element size. PATRAN's meshing capabilities are used to interactively refine the mesh.

The DSO supports seven types of performance measures: mass, volume, displacement, stress, compliance, frequency, and buckling. Among these, mass, volume, compliance, frequency, and buckling are global measures for the whole structure. Displacement and stress, however, are defined at specific points or elements in the structure and are considered local measures. Displacement performance measures can be defined by selecting nodes, degrees of freedom, and load cases.

Stress performance measures can be defined at Gauss points or averaged in an element. Also, for each loading case, stress measures are defined using material failure criteria, such as von Mises, maximum shear, or maximum or minimum principal stresses.

Structural performance measures are combined to define cost and constraint functions to set up the design optimization problem. The cost function, constraint functions with bounds, and design parameters with bounds form a design optimization problem that can be formulated for trade-off determination and design optimization.

Design Sensitivity Computation

The second stage of the design optimization process is design sensitivity computation. The sensitivity computation in the DSO capability employs the continuum DSA method, which is more efficient, accurate, and general that the finite difference method. [24] The design sensitivity coefficient matrix is computed for performance measures with respect to the design parameters defined in the pre-processing stage. Moreover, the sensitivity coefficients are computed outside the FEA codes using only post-processing data from FE analysis. The sensitivity computation in the DSO has been integrated and automated so that the program is executed and necessary data transferred and accessed without the need for designer interaction.

The adjoint variable method of continuum DSA is used to compute design derivatives for sizing application in the DSO. [25] For shape design applications, the DSO methodology introduces the concept of the design velocity field [26] to describe movements of material points resulting from a change in shape of the structural boundary. Velocity field computation can be accomplished using two methods, boundary displacement or isoparametric mapping.

Post-Processing

The post-processing design stage in the DSO is a four-step interactive design process, which includes design sensitivity display, what-if study, trade-off determination, and design optimization. [27-29] This interactive design environment allows the engineer to improve designs using the design sensitivity coefficients. The first three design steps help the engineer understand the structural behavior of the current design and suggest how better designs can achieved. The last design step launches a commercial optimization code. The post-processing stage in the DSO does not dictate a new design; instead, it provides sufficient design information and design suggestions for the engineer to make appropriate design decisions.

Design Sensitivity Display - The design sensitivity information can be used as design guidance. Graphical displays of the information using spreadsheets, bar charts, and color plots make it easy to use this sensitivity information. To help the user understand the structural behavior, the DSO also provides two normalization schemes—normalization with respect to mass and normalization with respect to performance measures.

What-if Study - The what-if study provides predictions of structural responses at perturbed designs using the design sensitivity information. In contrast to the lengthy finite element analysis for the perturbed design, structural responses can be obtained very quickly using what-if study. What-if results can be displayed using spreadsheets, bar charts, and PATRAN color plots. The DSO also provides model update capability, performing automatic data update, finite element analysis, and design sensitivity computation to correlate the structural system with the new perturbed design.

Trade-off Determination - The trade-off determination provides a design direction to correct constraint functions, reduce cost functions, or both, based on the design model at its current design. The direction can be used for what-if study, which is in turn used for design try-outs. In the DSO, four options are supported to perform design trade-offs: cost reduction, constraint correction, constraint correction with constant cost, and constraint correction with constant cost increment.

Design Optimization - The design optimization optimizes the structural design model using a nonlinear programming algorithm. In the DSO, an open software structure allows the user to easily integrate commercial optimization codes. The DSO performs structural model updates based on the new design, sends the new model for finite element analysis, updates the cost and constraint function values, computes design sensitivity information, and feeds the information into the optimization codes to interactively improve the design model. Currently, the DSO integrates VMA Engineering's Design Optimization Tool to perform design optimization.

The DSO workspace configuration employs remote facilities to provide flexibility in the design environment by permitting computationally intensive tasks, such as FEA, to be distributed from the engineering workstation to mainframe or supercomputers. In addition, the remote facility permits data files to be transferred to other graphical workstations so that designers can visualize model data. The remote facility thus enables the DSO to better utilize the graphical capability of the workstation and the computational power of the mainframe. The remote configuration is illustrated in Figure 3.8.

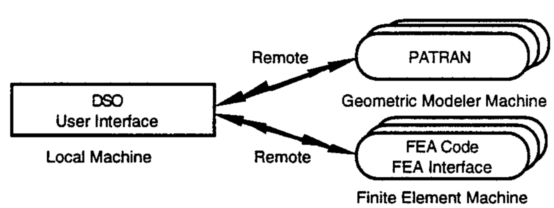


Figure 3.8 DSO Remote Configuration

With the remote facility, the DSO user interface runs on an engineering workstation that has an X server (defined as the Local machine), while PATRAN runs at another workstation (called the Geometric Modeler machine) that provides excellent graphics, and FEA jobs are sent to a supercomputer or a mainframe (called the Finite Element machine). Once the remote configuration is defined, the engineer can request the DSO to check the connection to make sure that the remote machines are available in the network.

The DSO requires two types of remote jobs: remote program execution and remote file transfer. To support these requirements, the remote facility currently uses BSD [30] remote commands such as `rlogin`, `rcp`, and `rsh`. Once a job is launched by clicking the menu button in a DSO menu, e.g., to execute a sensitivity computation, the remote facility parses shell commands defined in a command script [31] and communicates with the remote machines to carry out computations defined in the script. Since the program execution sequence is fixed for all computations, scripts are prespecified and stored in both local and remote machines. The concept of remote execution is illustrated in Figure 3.9.

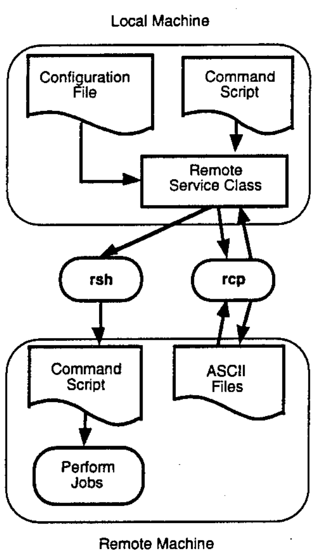


Figure 3.9 DSO Remote Execution

In order to promote software reusability and to encourage a building block approach for software construction, common data structures that are used in developing the DSO are grouped and implemented as foundation classes. This group consists of data structures for numerical computation, such as Vector, Matrix, and Complex Object; for persistent objects, such as CHR, CHF, and Table; and for object management, such as Sequence and List. The class hierarchy of foundation classes is shown in Figure 3.10.

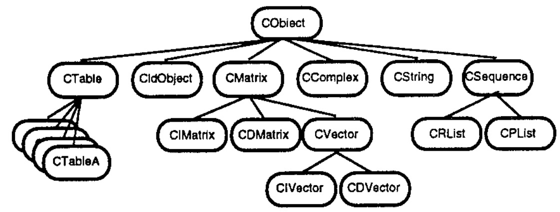


Figure 3.10 DSO Foundation Class Inheritance Hierarchy

The DSO uses a dedicated data management system to provide consistent and efficient access to the large set of data manipulated by the computation modules. A table-oriented data management concept is employed to develop the database. About 55 tables support software development and data storage during engineering design. Data viewing and access is simplified using an engineering spreadsheet. The spreadsheet is utilized in the DSO to browse important design data, to define and modify existing data, and to create formulas to define design parameters. The spreadsheets are developed using an object-oriented approach. [32] A general spreadsheet is developed as a framework, and specialized spreadsheets can be constructed by inheriting from and adding properties to the general spreadsheet. This spreadsheet-based user interface is implemented using OSF-Motif based on the X Window system.

Other Capabilities: DADS Translator

Other software capabilities developed under the DICE Phase 4 effort include the DADS Translator. The DADS translator is a tool to transform the input data format from the Dynamic Analysis and Design System (DADS) to Real Time Recursive Dynamics (RTRD) formulation. This effort initiated the first steps in establishing an interface between off-line dynamics modeling and simulation and real-time operator-in-the-loop driving simulation - linking CAE engineering analysis applications with the Iowa Driving Simulator within the framework of the integrated environment depicted in Figure 2.2. The DADS translator provides a conceptual foundation for the transformation of detailed engineering dynamics models employed in capabilities such as TVWS into models suitable for IDS operations, which employs recursive dynamics formulation to achieve real-time simulation. In this manner, the potential to develop recursive dynamics models correlated with design change/improvement developed and implemented in the CAE analysis environment, affords an essential first step in achieving a rapid, effective virtual prototyping capability.

DADS has been a widely used commercial dynamics simulation tool, and for engineers familiar with the DADS pre-processor and input format, the translator enables easy dynamic mechanism input modeling to RTRD-based simulations, without the need for in-depth knowledge of the RTRD input format. The translator is written in standard Fortran-77 code and can be easily ported to any platform capable of running RTRD code. Large mechanical systems, i.e. systems with a large number (>50) of dynamic bodies, can be accomodated by the translator by modifying internal array dimensions.

Two types of data are needed for the translator: the DADS input file and a topology analysis data file (see Figure 3.11).

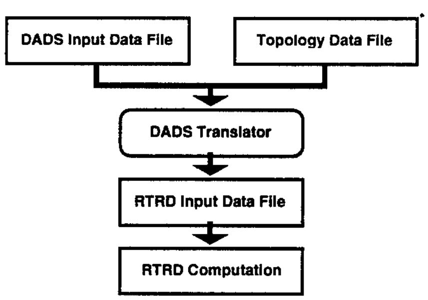


Figure 3.11 DADS to RTRD Translation Sequence

The DADS input file will need to contain only those elements supported by the translator. Translational, spherical, revolute, universal, spherical-spherical, and distance constraint joints are are supported in the DADS translator. Force elements supported include Translational Spring-Damper Actuator, Rotational Spring-Damper Actuator, and tire elements. Base body and revolute joint initial conditions are currently the only initial conditions supported. The topology model data file contains an upper triangular topology matrix whose entries consist of the joint type numbers that connect pairs of bodies.

Tracked Vehicle Concurrent Engineering Environment Integration

System integration in the TVCE environment combines the CAE analysis application described in the preceding into one functional unit. The integrated TVCE environment, depicted in Figure 2.2, possesses more countable and complete engineering capabilities than the sum of the isolated individual engineering tools. This is because the environment is self-contained and the applications are mutually supported. With these applications integrated into one system, engineers are able to base their design analysis process on any combination of three criteria: life prediction, dynamic simulation, and design sensitivity analysis. Thus, the integrated system enables global optimization of designs.

The TVCE environment is designed to be used in conjunction with existing CAD, CAM, and CAE applications. The function of the TVCE capability, as presented in this section, can be customized for incorporation into any existing environment. Figure 3.12 illustrates, for example, how the TVCE environment could be incorporated into a general product development enterprise. The TVCE environment has been designed for maximum extendibility - its open architecture is designed to easily incorporate technological improvements in both hardware and software.

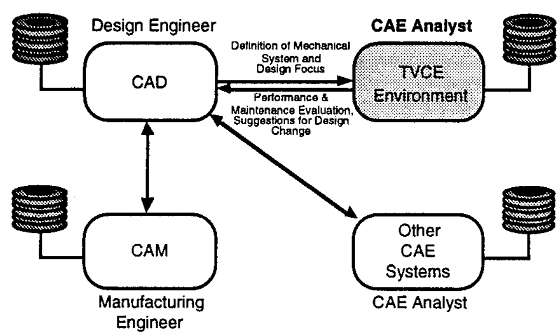


Figure 3.12 Utilization of the TVCE Environment in Industry

The function of the TVCE environment is to promote the use of simulation-based Concurrent Engineering (CE) to evaluate tracked vehicle design and suggest design modifications from dynamic performance and structural perspectives in a design team-oriented context. Concurrent Engineering is a concept which promotes the earliest possible integration of the enterprise's overall knowledge about a product. When CE is employed, design life cycle cost reduction can be achieved by:

- Promoting multi-disciplinary interaction through enhanced communications.

- Obtaining a globally optimized design through integrated systems.
- Reducing duplication of effort through project coordination.
- Keeping information on designs and simulation results current and organized through the use of a structured database system.

The TVCE environment is an invaluable resource for the increase of efficiency implied in the above. The future of engineering mechanical systems lies with the CE concept and this environment has been created as a framework for the CE design process.

The design of the TVCE environment focuses on data management, the wrapper, and information flow. Data management in the TVCE environment consists of constructing a unifying data model, providing data management, establishing version control, and providing a graphical user interface for the user. A wrapper provides the interface between the workspace and the global DDS of the TVCE environment. The wrapper also provides the front end graphical user interface, such as the data browser. Information flow in the TVCE environment is essential in promoting interdisciplinary collaboration.

The TVCE environment employs the CCS, TVWS, DSLP, and DSO workspace applications described previously, and three support entities that provide the individual applications with data storage, computational power, and communications/coordination. The global Design Data Server (DDS) provides the environment with data storage, data management, and version control. The engineering computation servers are specialized engineering applications that provide the workspace analysis capabilities with complex numerical analysis that cannot be conveniently or efficiently implemented directly in the application. The Design Process Management tools (see Section V) support project coordination among team members through a process-based communications methodology.

The integrated environment is realized through the wrappers and global DDS (see Figure 3.13). The global DDS maintains the product description which consists of finite element models, mechanical system component characteristics, and other items employed by the entities in the TVCE environment. The DDS acts as a server to the individual workspaces and support entities and provides the principal tools for implementing data management in the TVCE environment.

The issue of data management is tightly coupled with the integration of multiple applications into a single environment. Each workspace application employs its own description of all or part of the mechanical system together with information peculiar to the operational requirements of that workspace. The global database must be so structured and populated as to only support the input requirements for each workspace while concomitantly maintaining a consistent representation of the product, its behavior, and its analysis. To reduce data flow and management requirements, data used or generated by one application solely for the purposes of that application are not stored or managed globally.

The DDS was designed to handle all aspects of global data management. The DDS contains two modules, the Access Manager and the Data Manager. The Access Manager provides an interface between the communication channel and the global database. The Data Manager provides a catalogue of independent objects stored in the global database, manages the objects' versions, and lists the objects that do not have version histories. The global database is based on ROSE, which uses an object-oriented data structure. The object-oriented structure is particularly well suited for implementing a data model appropriate for mechanical system development using simulation-based design technologies. The data model developed for the TVCE environment database is illustrated in Figure 3.14, with a complete description of the database entities given in Appendix A.

While the DDS has been designed to bring together the diverse workspace applications used in product development and support automated file management and data file conversions in a server context, a client side mechanism is needed to complete transfer of data to the workspace tools. The wrapper software programs are designed to fulfill client side data translation requirements, thus providing the functional integration capability at the workspace level. Wrappers enable the standardized exchange of information between the wrapper application and the remainder of the TVCE environment, and also provide a standard user interface for engineers to access and browse the global DDS. A wrapper allows the engineer to :

- Browse and select objects of interest stored in the global database.

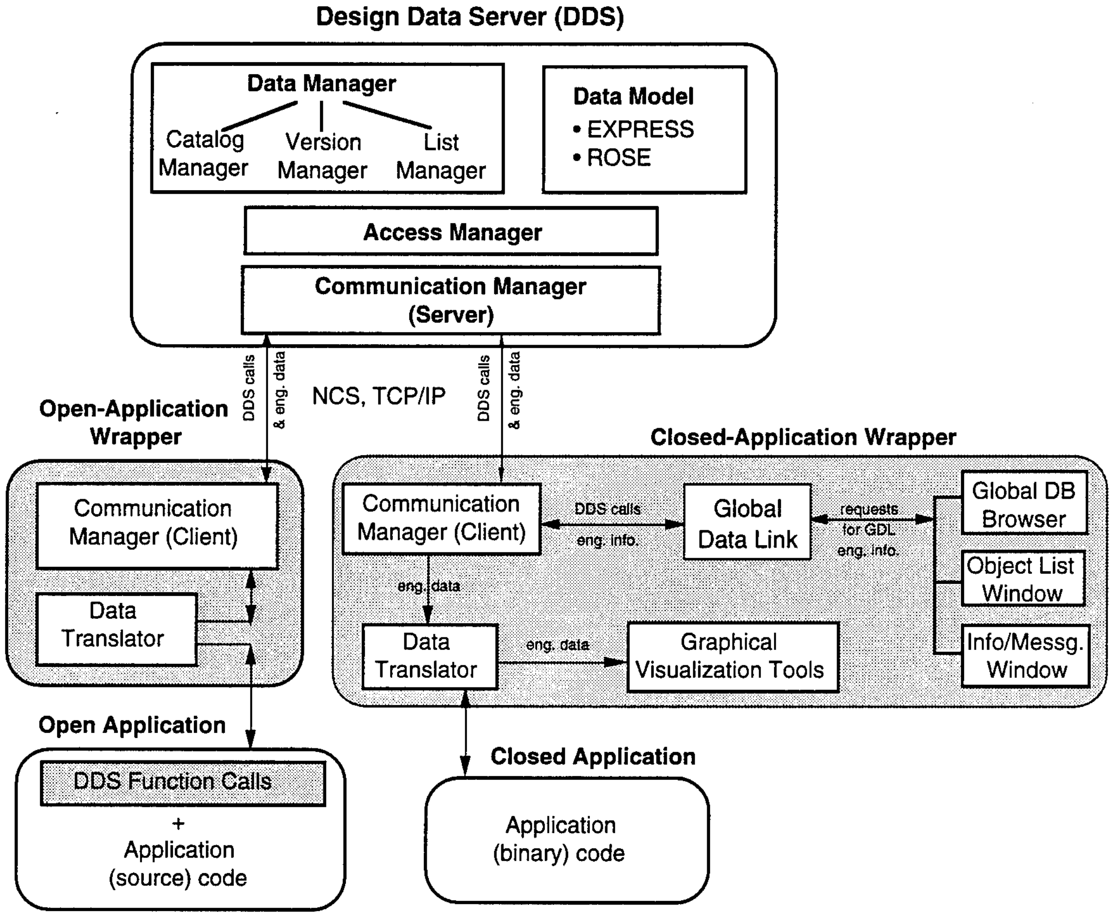


Figure 3.13 TVCE Integration Architecture and the DDS Structure

- Effectively preview engineering data of interest.
- Select objects to be sent from the application to the global database.
- Convert data between the global DDS and application formats.
- Invoke engineering applications.

Two types of wrappers are employed in the TVCE environment, open and closed application wrappers [33] (see Figure 3.13). An open engineering application is integrated into the TVCE by modifying the application to issue DDS calls directly. A closed application is integrated by using the wrapper as a medium to bring in needed data and export the application's output data to the global database. The closed approach is necessary when the application code cannot be modified.

A generic wrapper contains several modules: the client side of the Communications Manager (CM), the Global Data Link (GDL), the User Interface, the data translators, and the visualization tools.

The Communication Manager is the agent through which the other components of the wrapper communicate with the global DDS. The CM is composed of two separate modules. One of these, the client side, is linked to the wrapper. The other, the server side, is linked to the DDS. In the current implementation, these two sides use the Remote Procedure Call (RPC) network communication mechanism of the Network Computing System (NCS) to pass various data and requests between the wrapper and the DDS. The CM provides a set of functions that allows the wrapper to treat the DDS as a local database. In reality the DDS is a separate element, possibly executing on a geographically distant machine.

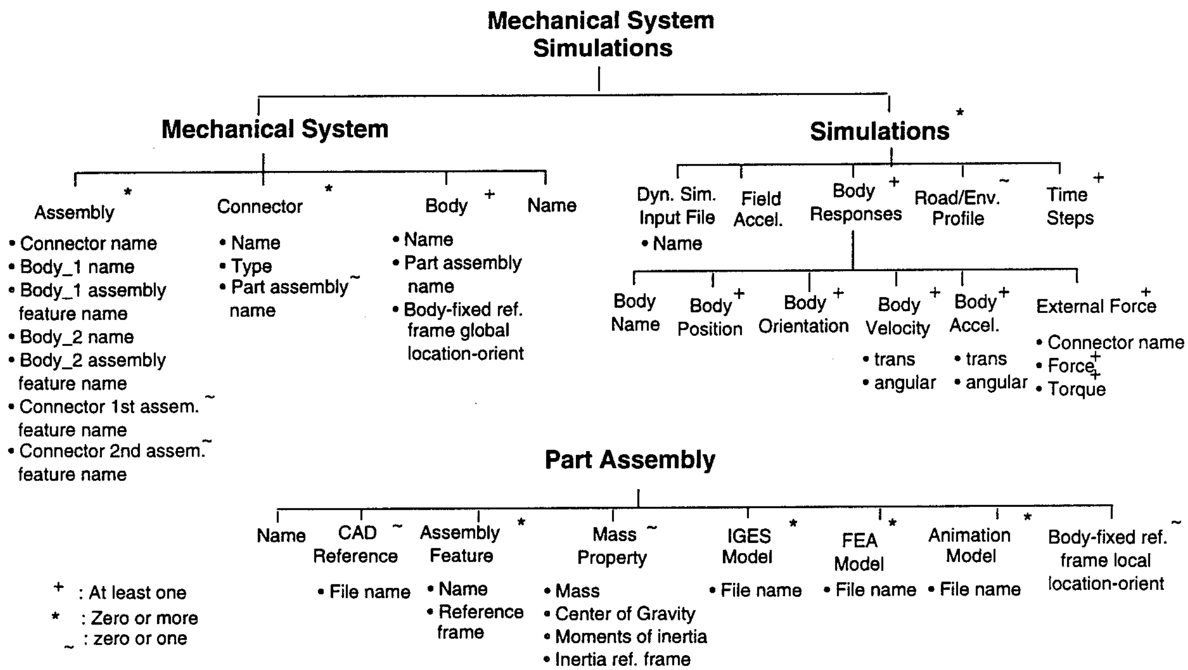


Figure 3.14 Tracked Vehicle Concurrent Engineering Environment Global Data Model

The Global Data Link provides an application with a view of the global database structures. It also allows engineers who are using that application to set the context for data to be sent into or obtained from the global database. Setting the context means informing the DDS about the objects of interest and particular versions of those objects. Since the DDS is shared by many users, it needs to keep track of each user's objects and versions of interest. This is done by the list of activation records maintained by the DDS, one record for each user. When the wrapper sets the context information (e.g. an object version), the DDS stores this information in the activation records. Later, when the wrapper calls the DDS to retrieve or transfer data, the stored context information is used by the DDS to obtain the data from the correct object. In this way, users can perform many operations on a particular object without setting the context for each of the operations.

It is important to note that the GDL only shows the portion of the data model in the global database that is of interest to the application. Furthermore, the GDL presents the data in a way that is meaningful to the application's users. The view of the global database can be tailored by modifying the GDL's data configuration files. The GDL is a tree representation of the data model of the global database.

The User Interface portion of the wrapper consists of a menu, the Global Database Browser, Database Object List window, and Information/Message utilities.

The Database Browser provides a graphical representation of the GDL and serves as the interface between the user and the global DDS. All the nodes of the browser form a view of the part of the global data model that is of interest to an application. Each node of the browser has a link to the corresponding GDL node. However, the browser does not store any information about the database object hierarchy. By separating the GDL and database browser, the user interface does not need to be changed if the object hierarchy of the global database changes.

The Information utility displays messages that are associated with the object selected from the Object List. The message helps users to recognize whether they have selected the desired object or whether additional steps must be completed before they can use the application. The information and messages that are displayed using this utility are obtained through a path similar to the one by which object instances are obtained from the global database.

The data translators have been developed either to translate data produced by an application to specific forms for the global database, or translate data stored in the global database to specific forms that the application can use. A typical data translation converts vectors (such as forces or geometry) from one coordinate system to another. For instance, the loading histories of a body are reported by a dynamic analysis application relative to the body fixed reference frame. The forces are used in structural analysis of the finite element model of the body, but the finite element model may be defined relative to a reference frame other than the body fixed reference frame. Hence there is a need for data translation. Another functionality of the data translator is to calculate resultant vectors from components of vectors, for example, calculating the forces in the xy plane from the force components in x and y directions. Such translations are needed before structural analysis engineers can select appropriate loads for their analysis models.

Finally, the visualization tools help engineers visualize data obtained from the global database and select data to be used for analysis. An example of such visualization tools is a 2D plotter, which is used to display loading histories at joints of a body to assist engineers in selecting the peak load. Crude graphical animation tools are also useful; they can animate the motion of a small number of mechanical components. For example, structural analysis engineers need to determine the location and orientation of the peak load on a wheel. They can do this by running an animated dynamic simulation and observing points or areas where the wheel contacts the road.

TVCE Environment Operations: Roadarm Example Application

Initial testing of the TVCE environment engineering workspace and integration tool capabilities was conducted internally at the Center prior to release of the software to the industrial partners for validation testing. A realistic tracked vehicle example was selected, based on a generic M1A1 Abrams main battle tank configuration. A track suspension component, the roadarm, was targeted for re-design using the TVCE engineering analysis capabilities. Figure 3.15 illustrates the tracked vehicle application used; the roadarm suspension component is highlighted in yellow. The basic test scenario, illustrated in Figure 3.16, consisted of the definition of the M1A1 system and roadarm models, using the CCS, followed by dynamic simulation using TVWS, to generate duty cycle data on the roadarm, with subsequent structural fatigue life prediction (DSLP) and design sensitivity analyses (DSO) performed to develop an improved roadarm design from fatigue life perspective. This section outlines in detail the activities performed during this exercise and presents a series of simulation results employed to develop the improved roadarm design. A description of the software testbed and computer hardware platform used to perform this exercise is given in Appendix B.

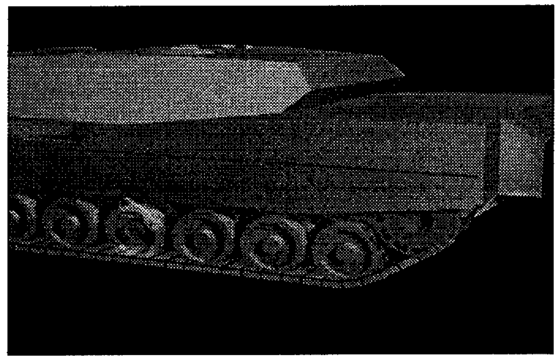


Figure 3.15 Tracked Vehicle Model

A pre-existing CAD model of the roadarm served as the starting point for the purposes of this exercise (see Figure 3.17 (a)). Conceptually, the CAD model serves as the communication referent among the engineering disciplines in the concurrent engineering environment during the product development process. The CAD model of the roadarm was generated using Unigraphics, and translated using UGII-PATRAN-Interface to the PATRAN neutral file format. The test engineer used PATRAN to read the neutral file, then created the PATRAN geometric and finite element models of the roadarm. The roadarm finite element model, shown in Figure 3.17(b), was generated using the same reference frame as that of the dynamic model. There are 310 20-node isoparametric finite elements, 1913 nodes, and about 5,700 degrees of freedom in the roadarm finite element model. The material S1005-1009 steel, was used for the roadarm. Three bodies of the tracked vehicle, hull (Hull), the 7th roadarm (ArmR7), and the 7th roadwheel (WheelR7) on the right side of the vehicle, were selected to add to the mechanical system using CCS. Note that the names given in parenthesis are specified by TVWS, and therefore, adopted as the naming convention for these bodies in this report. The mechanical system created using CCS is illustrated in Figure 3.18. After completion of the activities employing CCS, the information and data generated consisted of

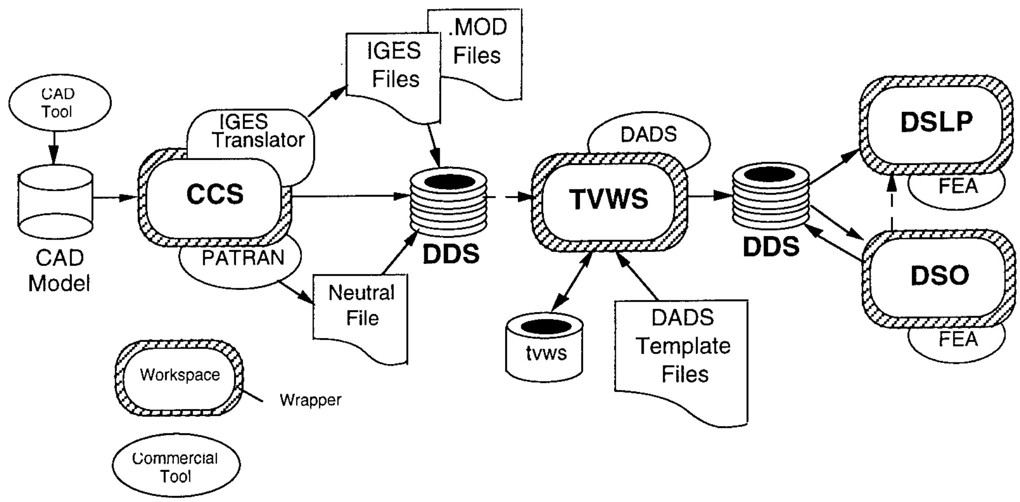
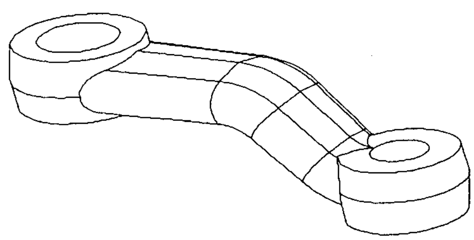
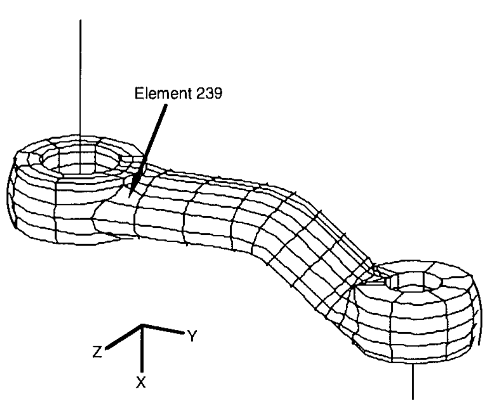


Figure 3.16 TVCE Environment Exercise Scenario

(1) three tracked vehicle mechanical system bodies existing in the DDS, (2) a "Model" directory containing CCS data files, (3) an IGES file of the roadarm existing in the DDS, and (4) a PATRAN neutral file of the Roadarm, also existing in the DDS.



(a) Roadarm CAD Model



(b) Roadarm Finite Element Model

Figure 3.17 Roadarm Models

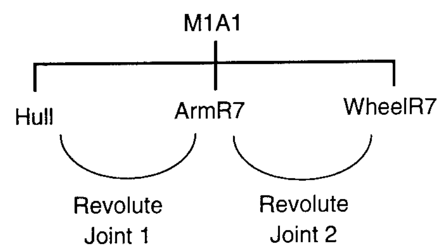


Figure 3.18 CCS Tracked Vehicle System Model

The example dynamic simulation model was defined in TVWS. A DADS input file of the tracked vehicle generated by the TVWS template files was used as the operational reference for dynamic simulation. A portion of the tracked vehicle model is illustrated in Figure 3.19. Note that body reference frame of ArmR7, x_{ra}-y_{ra}-z_{ra}, is oriented by rotating at an angle of -2.563 radian along x_{ra}-axis (which is parallel to the global X-axis). The center of gravity (CG) of the roadwheel, defined as a point of interest in DADS, is located at 20 in. (the length of the roadarm) along the positive y_{ra}-direction. The scenario defined for vehicle dynamic simulation identified a vehicle speed of 20 miles per hour in the forward direction (positive y-direction in the global frame shown in Figure 3.19), with no gun firing and no rotation of the turret. The road profile Aberdeen Proving Ground 4 (APG4) was selected for this simulation. The entire simulation lasted 12 seconds, with a step size of 0.05 seconds for a total of 240 time steps. After the TVWS/DADS simulation was completed, the information and data generated consisted of (1) dynamic simulation results for the tracked vehicle mechanical system stored in the DDS, (2) DADS pre- and post-processor and input data files generated by TVWS, and (3) a set of DADS output files.

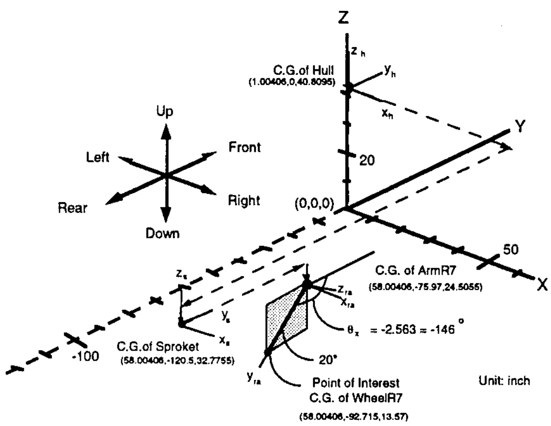


Figure 3.19 A Portion of the Tracked Vehicle DADS Model

From the simulation results, the peak load was found at time 9.4 seconds in the 12 second simulation. The loads reported by DADS at 9.4 seconds are listed in Table 3.2. Note that the loads were reported corresponding to the roadarm local reference frame, i.e., x_{ra}-y_{ra}-z_{ra}, as shown in Figure 3.19. At 9.4 seconds, the y-coordinate of the roadwheel center (WheelR7) is 3,309 in. From the road profile APG4, used in the simulation, at 9.4 seconds the tracked vehicle is heading up to an 8% slope and the WheelR7 is on a flat surface, as shown in Figure 3.20.

Table 3.2 Roadarm Dynamic Simulation Results at 9.4 Seconds

Items		Wheel Side	Hull Side	CG of Roadarm
Force	F_x	-114.81120	122.14820	
(lb.)	F_y	-20,612.988	19,830.537	
	F_z	-17,241.850	12,794.284	
Moments	M_x	0.0	248,706.34	
(lb.-in)	M_y	-1,411.3529	1,479.4565	
	M_z	-3,323.5974	941.63660	
Accel.	a_x			44.622192
(in/sec^2)	a_y			-5,434.92
	a_z			8,464.724
Orient. (rad)	q_x			-2.5567

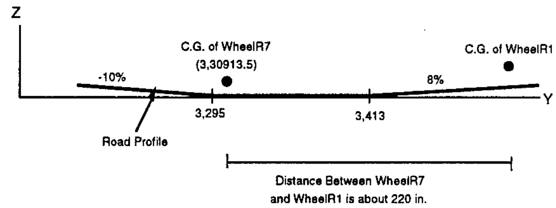


Figure 3.20 Portion of Road Profile and Vehicle Position

From the data found in Table 3.2, the orientation of the roadarm at 9.4 seconds was identified and the resultant force at the wheel end was acting approximately upward (see Figure 3.21), which is a meaningful result. The resultant forces in the x_{ra}-, y_{ra}-, and z_{ra}-directions were verified in equilibrium.

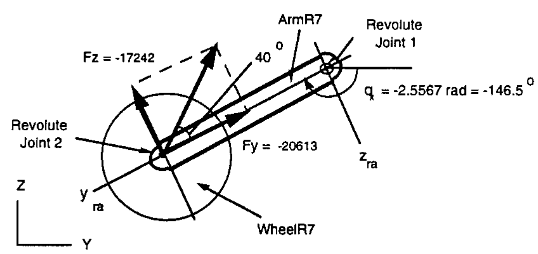


Figure 3.21 Peak Load Applied at Wheel End of Roadarm

After the dynamic analysis was completed in TVWS, the DSLP wrapper was used to (1) obtain the PATRAN neutral file of the roadarm from the DDS, (2) obtain dynamic simulation results from the DDS to initiate structural analysis and dynamic stress computation, and (3) execute DSLP for dynamic stress computation and life prediction. Dynamic stress computation consisted of:

- PATRAN model translation (translating the roadarm PATRAN model to an ANSYS finite element input data file for model analysis).
- ANSYS vibration analysis (performance of static analysis to obtain the mass matrix).
- Dynamic stress load vector computation (24 load cases to compute stress coefficients. Stress coefficients multiplied by the load history yields the dynamic stress).
- ANSYS stress coefficients analysis (solve the 24 load cases using ANSYS).

- Dynamic parameter computation (read in load history data).

The roadarm fatigue life due to crack initiation was computed at the corner nodes of finite element 239. The results are given in Table 3.3. The local strain approach, using S-W-T and Morrow mean stress corrections, was employed in DSLP to calculate fatigue life. In Tables 3.3 to 3.5, the unit measure is given as a "block". For this test case, a block corresponds to 12 seconds. The results of the fatigue life computation for this test predict a crack initiation life of 318 days (12 sec x 2.29E6) for a tracked vehicle running continuously.

Table 3.3 Roadarm Crack Initiation Fatigue Life

Nodes		Fatigue Life
Surface	34	6.79×10^9
	35	8.04×10^6
	1	2.29×10^6
	3	2.77×10^9
Interior	38	2.19×10^{12}
	40	1.98×10^8
	11	3.19×10^6
	39	5.31×10^8

Roadarm crack propagation fatigue life results were also computed for element 239 using FLAGRO, for an initial crack length of 0.01 in. Crack propagation fatigue life results are given in Table 3.4.

Table 3.4 Roadarm Crack Propagation Fatigue Life

No. of Cycles	Crack Length
300	1.13320×10^{-2}
600	1.28682×10^{-2}
900	1.46444×10^{-2}
1200	1.67037×10^{-2}
1500	1.90979×10^{-2}
1800	2.18897×10^{-2}
2100	2.51481×10^{-2}
2400	2.88091×10^{-2}
2700	3.31379×10^{-2}
3000	3.82776×10^{-2}

While DSLP was being used to compute dynamic stress and estimate roadarm fatigue life, the DSO workspace was being used to perform design sensitivity analysis of the roadarm to obtain a better design from a stress distribution perspective. Prior to initiating DSO analysis, the DSO wrapper was employed to import from the DDS: (1) the roadarm PATRAN neutral file and (2) the dynamic simulation results; identifying the peak load (worst case) for structural static analysis and design. After obtaining worst case structural responses, the DSO user identifies the area of concern (high stress area) using the finite element analysis tool and manually returned this information as a set of finite element nodes to DSLP. The DSLP user then predicted the roadarm fatigue life at these nodes. The DSO user obtained a new design for the roadarm and exported the new design back to the DDS as a new PATRAN neutral file. DSLP retrieved the new roadarm design FE model from the DDS and evaluated the fatigue life of the new roadarm.

The roadarm was parameterized by defining 10 design parameters characterizing five intersection surface movements in the x- and z-directions, as shown in Figure 3.22. The volume and maximum von Mises stresses defined at integration points of each finite element are defined as performance measures. There are 310 stress and one volume performance measures defined. The maximum stress is found at element 239. Eight corner nodes of the element 239 were sent to DSLP manually to calculate fatigue life, as described previously.

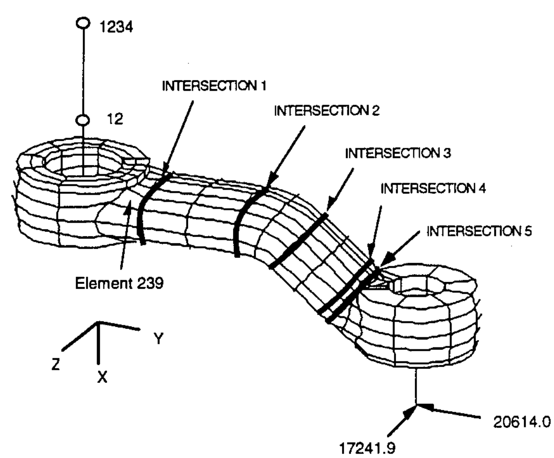


Figure 3.22 Five Intersections of the Roadarm

Design velocity field and sensitivity coefficients were computed using the isoparametric mapping and direct differentiation methods of the DSO, respectively. Computation of stress sensitivity coefficients at element 239 suggested that the stress at element 239 would be reduced by moving the first intersection in the positive x- and z- directions. A what-if study was performed using steepest descent direction of stress sensitivity coefficients at element 239 and a step size of 0.8 in. as the design change. From the results of the what-if study, a more homogeneous stress distribution was obtained for the design depicted in Figure 3.23. A new PATRAN neutral file was then generated by DSO containing the new geometric and finite element models of the roadarm for the perturbed design. The DSLP crack initiation fatigue life analysis was re-iterated for the new design yielding an improvement of 28.9 times the fatigue life of the original design (see Table 3.5).

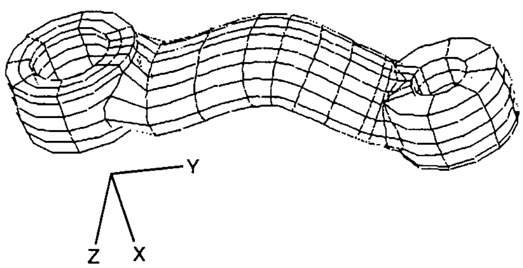


Figure 3.23 PATRAN Finite Element Model of Improved Roadarm Design

Table 3.5 Crack Initiation Fatigue Life of Modified Roadarm Design

Nodes		Fatigue Life
Surface	34	7.61×10^{9}
	35	1.61×10^{9}
	1	2.60×10^{11}
	3	9.37×10^{16}
Interior	38	1.19×10^{12}
	40	6.61×10^{7}
	11	5.49×10^{8}
	39	1.17×10^{12}

A series of operator and computer usage statistics were recorded during the performance of this exercise (see Appendix C) to quantify the duration and compute requirements. CAD and FE model generation duration was not included - these activities can require substantial effort, and were not timed in order to obtain a better assessment of TVCE environment operations. Discounting duplication of calculations and operations resulting from mistakes, as well as discussion and debugging time, roughly 24 hours, or three working days, were needed to achieve the definition of the new road arm design. During the exercise, the environment users were in actuality the tool developers who, therefore, are intimately acquainted with the functioning of the environment capabilities. It is expected that designers and engineers will likely require more time in the performance of similar activities, until a sufficient level of operational experience is attained. Some deviation in duration would also be expected for operation of the TVCE capability on other hardware platforms.

In addition, approximately 560 Megabytes of disk memory were required to accommodate the files generated by this example. A sampling of file sizes is also given in Appendix C. The largest files were produced by ANSYS FE analysis in DSO and DSLP. The FE models developed for this example represent a moderate to low number of elements/DoF in comparison with typical industrial models. It can be assumed that industrial applications require larger amounts of disk storage and run time for FE analysis.

Given the above data, however, it is reasonable to conclude that the TVCE capability entails a substantial achievement in improved efficiency of and reduction in the time required for the design and analysis process.

TVCE Environment Validation: Subcontract Applications

Three military tracked vehicle developers participated in TVCE validation exercises. Each of these three companies, the former BMY Combat Systems (now United Defense LP Combat Systems Division), the former FMC Ground Systems Division (now United Defense LP Ground Systems Division), and General Dynamics Land Systems Division, carried out a specific design and analysis application associated with an on-going tracked vehicle program/project. The following are non-proprietary summaries, adapted from the contractors' reports, of the validation exercises performed by each contractor, with some supplementary discussion regarding lessons learned from these exercises. For each example application described, a brief overview of the exercise scenario is provided, followed by contractor comments regarding the performance/utility of the individual software tools and the environment integration architecture.

United Defense LP Combat Systems Division (BMY)

In November of 1993, UDLP-CSD installed the TVCE software environment on a DEC Station to evaluate the utility of the environment using an application exercise performed in conjunction with the Breacher program. UDLP-CSD's Breacher Program has been designing a vehicle which can clear a safe path through a mine field. The vehicle is a modified M1 tank with a plow/blade assembly installed at the front of the vehicle as shown in Figure 3.24. The Breacher vehicle has an M1 suspension system and chassis. The vehicle has its own crew module, a plow blade, two pushbeams, three hydraulic actuators, and an excavating arm. The up and down blade position is controlled with a large life actuator which is attached to the center of the vehicle's front plate. The actuator is supported with a bracket which is welded to the front plate. This bracket assembly, illustrated in Figure 3.25, constituted the target component for analysis using the TVCE environment. UDLP-CSD attempted to employ all of the engineering workspaces, i.e., CCS, TVWS, DSO, and DSLP, in the TVCE environment during the performance of this exercise.

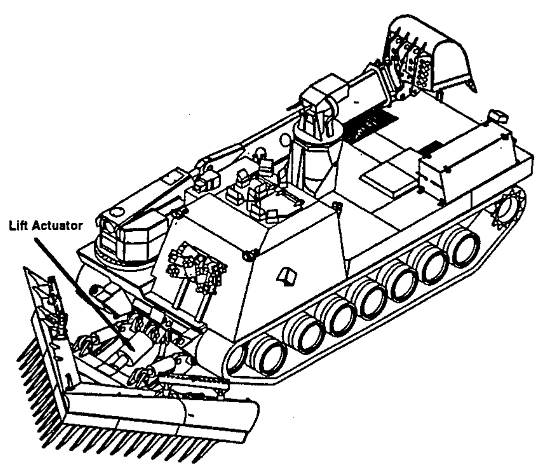


Figure 3.24 Breacher Tracked Vehicle

Exercise Scenario

The CCS functionalities were employed to generate a new mechanical system that included a hull, a roadarm, and a roadwheel. A dynamic simulation scenario was identified supporting the evaluation of vehicle dynamic performance over a rough (bumpy) terrain. Significant modification of the TVWS template dynamic modeling procedure was required to accommodate the unusual vehicle configuration imposed by the Breacher's plow/blade assembly. The objective of the dynamic simulation was to obtain roadarm load histories for the vehicle as it maneuvered over APG course #4 while in travel-lock position, where the blade/pushbeam/actuators subsystem is locked at the chassis. A mechanical system model containing a chassis (hull) and two track super elements was defined. Dynamic simulation was carried out as specified in the TVWS user documentation. Roadarm forces at the hull attachment point were obtained; load history data was exported to the global DDS using TVWS wrapper functionalities.

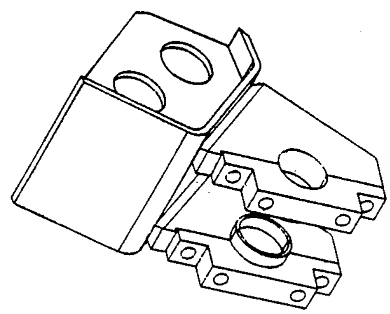
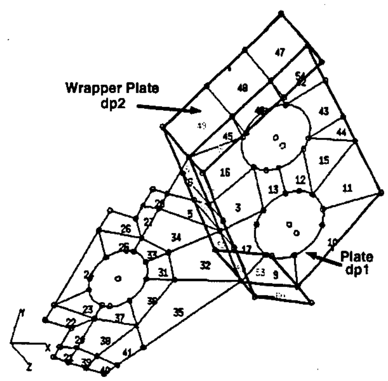


Figure 3.25 Breacher Plow/Blade Support Bracket Assembly

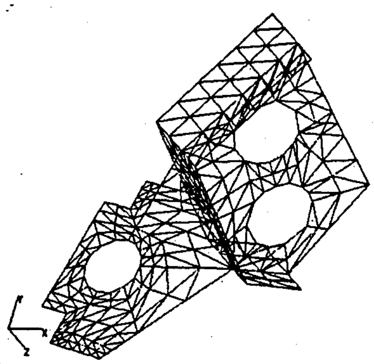
An attempt was made to import FEA model and load history data from the DDS into the DSLP which failed. As a result, successful operation of the DSLP workspace was not accomplished for this exercise.

For DSO evaluation, the sizing sensitivity analysis and optimization option was selected, since the bracket design is configured with structural steel plate. The PATRAN geometry modeler was launched in DSO to create geometric and finite elements of the bracket (see Figure 3.26). As illustrated in Figure 3.26, all elements in the FE model are ANSYS STIF63 Elastic Quadrilateral Shell elements; modification of the model using triangular elements was required to run the DSO analysis. The bracket plate with the two holes was selected for application of the sizing analysis. A one-half inch thickness was defined for the plate. Finite element analysis was launched in the DSO as specified in the user documentation. Volume and von Mises stress were defined as performance measures. Cost, performance constraint, and side constraint functions were defined using a combination of performance measures. Volume was defined as a cost function. Since the yield strength of the plate material is approximately 90,000 psi, with a safety factor of

3.0, an upper bound of 30,000 psi was defined for the performance constraint. The maximum feasible thickness for the plate is 1.0 inch, which constituted an upper bound for the side constraint. Both trade-off and what-if analyses were performed using the DSO; the results indicated some potential for defining a new bracket design. Attempts to launch the optimization routine failed, due likely to system problems rather than software problems.



(a) PATRAN Geometry Model



(b) Finite Element Model

Figure 3.26 Bracket Models

Evaluation

CCS - Generation of the simple hull, roadarm, roadwheel mechanical system model was inadequate for the UDLP-CSD application. Some limitations are evident in the DDS in handling/managing a multiple layered database - such management is required to update a roadarm as a member part of a track super element model. The CCS menu system appeared to work well, although some confusion was encountered and some procedures were inadequately defined. CCS menu function problems included (1) creation of a new element model using the CCS local directory, (2) exporting newly defined subsystem/mechanical system models and update of existing parts, (3) registry of system contents is not clear during export to the DDS, and (4) relation of CAD geometry files, .mod animation files, IGES geometry files, and PATRAN geometry files for a new elementary model is also unclear. A suggestion was also made to implement a capability in CCS that allows the deletion of unnecessary part, assemblies, subsystems, and system information. In general, however, the CCS tool and IGES translator capabilities work well in the integrated environment. Thorough training in CAD geometry modeling, PATRAN modeling, and terminology is required for engineers in order to obtain the full benefits of the CCS workspace, however.

TVWS - The current TVWS does not exhibit sufficient flexibility to support modeling of atypical vehicle configurations such as the Breacher. TVWS modeling and simulation development procedures are not well defined and the current terminology is somewhat confusing. Modification of template files is not flexible, and user responsibilities in managing/storing template information is unclear. The TVWS system was consistently organized in the execution of each analysis step, except for communication with CAE tools residing in remote locations.

DSLP - Although DSLP could not be exercised in the integrated operation, exercise of the stand-alone DSLP capability was accomplished without significant problems. A more thorough understanding of data transfer procedures between the DSLP and the DDS is required.

DSO - Once accessed, the software was reasonably straightforward to use. However, some delay in operation of the DSO was encountered in accessing software, locating and manipulating files, and software execution as a result of core dumps and software bugs. It is suggested that DSO be upgraded to accept quadrilateral finite elements. In general, however, DSO was found to be a powerful CAE tool, and has the potential to be used extensively in the UDLP-CSD design and analysis environment.

Conclusion

The general consensus with respect to TVCE functionalities indicated that a more transparent/informative method of operation needs to be developed. Most problems arose as the result of interface disparities between different computer platforms. As such, further investigation in software integration and networking is required. In addition, user-friendliness can be enhanced by establishing more effective guidelines for navigating the TVCE design/engineering sequence.

With respect to basic CAE workspace capabilities, most are too specialized for the journeyman engineer. Extensive knowledge of advanced dynamics, finite element theory, fatigue and fracture mechanics is required to fully utilize the simulation tools, suggesting a need for simplified operation methods and/or more exhaustive training of personnel. Despite the deficiencies , however, there is considerable potential for the application of the CAE tools, both stand-alone and integrated system, in achieving design optimization in a reduced amount of time, and eliminating guesswork.

United Defense LP Ground Systems Division (FMC)

The United Defense LP Ground Systems Division (UDLP-GSD) exercise of the TVCE environment was also carried out in late 1993/early 1994. The TVCE software was installed at UDLP-GSD on a Sun SPARCStation 2, using the SUN O/S 4.1.3 and the X-windows based Open Window, window management system. Remote computational analysis capabilities consisted of the ANSYS 4.4A1 finite element analysis code installed on an IBM RS6000, and the PATRAN 2.5 geometry modeler and finite element pre- and post processor, installed on a Silicon Graphics 240-40 4D platform.

Validation of the TVCE environment was performed for a Bradley Fighting Vehicle (BFV) application. The component targeted for analysis consisted of an aluminum bracket (Figure 3.27). The part is manufactured from 0.375 in. thick 5083-H321 or 5086-H32 aluminum, with a maximum yield strength of 24,000 psi. For the UDLP-GSD in-house problem, the bracket was assumed fixed in translation at the bolt holes at end "A", with the loads applied at end "B". The objective of the exercise was to employ the TVCE environment to optimize bracket thickness and analyze the bracket for fatigue. Using thickness as a design parameter, the bracket optimization strove to minimize volume and element stress, while obtaining a fundamental frequency above 60 Hz. The starting point for this exercise was a CAD model developed for the bracket using CADDS, from which an IGES file was obtained.

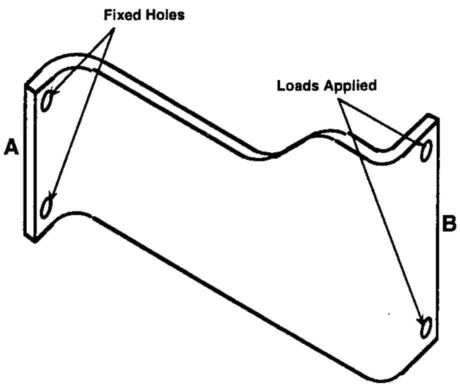


Figure 3.27 Aluminum Bracket (UDLP-GSD)

Exercise Scenario

The CCS workspace was used to define the basic mechanical system, body connections, and to develop bracket finite element models for use in DSO and DSLP analyses. A basic three body mechanical system was defined using CCS, as illustrated in Figure 3.28. For the purposes of this exercise, the bracket served as the trunnion joint, with the mass, inertia, and CG of everything attached at end bolts "B" considered as the "gun", and the centroid of the bracket bolts at "A" considered the "trunnion" joint. This configuration was defined to comply with naming conventions and dynamics model requirements in the TVWS capability. The actual gun weight and inertia was incorporated into the turret body. Bracket joints were used to connect the "gun" body to the turret (the "trunnion" joint) and the turret to the hull (the "ring" joint). In this manner forces at the bracket could be obtained.

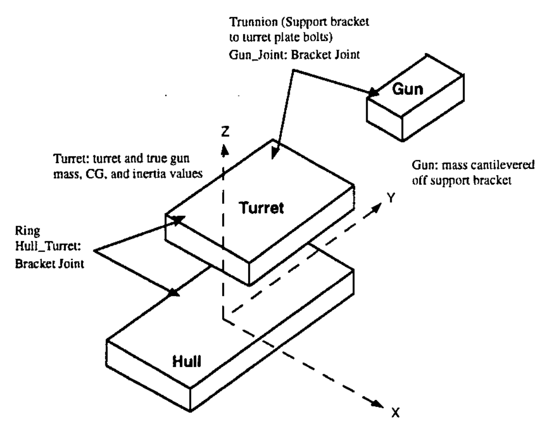


Figure 3.28 Simple Mechanical System Model (UDLP-GSD)

PATRAN was launched in CCS to develop two finite elements models of the bracket. The first model, developed for use by DSO, was composed of 1,892 triangular plate element, with 1,308 nodes. The second model consisted of 843 quadrilateral plate elements and 865 nodes, and was developed for fatigue analysis in DSLP. As DSLP requires the finite element model to be loaded at a joint location, the model included beams which connected the loaded bolt holes at "B" to the joint location where the load was applied. Beams were added to the restrained bolt hole centers at "A" so that boundary restraints were applied at only two locations. The defined mechanical system and PATRAN neutral files were then exported to the DDS for use in downstream analyses.

TVWS was used to perform the dynamic simulation. Template files were modified to correspond to the BFV, with "gun" body mass and inertia changed to correspond to the combined mass, inertia, and CG of elements attached to the end bolts "B". The turret body was modified to include the true gun weight and inertia, and the trunnion joint was modified to correspond to the bracket bolt "A" centroid location. Template modification yielded a vehicle with three main bodies: hull, turret, and gun, with two bracket joints, one between the hull/turret, the other between the turret/gun, as per the configuration defined using CCS. A 20 second simulation was defined for the BFV running at 10 mph over the TVWS course "bumpy1". Results for gun position and trunnion forces, as well as maximum gun force, torques, and accelerations were obtained from the dynamic analysis. After verifying the BFV simulation behavior, the results were exported to the DDS for use in the DSLP.

The DSLP capability was successfully employed for the bracket example, using crack initiation fatigue life prediction analysis. Fatigue crack initiation was predicted for 18 nodes of the finite element model, indicating a fatigue life of the component of approximately 138 days for the given simulation in continuous use.

A DSO sizing application for the bracket was performed, including design sensitivity analysis, tradeoff, and what-if studies, for a maximum vertical load of 1,078 lb. as identified from the dynamic simulation. 76 patches were used to develop the bracket finite element model described previously. Element/patch thickness was selected as the design parameter for a total of 76 design parameters. Volume, frequency, and all elements with von Mises stresses above the material yield strength were chosen as performance measures for a total of 34 design performance measures. Analysis results indicate that thickness change in the largest patches exhibits the greatest effect on bracket volume and frequency, whereas a thickness change in the highest stressed patches exhibits the greatest effect on stress.

Prior to performing trade-off analysis, cost and constraint functions were defined. Volume was defined as the cost function. von Mises stress with an upper bound of 24,000 psi and a fundamental frequency of 60 Hz were defined as performance constraints. Plate element thicknesses were selected as side constraints with limits of 0.125 in. to 1.00 in. The volume cost constraint was neglected in the trade-off analysis to obtain a feasible design. A design direction was obtained from the what-if study, using a perturbation step size of 0.10 in. The what-if study showed a feasible design change could be implemented yielding a decrease in stress for all highly stressed elements, with an increase in frequency and volume (neglecting the volume cost constraint).

Evaluation

TVWS - With practice, utilization of the TVWS capability became very straightforward in terms of changing test scenarios and template files, running analyses, and exporting results to the DDS. Some difficulties were encountered in the excessive generation of dynamic results files for each DADS run. The number of files generated can contribute to a significant depletion of disk space. In addition, the TVWS/DADS directory structure is not well suited to handle the large amount of object files.

The most significant difficulty encountered in operating TVWS, however, was a certain lack of flexibility in modifying template files to incorporate changes in run times, terrain files, force elements, and to add or modify components in the dynamic system model. An in-depth knowledge of the templates and dynamic modeling is required to accomplish such modifications. Likewise, a capability to incorporate UDLP-GSD developed DADS modifications was also lacking in TVWS. UDLP-GSD has developed a proprietary modification of the track super element in DADS that results in a more efficient and effective dynamic analysis. This modification would need to be in place in the TVWS/DADS capability to support routine dynamic analysis at UDLP-GSD.

DSLP - DSLP exhibits some sensitivity to the manner in which bar elements are added to the finite element model. Using PATRAN, bar/beam elements require a defined orientation; one option employs existing nodes to define a bar xy plane. If this method is used, however, the PATANS translator defines new nodes which causes DSLP to fail in the ANSYS stress coefficient calculation. A successful ANSYS run could be obtained only after the bar real definition was modified to show bar orientation by angular call out.

Performance of crack initiation computation at a particular node employs the dynamic stress superposition method in DSLP. DSLP has a capability to plot dynamic stress history for a node of interest, however, due to unknown circumstances, the plotting package did not work in the software installed on UDLP-GSD's platform. This did not impact the determination of fatigue life, only the ability to view and compare various nodal stresses prior to crack initiation analysis.

DSO - A few limitations and technical problems were encountered in setting up the design optimization problem and performing sensitivity/trade-off analyses. For one, the DSO capability sets requirements on the type of finite element models that can be employed and the methods used to develop them. Plate models using quadrilateral elements based on geometry curves and hand inputs using nodal coordinates are typically employed at UDLP-GSD. Sizing optimization in the DSO requires triangular plate elements that are directly mapped into the geometry or patches. Shape optimization not only requires direct mapping of elements to patches, but all patches must have lines which define their outlines. These requirements restrict the complexity and the size of models that can be used for analysis.

Most other difficulties consisted of minor operating bugs that nevertheless resulted in some system crashes during computation. Some computational errors and operating failure occurred due to the DSO's ability to analyze relatively smaller models than are typically developed at UDLP-GSD. CCAD was able to quickly determine the source of the problems and update DSO module capabilities to handle larger models.

Conclusion

During the performance of the application exercises at UDLP-GSD, it became readily apparent that to take full advantage of the TVCE capability, engineers would require in-depth experience in dynamic analysis,. design optimization, and fatigue analysis. However, if engineers have a fundamental working knowledge of PATRAN and DADS, and were performing basic analysis tasks, much less specific knowledge is required. Particularly in TVWS, basic dynamics analyses can be accomplished fairly quickly given the existing level of tutorial documentation and user interface prompts. Design optimization and fatigue analysis appear to require a more extensive theoretical background to set up the analyses and understand the results. In addition, the types of analyses that can be performed in DSO and DSLP vary to a large degree, restricting the ability to develop a system based on existing model templates that can be used in a "cut and paste" fashion.

Despite current limitations, the TVCE environment, when fully developed, will have a place in the suite of computer systems in the engineering environment at UDLP-GSD. The system database will be useful to store models and analytical data by vehicle family. As the system is utilized, engineers will be able to access the latest vehicle models rather than polling coworkers to determine where the latest version resides. With model and dynamic information available in the system, fatigue and optimization analyses will be easier and more rapidly performed.

General Dynamics Land Systems Division

During the period in which the contractors were exercising the TVCE environment, General Dynamic Land Systems Division (GDLS) had been engaged for some time in the development of a new tracked vehicle concept. This concept vehicle was the focus for the GDLS TVCE application exercise. Since the installation of the TVCE environment at GDLS occurred during the concept development stage of a new vehicle, however, a propitious opportunity was afforded to compare GDLS's existing design development practices with the TVCE development methodology. As such, the following presents GDLS's "As-Is" development in comparison with the "To-Be" process implementing the CCAD's TVCE capability. Due to the proprietary nature of the vehicle application, no description of specific vehicle characteristics or configuration is included in the following.

Exercise Scenario

As-Is: Vehicle dynamics analysis was performed using the DADS software in a stand-alone configuration.

The dynamics analysis effort consisted of the development of a full vehicle model to be used in the performance of bump course ride analysis. The model was constructed using the DADS pre-processor with a few modifications made to the verbose file. A total of five simulations were performed for various terrain conditions and velocities. Chassis force and torque time history data was extracted from each dynamic analysis run and transmitted to structural analysts to perform the necessary stress analysis and structural optimization. MSC/NASTRAN is the FEA tool employed at GDLS for stress analysis.

The NASTRAN and PATRAN software codes resided in the same hardware platform as the DADS server. Force and torque time history data was transmitted to the structural analysts by simply copying files through UNIX commands to the NASTRAN user's file directory. Stress analysis in the "As-Is" process was initiated with the development, using PATRAN, of a partial model of the side wall of the vehicle chassis. Design performance measures were defined for maximum stress and deflection of the side wall plates. Maximum deflection could not exceed 0.5 in. and von Mises stress could not exceed the yield limit of the material. A static stress analysis was performed using NASTRAN after extraction of load and torque data from the DADS analyses and applied to the FE model. Results of the stress and deflection analyses indicated that design changes to the side wall suspension interface were necessary to comply with the defined performance measure constraints. The principal design changes suggested consisted of reinforcement of wall ribs and/or increasing the thickness of wall plates. Minimizing the total weight of the structure was also an objective of design change.

After several trail and error attempts at increasing plate thickness and varying rib position, a design was obtained that met all defined performance measures. The design was not viewed as optimum since further reduction in stress, deflection, and weight could have been achieved had more time been available for this effort, or had a stress optimization tool been utilized. Approximately 350 man/hours were expended over the course of four to five weeks to complete the "As-Is" design process.

To-Be: Once the TVCE software was successfully installed on GDLS hardware, the "To-Be" process was initiated in TVWS by creating a file directory and copying all TVWS template files. The TVWS server was executed and a user catalog was created through the copy/paste function of TVWS. Vehicle catalog blocks were created, including terrain, hull, test scenarios, and test plans. The Test Scenario defined vehicle speed and terrain profile. The Test Plan defined execution time, test description, and path to test_scenario. Mass properties for vehicle model components were incorporated using TVWS capabilities, although mass property data defined using CCS was imported from the global DDS in a later session. After all necessary updates to TVWS template files were implemented, the DADS run was launched. The process initiated the TVWS DADS Input Generator to assemble the DADS input data file from existing DADS verbose files. Dynamic results were obtained and exported to the global DDS for use in DSO and DSLP analyses.

Structural analysis was initiated in DSO with the construction of a PATRAN model of the side wall structure of the vehicle. Existing PATRAN quadrilateral FE models could not be employed due to the triangular element restriction in DSO. The original vehicle model was transferred into PATRAN from the CADDS database using the IGES Translator. The model contained 909 elements and 490 nodes, and included the bottom plate (with bends) front glacis plate, sidewall, reinforcing ribs, top plate, and suspension unit hull opening. Updates to the model were performed outside the DSO using the 'patint' command. NASTRAN bulk data, necessary for the FEA run, was also created. The PATNAS translator was used to generate NASTRAN input data from the neutral file.

The FE model was then transferred to the DSO workspace. Prior to launch of the NASTRAN static analysis, model design parameters were defined for 37 patches using the 'parameterization' and 'linkage' options in the DSO sizing optimization capability. Five groups of patches with five independent thicknesses were defined as design parameters. The NASTRAN static analysis was then launched from DSO; a system failure occurred, at which point exercise of the DSO capability was terminated.

DSLP and CCS workspace capabilities were only minimally exercised due to a lack of sufficient time and resources to complete the exercise.

Evaluation

TVWS - Although definition of the vehicle model using the TVWS capability is somewhat complex, once the model has been developed, execution of the dynamic simulation and analysis is fairly simple and helps to expedite the design process. An estimated 30-40% in time savings was obtained for this exercise in comparison with conventional methods using DADS pre- and post-processors. Actual time savings was not representative, however, since the model was not developed from scratch in TVWS. Actual time savings will be dependent on the number of test profiles and the number of vehicle concepts employed in the design and evaluation process - the greater the number of test plans and scenarios, the more time and cost savings can be obtained using TVWS.

DSO - Although problems with the NASTRAN/DSO interface prohibited the successful completion of the structural analysis, the general feeling is that the DSO capability adds a new dimension to the stress analysis environment. GDLS has recognized the benefits offered by the use of the DSO, in terms of increased efficiency and risk reduction to design development, and plans to invest internal resources to bring this capability or other stress optimization tools to full operational status in the near term.

DSLP - As with DSO, it appeared that the FEA interface was designed principally for compatibility with ANSYS rather than NASTRAN. As such, utilization of DSLP for this exercise was quite limited. Although a thoroughly objective opinion cannot be presented, the DSLP process appears to be lengthy and require experienced fatigue analysis personnel to execute. Whereas the DSLP presents added capability to the existing CAE tool environment, the supplementary benefit may not justify the investment required to attain full operational status in-house.

Conclusion

The TVCE environment does accomplish CE objectives by linking dynamics, stress, and reliability perspectives, while adding new capability with respect to existing CAE tools. Increased efficiency is evident in system design and evaluation through application of the TVCE capability. It is estimated that 10-20% less time will be required to obtained stress-optimized designs using the DSO capability over conventional methods. Although the TVWS capability did not present any significant advantage in dynamic model development, definition and execution of dynamic simulation was enhanced. The DSLP presents a new capability for reducing risk in the design process, since no current life prediction capability exists at GDLS.

In general, however, the TVWS environment requires engineers of above average technical knowledge and ability to operate, and also requires experienced UNIX operators to support. Unlike most existing commercial software capabilities, e.g., DADS, ADAMS, ANSYS, NASTRAN, the DSO and DSLP analysis processes are fairly lengthy and should be simplified. In addition, the TVCE environment appears to require significant customization in order for it to be effective for at any one company. The difficulties encountered during evaluation of the TVCE environment are viewed as moderate, however. Overall, the TVCE represents a considerable technical accomplishment, and GDLS is well confident that the environment exhibits great potential to support implementation of a CAE-based Concurrent Engineering process in any industrial firm.

Lessons Learned

From the reports provided by the three contractors performing validation exercises, it is apparent that operational deficiencies, rather than conceptual or computational errors, constitute the majority of the difficulties present in the TVCE environment. It is evident that a higher degree of "user-friendliness" is required in order for a capability such as TVCE to be effective in an industrial setting, in particular with respect to eliminating complexity in analysis procedures and enhancing flexibility in dynamic and structural model development. As further development and refinement of workspace capabilities has been ongoing at CCAD since these validation exercises, many of these problems have been resolved and implemented in later versions of the workspace software. For example, all three contractors expressed concern regarding the triangular finite element limitation inherent in DSO. The current DSO capability fully supports analysis of quadrilateral finite element models, and both the DSO and DSLP interfaces with NASTRAN and ABAQUS FEA analysis routines have been strengthened.

Of principal concern to CCAD personnel were some difficulties evident in the implementation of data sharing, transfer, and data modeling in the TVCE integration functionalities. At this stage of development of the simulation-based Concurrent Engineering tool environment, it is evident that a more seamless and transparent means of defining and accessing the global data model to support CAE modeling and analysis is required. As will be seen in Section IV, this issue has been addressed in development of the next generation integrated tool environment architecture.

IV Interim Technology Developments

In the interim between the conclusion of the DICE Phase 4 effort and the initiation of DICE Phase 5, a parallel effort in CE environment development was occurring at the Center. This effort, *Simulation Based Design for Military System Supportability and Human Factors*, was sponsored by the Defense Modeling and Simulation Office (DMSO) to extend and refine concepts and technologies developed under DICE Phase 4 to enable CE for supportability and human factors. Center achievements under this effort resulted in a conceptually redefined integration methodology with respect to maintaining consistency between CAD product and CAE engineering analysis models. As well, two new CAE simulation workspace capabilities and numerous extensions to existing software tools were introduced in the integrated environment. As these technology refinements have played a significant role in the understanding of collaboration among CE environment users in DICE Phase 5, a brief overview of the DMSO workspace tools and environment architecture is provided in the following.

DMSO Project Effort

The DMSO project proposed the development of a qualitatively new simulation-based CE environment for use by all three military services, to bring realistic consideration of military system supportability and human factors into the early phases of the design process. The DMSO Integrated Concurrent Engineering Environment (ICEE) has built upon investments in innovative simulation technology applications by the National Science Foundation, the US Army, NASA, industry, and continuing research in the field of advanced computer aided engineering and driving simulation by The University of Iowa. The DMSO project has taken advantage of emerging methods and software for anthropomorphic modeling and simulation in support of maintainability analysis, facilities and methodologies enabling engineering-level consideration of the interaction between military personnel and vehicle systems, and recent advancements in computational mechanical system reliability analysis. Specific objectives defined for this effort included:

(1) Broaden the scope of applicability of simulation-based design to ground tactical vehicles, material handling equipment, construction equipment, and maintenance equipment that are of concern to all three services.

(2) Incorporate tools for maintainability evaluation and design for maintainability into a [CE] environment, using advanced anthropomorphic modeling methods and computer graphics that permit consideration of protective clothing, restricted vision, and special tools.

(3) Enhance the ability of military personnel-in-the-loop simulation under development to create realistic duty cycle information needed in design for durability and reliability early in the design process.

(4) Use simulator generated duty cycle information, heretofore available only after hardware has been developed and tested, early in the design process when design latitude remains to optimize military equipment for durability and reliability.

(5) Create and transfer to industry a portable and maintainable software system that is designed, implemented, and tested by organizations in all three services, using modern software engineering principles and computer-aided software engineering (CASE) tools and prevailing software standards.

The approach taken under this effort addressed extension of the DICE Phase 4 methods and software to support design development of multiple vehicle and general mechanical systems. The TVCE environment architecture was broadened as shown in Figure 4.1 to support a wider range of applications and design perspectives. The role of the CAD software component was targeted for special consideration in order to provide a capability to define components, structures, and other design characteristics of military equipment. The CAD-based product data model was extended to support part catalogues for supportability and human factors, and refined to enable transparent access to design data for all functional workspace capabilities.

Additional and extended tool capabilities included the development and implementation of the Maintainability Analysis Workspace, the Simulation and Visualization Analysis Workspace, and the Durability and Reliability Analysis Workspace. Wrapper functionalities were defined and implemented for each of these

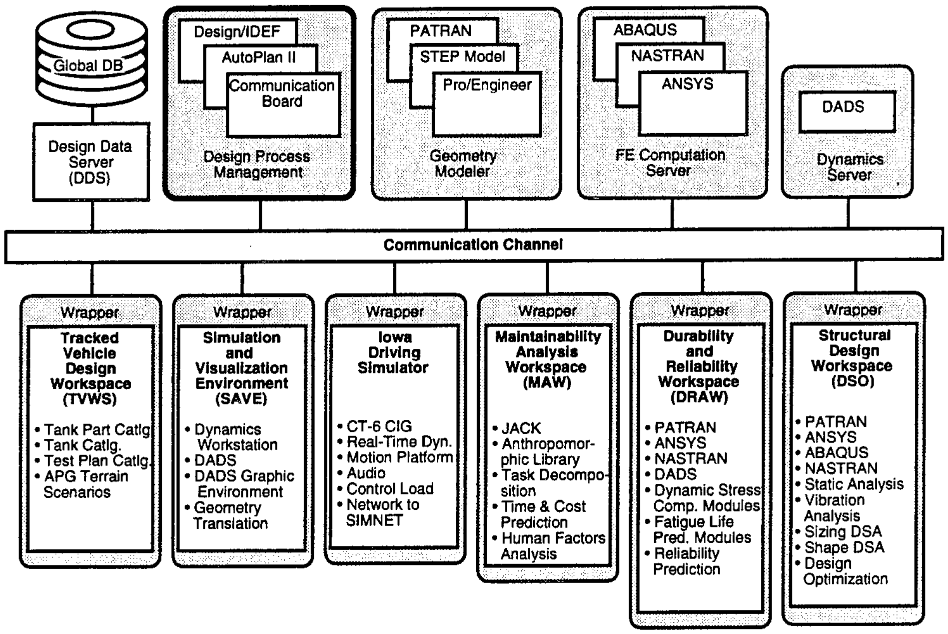


Figure 4.1 The Integrated Concurrent Engineering Environment

tool capabilities, addressing communication of information associated with duty cycles, loads, and design for human factors. A brief overview of the concept, function, and implementation in the ICEE for each of these tool capabilities is provided as follows. This section concludes with a detailed overview of refinements to the ICEE integration architecture.

Additional/Extended Workspace Capabilities

Simulation and Visualization Environment

The Simulation and Visualization Environment (SAVE) provides the dynamics modeling, simulation, and animation capability in the ICEE for wheeled and general mechanical systems. Corresponding to the function of the TVWS in the TVCE environment, SAVE provides the dynamic engineer with the ability to construct dynamic models, define a simulation scenario, including terrain characteristics and system operating parameters, launch a dynamic simulation in DADS, and analyze the results using animation and data reporting tools. The SAVE capability also provides the engineer with flexible body dynamics modeling, simulation, and animation through the incorporation of the Dynamics Analysis Workspace software from the DSLP environment developed under DICE Phase 4. In this manner, a unified, general purpose dynamic modeling and simulation capability is established in the ICEE environment that supports comprehensive consideration of rigid and flexible body dynamic performance for design development of the mechanical system.

The SAVE workspace provides four principal functions in the ICEE:

- Dynamic analysis of mechanical system performance.
- Generation of duty cycle information for reliability and life prediction.
- Generation of load history data for structural design sensitivity analysis.
- Hi-fidelity reproduction of driving simulation results.

Using the SAVE wrapper, body, joint/force element, geometry, and mass property data can be imported from the ICEE global database and used to define high fidelity dynamic system models for both rigid and flexible-body dynamic analysis. Construction of dynamic models is performed using the Center-developed Dynamic Workstation (DWS), a graphics based tool in the SAVE environment that enables the engineer to visually assemble selected bodies and define joint connections. [34] The DWS employs a connectivity graph [35] to enable the engineer to define a topological model of the mechanical system (See Figure 4.2). Connections (joint types) are defined between pairs of selected bodies by specifying joint definition frames, one frame for each body, and a type of joint. Assembly of the system in an initial configuration is carried out automatically in the DWS; a joint exercise utility in the DWS allows the engineer to kinematically verify the range of motion of the assembled bodies. Model definition is completed using a modified Newton algorithm with Moore-Penrose Pseudoinverse [36] in the system configuration adjustment utility in DWS, to close loops in a decoupled mechanical system model. In contrast to the template-based system assembly method used in TVWS, the SAVE/DWS environment permits the engineer to develop highly detailed dynamic models for any mechanical system configuration.

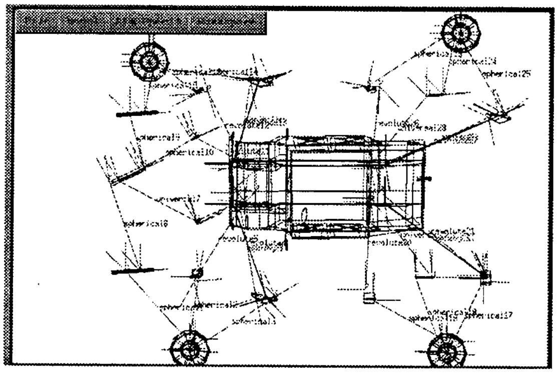


Figure 4.2 Topological Graph of Vehicle Assembly

Definition of simulation scenarios is accomplished much in the same manner as in the TVWS, with terrain profiles, test plans, and initial conditions specified in the DADS pre-processor utilities. Dynamic simulation and analysis is accomplished using the commercial DADS computation engine, with simulation results visualized using the DADS Graphical Environment (DGE) 2D plot and 3D animation utilities.

Flexible body dynamic analysis is carried out in SAVE using the Dynamics Analysis Workspace initially developed for the DSLP environment. The flexible body dynamic analysis methodology remains based on modal synthesis for component analysis, but has been adapted to take advantage of recently implemented modal synthesis analysis capabilities in the commercial DADS code. The commercial finite element modeler PATRAN is employed to load and create FE models and display deformation modes to assist engineers in selecting proper modes for dynamic analysis. ANSYS and NASTRAN codes are employed to carry out FE analysis. DADS is used to compute inertia relief forces for analyses employing both static attachment and rigid body modes and perform modal synthesis analysis using the DADS Intermediate Processor. Four types of flexible body modeling and analysis methods are supported in SAVE: (1) normal vibrational mode analysis, (2) static and vibrational mode analysis without rigid body modes, (3) static attachment and vibrational mode analysis with rigid body modes, and (4) static attachment or constraint mode analysis only.

Dynamic analysis generated by SAVE includes load, position, velocity, and acceleration for specified bodies in the dynamic model, as well as deformation modes and duty cycle information to support reliability and component life prediction analysis.

Under the DMSO and subsequent project effort, the SAVE capability has been developed to provide an interface with real-time driving simulation. A methodology (see Figure 4.3) for using DWS model editing capabilities has been developed for creating vehicle system models for NADSdyna applications; NADSdyna being the recursive dynamics formulation code used in the Iowa Driving Simulator (IDS) for real-time dynamic simulation. At present, NADSdyna mechanical system models developed using this methodology are implemented in an off-line (non-real time) simulation capacity only. Research is currently on-going in the development of model translation schemes from high resolution dynamics models (off-line) to lower resolution models compatible with achieving real-time dynamics computation for use in the IDS.

Figure 4.4 illustrates the SAVE software architecture as currently applied in the ICEE. The SAVE capability employs remote execution of the DADS system with automatic conversion of results files and data. A model-based scheme for file organization and storage is employed in SAVE; all data files are associated with specific models. File storage in SAVE is open permitting SAVE users to copy files in or out of model subdirectories and have changes immediately recognized by the SAVE environment. As a result, a higher level of flexibility is obtained in SAVE operation by eliminating the need for system data files or other hidden files to launch the SAVE capability.

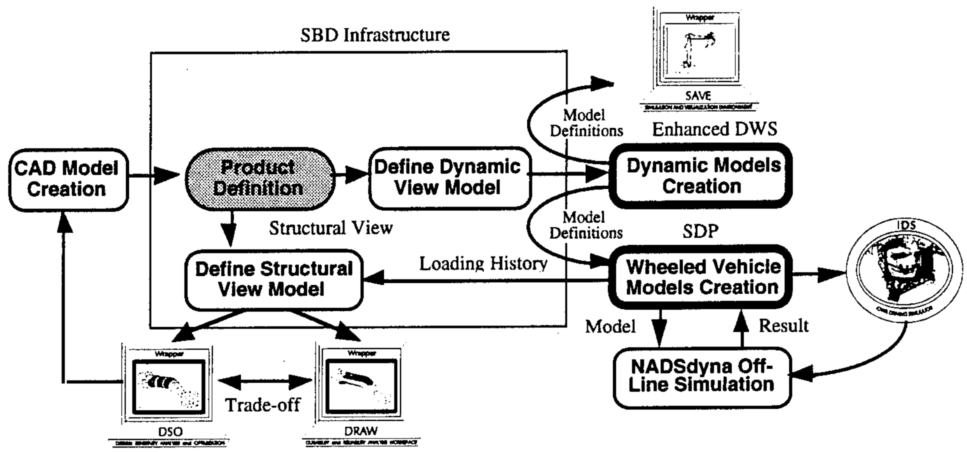


Figure 4.3 SAVE/IDS Engineering Scenario

Directory structures are created in each model that allow specific application codes, i.e. DWS or DADS, to view the model directory in the format defined for that application. In this manner, file access is greatly enhanced in the execution of specific codes. All operations performed during the use of the SAVE environment employ the current model principle. This utility is designed to simplify user interaction by eliminating the need for the user to repeatedly specify the model for each command.

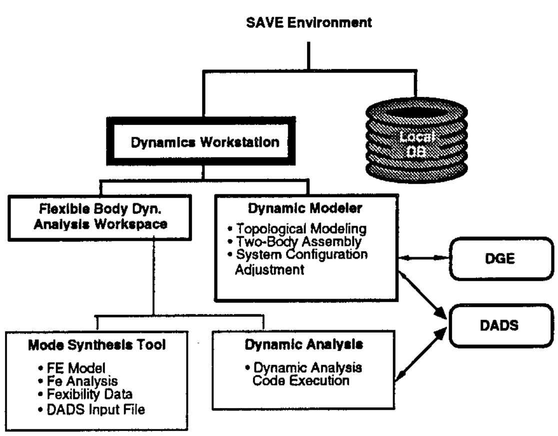


Figure 4.4 SAVE Software Architecture

Durability and Reliability Workspace

The Durability and Reliability Workspace (DRAW) extends the DSLP capability developed under DICE Phase 4 to employ load history data for computational analysis of fatigue failure in structural and dynamic components and probabilistic estimation that failure will not occur, i.e. reliability analysis. The durability analysis portion of the DRAW embodies the basic dynamic stress computation, fatigue crack initiation, and fatigue crack propagation methodologies and computational algorithms employed in the DSLP, discussed previously in Section III, with a number of computational and methodology improvements. The principal new capability in DRAW supports reliability prediction of selected classes of mechanical system parts and components.

Under DICE Phase 4, the fatigue life methodology was developed using a uniaxial local strain approach based on linear elastic stress time histories for fatigue crack initiation, and a stress intensity approach for crack propagation fatigue life prediction. During the evolution of the DSLP capability into the DRAW environment, durability analysis (fatigue life prediction) was upgraded to support the inclusion of the reliability analysis capabilities of the DRAW workspace. Modern fatigue life prediction methods employ knowledge of each stress and strain component throughout the loading history, at fatigue critical locations. Therefore, a procedure for utilization of multi-axial stress and strain estimation has been developed and implemented in the DRAW. New capabilities supporting this procedure include a multiaxial local strain approach, and critical zone and point searching, among others. The most recent developments in DRAW include the incorporation of an elastic-plastic stress-strain algorithm to provide a capability for computing a preliminary analysis of initiation life for all surface nodes in an FE model. This capability enables display of an FE model life contour for structural components that represents realistic magnitudes of stresses and strains at fatigue critical locations ("hot spots").

Reliability engineering techniques are used to estimate failure probability during mission time under prescribed conditions. The scope of reliability prediction segment is the systematic reduction, and/or control of potential hardware, software, and human failures throughout the life of a mechanical components under dynamic loading.

Reliability is defined as the *probability* that a component, device, equipment, or system will perform its intended function for a specified period of time under a given set of conditions. The obvious problems are: (1) the acceptance of the probability, which gives a numerical input for reliability assessment, (2) the required function, which is the concept of adequate performance for system parameters that deteriorate slowly with time, (3) the judgment necessary to determine the proper statement of environmental conditions, and (4) the duty time or the mission time which is the time period for certain service demanded of the item. There are four main parts in this segment:

- Peak-valley editing and cycle counting
- Bearing reliability analysis and assessment
- Gear reliability analysis and assessment
- Spring reliability analysis and assessment

The editing and cycle counting procedures are used to count the number of cycles in a dynamic loading block, and decide the maximum loading of each cycle for predicting reliability.

In assessing reliability, it is necessary to define and categorize different models and their corresponding probability statements of component failure. Based on *American Anti-Friction Bearing Manufacturers Association* (AFBMA) standards, the failure mode for a bearing is fatigue spalling and its failure probability density function is the Weibull distribution. The reliability model used for bearing reliability analysis is a failure rate model. According to the information provided by AFBMA, the effective load of bearing is calculated. Using the effective load, the rating of life expectancy for each cycle that is associated with 90 percent reliability is predicted. The Palmgren-Miner linear damage rule is used to sum the damage of the whole block. The adjustment factor for reliability is also provided that allows the user to predict the desired reliability or life. The flow chart for the bearing reliability analysis is depicted in Figure 4.5.

The limit states reliability model is used to predict gear and spring fatigue reliability. The governing failure mode of a gear is fatigue cracking of gear teeth roots. Standard AGMA data is used first to calculate the stress at the gear tooth root. The spring (helical spring and torsion bar) failure mode is also fatigue crack initiation. The performance criterion approach is used to evaluate the relationship between reliability and fatigue limit/ultimate stress. Using this information, the life of a gear associated with a certain reliability will be predicted. The flow chart of the gear and spring reliability analysis are also shown in Figure 4.5.

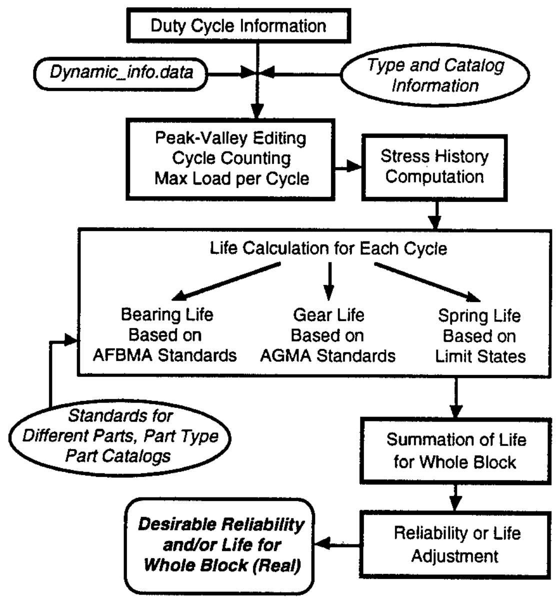


Figure 4.5 Reliability Prediction Workspace Flow Chart

The revised software framework for the DRAW workspace is illustrated in Figure 4.6. The DRAW architecture exhibits two major modifications from the DSLP architecture developed under DICE Phase 4, the removal of the Dynamics Analysis Workspace (DAW) and the incorporation of the Reliability Analysis Workspace. As described previously for the SAVE capability, the DAW flexible body dynamic analysis tools have been absorbed into the SAVE architecture to create a unified dynamics simulation

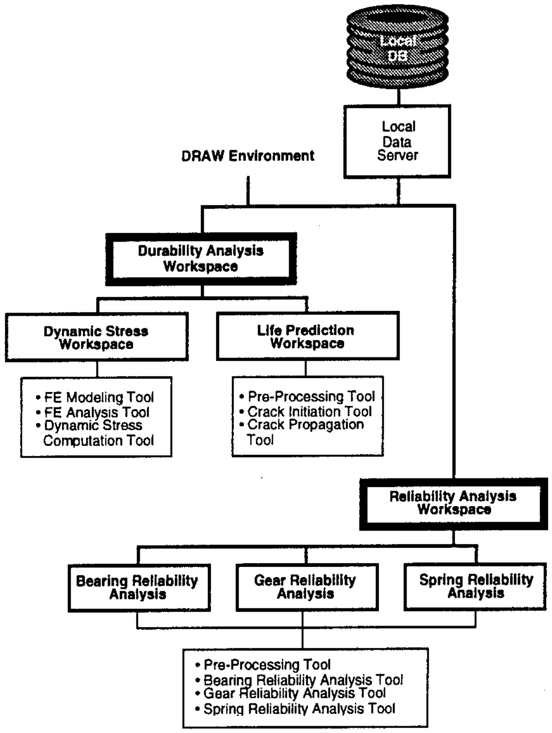


Figure 4.6 DRAW Software Architecture

workspace for general purpose rigid and flexible body analysis. The addition of the Reliability Analysis Workspace in DRAW has been enabled through the extension of the local data model for component reliability prediction as depicted in Figure 4.7.

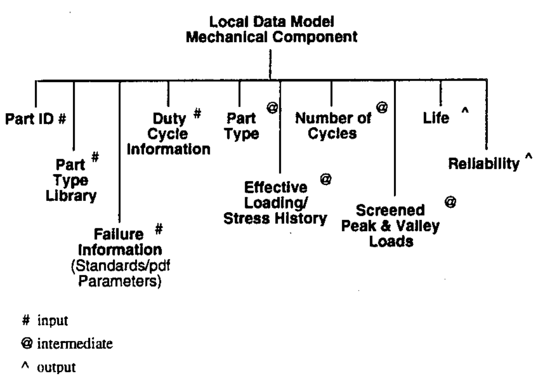


Figure 4.7 Component Reliability Prediction Local Data Model

As in the DSLP process, the DRAW analysis process entails four distinct segments: (1) finite element analysis, (2) dynamic stress computation, (3) fatigue life prediction, and (4) reliability prediction. Correlated to the DRAW architecture illustrated in Figure 4.6, these analysis segments implement the DRAW sub-workspaces as shown in Figure 4.8.

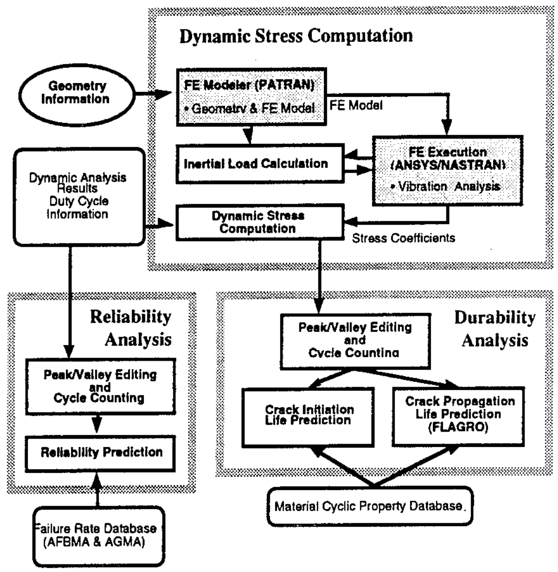


Figure 4.8 DRAW Analysis Process

Maintainability Analysis Workspace

Maintainability analysis of mechanical systems is usually carried out using wooden mockups and other physical models to demonstrate/verify pull space for machinery disassembly and maintainability access requirements. Typically, maintenance and support analysis of this type does not occur until late in the design process when design modifications entail high costs and lengthy implementation. The cost of design iterations resulting from maintenance and support considerations can be minimized, and even totally avoided, by incorporating good-practice design for maintainability rules and appropriate analysis early in the design process. The application of computerized modeling and simulation presents an effective means for enabling rapid, cost effective, maintainability analysis, promoting interaction between maintenance analysts and designers, and defining customer support requirements early in product development. Poor design decisions resulting in maintenance problems are thus avoided, as well as the need for expensive physical mock-ups.

The Maintainability Analysis Workspace (MAW) is CAE modeling and simulation tool capability intended to enable maintainability design consideration in the integrated CE environment by allowing for evaluation of mechanical system design maintainability, identification of design features that cause maintainability problems, and recommendation of design modifications to eliminate those problems. MAW capabilities supporting maintainability analysis include:

- Importation and preparation of the mechanical system design model for maintainability analysis.
- Definition of quantitative and qualitative maintainability requirements.
- Definition of maintenance personnel that meet anthropometric requirements.
- Application of the JACK® human modeling software for design, simulation, animation, and human factors analysis of maintenance tasks.
- Generation of a sequence of human motions and maintenance activities that fully describe the maintenance task.
- Prediction of the duration and cost of the given maintenance task.
- Simulation and animation of multiple maintenance technicians.
- Assessment of design maintainability and recommendation of design modifications to eliminate maintainability problems.

Figure 4.9 illustrates the maintainability analysis procedure defined for the application of the MAW capability. Maintainability analysis is conducted on a CAD-based design representation that includes both geometry of the mechanical system, fastener elements, and the external environment, and nongeometric data such as system, fastener, and tool orientation, component mass, assembly information, and access constraints. The maintainability model consists of a P-surf representation derived from the CAD product model.

Maintainability analysis in MAW involves the evaluation of issues related to the performance of maintenance tasks that deal with the repair or replacement of a part or subassembly. The basic maintenance task sequence consists of disassembly, access the target component, component repair or replacement, and re-assembly. A specific maintenance task is decomposed to the extent that simulation and animation of maintenance personnel carrying out the task is supported. The maintenance task framework supported

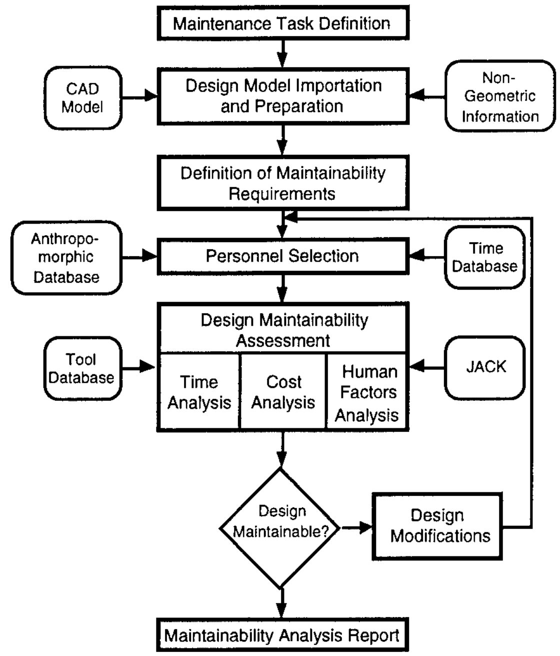


Figure 4.9 Maintainability Analysis Procedure

in MAW allows for a hierarchical decomposition into four levels:

(1) *Maintenance task level* - the basic task sequence; disassembly, replacement, re-assembly.

(2) *Disassembly sequence level* - the target component is accessed by a series of disassembly steps. The re-assembly sequence is assumed to be the inverse of the disassembly sequence.

(3) *Disassembly step level* - disassembly steps are activities aimed at disassembly of a part or subassembly, or to disconnect or disengage two parts. A disassembly step normally involves maintenance activities and motions of the technicians performing maintenance activities.

(4) *Macro motions and Macro models level* - macro models and motions are identified and sequenced for simulation and animation of disassembly steps. Macro models are used to simulate and animate maintenance activities and macro motions represent human motions such as bending, stooping, arm and hand motions.

Maintenance personnel are defined in the MAW capability according to a percentage of the population characterized in an anthropometric human database. The model database contains models corresponding to the 1st, 5th, 50th, 95th, and 99th percentile of both the male and female population. Human model are created using an anthropometric human figure scaling system, called SASS, which is available in the JACK software. An important aspect of the MAW analysis evaluates ergonomic issues, such as lifting, access, strength, etc., with respect to the ability of the selected percentile of the population to carry out the maintenance task.

Although a principal objective of the modern design methodology is to minimize the number of fasteners in a mechanical assembly, removal and installation of fasteners still represents the most common activities in the maintenance task. Consequently, identification of the tools employed in fastening operations, and the subsequent design of the mechanical system to accommodate these tools, is also an important element of the maintenance analysis carried out in MAW. The MAW capability employs an automated tool selection procedure that selects hand tools for a particular fastening operation. Criteria considered in tool selection include selecting a tool that is applicable to the fastening operation, selecting a tool that minimizes the time and cost of the operation, and selecting a tool that satisfies accessibility requirements.

Simulation and animation of the disassembly sequence enables the assessment of human factors issues, identify design features that inhibit performance of the maintenance task, and support prediction of the time required to perform the task. The MAW uses the JACK® human modeling and animation system to display the design model, model maintenance personnel, and simulate and animate human-design model interaction (see Figure 4.10). JACK is a general purpose human modeling system developed by the University of Pennsylvania that provides elemental human motions. Since the maintenance task typically involves hundreds of elemental motions, however, a hierarchy of macro motions has been implemented in MAW to support rapid modeling and simulation of maintenance tasks. A macro motion incorporates several basic human motions to represent a complex motion. For example, the macro motion for changing posture from standing to squatting consists of torso, pelvis, center of mass, left arm, and right arm motions. Elemental motions in JACK are parameterized and represented as macro motions in the MAW capability. A complete list of macro motions supported in the MAW is given in Table 4.1.

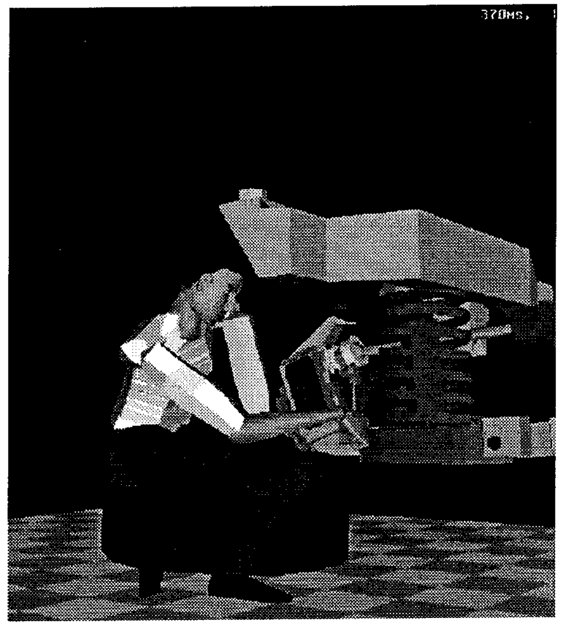


Figure 4.10 JACK Modeling and Simulation

Design development is supported using the MAW capability based on the results of time and cost prediction and human factors analysis. The design objective in MAW is to achieve a design that allows for easy and cost-effective maintenance. Cost, task time multiplied by labor rate, minimization considers such factors as assembly (fastener) configurations and specification of tools. Design change for human factors includes consideration of strength, work clearance, accessibility, and obstacles in the field of view. Design modification from a maintainability perspective generally comprise (1) replace a component with a new or modified design, (2) deleting a component from the design model, (3) moving a component to a new location, or (4) defining a new product configuration.

The MAW software framework is shown in Figure 4.11. Maintainability representation of the mechanical system design is extracted from the global product data model and supplied to MAW. MAW input data include: (1) geometry of the product design and environment in which maintenance operations are to be carried out, translated into a P-surf representation required for the display in the Jack environment, and (2) non-geometric information that involve orientation and location of each component required to assemble the system model in Jack, mass of components needed to support strength analysis, and information about fasteners as input data to the tool selection pro-

Table 4.1 Libraries of Macro Motions

Library	Macro Motions
Walk	
Basic Human Motions	Eye Motion Timed Head Control Arm Motion Timed Hand Control Finger Motion Torso Motion Pelvis Motion Center of Mass Motion Foot Motion Timed Foot Control Heel Motion Human Joint Motion
Human Postures	Stand Bending Climbing Crawling Kneeling on one knee Kneeling on both knees Prone Side Sit Squat Supine Walk Standing to Squatting Standing to Bending Squatting to Standing Bending to Standing
Object Manipulation	Move Object Adjust joint on object Attach Object
Maintenance Primitives	Tighten fastener with tool Loosen fastener with tool Grasp Tool Release Tool

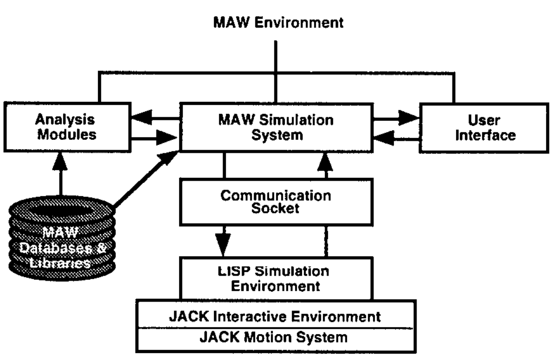


Figure 4.11 MAW Software Architecture

cedure. MAW Simulation System controls the maintainability analysis procedure, prepares input data, executes various analysis modules, and prepares and presents analysis results. Communication socket and Lisp Simulation Environment are used to establish and maintain a two-way communication between MAW and JACK. MAW output data are transmitted to the global database and include the final state of the JACK environment containing the mechanical system model, all information defined and generated in the maintainability analysis, including the complete animation of the maintenance task, and a report file containing the assessment of design maintainability and information on proposed design modifications.

Modification to Integration Architecture

A significant rethinking of the integration methodology and architecture occurred in the evolution from the TVCE environment to the ICEE (Figure 4.1) under the DMSO and subsequent project efforts. Developed in response to DICE Phase 4 contractor evaluation and data support requirements for the tool technologies described previously in this section, the integration architecture supports a more transparent data sharing and modeling methodology from an engineering user perspective. The architecture employs the basic utilities introduced under DICE Phase 4, namely the global database and Design Data Server (DDS), high level geometry modeling and computation servers, the communication channel, and workspace wrapper technologies. User interface technologies have been substantially upgraded, however, to support the Center's application of the "engineering view" concept that allows engineers from various disciplines to view and model the product from their own perspectives. [37] In addition, continuing research and development of the Center's capabilities in parametric-based, multidisciplinary design sensitivity analysis has resulted in extension of the global and local product data models to support both CAD and CAE (global and local data model) parametric geometry design representations.

The infrastructure employed in the ICEE has been designed to enhance correlation of various simulation models with a common CAD product representation, as initiated under DICE Phase 4. A base product definition is created from the CAD model and serves as the common source for definition of engineering views supporting the fundamental design perspectives which employ the CAE tools in the ICEE (see Figure 4.12). Engineering views are derived from the base definition to support subsequent analysis requirements

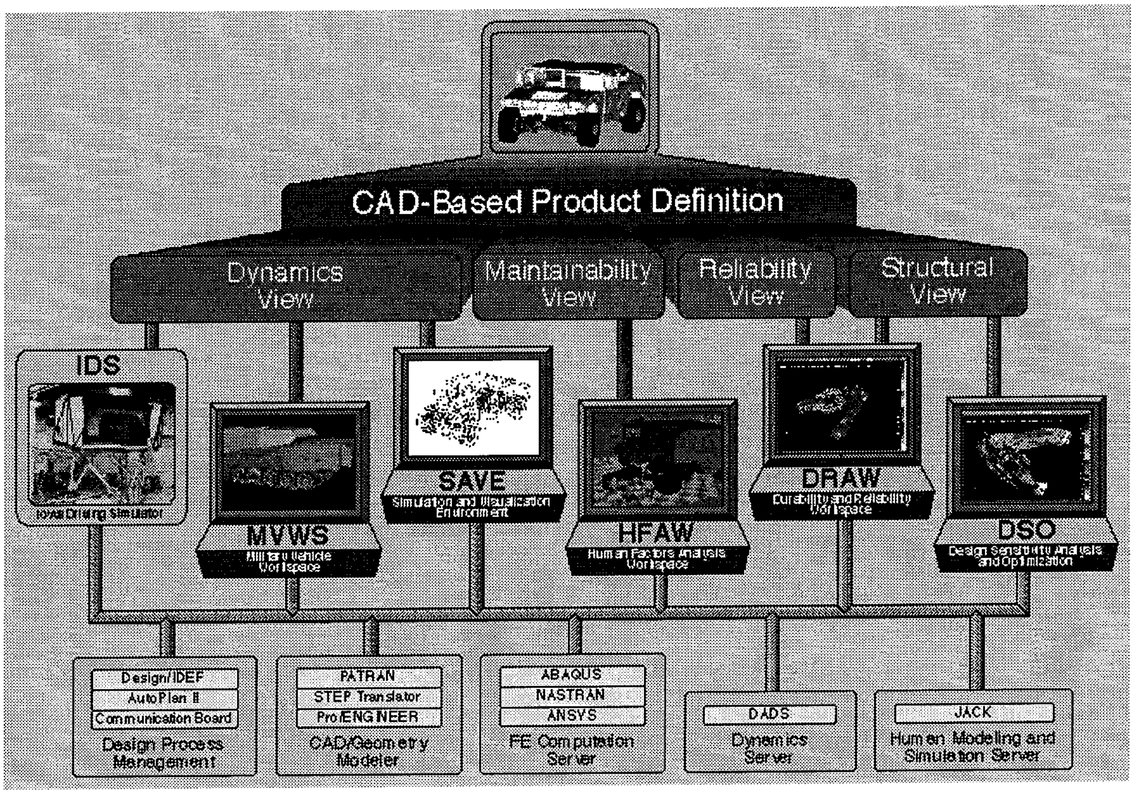


Figure 4.12 ICEE Engineering View Structure

for the workspaces in each engineering discipline. View models are correlated, or mapped, with the base definition to promote design collaboration and can be shared among multiple workspace capabilities supporting the same engineering analysis perspective. Workspace wrapper functionalities have been modified to enhance global data model access and data translation concomitant with the engineering view paradigm. To facilitate data sharing of standard parts, the concept of the handbook has also been introduced in the ICEE architecture.

To support multidisciplinary CAE analyses, a base definition is defined as the common ground among the CAE team members. The base definition contains two major types of information, an entity hierarchy and entity attributes. The entity hierarchy describes how the components of the system are grouped together (see Figure 4.13). If an entity in the hierarchy is an assembly, it can be expanded to display its components or collapsed without showing its components. The entity attributes for a part include mass, center of gravity (C.G.), moments of inertia, material properties, and geometry information. The default coordinate system defined in the CAD model is used as the local coordinate system for the part, with the C.G. reported relative to it. Moments of inertia are, however, reported relative to the C.G. coordinate system whose origin is located at the C.G.; x-y-z coordinates are parallel to local coordinates. A part is assumed to be formed from one type of material. Mechanical system geometry information is maintained in the original CAD format and later transformed to different formats in support of the various simulation model requirements. Parameters used to define the CAD geometry are extracted and employed in later use as a foundation for design parameterization and design trade-off.

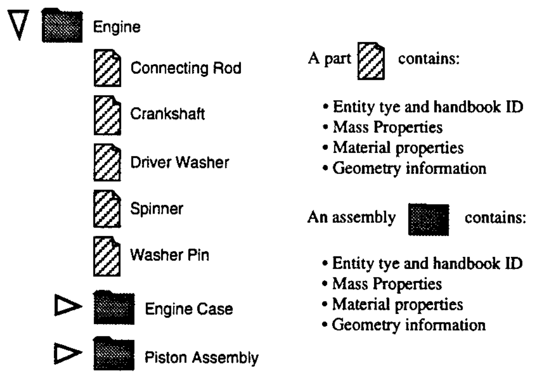


Figure 4.13 Product Base Definition
(Model Airplane Engine Example)

When a welded component is constructed of parts consisting of different materials, the model is treated as an assembly. Attribute information for an assembly differs from that of a part by the addition of assembly information describing the position and orientation of individual components relative to a local reference frame with material property information remaining undefined. Once all hierarchy and assembly information are defined, the global position and orientation of the individual part or assembly is calculated automatically.

The integration infrastructure has been designed to support engineers of different analysis disciplines to create their own simulation models. Because data requirements vary from discipline to discipline, the infrastructure must enable engineers to augment model data in the base definition with discipline-specific data. In doing so, the infrastructure must also promote consistency among model representations so that commonality is maintained and design trade-off can occur across disciplines via the base definition. To address these issues, the concept of engineering views has been introduced in the infrastructure. Engineering views provide an association with corresponding analysis disciplines that allow data augmentation to occur in a manner natural to the engineer. Furthermore, data created in the engineering views can be shared among engineers performing analyses from the same perspective. Consequently, duplication of effort is minimized.

Engineering views enable the maintenance of a consistent product data set for the mechanical system being evaluated. For each analysis discipline, a mapping between the view model and the base definition is established (see Figure 4.14). All engineering models, together with their respective simulation results, and the CAD model (the base definition) are correlated through these mappings, allowing meaningful communication among CAE analysts and design trade-off across disciplines. Mappings can also be employed to provide a foundation for automating engineering model (re)creation during iterative design analysis. Once a design change is proposed, each engineering workspace must re-evaluate the performance of the new design; heretofore, the engineering model has been regenerated from scratch, consuming a great deal of effort and resources. Mappings support a mechanism for retaining relationships between the engineering model and the base definition that can facilitate analysis model recreation, and therefore accelerate the design cycle. A principal objective of Concurrent Engineering is then achieved.

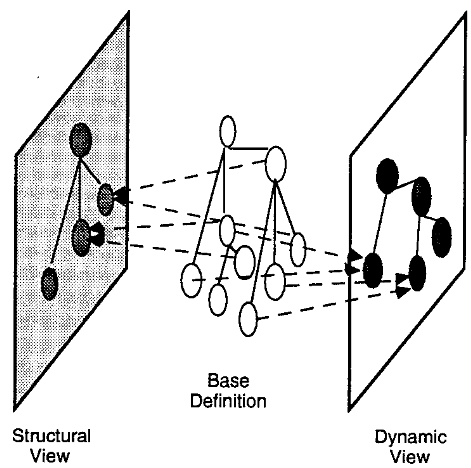


Figure 4.14 Base Definition/Engineering View Mappings

View model creation entails the major bottleneck during the simulation-based design process. A portion of the simulation model data can be created from the CAD model automatically, however, some of the model must be created by the engineer. For example, in definition of the dynamics view, the assembly hierarchy defined in the CAD model may not be suitable to represent the multibody mechanical system. From a dynamics perspective, for example, parts or assemblies may need to be regrouped into bodies and then connection joints, allowing relative motion between bodies, need to be defined and included in the dynamic model. Once the regrouping is performed, the composite mass, C.G., moments of inertia, and assembly information can be automatically calculated based on individual component mass properties and assembly information. Joint types and locations need to be specified by the dynamics engineer, however.

Certain data translations, such as dynamic animation file generation an IGES file, for example, may also be required during view model creation. View model creation has been defined for all analysis perspectives employing ICEE workspace capabilities, including dynamics, structural, reliability, and maintainability views.

With the implementation of the engineering view concept, the functionalities of the CAD/CAE Services (CCS) workspace developed under DICE Phase 4 are absorbed into the integration architecture. As a result, a greater degree of flexibility is enabled in CAE analysis by allowing engineers to independently develop product views rather than being limited to the product representation created using the CCS, and yet still maintain a requisite degree of consistency with the CAD design representation. Data translation utilities, such as the IGES Translator, and specification of additional CAE data are employed at the discretion of the engineer who requires the data, enabling each analysis discipline to determine the configuration and properties of the required model. The need for a single user familiar with all aspects of model requirements in all disciplines, as was the case under the DICE Phase 4 CCS development effort, is eliminated.

The use of handbooks as repositories for standardized parts information and material properties has been introduced in the ICEE. The current implementation of handbooks entails four categories. The first category includes screw, nut, and bolt information used in maintainability analysis. The second category contains gear and bearing information required in reliability analysis. Curve data describing the relationship between force and displacement or force and velocity, employed in dynamic simulation, constitutes the third category. Finally, material property data, including Young's modulus, Poisson's ratio, density, etc., are contained in the fourth category in support of structural and fatigue life prediction analyses. Standard part and material information identified in CAD models is integrated into the handbook when the model is imported into the environment. When a model is imported containing any of the standard parts described above, the ICEE will automatically verify if the part exists in the handbook, and if so, establish a link between the handbook and the part in the imported model.

The integration architecture and global data model have also been modified in anticipation of multi-disciplinary parametric design sensitivity analysis and trade-off methodologies whose development has been initiated under DICE Phase 5 (see Section V). In the ICEE, design parameters are associated with the dimensions of features in the parameterized CAD models. Design parameters are considered as attributes of entities in the base definition, and remain associated with an entity when regrouped in engineering views to create assemblies. The feature-based design parameters serve as a common language to support design trade-off across engineering disciplines where relevant performance of the mechanical system is measured. Wrapper functionalities have been modified to support transmittal of sensitivity coefficient data to the global database.

The software framework architecture of the ICEE is illustrated in Figure 4.15. The software framework is divided into three categories based on the functionalities supported: tool navigation support, data management, and design collaboration support.

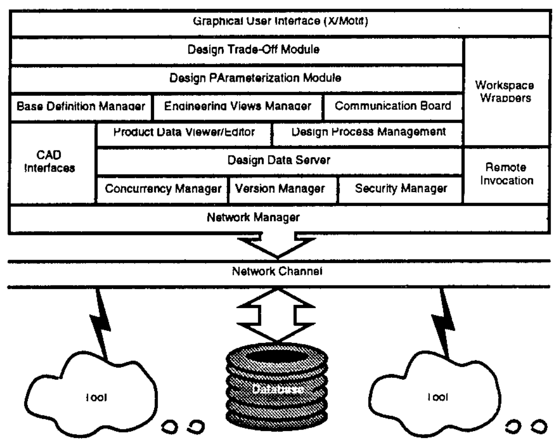


Figure 4.15 ICEE Software Framework Architecture

Tool navigation support includes the graphical user interface, remote tool invocation, and the network manager. The graphical user interface provides a central entrance, a well-organized menu structure, and user-friendly window layouts of the integrated environment and is developed using the X-Windows/Motif standard. The remote tool invocation utility allows the tool server to reside on a remote site and still respond to the user's input as though it was on the local machine. The network manager provides user transparent services to support distributed data transmission and remote procedure execution. Together the three modules facilitate smooth navigation in the integrated environment.

Data management includes CAD interfaces, the base definition manager, engineering views manager, workspace wrappers, the product data viewer/editor, the design data server, the version manager, the concurrency manager, the security manager, and the distributed database. The CAD interfaces are used to extract model information from CAD systems and to propagate design changes back to CAD systems. The base definition manager supports construction of the base definition from CAD model. The engineering views manager supports view model creation and invocation of workspace wrappers. The workspace wrappers prepare product data in the format required by the engineering workspace and also retrieve analysis results from the workspace. The product data viewer/editor allows engineers to browse and augment the product data either in the base definition or the engineering views. The design data server stores all design related data and provides access service to other components in the environment. The version manager, concurrency manager, and security manager maintain data integrity. The distributed database handles physical data storage.

Finally, the design collaboration utilities developed under DICE Phase 5 are implemented in the integration software framework and include the communication board, design process management, design parameterization, and design trade-off. The communication board provides a means for CAE team members to communicate about design tasks. Design process management allows product specific process definition and facilitates project tracking. The design parameterization module assists engineers to identify design parameters to facilitate effective design evaluation. The design trade-off module collects performance evaluation information from the engineering workspaces and assists engineers in obtaining optimal design.

V DICE Phase 5: Collaboration Technologies for Large Scale Mechanical System Concurrent Engineering

Experience gained during the performance of the DICE Phase 4 and DMSO project efforts indicates that challenges and impediments in enabling consistent, meaningful collaboration among diverse engineering analysis disciplines are on the critical path to achieving Concurrent Engineering goals. Achievement of a workable CE capability for large-scale mechanical systems is dependent on the resolution of a number of basic cultural, management, and computer science issues, and the development of appropriate engineering data and process modeling technologies that address these issues, within the context of integrated design and analysis operations. The implementation of such technologies is necessary to enable effective utilization of complex simulation based design tools that require a high level of discipline specific capability in a number of disparate disciplines, all of which must cooperate/collaborate to achieve a level of concurrency required to meet CE goals.

While numerous basic concepts and software tools have been developed for enterprise integration in support of "distributed TIGER teams" for Concurrent Engineering in DICE and related programs, these concepts and tools have required concerted evaluation of their efficacy to support large scale mechanical system CE. Initial achievements under DICE Phase 4 suggested some fundamental challenges in bringing these basic concepts to bear to support collaboration and enterprise integration of the diverse disciplines involving large scale mechanical structures, including complex mechanical system dynamic performance, soldier-system interaction, human factors involved in system operations and maintenance, reliability and failure effects analysis, and system design parameterization and optimization. Each of these disciplines requires support by specialists using large scale computer simulation and design support tools and an extraordinarily complex database of product, process, and multiple model information required to achieve the level of concurrency required to substantially speed the process of system design and evaluation through many iterations of design refinement.

Whereas developments in simulation tool applications and product data modeling under DICE Phase 4 and the DMSO projects have in a large measure defined the computational analysis and product modeling conditions under which enterprise integration must occur, the DICE Phase 5 effort at the Center has sought to conceptualize and implement specific methods and mechanisms by which focused design data exchange is enabled and managed to achieve concurrent mechanical system design and engineering. A basic tenet of implementing a CE capability is to provide an environment in which diverse element of an enterprise can work together to achieve consensus pertaining to the success of the design effort; i.e. Concurrent Engineering must facilitate collaboration for consensus. Implicit to this concept is the assumption that diverse perspectives within the enterprise will maintain disparate and conflicting priorities with respect to their individual areas of expertise. Consequently, any CE environment must provide a forum in which conflicting activities can express their concerns, develop an awareness of the impact of their concerns on the success of the enterprise, and engage in meaningful and effective activities to expedite the resolution of conflicts in a manner that promotes the success of the enterprise. These requirements for effective CE of mechanical systems necessitate compliance with a number of issues that are intrinsic to the integrated CE capability. First, lines of communication between all activities of a complex enterprise must be established. Second, each participating discipline must be able to express its concerns in a language understood by all remaining perspectives and is consistent with the communication capabilities of the environment.. Third, assessment of the impact of competing concerns requires a capability to prioritize, or manage, these concerns to maximize compliance with each area of expertise, while minimizing adverse impacts on the remaining areas. Finally, appropriate processes that describe the enterprise and adapt the concerns of each activity are necessary to effect collaboration leading to the success of the enterprise.

As described previously in Section II, the Center adopted a two phased approach to enhance collaboration among designers and analysts employing an integrated CAE tool environment. By defining a parametric methodology that enables multidisciplinary design sensitivity analysis, trade-off, and optimization, (Phase I) and implementing design process management methods and tools appropriate for ICEE operation (Phase II), the Center has developed a consistent, comprehensive technique for promoting coordinated interaction among ICEE users that focuses the design evaluation and optimization effort to achieve specific design goals. This section presents a detailed overview of the concepts, methods, and tools developed and implemented under each of these phases and concludes with a realistic example of how these technologies, together with the CAE tool capabilities developed under DICE Phase 4 and the DMSO project, are employed in full scale application.

Mechanical System Design Parameterization

Parametric modeling is capability employed in most advanced CAD systems to express models in terms of assigned dimension variables and features, as well as other physical characteristics. Parameterization also allows the designer to relate physical characteristics in a CAD model to each other, enabling a change in one feature to be automatically expressed in another feature according to a pre-defined relationship. These relationships, or associativity, can be defined at all levels of the product hierarchy, between features that represent a single part, between parts, between components, assemblies, etc., up through the system level. In this manner, model development and implementation of changes in the model can be accomplished with considerable ease, once all parameters and associations between parameters are defined.

Model parameterization is beginning to see substantial implementation in the development of analysis models for CAE applications as well. It is the advent of both CAD and CAE parametric modeling capabilities that provides the basis for enhanced collaboration as presented under this effort. Design parameters constitute the basis for the establishment of a common "language" that can be understood by all disciplines in the design enterprise. Through the definition of a common design parameter set, each discipline can develop design and/or analysis model representations according to the needs of that discipline that exhibit a fundamental commonality with all other disciplines, so long as a representation incorporates design parameters contained within the set. By this, design changes suggested by any one discipline can be easily propagated to other disciplines and modeling of design changes within disciplines can be substantially accelerated. Application of design parameters for CAD and CAE model development also enables the development and implementation of powerful, robust computational methodologies that permit analysis of performance, definition of constraints, design sensitivity analysis, and design trade-off and optimization with respect to these defined design parameters in a multidisciplinary environment. The methodology developed under this DICE Phase 5 effort presents a scenario for simulation-based mechanical system design modeling, evaluation, sensitivity analysis, and trade-off as discussed in the following.

The fundamental method for simulation-based design using the ICEE consists of six phases: design evaluation, definition of performances measures, identification of costs and constraints, design sensitivity analysis, design trade-off, and design propagation as illustrated in Figure 5.1. The objectives of the design evaluation phase are to bring a CAD product definition into the environment, create simulation models, and perform multidisciplinary design evaluation to assess the performance of the mechanical system. Based on the evaluation results, aspects of system performance to be improved are defined as performance measures, and geometry dimensions in the CAD models that are to be varied to obtain desired system performance are defined as design parameters. In the design sensitivity analysis phase, CAE analyses are employed to calculate the design sensitivity of performance measures with respect to these design parameters. Design sensitivity information is used to conduct design trade-off with the goal of obtaining an improved design. Then design changes are propagated back to the CAD and CAE models to iterate the design process. Effective application of this methodology is, however, contingent on the definition of suitable parameterized CAD and CAE models, a consistent parametric mapping scheme between the CAD and CAE models, and the extension of the global CAD product model to include parameter and associativity objects at all levels of the product model hierarchy, to enable collaborative design change propagation among disparate design disciplines using the ICEE environment.

Parametric Design Modeling

While parametric modeling is widely used in design development, the application of this technique in support of the methodology presented here for large scale mechanical system CE, has required considerable investigation to determine appropriate parameters that can be used to represent the mechanical system according to the needs of the analysis tools comprising the ICEE. Most dimension and feature parameters are employed. However, to model the multibody mechan-

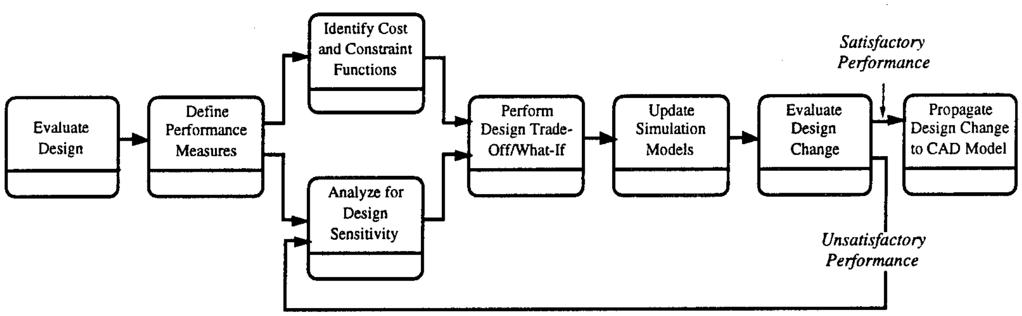


Figure 5.1 The Fundamental Design Evaluation and Optimization Methodology

ical system, parameters representing mass and material property attributes and connectivity between parts, components, and assemblies are required as well as formulation of legitimate associativity algorithms for dynamic connections. The design parameterization concept specified in this section employs two types of parameters, geometric and material properties. The geometric parameters describe the geometric shape of the mechanical system in a Computer-Aided Design (CAD) tool. The material properties determine mass properties of the mechanical system, and constitutive behavior of the components of subsystems of the mechanical systems. The design parameterization concept has been primarily developed to support multi-disciplinary design trade-off and design re-iterations for the mechanical system simulated in the ICEE to achieve better engineering performance.

In this DICE Phase 5 project, the initial focus has been on design change occurring at the component (part) level. Development of design change for an assembly of the mechanical system is occurring under on-going research projects at the Center. For example purposes, a simple engine connecting rod, shown in Figure 5.2, will be used to illustrate the design parameterization concept described in this section. The CAD system, Pro/ENGINEER, used to create the model illustrated in Figure 5.2, has been selected and implemented in the ICEE to support the parametric modeling needs as determined for this methodology.

In Pro/ENGINEER, the geometry of a CAD model is defined by a set of geometric features, the associated dimension parameters, and a set of parameter constraints that describe the features and relationships between features. Figure 5.3 shows the hierarchical relationship of features and dimension parameters in a parameterized and constrained part. A parameterized part contains a number of features, where a feature is

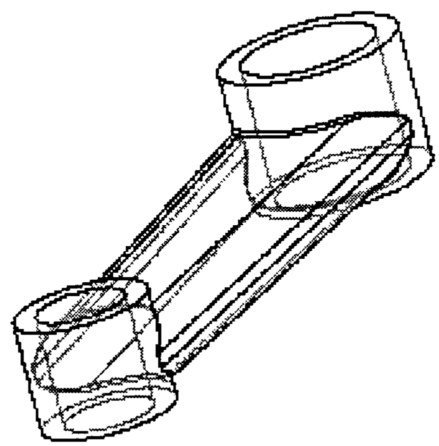


Figure 5.2 An Engine Connecting Rod

described by a set of dimension parameters. A feature comprises a simple 3-D manufacturing construct, and may be created by protruding a 2-D sketch. A sketch could be any simple or complicated 2-D contour. There are many protrusion methods to create a 3-D model based on a 2-D sketch, such as extrusion, revolving, sweeping, and blending. The manufacturing features can be a hole, shaft, round, chamfer, neck, cut, etc. The geometric shape of a feature is determined by its dimension parameters.

Another important aspect of part definition is the feature relationship. As shown in Figure 5.3, the feature relationships consist of three components. The first is the Feature Sequence, which is the feature creation order during the modeling process. The second component is the Feature Hierarchy which determines the parent/child relationship between features. A child feature is defined relative to its parent feature. The third component is the feature placement relationships, which can be classified into two types. The first type is relative locations that locate a child feature relative to its parent feature. The other type is the relative locations which predicates an alignment between two features.

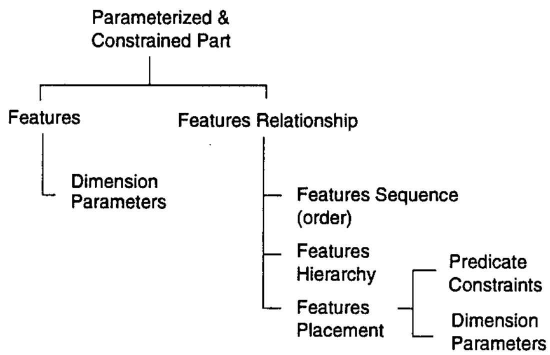


Figure 5.3 Hierarchical Relationship of a Parameterized and Constrained Part

The geometry of a component in a mechanical system can be changed by the following methods:

(1) Changing values of the dimension parameters, including feature geometric dimensions and placement dimension.

(2) Changing the feature types, e.g., from a circular hole to a square hole, in which the associated geometric parameter set is also changed.

(3) Changing the feature relationships, e.g., adding or removing an alignment relationship between two features.

Under this research effort, design changes have been restricted to the first method. The dimension parameters that describe feature geometry and feature placement relation are considered as candidates for design parameters, and design changes modify values of these parameters instead of modifying their definition. The arm of the connecting rod is used to illustrate CAD parameters, design parameterization, and design change propagation methods in this example. The arm has two features defined in Pro/ENGINEER, protrusion and round. Dimensions associated with these features, as shown in Figure 5.4, determine its shape. In the model creation process, there are nine parameters defined to create the two features. Among these parameters, assignment relations have been defined as listed in Table 5.1; parameters on the right side of the equation are candidates for design parameters.

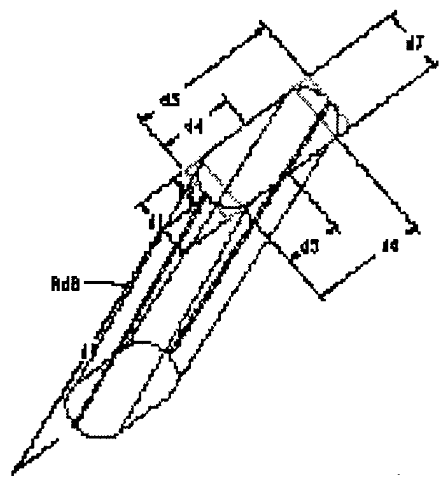


Figure 5.4 Connecting Rod Parameters

Table 5.1 Parameters and Relation of Features in Connecting Rod

Feature Names	Parameters and Relations	Values (Meter)
Protrusion	d_0	0.042862
	$d_1 = 1.0 \times d_7$	0.004762
	$d_2 = 0.5 \times d_7$	0.002375
	$d_3 = 2.0 \times d_4$	0.012573
	d_4	0.006286
	d_5	0.004731
	$d_6 = 2.0 \times d_5$	0.009525
	d_7	0.004762
Round	Rd_8	0.002286

Material properties, including Young's modulus, Poisson's ratio, and mass density are considered as material design parameters. Young's modulus and Poisson's ratio affect structural flexibility and stress-strain behavior of the component. Mass density contributes to dynamic behavior of the mechanical system, and human factor analysis on maintainability tasks. Material properties are defined in CAD models and can be retrieved and brought into the ICEE product model.

Design parameters are selected from existing feature geometric and placement dimension parameters as well as material properties generated in the CAD models of the components of interest. In addition, design parameter linking can be performed to link changes between design parameters external to the CAD tool. Design parameter linking implemented under this effort can be expressed as:

$$d_i = \alpha d_j \tag{1}$$

where change of design parameter d_i depends on change of design parameter d_j with a constant ratio α, with d_i referred to as a dependent design parameter and d_j as an independent design parameter. An independent design parameter may have more than one dependent design parameters. A dependent design parameter, however, cannot be linked to two independent design parameters.

For the connecting rod, parameters d_7 and Rd_8 are selected as design parameters. These design parameters are linked by:

$$Rd_8 = 1.0 \times d_7 \tag{2}$$

In addition to linking, proper upper and lower bounds must be defined for each design parameter to prevent undesirable design changes.

Design Evaluation

In the design evaluation phase, a CAD product model of the mechanical system is first brought into the ICEE as the base definition as described in Section IV. After the base definition is generated, engineers create various view models, including dynamics, structural, reliability, and maintainability. The engineering views support view model generation derived from the base definition. After view models are created, engineering workspaces are used to launch performance and analysis simulations.

Simulations include dynamics, structural, durability, reliability, and maintainability. Simulation results are exported to the global database through workspace wrappers as well as posted to the Communication Board of the infrastructure for use in defining a design model. With the simulation results, engineers in the CAE team exchange design information through the Communication Board, identify problem areas in the mechanical system and define them as performance measures, and define design parameters from geometry dimensions in CAD models for the mechanical system.

Performance Measure Definition

Problematic system performance is identified and defined as performance measures. System performance in various areas is considered as performance measures, including dynamic, structural, fatigue, reliability, and maintainability. Possible performance measures in each area are described as follows.

Dynamic Performance Measures

In general, the dynamic performance measures include the acceleration of body which would effect the ride quality, stability, and obstacle performance; the distance between two bodies which would effect the road holding ability, the operation range of the actuator between the bodies; reaction forces applied to the joints and the external forces generated by the spring, damper, actuator, and tire.

Structural Performance Measures

Structural performance measures are defined for components or subsystems of interest in the mechanical systems. Structural performance measures consists of global and local measures. Global measures include structural mass, volume, natural frequencies, and buckling load factors. Local measures include displacements measured at certain finite element nodes in certain directions and stresses measured at certain finite elements with certain failure criteria.

Fatigue Performance Measures

For fatigue, the number of blocks of dynamic simulation cycles before crack initiation in the components or subsystem in the mechanical system are considered as fatigue performance measures. Also, the number of blocks needed for a crack to extend to a prescribed length can be considered as performance measure definition for crack propagation.

Reliability Performance Measures

Reliability of the survival of a standard mechanical part, such as a gear, bearing, and spring, in the mechanical system under a prescribed mission cycle and failure criteria is considered as the performance measure from the reliability perspective.

Maintainability Performance Measures

Maintainability performance measures include the time and cost of the maintenance task and ability of human personnel to carry out the maintenance task. The time measures include the total maintenance task, total time for each technician, total disassembly sequence time, average disassembly step time, maximum disassembly step time, and average and maximum macro model and macro motion time for each disassembly step. Corresponding cost is assigned to each of these measures. Human factors analysis is performed to identify problems related to the interaction between maintenance personnel and the design model in a maintenance task. The human factors problems, related to maintainability of the mechanical system design, may involve the inability of the maintenance technician to produce required strength (torque), unavailability of work area clearance required to carry out the task, accessibility problems, and problems related to visual requirements of the technician in performing the task.

Cost and Constraint Function Definition

Performance measures serve also as a performance pool, a part of which can be selected as cost and constraint functions to set up the design problem. In the ICEE, cost and constraint functions can be defined as a combination of various performance measures, i.e.,

$$\phi^k = a_i \psi_i^{c_i} \qquad (3)$$

where ϕ^k is either a cost function or the k^{th} constraint function, ψ_i is a performance measure; a_i and c_i are real and integer coefficients, respectively; and n is the number of performance measures employed to define the cost or constraint function. The cost function, constraint functions with bounds, and design parameters with bounds form a design optimization problem that can be sent for trade-off determination and design optimization.

Design Sensitivity Analysis

Design sensitivity analysis measures the influence of variations in design on system and component performance. It complements simulation tools by showing which design characteristics should be modified to most effectively improve performance. Design sensitivity analysis theory for structures is well established.[25] Design sensitivity analysis of dynamics performance and reliability-based design sensitivity analysis theory are currently under development.[38-40] Progress in design sensitivity analysis for structural durability was made very recently. [41]

Based on the design model definition, engineering workspaces may be used to carry out design sensitivity analysis, either analytically or by using the finite difference approach. The finite difference approach requires additional analysis. The design parameters of the mechanical system are perturbed and the sensitivity coefficients are calculated using Eq. 4;

$$\frac{\partial \Psi^i}{\partial b_j} \approx \frac{\Delta \Psi^i}{\Delta b_j} = \frac{\Psi^i(\mathbf{b}+\Delta b_j) - \Psi^i(\mathbf{b})}{\Delta b_j} \qquad (4)$$

where Ψ^i is the i^{th} performance measure and b_j is the j^{th} design parameter. Using workspace wrappers, engineers send the sensitivity coefficients back to DDS for design trade-off.

To achieve a reasonably fast turnaround in design sensitivity analysis when using the finite difference approach, methods must be developed to quickly create simulation models for the perturbed designs. Currently, the mapping and quick finite element model creation methods have been developed using design velocity fields for structural areas.[26] Research is continuing under ARPA's *Integrated Product and Process Development* project (see Section VII) to develop mapping schemes between CAD design parameters and parameterized view models for other simulation areas. The mapping methods that speed up simulation model creation will also significantly reduce turnaround time in the design iteration process.

Design Trade-Off

The lead engineer uses the communication utility of the infrastructure to review sensitivity coefficients, either in a matrix form or bar charts, conduct design trade-off, perform what-if studies, and make decision among the proposed designs.

Very often in the design process, engineers must perform trade-off between cost and constraint functions. Before performing design trade-off, cost function to be minimized and a set of constraint equations to be restricted in the design process must be defined first. The cost and constraint function definitions are as explained previously. The trade-off analysis assists the engineer in finding the most appropriate design direction under certain design requirements. The infrastructure allows the selection of a design direction using four options: (1) cost reduction with a feasible design, (2) constraint correction neglecting cost, (3) constraint correction at constant cost, and (4) constraint correction with specified cost increment.[42] After the design direction is found, the engineer can carry out what-if studies.

The what-if analysis provides a first-order prediction of the system or component performance measures by Taylor series expansion about the current design, i.e.,

$$\Psi^{i}(\mathbf{b}+\delta\mathbf{b}) \approx \Psi^{i}(\mathbf{b}) + \frac{\partial\Psi^{i}}{\partial b_j}\,\delta b_j \qquad (5)$$

where δb_j is the design perturbation of the j^{th} design parameter. The what-if study gives quick first-order approximation for structural performance measures at the perturbed design, without going through model generations and simulations.

Once a satisfactory design is found after trying out different design alternatives in an approximation sense, engineers can use the infrastructure to update design parameters and propagate design changes to the view models and simulation models to conduct the design evaluation phase for the new design. The design phases are repeated until a satisfactory design is obtained.

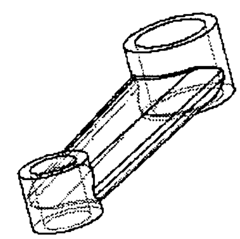
Design Change Propagation

After a new design is obtained from design sensitivity analysis and design trade-off, the new design can be propagated back to Pro/ENGINEER to regenerate updated CAD models for the new design.

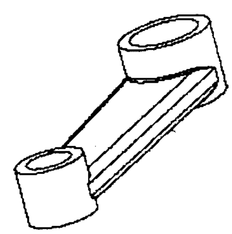
In the ICEE, the new design is represented as a vector of new design parameter values, including both independent and dependent parameters. These design parameter values will be verified to make sure they are within corresponding upper and lower bounds. The modified design parameter values are then mapped back to the geometric and material parameters in CAD models that are selected as design parameters.

For the engine connecting rod example, the design parameter d_7 is assumed to change from 0.004762 to 0.004002 meters (-0.00076). Based on design parameter linking defined in Eq. 2, Rd_8 will change the same amount from 0.002286 to 0.001526. The other parameters, including d_1 and d_2, are changed according to d_7 due to the relationships built in the CAD model. The changes of geometric parameters are summarized in Table 5.2.

The new design parameter values listed in Table 5.2, including d_0, d_4, d_5, d_7, and Rd_8 are determined by the ICEE users, i.e., the design team members. The new values will be sent back to Pro/ENGINEER to propagate such changes. In Pro/ENGINEER, new values of d_1, d_2, and d_3 are determined using the relationships built within Pro/ENGINEER. A new CAD model can be generated as shown in Figure 5.5(b).



(a) Current Design



(b) New Design

Figure 5.5 Connecting Arm Designs

Table 5.2 Changes in Connecting Rod Geometric Parameters

Feature Name	Parameters and Relations	Current Values	New Values	Changes
Protrusion	d_0	0.042862	0.042862	0.0
	$d_1 = 1.0 \times d_7$	0.004762	0.004002	-0.00076
	$d_2 = 0.5 \times d_7$	0.002375	0.001995	-0.00038
	$d_3 = 2.0 \times d_4$	0.012573	0.012573	0.0
	d_4	0.006286	0.006286	0.0
	d_5	0.004731	0.004731	0.0
	$d_6 = 2.0 \times d_5$	0.009525	0.009525	0.0
	d_7	0.004762	0.004002	-0.00076
Round	Rd_8	0.002286	0.001526	-0.00076

Design Evaluation and Optimization Process

Having defined the basic methodology for the application of parametric modeling, design sensitivity analysis, and change propagation techniques to support multidisciplinary design collaboration in the preceding, the following presents the specific application of these techniques in the Design Evaluation and Optimization Process (DEOP) identified for the utilization and implementation of the ICEE tool capabilities. This process expands the parametric methodology to capture the essential activities that define and focus the design effort to achieve specific product-oriented objectives for mechanical system engineering as supported by the ICEE. These activities include ascertainment of design goals, characterization of product user operations, determination of the scope, i.e., design considerations, of the design effort, and determination of an appropriate level of modeling and simulation required to achieve the specified design objectives, using the ICEE.

The process described in the following has been defined to be consistent with the current capabilities and functions of the ICEE. As such, the process embodies certain limitations representative of the spectrum of design considerations that can be supported by the ICEE tool capabilities. The process also represents a limited segment of the mechanical system product life cycle for which the ICEE tools are currently identified to support. A general form of the process is presented, which can be assumed to support any consistent mechanical system design and analysis problem. The process model follows the IDEF format,[43] which supports a more accurate characterization of activity attributes, i.e., input, output, control, and mechansim. Also, the IDEF format enables modeling of feedback loops to capture the iterative nature of the simulation-based design process. The IDEF format employs a muli-level process model hierarchy wherein each level successively decomposes the process into greater detail.

Level 0

At this stage of development, the fundamental utilization of the ICEE is to analyze an existing mechanical system design with respect to a defined set of performance objectives, and produce a new design that satisfies those objectives. As expressed in the highest level of the IDEF0 format (Level 0), the DEOP is as given in Figure 5.6. The principal input to the DEOP is an existing mechanical system design. The process as given does not assume any specific format for representation of the existing mechanical system, only that some representation(s) exist. These could be in the form of CAD solid models, 2-D design drawings, or even physical models - anything which the design engineer can use to create the models that will be analyzed using the CAE simulation tools in the environment. Of course, it is preferable that at least 2-D design drawings of the mechanical system and components exist, as it is prohibitively time consuming and costly to develop the analysis models required for this process from physical representations.

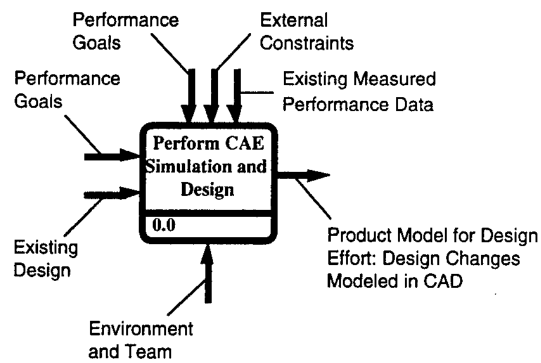


Figure 5.6 Level 0 of the Design Evaluation and Optimization Process

Performance goals in this context are presumed to be qualified statements of performance objectives for the mechanical system as a whole. A qualified statement of performance could be something as ambiguous as "longer life", "easier maintenance", "better handling", or something as specific as "a 10% reduction in weight with no degradation in existing system handling." In either case, the performance goals are typically characteristics that the existing design does not exhibit. Performance goals are considered to be both inputs and controls to the CAE design and analysis process, as they provide both a referent against which the success of the design effort is measured (control) and inputs to be transformed into specific quantified constraint data for use in CAE analysis formulations.

Two other control data are defined for the DEOP, external constraints and measured performance data. Measured data provides a referent against which the accuracy of the CAE models and simulations is determined. External constraints consist of any number of factors which may influence the definition of the CAD and analysis models in used in the design process as well as determining acceptable design changes from the aspect of "other" disciplines not represented in this process. For example, external constraints may dictate the definition of design features that are necessary from a manufacturing perspective, or may dictate the use of certain materials from a cost perspective. Although "external constraint" represents a broad set of unknown controls, its inclusion in the process model is intended to allow for these other factors.

The mechanism by which the DEOP will be accomplished is actually a composition of two resources, the integrated engineering environment and the users of this environment. Since the DEOP is defined to support implementation of the integrated environment by the design team (users), for any activity represented in this process either or both of these resources will be the mechanism for that activity. As such, a unique mechanism is not defined for each function or activity in the remainder of this process model.

Given the inputs and controls defined above, the DEOP is structured to yield a product model that incorporates design changes modeled in CAD that represent an improvement in performance which satisfies the objectives of the design effort. The product model yielded by this process will represent a level of detail consistent with those systems, subsystems or components, and the respective assembly hierarchy, targeted for design improvement by the DEOP. The product model constitutes 3-D solid CAD models of the improved design as well as the assembly hierarchy for the system and mass and material properties for each component in the assembly.

Level 1

Figure 5.7 illustrates Level 1 of the DEOP. Level 1 represents the principal activities (functions) that characterize the design and analysis process. This sequence embodies the basic logical sequence for any design effort, whether that effort employs CAE simulation analysis or more traditional design and analysis processes. In essence this sequence consists of (a) characterization of how the current system performs (Activity 1.0), (b) specification of how the system *should* perform (Activity 2.0), (c) determining whether the current design can meet the new performance requirements and where it fails to meet those requirements (Activity 3.0), and (d) improving the design of the current system so that it meets the specified performance requirements (Activity 5.0).

The Evaluation of Existing Product Design and Design Improvement, Activities 3.0 and 5.0 respectively, employ extensive use of CAE simulation technologies in this process, although these functions can be accomplished using more traditional design-build-test cycles. Activity 4.0 in Figure 5.7, however, introduces the key function that employs the results of the CAE simulation and analysis to define a modeling scheme that focuses the design improvement effort to achieve the defined performance objectives. The Definition of the Baseline Design Model (Activity 4.0) employs CAD and CAE parametric modeling as a means to suggest design changes that address specific performance concerns. The use of parametric modeling also establishes collaborative design development within the context of the DEOP by promoting rapid design change analysis within each analysis discipline and meaningful dissemination of design change suggestions across analysis disciplines. A brief description of each principal activity in Level 1 of the DEOP is given in the following.

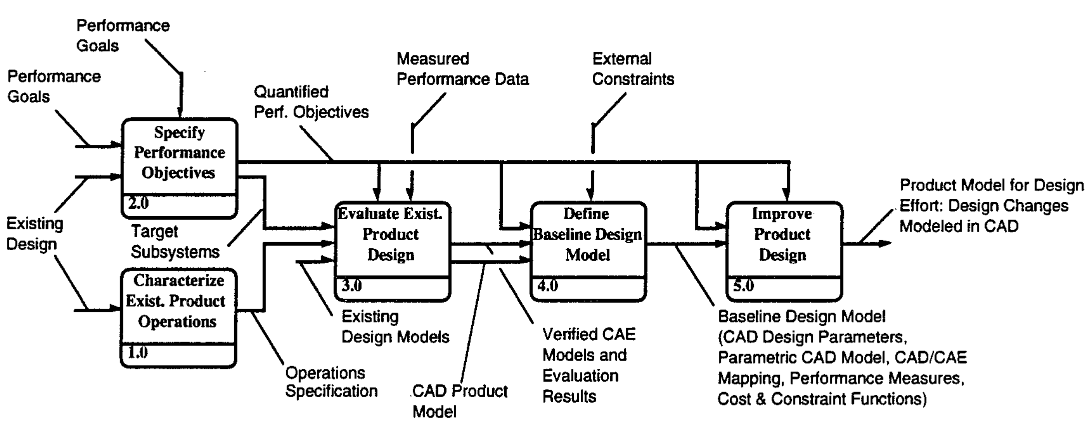
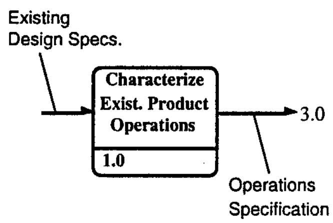


Figure 5.7 Level 1 of the Design Evaluation and Optimization Process

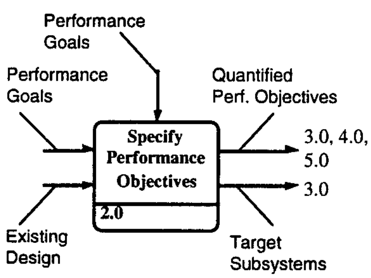
<u>Activity Name</u> | <u>Description</u>

1.0 Characterize Existing Product Operations



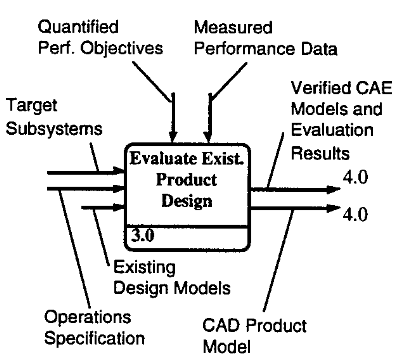
The objective of this activity is to identify the operations characteristics of the mechanical system currently in use. These characteristics, based on the existing performance specifications for the current product design, include aspects of the external environment, i.e., terrain, weather, etc., the spectrum of control factors and inputs employed by the user to operate the system, and the range of external loads applied to the system in performance of its functions. This information constitutes the operations specifications to be used to define the scenario for the downstream CAE simulations. This activity may entail the use of driving simulation to accurately quantify these characteristics for engineering use, whereupon the creation of real-time driving simulation scenario database and vehicle models, and the utilization of driving simulation facilities, become necessary sub-activities of this activity, provided that these models are not already available.

2.0 Specify Performance Objectives



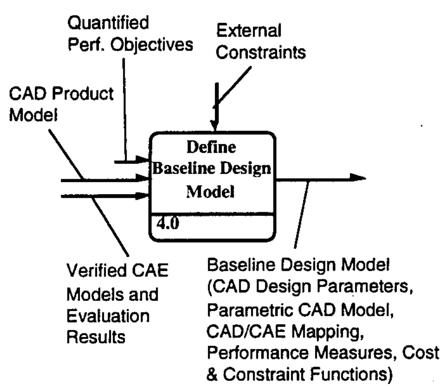
This activity is performed to identify and quantify those mechanical system performance characteristics that the user wants to improve and the degree of improvement. This information will be used to define the engineering problem to be solved through implementation of the Simulation and Design process. Through identification of performance improvement requirements, the affected systems, subsystems, components, or parts will be targeted, and the design and analysis disciplines most appropriate for achievement of a solution will be determined. Quantified performance objectives with respect to each analysis discipline will be determined. These quantified performance objectives will comprise the control referent against which the simulation based design suggestions will be assessed.

3.0 Evaluate Existing Product Design



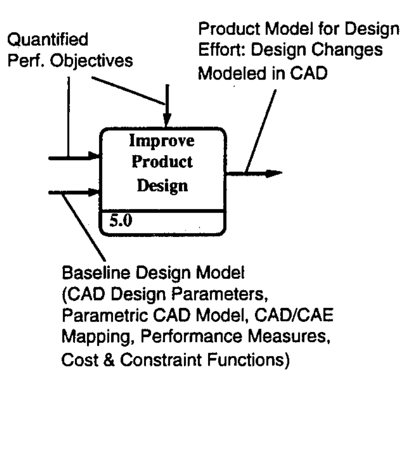
The objectives of this activity are to develop the product model for the design effort, verify the adequacy/accuracy of the CAD design and CAE simulation models for the baseline product, determine the analysis criteria for evaluation of product performance simulations referenced to quantified performance requirements, and assess the performance of the baseline product with respect to these specified objectives. Definition of the product data model for the existing mechanical system will be commensurate with the data requirements of the analysis tools to be employed in the design effort and the level of detail required for the design problem. The product model created in this activity will represent the minimum baseline system configuration necessary to meaningfully describe, analyze, and improve the target systems, subsystems, components, or parts. The CAD modeling portion of this activity will be performed to define system, subsystem, component, and part CAD design parameters to provide a basis for downstream CAD/CAE parametric model mapping. CAE analysis models will be derived from the CAD-based product model and verified to assure accurate representation of system performance. The output of this activity will be the verified CAE analysis models and a series of CAE simulation analysis results identifying problem areas in the target systems, subsystems, components, or parts representing non-compliance with performance objectives.

4.0 Define Baseline Design Model



This activity will be performed to define the baseline product design model. The design model represents the design change scheme that will be used to implement CAE analysis based design change suggestions in the CAD model. This activity will use the identified problem area information from the preceding activity and the parameterized CAD model from activity 3.0 to establish a parametric mapping scheme between the CAD and CAE analysis models that specifically targets design change to alleviate problem areas. This activity will also be performed to establish the multi-disciplinary design trade-off criteria that describe the range of allowable design changes, with respect to specified performance objectives, for each analysis discipline. These trade-off criteria will then be used as limiting factors for determination of the optimized system, subsystem, component, or part design.

5.0 Improve Product Design



This activity effectively embodies the basic methodology for the application of the parametric modeling presented earlier. This activity represents an iterative series of activities to be performed in order to evaluate the compliance of system, subsystem, component, or part design change improvements with defined mechanical system performance objectives. The objective of this activity is to determine an optimal design model configuration through systematic sensitivity analysis, multi-disciplinary design trade-off, design model change, mechanical system simulation, and performance evaluation. Once an optimal CAE analysis model configuration has been obtained, this activity culminates in design change propagation of the improved configuration to the product CAD design model via the mapping scheme defined in the preceding activity.

Further decomposition of the DEOP has been accomplished through Level 4 (see Appendix D). Level 2 of the DEOP deconstructs the fundamental Level 1 design and analysis sequence into process activities which characterize the modeling and simulation requirements supporting each major activity described in the preceding. Level 2 of the DEOP formalizes and extends the application of the parametric design evaluation and optimization methodology illustrated in Figure 5.1 specifically for the ICEE environment. The Level 3 process model decomposes the general product modeling and simulation activities into specific activities to be performed for each ICEE workspace tool (analysis perspective). Finally, the Level 4 model identifies the application of the product view structure methodology for each analysis perspective and indicates the sequence of model derivation from CAD to CAE model for each tool capability.

Having defined the design process, the remainder of the DICE Phase 5 effort focused on the development, implementation, and application of software technologies that allow the design team to execute the DEOP, manage a design project that employs the DEOP, and communicate data requirements and design change suggestion in a manner commensurate with the flow of the DEOP.

Design Process Management Technologies

The ICEE supports teams of CAD and CAE design developers and analysts. Effective utilization of the ICEE tools comprising this environment requires a well-defined process and process management methodology. As has been described in the preceding, the Design Evaluation and Optimization process has been defined from the application of the parametric methodology under the first phase of this DICE effort. The second phase of the Center's research in collaboration technologies, then, targets the development of process management methods and tools to enhance collaboration and concurrency among users and activities for this process. The following presents the development and implementation of the Design Process Management (DPM) methodology applicable to the performance of the DEOP activities in a team user environment composed of design and analysis engineers who employ the integrated CAE tool capability. A DPM tool suite has been developed, extending the CAE software environment integration architecture, that supports the implementation of this methodology. The DPM tools and methodology are specifically targeted for use by the manager or project leader of this team environment to assist in development, dissemination, and management of the product design effort as implemented by the defined process.

Process Management for Enhanced Concurrency and Collaboration

Effective utilization of the ICEE and collaboration among environment users is contingent on a number data generation and communication factors intrinsic to the operational requirements of simulation based design tools and the concurrent design development process in general. As described in Sections III and IV, the ICEE consists of a number of individual CAE tool applications integrated via a global product database, a database server, and a series of workspace wrappers. Although common CAD product model data is available, via the global database, to each of the CAE tools in the environment as input, each of the CAE tools also exhibits certain additional data requirements which must be fulfilled before analysis can be accomplished using that particular tool. In addition, the results of the design analysis associated with a particular CAE tool are typically expressed in a form more conducive to the design perspective pertaining to the use of that tool, and may not directly support or be compatible with the data requirements of another CAE tool in the ICEE. In consequence, at any stage in the simulation based design process, a requirement exists to produce analysis output in a form that is usable by downstream simulation analyses, in order to facilitate ICEE operations. Enhanced collaboration among users of the ICEE, *vis-à-vis* the DPM, will occur when users can identify data sources that meet their analysis requirements and communicate their input data needs.

Given the potential for increased exploration of design alternatives represented by the CE methodology, it is also likely that simulation tool data requirements will vary throughout the design and analysis process. Again, collaboration is enhanced when environment users can identify their data sources and communicate specific requirements corresponding to variations in analysis scenarios.

The process management methodology envisioned for the ICEE employs process definition, characterization, analysis for concurrency, and progress tracking to provide enhanced collaboration among environment users. Process definition enables teams and team managers to specify and capture data generation, design, analysis, and design evaluation activities in the

Concurrent Engineering process and represent data flow between design activities and perspectives. A design process model can be graphically displayed to all environment users to provide them with an awareness of where data comes from and where it goes, thus defining responsibilities and obligations in the CE process. Process activities can be characterized by user data requirements, by operational parameters such as time and resource requirements, and by activity dependencies; providing information supporting the determination of a process plan in compliance with the time frame and resources specified for completion of the design project. Process analysis allows the design project manager to identify potential bottlenecks to concurrency in advance of process implementation and aid in the definition (optimization) of contingency process plans. Finally, progress tracking allows the environment users and manager to review and update the process plan, using metrics which quantify process characteristics, to correlate design process implementation with the specified process plan.

Application of the process management methodology in the Simulation Based Design environment, then, has been accomplished through appropriate correlation of the process definition, characterization, analysis, and tracking functions and the identification and implementation of software capabilities supporting these functions, with the operational requirements of the simulation based design tools and the iterative concurrent design process. By this approach, environment users are provided with a frame of reference, with respect to project planning and environment operations, supporting communication and collaboration to achieve project objectives and adhere to process schedules and milestones.

Design Team Organization

The Concurrent Engineering methodology promotes a multi-disciplinary team approach to product design development. The integrated simulation-based design tools represented in Figure 4.1 are designed for use by members of design, structural performance analysis, dynamic performance analysis, and product support functional disciplines comprising the design team. Since the Integrated Concurrent Engineering (ICE) tools are potentially useful by a number of functional disciplines, the concept of operations described in the sections below assumes that individual ICEE tool utilization is consistent across design disciplines, and therefore, process specifications and management for any individual ICEE tool will be consistent across disciplines as well.

The structure of the product development team envisioned for utilization of the ICE tool environment integrates both product and functional organizational characteristics (see Figure 5.8). The application of any individual CAE analysis tool can be construed as predominantly supportive of a functional perspective in the design organization. As such, a selection of stand-alone CAE tool capabilities would presumably be applicable to the strict functional team organization illustrated at right in Figure 5.6. The ICEE tool

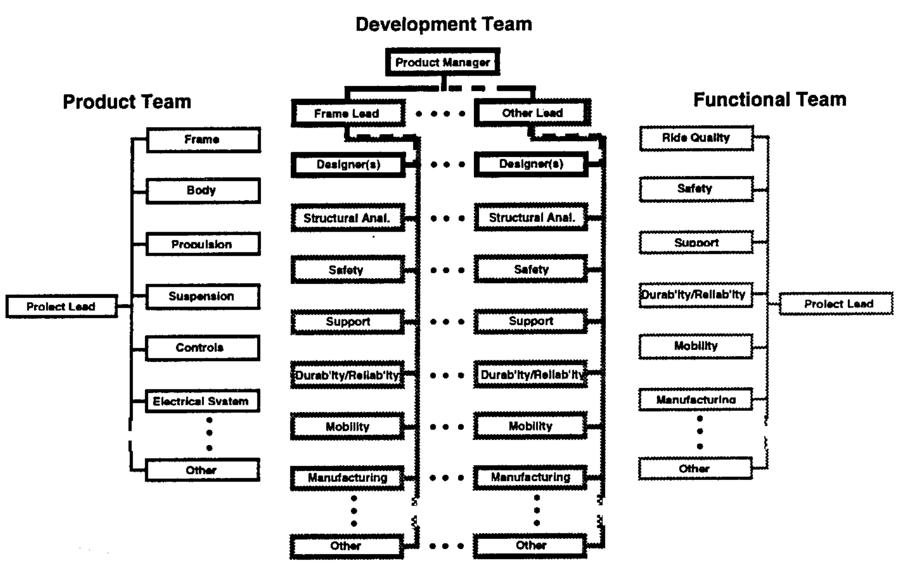


Figure 5.8 Product Design Development Team Structure

environment supports a design methodology that employs CAE analysis to suggest and propose design changes during the development process. By extension, it can be assumed that the ICEE is more supportive in a design capacity and therefore will be employed by a team exhibiting a greater product focus, as illustrated by the Development Team in Figure 5.8. As a result, the Development Team structure is representative of a matrix organization, addressing both functional and product development concerns.

At the current stage of development, ICEE operations correlate most directly with the functioning of the subsystem level team structure in the Development Team illustrated in Figure 5.8. The subsystem, i.e. frame, body, propulsion, etc., level fundamental team structure consists of representatives of functional disciplines, a design subgroup representing the product perspective, and a subsystem lead responsible for establishing the focus for and management of the team's design development efforts. By extension, the ICEE tool environment is supportive of the system Development Team structure as whole since the environment can presumably be employed to support any of the subsystem level teams. Collaboration and team management issues become more a matter of scale rather than disparate application of the ICEE tool capability between subsystem teams.

Figure 5.9 illustrates an example of the disciplines represented in the design team that can be supported with current ICEE tool capabilities. Functional and design applications supported by these capabilities are, as indicated previously, limited to design, structural, dynamic, and support analyses which are consistent with the usage of the CAD and CAE tools designated in Figure 5.9. Consequently, DPM functionalities are correspondingly limited, in an operational context, to management and collaboration among ICEE users employing these specific tool capabilities.

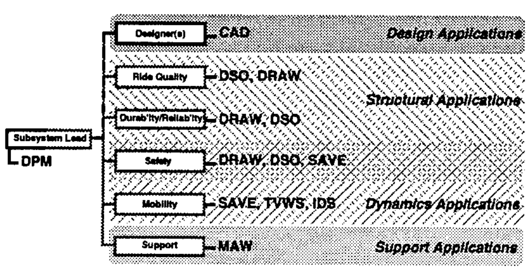


Figure 5.9 Design Team ICEE Tool Usage

Process Management Methodology

Once a determination is made associating specific ICEE tools with specific functional disciplines in the team environment, certain functions characteristic of that tool are imposed on the team members to effectively operate the ICEE tool within the context of the CE methodology and integrated environment utilization. These characteristic functions will be used to define the environment operational process supporting design development. Definition and management of this process provides the foundation for the collaboration methodology described earlier, therefore, development of the DPM capability targets the design team leader (manager) as the principal DPM tools user. Specific process management functions performed by the team leader addressed and supported through software development and implementation under this research effort include (1) determination of design project areas of concern, (2) designation of relevant project team personnel, (3) definition of the design and analysis process for the project effort (4) analysis and modification of the design process as required to promote concurrency/collaboration, and (5) assessment of project status. By exercising these functions, the project manager establishes the working team and defines their activities and responsibilities, their obligations to each other with respect to data requirements, and assesses the continuing effectiveness of the design development team in meeting project objectives, in a CE operations context.

Given these management functions, then, a series of activities, performed by the project manager or team leader, has been determined that represents a segment of the process management methodology envisioned for application to ICEE operations. These activities and the sequence in which they are performed are illustrated in Figure 5.10. A brief description of each activity follows.

(1) *Define Areas of Concern.* The project manager or team leader performs this activity to identify those functional disciplines, i.e., structures, dynamics, maintainability, etc. whose participation is required in the design development effort, given the objectives of the design project. In essence, the project manager or team leader is defining the organizational structure of the development team.

(2) *Determine Project Team Assignments.* Having determined which functional disciplines are re-

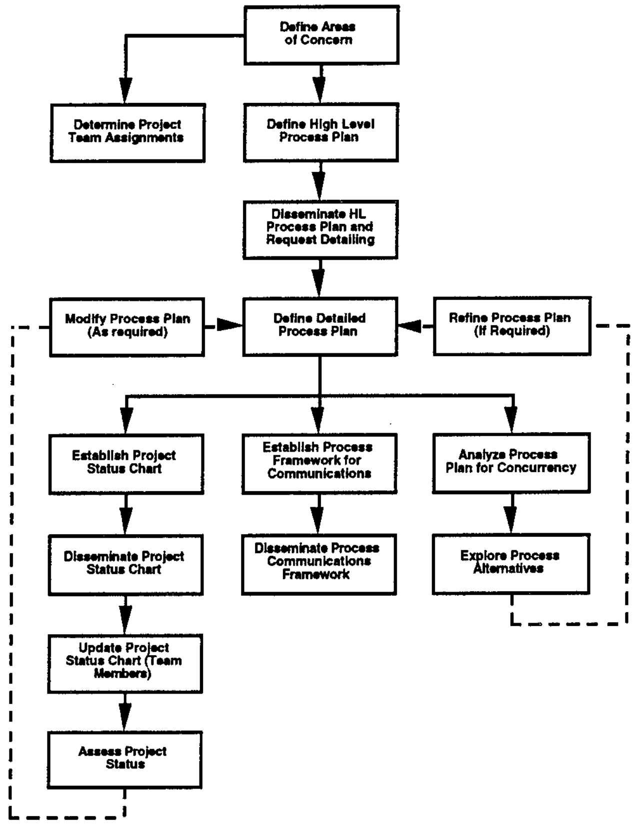


Figure 5.10 Process Management Methodology as Exercised by the Team Leader

quired for the design development effort, the project manager or team leader identifies and implements specific personnel assignments representative of the various functional disciplines. This activity results in the actual formation of the development team.

(3) *Define High Level Process Plan.* Having determined which functional disciplines are required for the design development effort, the project manager or team leader defines a general process plan for the design development effort. This process plan is used to identify the basic framework for the design development effort and a gross level of interaction between functional disciplines.

(4) *Disseminate High Level Process Plan and Request Detailing.* The project manager or team leader disseminate the general process to the team members and requests details regarding design and analysis activities to be performed by each functional discipline comprising the team, particularly requirements for input data that must be provided by other sources and design, analysis output required by other disciplines, and output which will be used to evaluate design alternatives. Team members will also need to provide information regarding estimated durations, resource requirements, etc. associated with their particular functional application.

(5) *Define Detailed Process Plan.* The project manager or team leader assembles the detailed design and analysis sub-processes to form the operational process plan for the design development effort. This process plan will identify a fine level of interaction between analysis disciplines based on data requirements, analysis output, and design evaluation activities.

(6) *Analyze Process for Concurrency.* Once a detailed process plan has been defined, the project manager or team leader will analyze the process to determine potential bottlenecks, critical activities and data flow requirements, and to identify the level of concurrency among activities in the process plan.

(7) *Explore Process Alternatives.* Once potential process bottlenecks, critical activity and data flow requirements, and the level of concurrency among activities have been identified, the project manager or team leader will postulate alternative data sources, activity sequencing alternatives, and design and analysis method alternatives and reanalyze process alternatives to optimize the level of concurrency obtainable, commensurate with overall project objectives.

(8) *Refine Process Plan.* Provided that process analysis indicates a workable alternative is possible, the project manager will refine the detailed process plan to incorporate activities and activity sequencing, commensurate with achieving optimal concurrency among design and analysis activities.

(9) *Establish Process Framework for Communications.* Since the hardware platforms supporting ICEE operations is based on a distributed computer network, the project manager or team leader will define a framework that specifies team members and member locations. This information will allow team members to establish communication links to enhance collaboration in the design development effort. The communications framework will be based on the design development effort process plan.

(10) *Disseminate Process Communication Framework.* Once a framework for team communications has been established, the project manager or team leader will implement this framework in the distributed, networked design environment.

(11) *Establish Project Status Chart.* Having defined the ICEE operations process for design and analysis, the project manager or team leader will develop a project status chart correlating to the performance of process activities. This chart will include information such as planned activity start and finish dates, actual activity start and finish dates, percent activity completion, etc.

(12) *Disseminate Project Status Chart.* The project manager or team leader will distribute the project status report chart to the design development team members.

(13) *Update Project Status Chart (Team Members).* Team members will update activity status as required, corresponding to changes in start, finish, degree of completion of process activities.

(14) *Assess Project Status.* The project manager or team leader will evaluate changes in process activity duration/completion with respect to achievement of overall project goals.

(15) *Modify Process Plan.* If required, the project manager will modify the detailed process plan to assure that changes in design development process activity duration/completion comply with overall project objectives. If necessary, the project manager or team leader will re-analyze the modified process plan for concurrency (repeating Steps 6 through 8 above), re-establish the process communications framework (repeating Steps 9 and 10), and update the project status chart (repeating Steps 10 and 12).

Given the activities to be performed by the team leader to manage the DEOP, requirements for a suite tool capabilities were defined, a complete listing of which is given in Appendix E. User interface requirements, consistent with the easy, effective utilization of the Design Process Management (DPM) software were also identified. Requirements for process management activities predominantly supportive of team communications, notably dissemination of process plans, project status, and the communications framework, are detailed later in this section.

DPM Software Tools

From capability requirements listed in Appendix E, six principle software tools and utilities were identified to comprise the DPM capability: (1) the Development Team Organization Modeler, (2) the Design Development Process Modeler, (3) the Group Technology Process Analysis package, (4), the Project Status Utility, (5), the Communications Framework Modeler, and (6) the CAE workspace wrappers. A number of either commercially available or preexisting software packages were obtained to fulfill Process Modeling, Group Technology, and Project Status functions. Most of the additional software development effort under this project consisted of integration of these existing software capabilities with each other, and with the ICEE. Figure 5.11 illustrates the proposed design process management software structure, incorporating the existing software packages, based on input/output specifications as per data requirements and data generation capabilities corre-

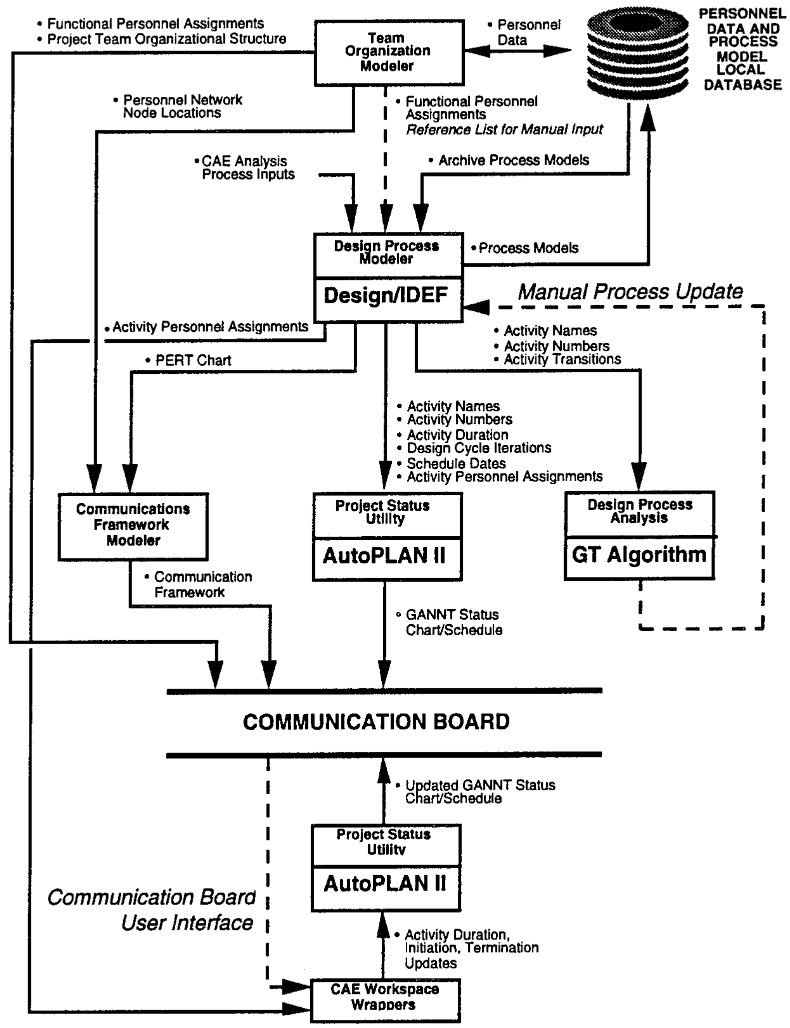


Figure 5.11 Design Process Management Software Architecture

sponding to each tool or utility. The design process management software interaction has been so structured to specifically yield project status representations and a process-based communications framework, which in combination with the Communications Utility described later in this section, supports the design team leader in implementation of the project management methodology and provides the design team members with the requisite level of design project awareness and interactive communications to enhance collaboration in the ICEE. A brief description of the purpose and capabilities of each tool in the DPM suite is provided in the following.

Design Development Team Organization Modeler - The purpose of the Design Development Team Organization Modeler is to provide the design project team leader with a capability to construct a graphical representation of the design team organization and textually specify team member assignments and responsibilities. This graphical organizational structure representation and team member assignments are disseminated via the communications board.

Design/IDEF Process Modeler - The purpose of the Design Development Process Modeler is to provide the design project team leader with a capability to graphically model the activities representing the operation of the ICEE tools, specify the input/output data relationships between tools/activities, and identify the team member(s) responsible for performance of an activity/operation of a tool in the process. Given the iterative nature of the Design Evaluation and Optimization Process, the relationships between activities/tool operation include iterative loops representing redesign cycles in the development project. The commercial process modeling package, Design/IDEF, has been selected as the Design Development Process Modeler due to its ability to model iterative loops, specify resources (personnel), produce an output file, and permit manual modifications to an existing Design/IDEF process model.

Group Technology Process Analysis Algorithms - The purpose of Group Technology (GT) Process Analysis software is to provide a means to analyze a given process for the identification of activities which may impede or inhibit concurrent performance of all the activities comprising the design process. It is also the purpose of this software to allow the user to explore various options to improve the level of concurrency among the activities comprising a given process. This software tool capability is composed of a suite of algorithms that applies GT methodologies to activity relationships in order to identify bottlenecks in the design process. The GT analysis algorithms employ a matrix representation of the process model as analysis input. The analysis performs a re-ordering of the matrix to group activities and identify potential bottlenecks according to GT methodologies. The output of this analysis is a reordered matrix which expresses a higher degree of concurrency among process activities.

With respect to the term "explore," the operational means by which alternative activity relationships can be determined to enhance the degree of concurrency in the process are the addition/deletion of activities and transitions within the context of the matrix input format for the analysis algorithms, followed by a reanalysis of the modified process as expressed in the input matrix using the GT algorithms. This capability is intended to assist in the rapid determination of an optimal process plan without the need for the laborious and time consuming manual iterative modification of the Design/IDEF process model.

The GT Process Analysis software to be used is a product of the Department of Industrial Engineering, the College of Engineering, The University of Iowa. The fundamental algorithms currently exist and are available for incorporation into the CE environment.

It should be noted that the GT analysis software utilities will not automatically resolve, through activity and activity relationship modifications, an optimal process. The GT analyses simply provide a tool for the user to accomplish this function. A high degree of user interaction is required in the operation of this software capability.

AutoPLAN II Project Status Utility - The purpose of the Project Status Utility software is to provide the users of the Simulation Based Design environment with a schedule that expresses, in GANTT format, the current status of the design project, correlated with the process defined for implementation of the design project. The project status utility employs activity data output from Design/IDEF, i.e., activity names, activity numbers, activity durations, activity initiation dates, and personnel assignments, to create an initial schedule for the design project, thus establishing a schedule correlated with the defined process plan. In order to express the iterative nature of the design process, the project status utility is structured to decompose, or "unroll," activity cycles embedded in the design process and sequentially schedule repeating activities according to a defined number of iterations specified in the process model.

This utility can also express changes in implementation of activities in the design process, e.g., actual start dates, finish dates, and percent completion, and graphically compare current status with the schedule originally planned for the design project.

The commercial software AutoPLAN II has been implemented as the project status utility, for its data import/export capabilities and process schedule representations. The output of the project status software is a graphical representation (GANTT chart) of the project schedule that can be viewed by each user in the ICEE.

Communications Framework Modeler - The purpose of the communication framework modeler is to associate the personnel responsible for individual activities in the design process with a network location in the ICEE. In other words, the communication framework modeler relates a network address with each activity in the design process according to who is responsible for the performance of that activity. In this manner, a process based communications framework is enabled that establishes a basis for interaction, communication, and collaboration between personnel who generate specific design data and personnel who have a requirement for that design data.

CAE Workspace Wrappers - The CAE workspace wrappers are the principal communications (integration) interface that enables each ICEE tool to communicate with the rest of the environment. The CAE workspace wrappers employ a suite of capabilities enabling and assisting in transfer of design data. The CAE wrappers have been modified to incorporate functionalities supporting communication and collaboration as per requirements identified in Appendix E. These functionalities include a user interface to access the communications utility, by which means the CAE tool user can view project status, view the communications framework, and establish communications links, and a user interface that permits the CAE tool user to update the status of those activities for which he is responsible.

Integration of the Design Process Management (DPM) software with the ICEE has been accomplished via the Communication Utility, or Communication Board, as depicted in Figure 5.12. Integration of the DPM software via the Communication Utility has been primarily a function of enablement, i.e., integration using the Communication Utility to enable the communication environment defined in the DPM Communications Framework Modeler. Other functionalities supporting integration of DPM with the ICEE will simply consist of graphical display, e.g. project status chart display, again employing the Communication Utility as the interface. Communication Utility functionalities and requirements are described in the following.

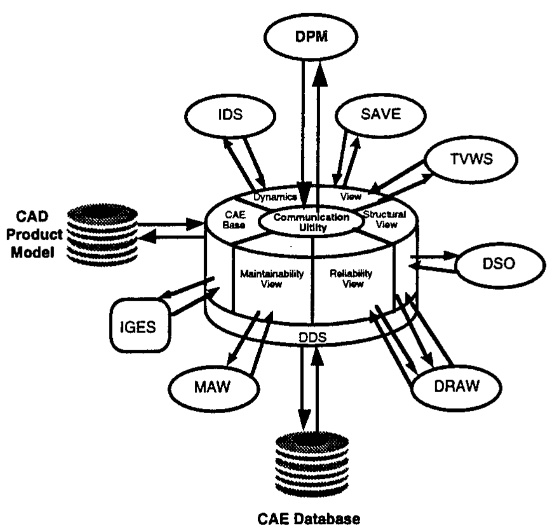


Figure 5.12 DPM Integration with the ICEE

Communications Implementation

Having identified a methodology by which an appropriate framework can be established to promote collaboration among ICEE users, a software capability enabling communication links in the networked workstation platform environment has been implemented. Communication links between environment users are able to support an exchange of design information and design development rationale commensurate with the needs of the multi-disciplinary design team. Although a plethora of design data exchange capabilities are currently available from the commercial software community, including textual, graphical, shared interactive, audio, and video communications, a nominal textual and graphical data exchange capability has been implemented to support the communications utility for this effort; demonstrating the fundamental concept of operations for collaborative communications in the distributed, multi-disciplinary Concurrent Engineering environment.

Design collaboration in the ICEE is a function of design change proposition and evaluation using the physics based simulation tools comprising the ICEE. The principal method for proposing design change is the application of the parametric methodology described earlier. Using this methodology, ICEE users are able to express mechanical system and component design in terms of geometric and material parameters and propose design changes through modification of parametric values. Effective collaboration using the parametric methodology, however, requires a capability that allows users to unambiguously communicate their rationale for proposing design changes and to specify their data requirements in order to perform design change evaluation. As such, in order to maintain an appropriate focus for achieving design project objectives, communication of design rationale and requests for design data must be well-correlated with the design process and operational needs of the ICEE.

In the ICEE, the design process is structured around the input/output data requirements of activities employing the use of the individual ICEE tools. As a result, communication links between environment users will support requests for analysis and design data input and notification of analysis data output when an activity or analysis operation has been completed. Consequently, a certain level of control is imposed on communications supporting the CE process. The communications framework provides a reference for determining which engineering disciplines should be communicating and what information should be communicated at any given point in the design process, supported by a network link that facilitates this level of control. While this method may seem at odds with the objectives to promote multidisciplinary collaboration, it promotes a methodology that requires the process by which collaboration is achieved to be well-defined, minimizing ambiguity in communications and maintaining a focus for the design effort. Given the objectives of Concurrent Engineering and the need to promote collaboration among designers and engineering analysts, it then becomes critical that the design process, or ICEE environment operational process, represents all necessary interactions between team personnel, particularly in the definition of design evaluation and design change activities.

An appropriately structured communications utility serves to capture the design development rationale for future reference, promoting the establishment of a design audit trail. Textual and graphical communications are archived in the global database with references for each party sending or receiving design communications, date, time, and subject. This communications capability, in effect, enables an e-mail like network link, adapted to support transmission of design graphics, and operating as an integral element of the ICEE.

The specific activities performed by all development team members in the utilization of the Communication Utility are (1) access the Communication Utility, (2) display organization structure, communications framework, and AutoPLAN II project status, (3) establish network communication links using the organization structure and communication framework displays as references, (4) create text, import graphics, (5) transmit design development rationale messages, (6) receive messages and read, store, or reply, and (7) update project status.

Operation of the communications utility to establish network links is accomplished by accessing the communications utility through an appropriate interface implemented in the individual CAE workspace wrappers, each environment user is presented with an organizational tree diagram and the process-based communications framework, defined using the Design Process Management software. With the organizational diagram and communications framework as references for communications, the user identifies himself in the organizational structure and then all other team members with whom he desires to communicate. To establish a communication link using the process-based communications framework as a reference, the user simply identifies his current activity, establishing himself as the point of origin for this communication and then identifies the downstream or upstream activity from which he either requires information or provides data for, respectively.

Once a link is established, the user is presented with a window wherein he can create text and import graphics files using typical e-mail functions. The user can identify a subject for this communiqué, with date, time, and transmission/reception locations (i.e., from and to) automatically retrieved from the operating system and network nodes. Once the message has been sent, a copy is automatically archived in the user's local database and the global database, retrievable by the user sending the message, the user receiving the message, and the team leader. When a message is sent, the user receiving the message is provided with audible and graphical notification of an incoming communication.

As communications are based on the design or environment operational process, notification of activity status is incorporated in the communication utility functionalities. Environment users are provided with a means to updated activity status as represented in the AutoPLAN II status chart. This information, provided by each user through a suitable user interface, is also used to display activity status in the communications framework. Activities in the communication framework display are color-coded for three levels of completion: not started, on-going, and completed, with a change in status from not-started to on-going referenced to the actual start date for that activity, and a change in status from on-going to completed referenced to a 100% complete status report for that activity.

Finally, in addition to graphical display and access of organizational structure and communication framework, the user employs the communications utility to display project status representations via the AutoPLAN II software.

A complete list of communication software requirements corresponding to the above concept of operations is given in Appendix F. From these requirements, development of the communications capability was based on the implementation of the Netscape software and World-Wide-Web protocols. The Communication Board establishes a Web site project page for each design project to be performed, presenting each member of the project team with a hierarchical listing of all process activities. By linking activities with individual user login identification, each user is presented with a project page that discriminates those activities for which he is specifically responsible (see Figure 5.13). Each activity listed in the project page is characterized by prior activity dependencies, duration, start and finish dates, status, and results to be reported when the activity is completed. Each activity description in the project page also embeds an e-mail link to the team member responsible for that activity. A status page is established for each activity (see Figure 5.14) that allows the team member to update the current status of the activity, append documentation and/or notation regarding activity performance, and post results by providing a path link to data files in the user's database. Status updates posted in the project web site are automatically input to the AutoPLAN II software, providing the team leader or members with an up-to-date overview of total project status. Other Communication Board capabilities include static graphical representations of the IDEF modeled process diagram, and functionalities employed by the team leader to define/modify a distributed team organization based on Internet e-mail addresses.

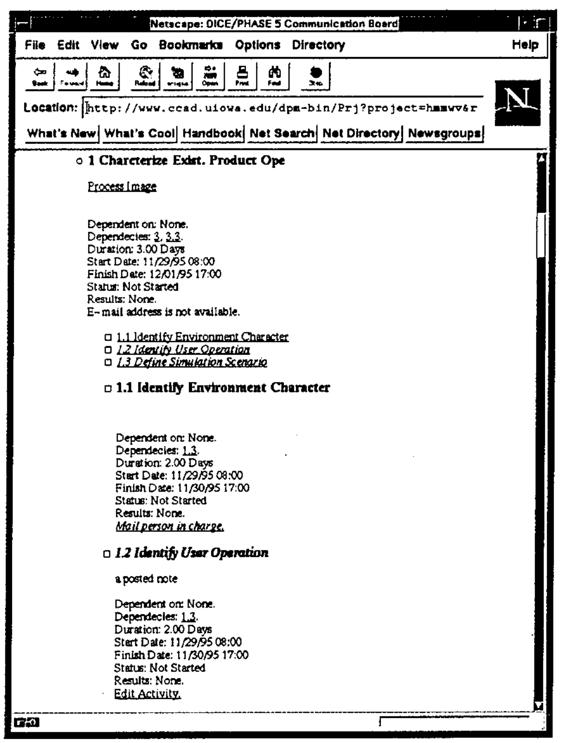


Figure 5.13 Communication Board Web-Site Project Page

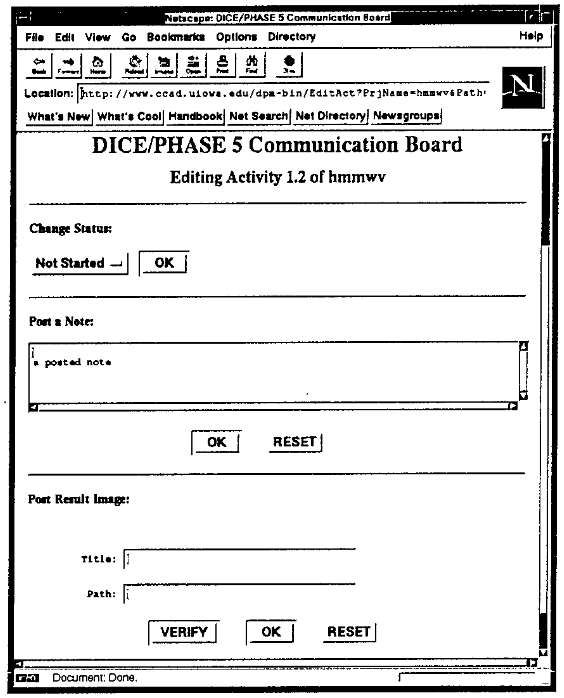


Figure 5.14 Communication Board Status Page

By basing the implementation of the Communication Board on the widely used and available Netscape software, several advantages are gained. The foremost advantage is that communication among distributed team members becomes platform independent. Since web protocols are standard for all Internet user access, no matter the system used, connection to the Communication Board is simply a matter of installing the shareware Netscape software on the user's computer system. In addition, by employing the flexibility of web page programming and graphical user interface, a comprehensive front-end navigator can be defined and implemented to access and operate the totality of the ICEE tool capabilities. Development of such a web site navigator for the CCAD testbed (see Section VI) is currently on-going under other project efforts at the Center.

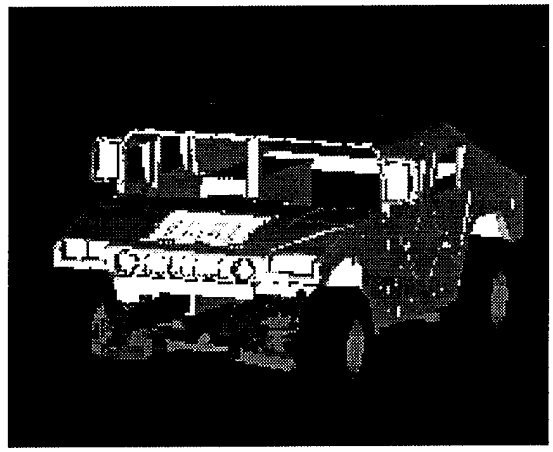
HMMWV Design Evaluation and Optimization Example Application

A realistic design evaluation and optimization problem has been defined to support verification of the DICE Phase 5 collaboration methodologies and technologies. This section outlines the application of DICE Phase 5 methods to assess potential impact to a US Army High Mobility Multi-purpose Wheeled

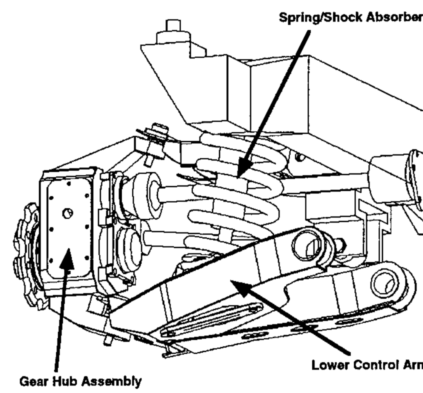
Vehicle (HMMWV) system as the result of a proposed change in the armor configuration. This exercise has been performed in conjunction with the Center's *Integrated Product and Process Development* (IPPD) project under sponsorship from ARPA (see Section VII). As the majority of software implementation for specific multi-disciplinary parametric DSA computational methodologies has occurred under the ARPA IPPD effort, a complete discussion of analysis results obtained from the exercise of this problem will be detailed in the ARPA IPPD final report to be published by the Center. The following, however, presents an overview of the HMMWV design problem, as defined through application of DICE Phase 5 concepts, including the identification of parametric design variables, as well as a process defined to solve the problem, that can be managed using the DICE DPM tool suite.

The basic objective of this exercise is to evaluate the impact of an additional 2,900 pounds of armor to the dynamic performance, durability, reliability, and maintainability of the HMMWV's suspension subsystem and identify design changes that will mitigate degradation in performance of affected suspension components. A product model of the HMMWV has been developed that is commensurate with the model requirements for this exercise; approximately 200 parts and assemblies have been defined in CAD that provide a gross representation of all HMMWV elements other than suspension and a detailed representation of the suspension subsystem components (see Figure 5.15). This product model, then, provides the base definition from which all CAE analysis views are derived. Durability and reliability analyses during this exercise target the HMMWV lower control arm and gear hub assembly, respectively, requiring model representations at the part level, and standard fastener geometric information for screws, nuts, etc., has been included in the product model sufficient to support maintenance task simulation for the suspension subsystem.

The basic scenario selected for this exercise evaluates the performance of the HMMWV traveling at a constant 20 mph over the Aberdeen Proving Ground 4 test course, a moderately bumpy environment which the test vehicle traverses in a straight path (see Figure 5.16). A dynamic model of the HMMWV is defined through the creation of the dynamics view, which groups parts and or assemblies to create dynamic bodies and defines joint connections between them. For this HMMWV example, a 14 body dynamic model



(a) HMMWV System



(b) HMMWV Suspension Detail

Figure 5.15 HMMWV CAD Product Model

will be employed as illustrated in Figure 5.17. To evaluate the impact of the additional armor loading on the vehicle, two dynamic simulations are to be performed, the first with an unarmored HMMWV dynamic model (5,558.5 lbs.), the second with the armored configuration model (8,458.5 lbs.). As well as providing suspension duty cycle data, this provides a direct comparison of dynamic performance between the two configurations, enabling dynamics engineers to assess gross component dynamic performance. For example, in this case, the additional armor loading resulted in metal-to-metal contact in the shock absorber, requiring adjustment of the spring constant in the armored configuration dynamic model to avoid unacceptable shock absorber operating conditions and to obtain reasonable load distribution to other suspension components. This situation introduces a necessary upgrade to the vehicle, i.e., substitution of a stiffer spring in the suspension system, when armor is added to the configuration. This implies that a significant maintenance task for the suspension subsystem will in all likelihood be required to add armor to already fielded systems. The Maintainability Analysis Workspace (MAW) capability will be employed to simulate this maintenance task and assess cost, time, and human factors issues.

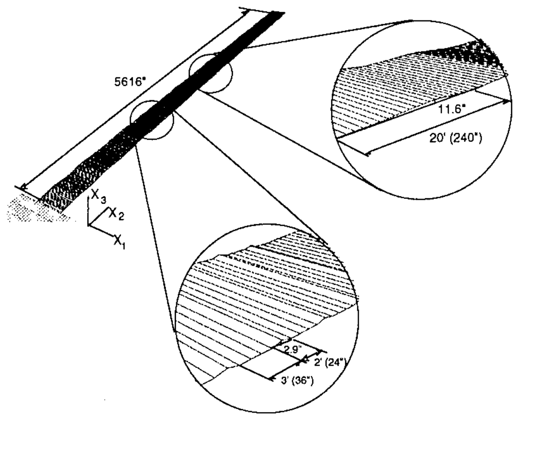


Figure 5.16 APG4 Computer Model

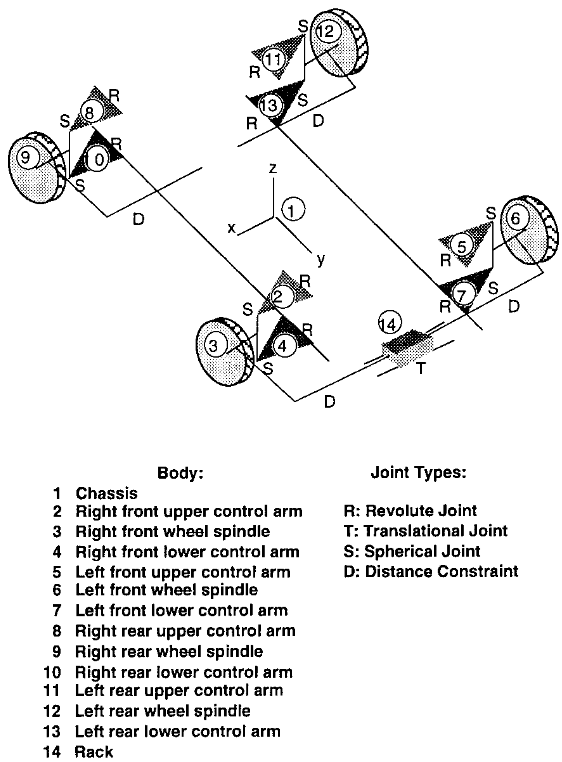


Figure 5.17 HMMWV Dynamics Model

Dynamic simulation results will also be used to support human factors analysis in terms of ride quality due to vertical acceleration at the driver's seat for the armored configuration. Vertical acceleration data from dynamic simulation will be transformed from time domain to frequency domain using the Human Factors Analysis workspace (software tools implemented in an upgraded MAW capability) for comparison with accepted standards of driver comfort in terms of vibrational frequency.

Structural analysis in this exercise targets the natural frequency and buckling load of the lower control arm (see Figure 5.18). Durability analysis will be used to predict crack initiation fatigue life for the lower control arm. The design parameterization methodology developed under DICE Phase 5 will be applied to lower control arm structural and durability analyses to achieve an optimized control arm design. The parametric design of the control arm considers two geometry design parameters (Figure 5.19(a)), eight thickness parameters, corresponding to thicknesses of the eight parts that form the lower control arm (Figure 5.19(b)), and one material parameter. Design optimization and trade-off within structural and durability analyses will consider the fatigue life of the lower control arm at 10 critical nodes in the finite element model, with the objective function being the weight of the lower control arm. Structural view definition supporting both structural and durability analysis will consist of conversion of the lower control arm CAD geometry into PATRAN geometry, retrieval of the duty cycle data generated in dynamic analysis, and using this, definition of load and boundary conditions that are consistent with the dynamic model.

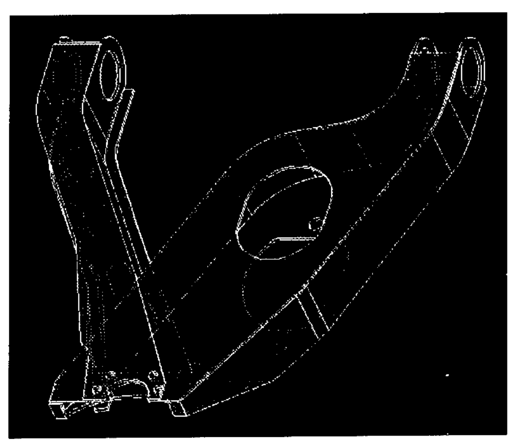
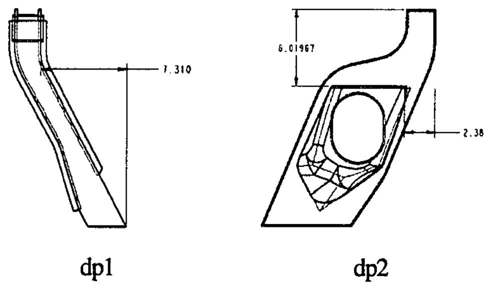
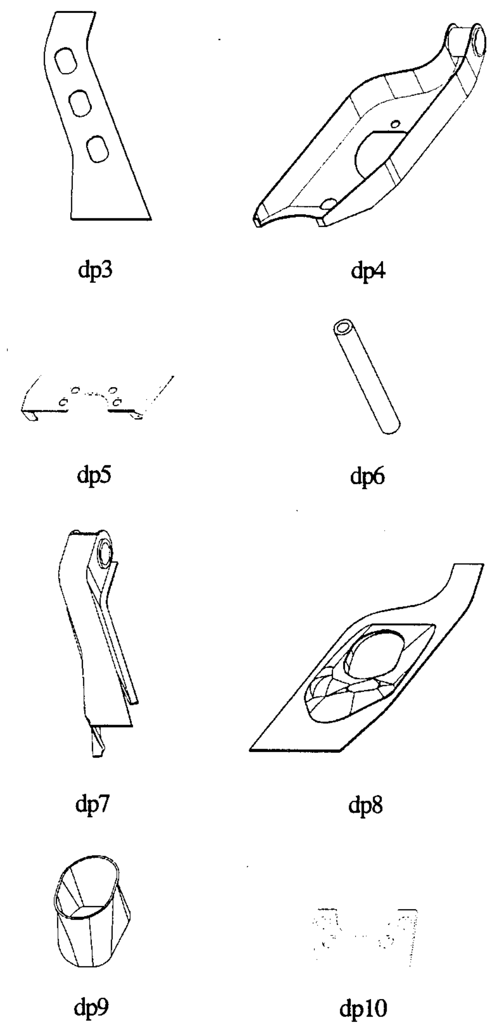


Figure 5.18 HMMWV Lower Control Arm



(a) Geometry Design Parameters

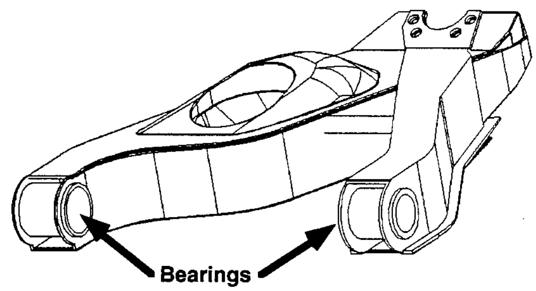


(b) Plate Thickness Parameters for Parts

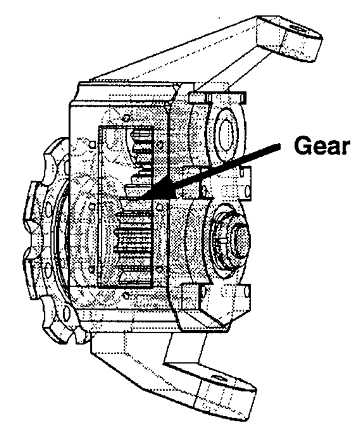
Figure 5.19 Lower Control Arm Design Parameters

Design trade-off and optimization at the multidisciplinary level will consider the geometry parameters dp1 and dp2, using shock absorber forces and the removal time for the lower control arm as performance measures.

Reliability analysis in the HMMWV example will target several components in the suspension subsystem including lower control arm bearing (Figure 5.20 (a)), the spring, and gears in the gear hub assembly (Figure 5.20(b)). A reliability view will be constructed that employs load history data for the armored configuration from dynamic simulation, and life estimates will be calculated for each target component corresponding to a 99% reliability requirement. Minimum life requirements will be identified for each component and used for comparison with predicted reliability results to assess component reliability performance. Deviation from minimum life requirements will provide a basis for developing design change suggestions from the reliability perspective.



(a) Lower Control Arm



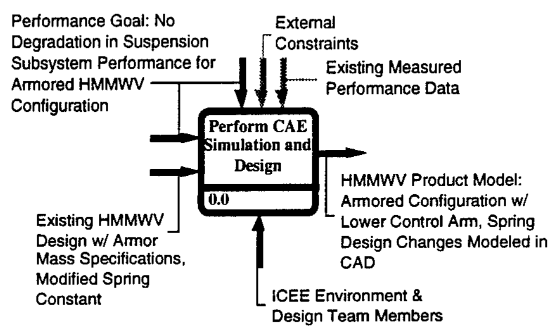
(b) Gear Hub Assembly

Figure 5.20 Target Components for Reliability Analysis

Given the HMMWV design scenario described in the preceding, a design evaluation and optimization process, based on the generic process described earlier in this section, has been defined to capture the interaction and design data requirements between the various engineering users and tool capabilities (see Figure 5.21, Appendix G). The process employs the multi-level IDEF hierarchy to successively decompose the HMMWV scenario into specific activities that individual tool users will perform in coordination to support and disseminate modeling and simulation data required to successfully resolve the HMMWV design problem.

The process depicted in Figure 5.21 and Appendix G has been implemented for the HMMWV validation efforts using the DPM tool suite. Relevant process data (projected vs. actual activity durations, start times, end times, etc.) for this exercise are currently being compiled and tabulated during the on-going IPPD project effort. Reporting of process metrics and conclusions with respect to the effectiveness of the DPM methodology to support collaboration among engineers performing the HMMWV example will be included in the IPPD final report to be published in April, 1996.

Level 0



Level 1

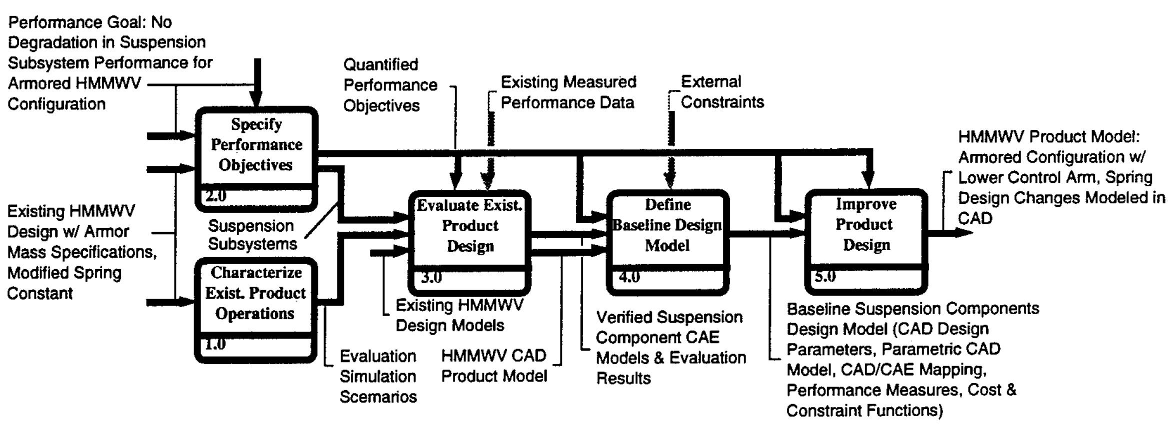


Figure 5.21 HMMWV Example Design Evaluation and Optimization Process (Levels 0, 1)

VI The CCAD Testbed

The CCAD Testbed is composed of a number of Center-developed software tools and a set of commercial CAD and CAE software codes. The Center-developed software tools include five CAE workspaces, an IGES Translator, and a software framework for system integration.

System Integration

The system integration framework consists of a global data model, a Design Data Server (DDS), a communication channel, and a system user interface (SUI) that integrates the design and analysis activities into the Testbed. The SUI acts as the entry point for data generated outside the Testbed (from CAD) and provides capabilities for CAE engineers to create CAE models using various CAE Views to support their analyses.

Global Data Model - The global data model is a generic CAE model definition of mechanical systems to support CAE analysis and design in the Testbed.

DDS - The Design Data Server is designed to handle all aspects of global data management in the concurrent engineering environment.

Communication Channel - The communication channel consists of a Communication Board (CB) that supports communication among CAE engineers, and a Client-Server type data communication protocol based on CORBA architecture to support a distributed DDS in the concurrent engineering environment.

System User Interface (SUI) - The SUI of the CCAD Testbed integrates workspaces and CAE tools into the environment by providing capabilities to: (i) interfacing with CAD product definition to create the Base for CE activities, (ii) creating CAE models to perform analyses, (iii) launching workspaces or CAE tools for analysis and retrieving data back to DDS or visualizing the results, and (iv) supporting communication among CAE engineers to perform design trade-off.

Recent development and implementation of software capabilities in the Integration Architecture includes:

- Netscape-based Communication Board
- Design parameterization and design trade-off
- Design change propagation for parts
- Design process management

Center-Developed CAE Tools

SAVE 2.0 - The Simulation and Visualization Environment (SAVE) is a software workspace that provides capabilities to model, simulate, and visualize a multibody mechanical system. Software capabilities implemented in the SAVE include:

- Computational inverse kinematic position analysis
- Preliminary interactive workspace analysis
- Preliminary singularity analysis
- NADSdyna connection
- Assembly capability for design modification

DRAW 3.2 - The Durability and Reliability Analysis Workspace (DRAW) is a software environment that provides capabilities to perform dynamic stress computation, fatigue life prediction, and reliability analysis of structural components using the duty cycle information obtained from multibody dynamic simulation. The DRAW can also be used to perform reliability analyses for standard mechanical components, such as bearings and gears, using the duty cycle information obtained from multibody dynamic simulation. Software capabilities implemented in the DRAW include:

- Finite element information based surface node searching
- ANSYS 5.2 connection
- Efficient preliminary life prediction
- Fatigue critical node list
- Approximate elastic-plastic multiaxial stress-strain estimation
- Spring reliability assessment

DSO 3.3 - The Design Sensitivity Analysis and Optimization (DSO) tool is a software environment that provides capabilities to perform structural analysis, design sensitivity analysis, what-if studies, design trade-off determination, and design optimization. Software capabilities implemented in the DSO include:

- ANSYS 5.2 connection
- ABAQUS 5.4 connection
- Sizing DSA for ABAQUS quad finite element
- Sizing DSA for MSC/NASTRAN quad finite element

IGES Translator - The IGES Translator is a software tool that provides capabilities to clean up and translate CAD-generated IGES files into various formats to support CAE modeling and visualization.

MAW 2.1 - The Maintainability Analysis Workspace (MAW) is a Computer-Aided Engineering environment designed to support maintainability analysis of evolving mechanical systems design from early in the design process. Software capabilities implemented in MAW include:

- Maintenance personnel selection
- Operability analysis
- Automated tool selection
- Generation of disassembly sequence
- Generation of maintenance manuals

MVWS 2.0 - The Military Vehicle Workstation (MVWS) also the Tracked Vehicle Workstation (TVWS) is a concurrent engineering tool used to assemble, perform, and analyze dynamic simulation for tracked military vehicle systems at the journeyman engineering level. Software capabilities implemented in the MVWS include:

- DADS modification to allow analyzing tracked vehicle on a moving body
- Cut/paste 2D plots
- 2D plot configuration files
- DADS 8.0 connection

Commercial CAD and CAE Tools

The CCAD Testbed uses Pro/ENGINEER as the CAD tool, DADS as the dynamic analysis tool, PATRAN as the structural modeling tool, ANSYS, MSC/NASTRAN, or ABAQUS as structural analysis tools, Design Optimization Tool (DOT) as design optimization tool, and FLAGRO as life prediction tool.

Software Operation Perspective

The CCAD Testbed will be running on UNIX and X/Motif-window environment over a set of heterogeneous machines. The Testbed supports a CAE team to perform design and analysis activities. The Testbed should allow multiple users to access the environment to create views, launch tools, and retrieve analysis results. However, only one user is allowed to modify the base definition at one time.

Hardware and Software Platforms

ICEE Testbed Hardware and Software specifications are given in Table 6.1. Remote access to the CCAD Testbed over the Internet will require outside users to have proper network access and have X11 R5 and Motif 1.2 to display the Testbed environment. User documentation is available for each Testbed workspace and is listed in Table 6.2.

Table 6.1 ICEE Testbed Specifications

	Tool	Hardware Platform	Operating System
ICEE 0.2		HP 9000/755	HP-UX A.09.05
	Pro/Engineer 15.0	HP 9000/755	HP-UX A.09.05
	Design/IDEF 3.1	HP 9000/755	HP-UX A.09.05
	AutoPLAN II 1.1	HP 9000/755	HP-UX A.09.05
	Netscape 2.0	All	
	P3/PATRAN 1.2	HP 9000/755	HP-UX A.09.05
	ANSYS 5.2	HP 9000/755	HP-UX A.09.05
	MSC/NASTRAN 67R2	HP 9000/755	HP-UX A.09.05
	ABAQUS 5.4	HP 9000/755	HP-UX A.09.05
	DADS 7.5	HP 9000/755	HP-UX A.09.05
SAVE 2.0		HP 9000/755	HP-UX A.09.05
DRAW 3.2		HP 9000/755	HP-UX A.09.05
	FLAGRO 2.01	HP 9000/755	HP-UX A.09.05
DSO 3.3		HP 9000/755	HP-UX A.09.05

Table 6.1 (Con.) ICEE Testbed Specifications

	Tool	Hardware Platform	Operating System
	DOT 3.0	HP 9000/755	HP-UX A.09.05
IGES Translator		HP 9000/755	HP-UX A.09.05
MAW 2.0		SGI	IRIS System V.4
	JACK 5.9	SGI	IRIS System V.4
MVWS 2.0		SUN	Sun/OS
	Informix 2.1	SUN	Sun/OS
	NRMM II	HP 9000/755	HP-UX A.09.05
Adobe PDF Viewer		All	Sun/OS
XV 3.0		HP 9000/755	HP-UX A.09.05

Table 6.2 ICEE User Documentation

	Testbed	ICEE	SAVE	DRAW	DSO	MAW	MVWS
Testbed Access	x						
Overview				x	x	x	x
Tutorial		x	x	x	x	x	x
User's Reference		x	x	x	x	x	x
Example Manual			x	x	x	x	
Installation			x	x	x	x	x
Concept Manual		x		x	x	x	x

VII Continuing Concurrent Engineering Research Efforts

While accomplishments in tool development, integration, and coordination under the Center's DICE effort conclusively demonstrate the feasibility of applying simulation-based technologies to achieve Concurrent Engineering goals for large scale mechanical system design, substantial research and development remains to be conducted to achieve truly seamless, distributed, and rapid multidisciplinary product development. In particular, when given the enormous variety of computer modeling and analysis systems available to the engineering user community, development of computational methods and technologies supporting robust application of multidisciplinary parametric-based design sensitivity analysis, trade-off, and optimization in a fully integrated, interoperable environment is still in its infancy. While great strides have been made by large industrial firms such as Boeing in the application and utilization of simulation-based design technologies, further necessary R&D is required to enable smaller industrial firms, which constitute the majority of US engineering and manufacturing capability, to afford and competitively utilize this technology.

To this end, the Center for Computer Aided Design is continuing in the development and dissemination of leading edge technologies that build on DICE accomplishments to achieve robust, seamlessly interoperable simulation-based design. Currently, the Center is engaged in a number of on-going research efforts in distributed CAE application and immersive driving simulation supporting concurrent product development through virtual prototyping. Two of these efforts build directly on the accomplishments of the Center's DICE engineering capabilities, and are being performed to (1) implement multidisciplinary design sensitivity analysis and trade-off based on the DICE Phase 5 parametric methodology and (2) address the development and application of standards-based product modeling protocols to support seamless parametric CAD design change in an industrial manufacturer/supplier environment. Sponsored by, respectively, the Advanced Research Projects Agency's (ARPA) *Integrated Product and Process Development* program and the National Science Foundation (NSF), these efforts will address computational and product modeling and data sharing limitations that currently exist for the distributed product development enterprise. A brief overview of each of these Center research efforts is presented in the following.

ARPA Integrated Product and Process Development

A principal objective of the ARPA Integrated Product and Process Development (IPPD) project is to enhance engineering utility, design change propagation, and design collaboration in the multidisciplinary design environment through the extension of the CAE tool capabilities in the ICEE to support computational design sensitivity methodologies. Design sensitivity methodologies will be defined for dynamic, reliability, and maintainability analyses with software functionalities to be developed and implemented in the SAVE, DRAW, and MAW workspaces.

The sensitivity of dynamic, structural, reliability, and maintainability performance to variations in design parameters will be determined for input to design optimization. In order to suggest design change among a multidisciplinary team, design parameters will be selected among the CAD geometry and material parameters, and performance measures defined to support consistent evaluation and design trade-off in the design development process. Specific performance measures defined for each area include:

Dynamic Performance - Dynamic performance measures include the acceleration of a body, which effects the ride quality, stability, and obstacle avoidance performance; the distance between two bodies which would effect the road holding ability, the operation range of the actuator between the bodies; reaction forces applied to the joints and the external forces generated by the spring, damper, actuator, and tire.

Structural Performance - Structural performance measures will be defined for components or subsystems of interest in the mechanical systems. Structural performance measures consist of global and local measures. Global measures include structural mass, volume, natural frequencies, and buckling load factors. Local measures include displacements measured at certain finite element nodes in certain directions and stresses measured at certain finite elements with certain failure criteria.

Fatigue - For fatigue, the number of blocks of dynamic simulation cycles before crack initiation in the components or subsystem in the mechanical system will be considered as a fatigue performance measure. Also, the number of blocks needed for a crack to extend to a prescribed length can be considered as a performance measure for crack propagation.

Reliability - Reliability of survival of standard mechanical parts, such as gears, bearings, and springs, in the mechanical system under a prescribed mission cycle and failure criteria is considered as performance measure for the reliability analysis perspective.

Maintainability - Maintainability performance measures include the time and cost of the maintenance task and ability of human personnel to carry out the maintenance task. The time measures include the total maintenance task, total time for each technician, total disassembly sequence time, average disassembly step time, maximum disassembly step time, and average and maximum macro model and macro motion time for each disassembly step. Corresponding cost is assigned to each of these measures. Human factors analysis is performed to identify problems related to the interaction between maintenance personnel and the design model in a maintenance task. The human factors problems, related to maintainability of the mechanical system design, may involve the inability of the maintenance technician to produce required strength (torque), unavailability of the work clearance required to carry out the task, accessibility problems, and problems related to visual requirements of the technician in performing the task.

The computational methodology for Design Sensitivity Analysis (DSA) in each area will be based on a finite difference approach. A more robust connection between CAE simulation tools and a parametric CAD system will be explored to support rapid design change propagation based on the parametric DSA, trade-off, and optimization capabilities, using the structural analysis tool capabilities for demonstration. The methodology for design change propagation will allow the design engineer to compute design sensitivity coefficients of the structural performance measures, such as stress, evaluated using the finite element analysis tools, with respect to design parameters defined in the CAD model. The change propagation approach comprises enhancement and implementation of the DICE Phase 5 design parameterization method to tie structural DSA and optimization to the Pro/ENGINEER CAD tool, and development of a design optimization method that supports structural geometric and finite element model update in the optimization process.

Integration architecture extensions will be developed and implemented in the ICEE to support the multidisciplinary DSA and change propagation capabilities, with specific attention given to the implementation of spreadsheet capabilities to display design sensitivity and performance measure data in such a manner as to allow environment users to execute design change.

The multidisciplinary DSA/design trade-off methodology and capabilities developed under this IPPD effort will be demonstrated using the HMMWV scenario described in Section V.

NSF Information Integration for Simulation Based Design and Manufacturing

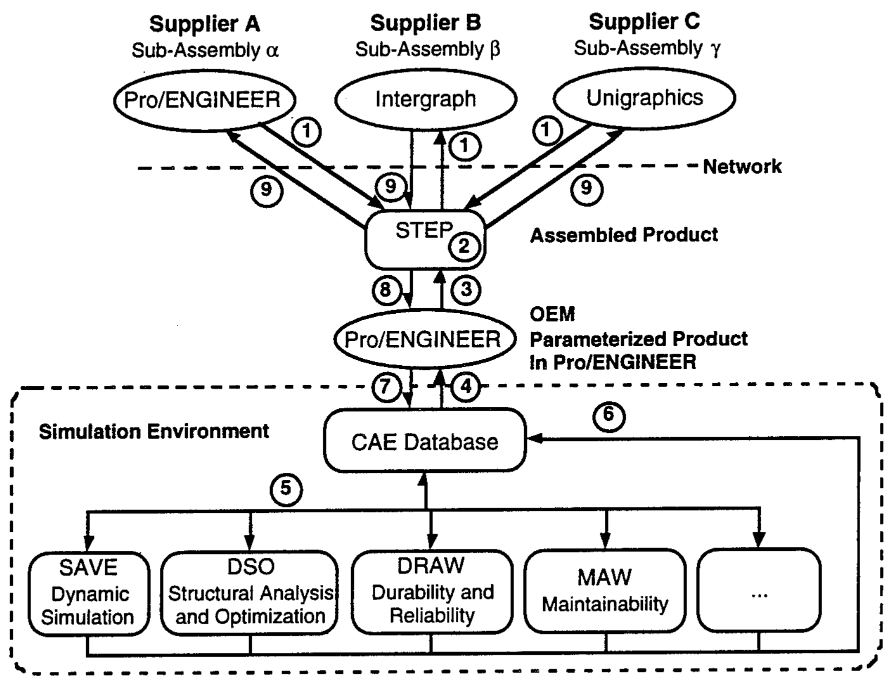
The objective of the NSF Information Integration for Simulation Based Design and Manufacturing project is to define the requirements to be imposed on product models so that simulation-based concurrent design of complex mechanical systems can be performed over a network between multiple, distributed design perspectives. Design changes resulting from simulation analysis must be merged into new versions for further simulation and analysis and/or transition to manufacturing simulation and planning. The rapid expansion of high performance networks means that it will soon be possible for an Original Equipment Manufacturer (OEM) to create a complete product model, using on a given CAD system, by assembling subsystem/part models provided by multiple suppliers created using different CAD systems. Using virtual prototyping and simulation-based design technologies, the OEM will be able to create simulation models to evaluate and tune a design simultaneously from multidisciplinary perspectives.

This effort, then, addresses a major barrier to an OEM and its suppliers coordinating in a virtual enterprise, i.e. that of typically incompatible design and analysis capabilities, particularly CAD systems, that each member of the enterprise employs in its operations. As a result, a substantial likelihood exists that data produced by the modeling and analysis capabilities of one member cannot be easily read by any other member of the enterprise. Efficient, error-free exchange of data is necessary to provide engineering users and tool applications with the appearance of a unified database.

Conducted in partnership with the Rensselaer Polytechnic Institute, this project will exploit the STEP Standards for exchange of product definition data to establish interoperability of application systems in a virtual enterprise consisting of a ground vehicle OEM and several subsystems/parts suppliers. The research effort will focus on the development and verification of the design scenario illustrated in Figure 7.1 that implements an integrated simulation based design capability in an OEM/supplier environment.

The research required to accomplish this scenario will advance the state-of-the-art in integration of engineering systems in wide area networks such as the Internet. This effort will begin to create the technology needed to break the barriers to the establishment of virtual enterprises, particularly the lack of interoperability between application systems. In addition, this research will advance the state-of-the-art in neural database design for integration of engineering systems and distributed control systems for engineering that manage the process of implementing engineering changes and the overall work flow for design and manufacturing. Benefits to the OEM/supplier enterprise will entail, among others:

- Reduced costs through the elimination of design and manufacturing mistakes using simulations.
- Reduction in product development time by enabling seamless connection between the OEM and its suppliers.
- Reduction of the number of prototype test cycles and associated time and cost through the use of simulations.
- Enablement of complete product and process design iterations in days, permitting a fundamental understanding of trade-off and systematic optimization of product quality.



1. Get CAD models of sub-assemblies from suppliers through network
2. Assemble CAD models of sub-assemblies to create product model data using STEP
3. Translate STEP model into Pro/ENGINEER
4. Retrieve Pro/ENGINEER data to create CAD database
5. Carry out simulation based design activities using Iowa tools
6. Suggest a new design based on simulation results and update CAE database
7. Update Pro/ENGINEER model for suggested design
8. Update STEP model for the suggested design
9. Propogate design changes back to suppliers' CAD models

Figure 7.1 OEM/Supplier Design Scenario for Simulation-Based Concurrent Engineering

VIII Summary of Results and Conclusions

The technological developments and demonstrations described in the preceding sections of this report show that simulation-based design presents a powerful and viable means for achievement of Concurrent Engineering (CE) goals for complex systems. Indeed, during the course of the DICE Phase 4 and Phase 5 efforts at the Center for Computer Aided Design, a number of initiatives in this area by both industry and government demonstrated the need for and the willingness to invest in the development and implementation of state-of-the-art simulation and integration technologies. New product development programs, notably Boeing's 777 aircraft design and production effort, have shown that integrated simulation based design tools can be employed to achieve CE objectives in cost reduction, reduced time-to-market, and improved product quality. The Center's efforts under DICE have demonstrated that simulation-based design technologies can achieve that same objectives for ground vehicles and other complex mechanical systems, addressing a wide variety of realistic product development concerns, using existing CAD, CAE, and database capabilities. The techniques and methodologies presented in this report show that simulation-based Concurrent Engineering holds great potential to assist in the establishment of a new era of product development, in terms of virtual prototyping and virtual enterprising, for a wide range of commercial and defense product engineering.

Specific objectives in simulation-based CE achieved by the Center under DICE Phase 4 and Phase 5 are summarized in the following. Some additional considerations regarding the potential impact on virtual enterprising and prototyping are also provided to conclude this report.

DICE Phase 4

One of the principal achievements of the DICE Phase 4 effort has been the implementation of tool integration technologies to create computationally intensive software capabilities for dynamic and structural analysis at a workspace level. As a result of this research, it has been shown that diverse computational algorithms and codes can be brought together in software environments that support robust analysis of industrial level problems to achieve specific design solutions. An excellent example of this has been the Dynamic Stress and Life Prediction workspace development under DICE Phase 4. This tool capability employs a computationally intensive variety of dynamic analysis, finite element analysis, and life prediction codes. Traditional engineering methodologies have treated these types of analyses separately, with design development occurring only after intensive and time-consuming interaction among expert users of each of these tools. Through workspace integration, these resources are at the disposal of a single user who can employ them to resolve the higher level engineering problem. Similar achievements have likewise been attained for all of the workspace capabilities developed under DICE Phase 4, achieving powerful computational capabilities supporting dynamic simulation, structural analysis and design, and life prediction. By extension of integration methods and technologies to encompass multiple design disciplines and workspace capabilities, it is reasonable to conclude that the engineering team, or product development enterprise will enjoy even more substantial advantages in resolving specific product design issues in a globally optimized manner.

Another important aspect of the Center's DICE Phase 4 effort has been demonstration of the feasibility of applying existing network, database, and communication technologies to support data intensive, large scale mechanical system design and engineering. The information required and generated by the design and engineering disciplines supported by the Tracked Vehicle Concurrent Engineering (TVCE) environment comprises an enormous amount of specialized data that needs to be consistently and comprehensibly processed and correlated to effect focused design efforts. Particularly in light of the substantially increased number of design alternatives that can be explored as the result of simulation-based design tools, the potential for information overload in the design effort presents a real problem, unless a workable information control approach is found and implemented. Integration technologies developed under the DICE effort have demonstrated that an appropriate means of information management is possible; through the use of object-oriented databases, database modeling, version control, and networked communications, design and engineering data of an industrial degree of quantity and quality can be organized in a manner that facilitates practical CE.

Finally, by eliciting the participation of industry in the project, it has been shown that the tools and methods developed are indeed applicable in an industrial setting. The exercises used to validate the TVCE capabilities are commensurate with the level of engineering problem to be found in defense and commercial vehicle development - engineering perspectives supported by the TVCE tool capabilities are those of critical concern to both military and commercial product users.

DICE Phase 5

While the application of simulation-based design technologies such as those developed under DICE Phase 4 is continuing to gain prominence in the engineering community at large, the full impact to the design and engineering process and the engineer's activities has yet to be completely understood. The Center's DICE Phase 5 effort has addressed this issue by attempting to capture the process in which simulation-based design tools are used. In order to achieve a meaningful degree of collaboration among users of the integrated CE environment, it is essential that the interdependence among individual users and distinct design perspectives be well-conceived, so that a focused, manageable design effort can be initiated. The Design Evaluation and Optimization Process developed under this DICE Phase 5 effort, while targeting operation of the design environment developed by the Center, nevertheless substantiates the relevance of the product modeling and simulation requirements and activities necessary to achieve multidisciplinary interaction. While the Center's Design Evaluation and Optimization Process (DEOP) captures only a small segment of the product life cycle, it can be concluded that the application of simulation-based design technologies will provide a means to achieve a more thorough comprehension of the design and analysis process, as well as a more consistent implementation of that process in product development enterprises.

The basic product design methodology around which the Center's DEOP has been developed applies CAD parametric solid modeling to provide a means by which engineers can engage in focused design change and evaluation. By enabling analysis engineers to view the product in terms of parameterized models, the design change methodology introduces parametric design change for CAE analysis. Heretofore, parametric solid modeling has been principally the province of CAD designers. In this manner, a fundamental link promoting design change propagation is established between design and engineering analysis - a substantial achievement in effecting meaningful collaboration. In addition, parametric methodology permits analysis engineers a means to rapidly evaluate the impact of design change suggestion, by enabling the creation of consistent computational association of design parameters with performance measures relevant to vehicle design objectives. Given the design parameter set and defined performance measures, multidisciplinary design trade-off and optimization can be attained by application of design sensitivity analysis in each design perspective.

This DICE Phase 5 R&D culminated in implementation of tool technologies supporting coordination and management of the design team and their respective activities in the design process. Extending functionality in the environment integration architecture, these capabilities enable the project manager to focus the design project to achieve customer defined objectives and track the project effort to assure compliance with development goals in terms of time and resource expenditure.

Conclusion: Virtual Prototyping and Virtual Enterprise Implications

The basic purpose of physical prototypes in the traditional design-build-test cycle is to verify that the product does in fact satisfy the requirements of the end user before full scale production is initiated. Consequently, the physical prototype is subjected to rigorous testing and evaluation under conditions the product will see in actual service. Unfortunately, many times during testing and evaluation, the prototype exhibits deficiencies in compliance with customer objectives, which necessitate a redesign of the affected systems, subsystems, or components. At this stage of product development, such redesign can be enormously expensive and time consuming. Virtual prototyping, i.e., computer models and simulations, provides a means by which vehicle systems can be modeled and analyzed early in the product development process to elicit customer interaction and focus the product development effort to achieve customer-defined goals, thereby eliminating much of the need for late stage redesign and the associated incurred costs.

The current, predominant view of virtual prototyping is centered on assessing direct interaction between the human user and system level product models, as with operator-in-the-loop driving simulation and visual virtual mock-ups. In each case, a high level evaluation of human-system interaction can be obtained. However, the impact of such interaction at subsystem, component, and part levels is more indeterminate. The development of the integrated CAD and CAE modeling and simulation technologies under the DICE effort at the Center introduces an added dimension to the concept of virtual prototyping. By linking a capability such as the Integrated Concurrent Engineering Environment (ICEE) with virtual prototyping capabilities such as the Iowa Driving Simulator, a comprehensive test program, comparable to that performed during physical prototype testing, can be implemented at the outset of the product design cycle, without the need for construction of costly physical hardware. In addition, the design change propagation capabilities of the ICEE can be employed to establish continuous fine tuning of the product design, through customer interaction via the virtual prototype.

In addition to the potential for improved virtual prototype testing and evaluation, integrated simulation-based design environments such as the ICEE will have a significant impact on the realization of the virtual enterprise. In today's intensely competitive commercial and defense environments, the establishment of new product ventures can be prohibitively expensive when organizations are faced with initial start-up costs, insufficient expertise in critical engineering disciplines, etc., particularly among small engineering and manufacturing firms. Many times, real product needs cannot be addressed as a result of the inability to assemble the necessary personnel and resources needed to develop and manufacture products, particularly specialized systems, and achieve an acceptable time to market for these products. Hence, the concept of the virtual enterprise, wherein geographically distributed personnel and resources are connected through wide area computer networks, is increasing in importance. The technologies developed under this DICE effort can provide a basis for the establishment of the virtual enterprise; the tool and environment integration techniques developed under DICE Phase 4 show how distributed computing resources can be integrated to create necessary design and analysis capabilities, and team, project, and process management methodologies developed under DICE Phase 5 demonstrate that a coordinated product development effort can be implemented and maintained among distributed personnel. As such, two critical aspects of virtual enterprise integration have been addressed under the Center's DICE efforts.

References

[1] Clark and Fujimoto, *Product Development Performance*, Harvard Business School Press, Boston, 1991.

[2] CALS Technical Report 002, Application of Concurrent Engineering to Mechanical Systems Design, 1989.

[3] Haug, E.J., "Integrated CAE Tools in Design for R&M," Technical Report R-77, Center for Computer Aided Design, The University of Iowa, 1989.

[4] Wu, J.K., Choong, F.N., Choi, K.K., and Haug, E.J., "Data Model for Simulation-Based Design of Mechanical Systems," *International Journal of Systems Automation: Research and Applications (SARA) 1*, 1991, pp. 67-87.

[5] Pugh, S. Total Design, Addison-Wesley, New York, NY, 1990.

[6] "Software Design Document for the Information Sharing System of DICE," Version 1.0, Concurrent Engineering Research Center, West Virginia University, Morgantown, WV, 1992.

[7] Ciarelli, K. "Integrated CAE System for Military Applications," Proceedings of the 16th Design Automation Conference, September, 1991, Chicago, IL, pp. 15-24.

[8] Wu, J.K., Koppes, B.A., Ciarelli, K.J., Calbeck, D., and Smuda, B., "An Integrated CAD/CAE System for Vehicle Design and Evaluation," Technical Report R-128, Center for Computer Aided Design, The University of Iowa, June 1992.

[9] Shin, S.H., Yoo, W.S., and Tang, J., "Theoretical Development and Computer Implementation of the DADS Intermediate Processor," Technical Report No. R-138, Center for Simulation and Design Optimization, The University of Iowa, July, 1992.

[10] Yim, H.J., Dopker, B., and Haug, E.J., "Simulation of Stress Histories and Fatigue Life Prediction of Mechanical Systems, Components," Technical Report No. R-8, Center for Simulation and Design Optimization, The University of Iowa, July, 1988.

[11] Owen, D.R.J., and Fawkes, A.J.F., "Engineering Fracture Mechanics; Numerical Methods and Applications," Pineridge Press Ltd., West Cross, Swansea, UK, 1983.

[12] Forman, R.G., and Hu, T., "Application of Fracture Mechanics on the Space Shuttle," *Damage Tolerance of Metallic Structures*, ASTM STP842, Philadelphia, PA, 1984.

[13] Newman, J.C., "A Crack Opening Stress Equation for Fatigue Crack Growth," International Journal of Fracture, Vol. 24, No. 3, Mar, 1984.

[14] "Fatigue Crack Growth Computer Program 'NASA/FLAGRO'," Revised March, 1989, Structures and Mechanics Division, NASA L.B.J. Space Center, Houston, TX, 1989.

[15] Haftka, R.T. and Grandhi, R.V., "Structural Shape Optimization - A Survey," *Computer Methods in Applied Mechanics and Engineering*, Vol. 57, 1986, pp. 91-106.

[16] Choi, K. K, Santos, J. L. T., and Yao, T-M., "Recent Advances in Design Sensitivity Analysis and Its Use in Structural Design Process," *SAE Transaction*, Paper No. 88073, 1988.

[17] PDA Engineering, *PAT/ANSYS Application Interface*, Software Products Division, 1560 Brookhollow Drive, Santa Ana, CA, 1989.

[18] PDA Engineering, *PAT/NASTRAN Application Interface*, Software Products Division, 1560 Brookhollow Drive, Santa Ana, CA, 1989.

[19] PDA Engineering, *PAT/ABAQUS Application Interface*, Software Products Division, 1560 Brookhollow Drive, Santa Ana, CA, 1989.

[20] Choi, K. K., Santos, J. L. T. and Frederick, M. C., "Implementation of Design Sensitivity Analysis with Existing Finite Element Codes," *ASME Journal of Mechanisms, Transmissions and Automation in Design*, Vol. 109, No. 3, pp. 385-391, 1987.

[21] Zienkiewicz, O. C. and Zhu, J. Z., "A Simple Error Estimator and Adaptive Procedure for Practical Engineering Analysis," *International Journal for Numerical Methods in Engineering*, Vol. 24, 1987, pp. 337-357.

[22] Rank, E. and Zienkiewicz, O. C., "A Simple Error Estimator in the Finite Element Method,"

Comm. in Applied Numerical Methods, Vol. 3, 1987, pp. 243-249.

[23] Chang, K. H. and Choi, K. K., "An Error Analysis and Mesh Adaptation Method For Shape Design of Elastic Solids," *Computers and Structures*, Vol. 44, No. 6, 1992, pp. 1275-1289.

[24] Haftka, R. T. and Adelman, H. M., "Recent Developments in Structural Sensitivity Analysis," *Structural Optimization*, Vol. 57,1990, pp. 137-151.

[25] Haug, E. J., Choi, K. K., and Komkov, V., *Design Sensitivity Analysis of Structural Systems,* Academic Press, New York, 1986.

[26] Choi, K. K. and Chang, K. H., "A Study on Velocity Field Computation for Shape Design Optimization," submitted to *Journal of Finite Elements in Analysis and Design,* December 1992

[27] Santos, J. L. T. and Choi, K. K., "Integrated Computational Considerations for Large Scale Structural Design Sensitivity Analysis and Optimization," *GAMM Conference on Discretization Methods and Structural Optimization,* University of Siegen, FRG, October, 1988.

[28] Santos, J. L. T., Godse, M. M., and Chang, K. H., "An Interactive Post-Processor for Structural Design Sensitivity Analysis and Optimization: Sensitivity Display and What-If Study," *Computers and Structures,* Vol. 35, No. 1, pp. 1-13, 1990.

[29] Choi, K. K. and Chang, K. H., "Shape Design Sensitivity Analysis and What-if Workstation For Elastic Solids," *AIAA 32nd SDM Conference,* Paper No. 91-1206, April 1991.

[30] Apollo Computer Inc., *BSD Command Reference,* 330 Billerica Road, Chelmsford, MA 01824, 1988.

[31] Anderson, G. and Anderson, P., *The UNIX C SHELL Field Guide,* Prentice-Hall, Englewood Cliffs, NJ 07632, 1986.

[32] Meyer, B., *Object-Oriented Software Construction,* Prentice Hall, Englewood Cliffs, NJ 07632, 1988.

[33] Wu, J.K., Choong, F.N., Kvidera, D.A., Jiang, X., Hsieh, L.J., McGee, B., and Sangareddi, K. "Application Wrappers in a Simulation-Based Concurrent Engineering Environment," Technical Report R-175, Center for Computer Aided Design, The University of Iowa, 1994.

[34] Wu, J.K., Fogle, A., Wang, J.Y., and Lu, J.K., "A Dynamics Workstation," Submitted to the 1989 ASME Design Automation Conference.

[35] Tsai, F.F., "Automated Code Generation Methods for High Speed Simulation of Rigid Body Dynamic Systems," Ph.D. Thesis Proposal. Deptartment of Mechanical Engineering, The University of Iowa, 1988.

[36] Lin, T.C., Wang, J.Y., Luh, C.M., and Yeh, H.C., "A Two-Stage Assembly Algorithm for Constrained Multibody Systems," Center for Computer Aided Design, The University of Iowa, 1995.

[37] Choi, K.K., Chang, K.H., Wang, J.Y., Tang, J., Ogarevic, V., Tsai, C.S., and Vujosevic, R., "Simulation-Based Concurrent Engineering Tools and Infrastructure," *The First World Congress of Structural and Multidisciplinary Optimization*, Goslar, Lower Saxony, Germany, 1995.

[38] Bestle, D. "Optimization of Automotive Systems," Concurrent Engineering Tools and Technologies for Mechanical System Design (E.J. Haug, ed.), Springer-Verlag, Heilberg, 1993.

[39] Tak, T. and Kim, S.S., "Design Sensitivity Analysis of Multibody Dynamic Systems," Technical Report R88, Center for Computer Aided Design, The University of Iowa, Iowa City, IA, 1990.

[40] Chang, K.H., Yu, X., and Choi, K.K., "Structural Reliability Analysis Using Design Sensitivity Analysis and Optimization (DSO)," *The First World Congress of Structural and Multidisciplinary Optimization*, Goslar, Lower Saxony, Germany, 1995.

[41] Chang, K.H., Yu, X., and Choi, K.K., "A Hybrid Sensitivity Analysis Method for Structural Durability Design," (in preparation).

[42] Arora, J.S., Introduction to Optimal Design, McGraw-Hill, 1989.

[43] Mayer, R.J., ed., *IDEF0 Function Modeling*, Knowledge Based Systems, Inc., 1990.

Relevant DICE Publications

The following is a list of technical reports and papers published or submitted for publication by CCAD personnel under support or partial support from the DICE Phase 4 and Phase 5 research projects.

Chang, K.H., Choi, K.K., Tsai, C.S., Chen, C.J., Choi, B.S., and Yu, X. "Design Sensitivity Analysis and Optimization Tool (DSO) for Shape Design Applications," *Computing Systems in Engineering*, Vol. 6, No. 2, pp. 151-175, 1995.

Chang, K.H., Perng, J.H., and Choi, K.K., "Introduction to the Design Sensitivity Anlaysis and Optimization Tool (DSO) for Sizing Design Appplications," Technical Report R134, Center for Computer Aided Design, The University of Iowa, Iowa City, IA, April, 1992.

Choi, K.K. and Chang, K.H., "A Study on Velocity Field Computation for Shape Design Optimization," *Journal of Finite Elements in Analysis and Design*, 15, pp. 317-341, 1994.

Choi, K.K., Wu, J.K., Chang, K.H., Tang, J., Wang, J., and Haug, E.J. "Large Scale Tracked Vehicle Concurrent Engineering Environment," *STRUCTURAL OPTIMIZATION 93, The World Congress on Optimal Design of Structural Systems* (Ed. J. Herskovitz), Rio de Janeiro, Brazil, August 2-6, 1993, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1995.

Choi, K.K., Chang, K.H., Wang, J.Y., Tang, J., Ogarevic, V., Tsai, C.S., and Vujosevic, R., "Simulation-Based Concurrent Engineering Tools and Infrastructure," *The First World Congress of Structural and Multidisciplinary Optimization*, Goslar, Lower Saxony, Germany, 1995.

Shin, S.H., Yoo, W.S., and Tang, J., "Effects on Mode Selection, Scaling, and Orthogonalization on the Dynamic Analysis of Flexible Multibody Systems," *Mechanics of Structure and Machines*, Vol. 21, No. 4, 1993.

Tang, J. "An Integrated System for Dynamic Stress Computation and Fatigue Life Prediction of Mechanical Systems," *Proceedings of the 1993 ASME PYP Conference*, Denver, CO, 1993.

Tang, J., "Introduction to Integrated Systems for Dynamic Stress and Life Prediction of Mechanical Systems," Technical Report R171, Center for Computer Aided Design, The University of Iowa, Iowa City, IA, December, 1993.

Tang, J., "A General Methodology for Durability and Reliability Analysis of Mechanical Components," Technical Report R180, Center for Computer Aided Design, The University of Iowa, Iowa City, IA, 1994.

Tsai, C.S., Chang, K.H., and Wang, J., "Integration Infrastructure for a Simulation-Based Design Environment", Proceedings of the Computers in Engineering Conference and the Engineering Data Symposium, *ASME Design Theory and Methodology Conference*, pp. 9-20, Boston, MA, August 1995.

Wu, J.K., Choong, F.N., Twu, S.L., Kim, S.S., and Baek, W.K., "A Global-Local Data Model Scheme for Integrating Disciplines in a Concurrent Engineering Environment, Center for Computer Aided Design, The University of Iowa, Iowa City, IA, November, 1991.

Wu, J.K. and Choong, F.N., " A Process Modeling System for Concurrent Engineering of Mechanical Systems," *Proceedings of the 1992 AI in Design Conference*, 1992.

Wu, J.K. and Choong, F.N., "Tool Integration and Process Modeling for Concurrent Engineering of Mechanical System Design and Analysis," Technical Report R137, Center for Computer Aided Design, The University of Iowa, Iowa City, IA, June, 1992.

Wu, J.K., Choong, F.N.,, Kvidera, D.A.,, Jiang, X., Hsieh, L.J.,, McGee, B., and Sangareddi, K., "Application Wrappers in a Simulation-Based Concurrent Engineering Environment," Technical Report R175, Center for Computer Aided Design, The University of Iowa, Iowa City, IA, 1994.

Appendix A: TVCE Global Data Model Entities

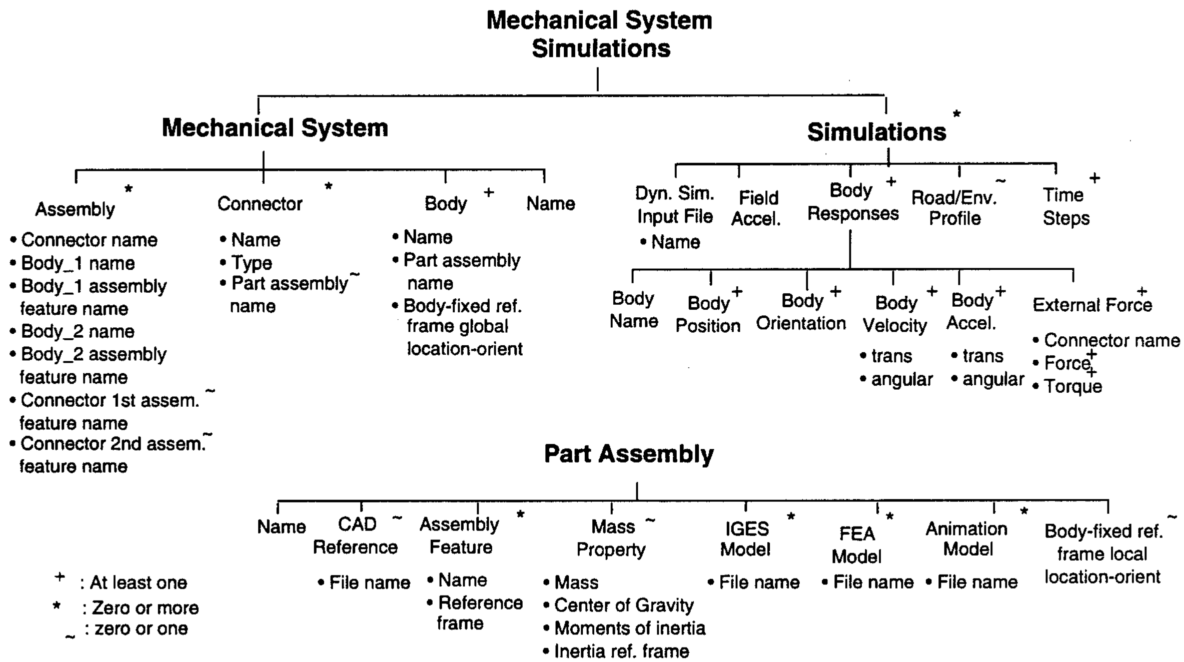


Figure A-1 TVCE Environment Global Data Model

This appendix provides a brief explanation of the global data model, shown in Figure A-1.

The global database has two catalogs: the mechanical system simulation catalog and the part assembly catalog. A mechanical system simulation contains several versions of mechanical systems. Each version of a mechanical system may have results obtained from several dynamic simulations.

A ***mechanical system*** is composed of several bodies, connectors, and system assemblies between bodies and connectors. The simplest mechanical system has a body. A ***body*** is a part assembly whose motion in three-dimensional space is important from the point of view of the whole mechanical system dynamics, and whose loading histories can be determined from dynamic analysis. A ***part assembly*** is composed of parts that are rigidly assembled, for instance through welding, rivets, and bolts and nuts. A ***part*** is an entity that has homogeneous material properties and specific geometry.

A ***connector*** functions as a kinematic joint or a force element in a mechanical system from the dynamics point of view. The type of a connector can be revolute, translational, cylindrical, universal, or spherical joint (as a joint), or translational or torsional spring, damper, or actuator (as a force element). Because the connector has a specific geometry and material properties, it is treated as a part assembly in this data model.

A ***system assembly*** defines the connectivity between a pair of bodies through a connector. The connectivity information for a system assembly identifies three things: the connected bodies, the features where the bodies are connected, and the name of the connector that joins them. The degrees of freedom at each connection can be deduced from the connector names.

In the real world, a part assembly can be composed of several (member) parts, but in this global database the ***part assembly*** does not contain information about its member parts.

For a part assembly, the TVCE global database stores:

- the name of the part assembly,
- the CAD reference of the part assembly (which is a file name and its path in a CAD system),
- names and reference frames of assembly features that are used to connect the part assembly with connectors or other bodies,
- composite mass property data of the part assembly (including total mass, total moments of inertia, center of gravity, and the reference frame used to define the moments of inertia),
- IGES representations of the part assembly,
- FEA representations of the part assembly (which contains material property data and geometry that structural analysis applications would need),
- animation representations of the part assembly (i.e., mod files), and
- the location and orientation of body fixed reference frame relative to the geometry construction reference frame of the part assembly.

For a body, when an assembly feature reference frame is used to connect the body with another body through a joint, this frame is a joint reference frame. Similarly, if the assembly reference frame is used to connect the body with a force element, the reference frame is a force element reference frame. In a body, the assembly feature reference frames are described relative to the body fixed reference frame. For a part assembly, the body fixed reference frame is defined relative to the geometry construction reference frame. For dynamic analysis, the body fixed reference frame is defined relative to the inertial reference frame. The geometry construction reference frame is used in defining finite element models for structural analysis.

For each ***simulation***, the global database stores a general message regarding the simulation, the dynamic analysis (DADS) input file name, the profile of the road (if this exists), the time step history, the field acceleration vector (e.g. gravitational acceleration), and body responses of each body of interest. For a body response, the global database stores the body name, body position and orientation histories, a body velocity history, a body acceleration history, and force and torque histories at each assembly feature.

Appendix B: Tracked Vehicle Example Testbed

The TVCE testbed environment used to support the generic M1A1 Abrams tracked vehicle exercise consisted of a SUN Sparc10 workstation as the "local" machine where most TVCE software is located, an HP server as the "remote" machine where number computationally intensive applications, such as CAD modeling, dynamic analysis, and finite element analysis are performed, and a number of color X-terminals to operate and display the software. The machine configuration defined for the test is illustrated in Figure B-1.

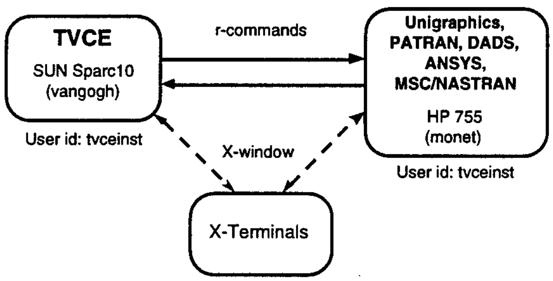


Figure B-1 Machine Configuration

The software located in the testbed on the SUN (/ccad/bin) workstation consists of the four TVCE workspaces: CCS1.0 (include IGES2mod), TVWS, DSLP2.0, and DSO3.0; DDS (Design Data Server), and two data wrappers DSLP_wr and DSO_wr. Note that the CCS and TVWS wrappers are implemented as part of the CCS and TVWS executables, respectively. Also, the TVWS Informix database containing the tracked vehicle definition utilized for this test was located in the /ccad/db directory.

The commercial CAD and CAE tools supporting the TVCE Testbed were located at the remote machine. These tools included Unigraphics 10, PATRAN2.5, DADS6.5, ANSYS 4.4a, and MSC/NASTRAN 67R2. The remote utilities, including script files, the remote program, and finite element interface programs were stored in the /ccad/lib directory. TVWS DADS template files were located in the /ccad/lib/tvws directory. Table B-11 lists the various sizes of the TVCE software elements. Table B-1 shows that a total of 50 Megabytes disk space is required to accommodate the TVCE software.

Table B-1 Size of TVCE Software

Software	Modules	Local	Remote
DDS	Database	9.266*	0.0
	Wrappers	3.468	0.0
CCS	ccs	1.900	0.0
	iges2mod	1.491	0.0
TVWS	tvws	4.727	0.0
	data and files	8.543	0.447
DSLP	dslp	6.358	1.149
DSO	dso	13.061	0.781
Others		0.127	0.021
Total		48.941	2.398

* Unit: Megabyte

Appendix C: TVCE Environment Test Operational Statistics

Table C-1 Operator and Computer Time Requirements

TVCE Software	Job	Human Efforts	Computer (Clock) Local	Remote
CCS	Creating mechanical system	30 min		0
	UG to IGES	0	45 min	0
	IGES Translator	-	-	-
	Export to DDS	10 min		0
Subtotal		40 min	45 min	0
TVWS	Import mass and create dynamic model	30 min		0
	DADS job	0	0	3 hours
	Export to DDS	10 min		0
Subtotal		40 min	0	3 hours
DSLP	Import from DDS	5 min		0
	ANSYS static run	0	0	1 hour
	ANSYS stress coefficients	0	0	12 hours
	Dynamic stress computation	26 min		0
	Life Prediction	0	32 min	0
Subtotal		31 min	32 min	13 hours
DSO	Import/Export/2-D plot for peak load	30 min		0
	Define design model	40 min		0
	ANSYS static run	0	0	26 min
	Sensitivity Computation	0	20 min	0
	ANSYS restart run	0	0	30 min
	Sensitivity Display	20 min		0
	What-if study	20 min		0
Subtotal		110 min	20 min	56 min
Total	(in hours)	3.7	2.7	17

Table C-2 Generated File Sizes

TVCE Software	Files	File Sizes Local	Remote
DDS	DDS Database	8.116	0.0
CCS	Local files	3.448	0.0
	IGES files	17.393	17.393
	PATRAN files	3.010	3.010
TVWS	DADS result files	8.444	0.0
	DADS job	0.0	29.269
DSLP	Local files	15.931	0.0
	PATRAN files (2)	3.886	3.886
	ANSYS static runs (2)	0.0	110.980
	ANSYS stress coefficients (2)	0.0	189.718
DSO	PATRAN files	1.943	1.943
	DSO database	8.914	0.0
	ANSYS static run	0.0	55.490
	ANSYS restart run	0.0	76.668
Total		71.085	488.357

* Unit: Megabyte

Appendix D: Design Evaluation and Optimization Process, Levels 2-4

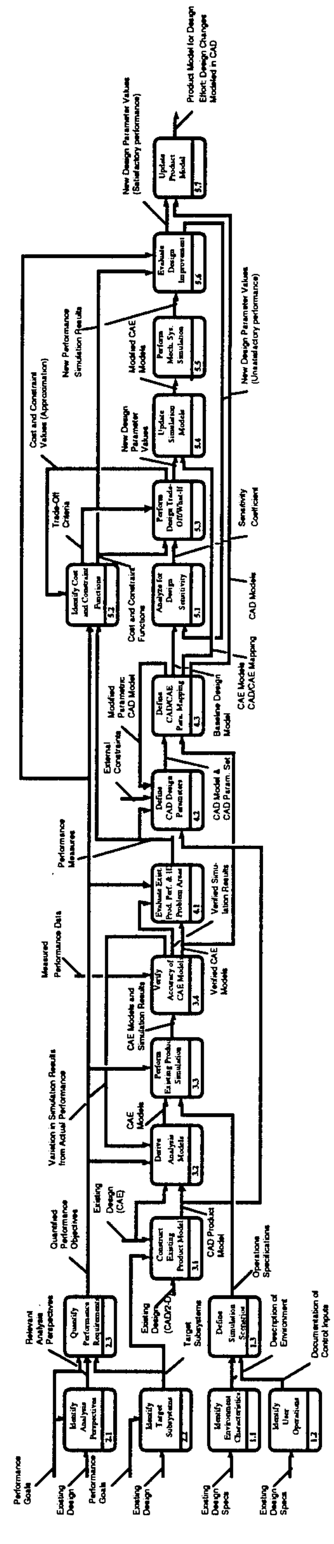
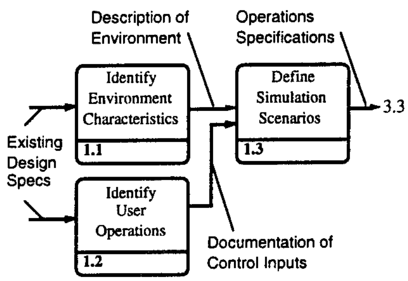


Figure D-1 Level 2 of the Design Evaluation and Optimization Process

Segment Activities

Description

1.1-1.3 Identify Environment Characteristics (1.1)



The objective of Activity 1.1 is to describe the external environment in which the mechanical system operates to enable the construction of realistic performance simulation scenarios. The information to be described in this activity includes road and terrain profiles, soil and surface conditions, and external load carrying conditions such as earth-moving, equipment hauling, equipment tie-downs, etc. Additional information that may need to be described in this activity would also include identification of the mechanical system operating envelope, obstacles, and motion characteristics of the environment in which the system resides, such as HMMWV transport on a rail car (the rail car is moving), or loading an M1A1 tank onto a ship (the ship is moving in a given sea state). Environmental characteristics representing extreme conditions should also be identified in order to characterize "worst case" scenarios.

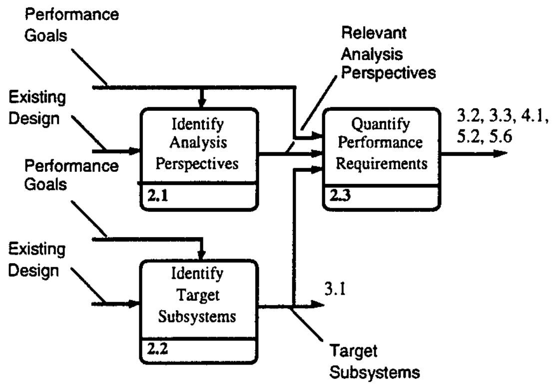
Identify User Operations (1.2)

The objective of this activity is to identify the control inputs the mechanical system user would employ to operate the system in the achievement of the tasks for which the system is to be designed. This information would include mechanical system velocity and acceleration, braking, and steering. This information will be used in conjunction with the environment description produced in activity 1.1 to define the mechanical system operation scenarios to be simulated in downstream dynamic CAE analyses. The output of this activity, then, will consist of descriptions of specific vehicle paths, together with velocity and acceleration values, representing the complete range of control functions to which the system is subjected during the performance of the tasks for which it is to be designed.

Define Simulation Scenarios (1.3)

System evaluation for the existing designs will require the definition of a specific, consistent, and comprehensive set of scenarios that represent the spectrum of system operations. Using the external environment description and operator control actions documented in the previous activities, test scenarios will be defined supporting evaluation of system performance with respect to both existing specifications and defined objectives. A scenario can include such elements as road profiles, vehicle paths, constraints on vehicle motion, environ-mental conditions, which should be specified to produce performance data relevant to the goals of the design effort. The result of this activity will be a specific sequence of vehicle operations which will provide a basis for meaningful comparisons between various system design configurations.

2.1-2.3 Identify Analysis Perspectives (2.1)



This activity will be performed to qualify specific areas of improvement for the mechanical system in question in an engineering design and analysis context. Typically, mechanical system design improvement is expressed in generic terms such as improved safety, performance, reliability, reduced cost, etc. For successful achievement of the mechanical system design improvement and evaluation process, it is necessary to qualify these generic objectives relevant to specific engineering analysis disciplines. For example, an improvement in system safety may represent an improvement of the system structural performance, or it may represent an improvement in vehicle [dynamic] stability. This activity will be performed to identify those design and analysis disciplines whose input and expertise is required to achieve the specified performance objective.

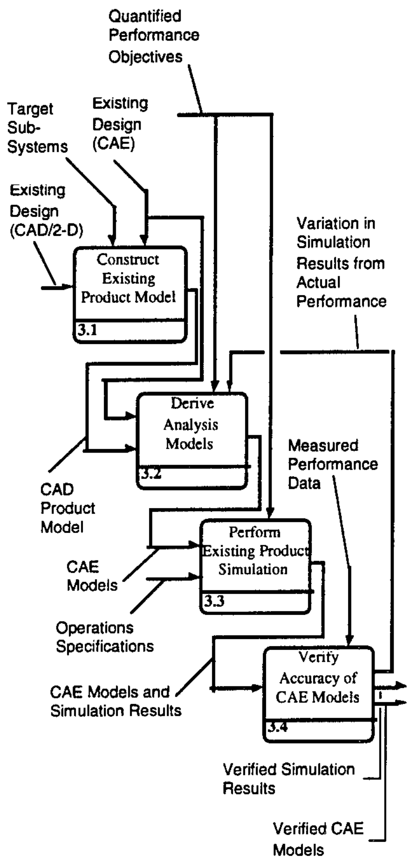
Identify Target Subsystems (2.2)

The objective of this activity is to identify the specific mechanical system parts, components, or subsystems for which change in design will yield performance improvement sufficient to satisfy the specified performance objectives. For example, an improvement in ride quality may target the performance of the vehicle's suspension subsystem, whereupon any or all of the constituent components of this subsystem may be considered for design improvement. Specification of the particular "components of interest" is required in order to identify quantifiable performance goals as well as to control the level of detail required of the product model hierarchy supporting the design and analysis effort.

Quantify Performance Requirements (2.3)

In order to provide the design effort with a relevant referent to determine success, it is necessary to relate the qualified performance requirements in terms of target values that are meaningful with respect to the analysis disciplines identified for the design effort and the CAE tool capabilities supporting these disciplines. A meaningful, consistent set of quantified performance objectives is essential to maintain the focus of the design improvement effort throughout the DEOP process, as this data provides control referents for defining appropriate analysis models (Activity 3.2), defining the scope of the simulations required to assess compliance with these objective (Activity 3.3), evaluating the performance of the existing design and pin-pointing problem areas (Activity 4.1), as well as providing direct input for formulation of trade-off criteria (Activity 5.2), and providing a referent for evaluating design improvement (Activity 5.6). Since all analysis perspectives identified for the effort will be required to perform all of these activities, quantified performance objectives must be established for each analysis discipline.

3.1-3.4 Construct Existing Product Model (3.1)



The objective of this activity is to construct a model of the existing product that provides the minimum sufficient information to support downstream CAE analysis and design modeling requirements for the design effort. Target subsystems and existing CAE models are employed as controls for this activity to determine the minimum level of detail for the product model. For instance, for a design effort targeting improvement of a component in the suspension subsystem, detailed CAD geometry, mass, and material data is required for only that component to support downstream structural analyses, as well as a detailed assembly hierarchy for the suspension subsystem as a whole to support downstream dynamic analysis. Fastener information for the suspension subsystem would also be detailed to support maintenance analysis requirements if included in the design effort. Only basic geometry, mass property, material property, and joint location data is required to model the remainder of the suspension subsystem, the remaining vehicle systems, and the product model hierarchy. Existing CAE model data of the target subsystems or components (if available) should be employed to assure accurate construction of the product model. The output of the activity is a "base" CAD product model of the minimum system configuration from which analysis models appropriate for the design effort can be derived.

Derive Analysis Models (3.2)

Derivation of analysis models is based on both the modeling requirements of the particular analysis tools to be used and the view concept supported by the integration architecture of the CAE Simulation Environment. Each analysis discipline engaged in the design effort requires a representation of the product system that is consistent with the needs of that analysis discipline. As such each analysis discipline will need to augment the data provided in the base CAD product model produced in the preceding activity, and may be required to redefine the assembly hierarchy of the base product model. Each analysis discipline, e.g., structural, dyna-mics, maintainability, etc. will establish a "map-ping" between the base product model of the existing product and its particular view structure. This mapping will provide the basis for cross disciplinary design change communication downstream, therefore it is essential that each analysis discipline maintain a consistent mapping scheme throughout the simulation and design process. This activity also entails the creation of dynamic, structural FEA, geometric polygon, etc., models of the existing product design depending on the analysis disciplines involved in the design effort.

Perform Existing Product Simul-ation (3.3)

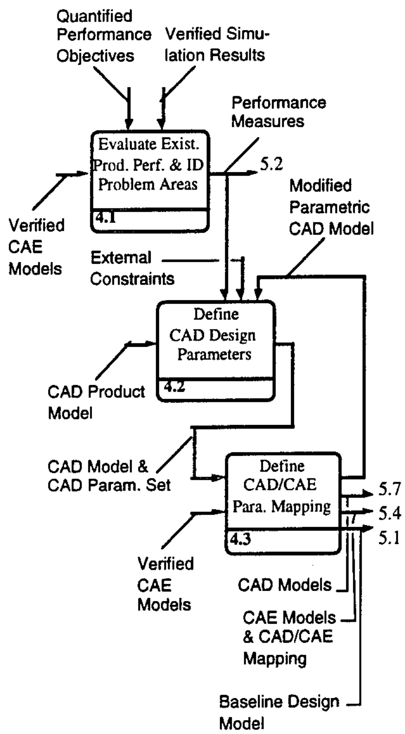
Having constructed the analysis models in Activity 3.2 and obtained the operations specification for the simulation scenarios in Activity 1.3, the objective of this activity is to assembly the CAE analysis simulations for the existing product design, launch them, and obtain the analysis results. Depending on the types of simulations needed for the design effort and the availability of pre-existing data, some simulations may require output data from other simulations before they can be executed. For instance, simulation of component durability, i.e., life prediction, requires a dynamic load history to calculate dynamic stress. Unless this data is already available, this analysis will require the output of a dynamic simulation and analysis. Definition of an appropriate set of initial conditions for each analysis simulation during this activity is also required prior to launching simulations. Each simulation should be structured to yield data that will provide a meaningful comparison with the quantified performances objectives, therefore these objectives are considered as a control factor in the definition of appropriate simulations and formulation of output results.

Verify Accuracy of CAE Models and Simulations (3.4)

The objective of this activity is verify the accuracy of the models and simulations used to reproduce the performance of the existing product as evidenced by the results of the preceding activity. Verification (control) of the simulation results naturally presumes a basis for comparison in the form of actual, measured data. Unless, however, extensive test data is available for this activity, the engineers and analysts supporting the design effort must rely on their objective judgment to assure the accuracy of the CAE models and simulation results. Should this judgment or variations in the simulation results from measured performance data indicate otherwise, a control feedback loop is defined to re-examine the derivation of the analysis models and simulations. The CAE process as given assumes that activities 3.2, 3.3, and 3.4 will be iterated until a satisfactory level of accuracy in simulation results is obtained.

4.1-4.3 Evaluate Existing Product Performance and Identify Problem Areas (4.1)

This activity performs the initial evaluation of the existing product as referenced to the quantified performance objectives. Whereas in preceding activities



the specified objectives have acted as controls to structure the models and simulations to produce data appropriate for comparison with the objectives, this activity actually performs this comparison. As a result, this activity presents the first instance of design decision-making in the DEOP. A decision must be made as to whether or not the existing product satisfies the stated performance objectives. This is unlikely, since as previously stated, performance goals typically imply charac-teristics the existing product does not exhibit. However, should the existing design satisfy these objectives, then the SDP process will terminate at this stage.

Should the existing product not satisfy the performance objectives, then this activity will also identify specific problem areas which represent insufficient performance. Since the analysis models and simulations have been structured to yield data relevant to performance objectives, this data then serves as the basis for the establishment of performance measures to define the CAD design parameter set in the succeeding activity and support the determination of design sensitivity analysis formulations in the design improvement phase.

Define CAD Design Parameter Set (4.2)

The objective of this activity is to define/select CAD parameters in the product model of the target subsystems or components which will be employed to modify the existing product design in the identified problems areas. In effect, this activity reduces the CAD design parameter set to a set that targets design change with respect to the identified problems areas and can be analyzed with respect to the defined performance measures. The output of this activity is the CAD models of the target components together with the functional design change parameter supporting design trade-off and multi-disciplinary design optimization during the remainder of the design process.

Define CAD/CAE Parametric Mapping (4.3)

The objective of this activity is to parameterize the CAE analysis models and establish a mapping scheme between the CAD design parameters defined in Activity 4.2 and these CAE model parameters. This mapping scheme will be structured to enable rapid update of the analysis models, propagation of design change suggestions across analysis disciplines using the CAD model from the preceding activity as an intermediary, and enable design changes implemented in the CAE models to be propagated back to the CAD product model. It is assumed that a number of iterations will be required to establish appropriate and workable CAE parametric models and a parametric model mapping scheme. Therefore the SDP incorporates a control feedback from this activity to the preceding activity (Definition of CAD Design Parameters) to assist in the selection of CAD design parameters that can be mapped to a CAE model representation. The aggregate of the parametric CAD design model, the parametric CAE analysis models, and the mapping scheme constitute the essential elements of the baseline design model produced by this activity. The baseline design model will provide the basis for optimized product design determination during the remainder of the DEOP.

5.1-5.7 Analyze for Design Sensitivity (5.1)

The objective of this activity is to compute design sensitivity coefficients of system performance measures identified in Activity 4.1 with respect to CAD model design parameters identified in Activity 4.2. The design sensitivity coefficients specify the influence of design parameters on performance measures - the methodology to be used to support design modifications. The design sensitivity coeffi-cients computed in this activity will be used to support design trade-off and what-if studies. Design sensitivity coefficients will be computed by those analysis disciplines that support the identified performance measures. The design sensitivity coefficients will be propagated to the global mechanical system product model and assembled at the system level.

Identify Cost and Constraint Functions (5.2)

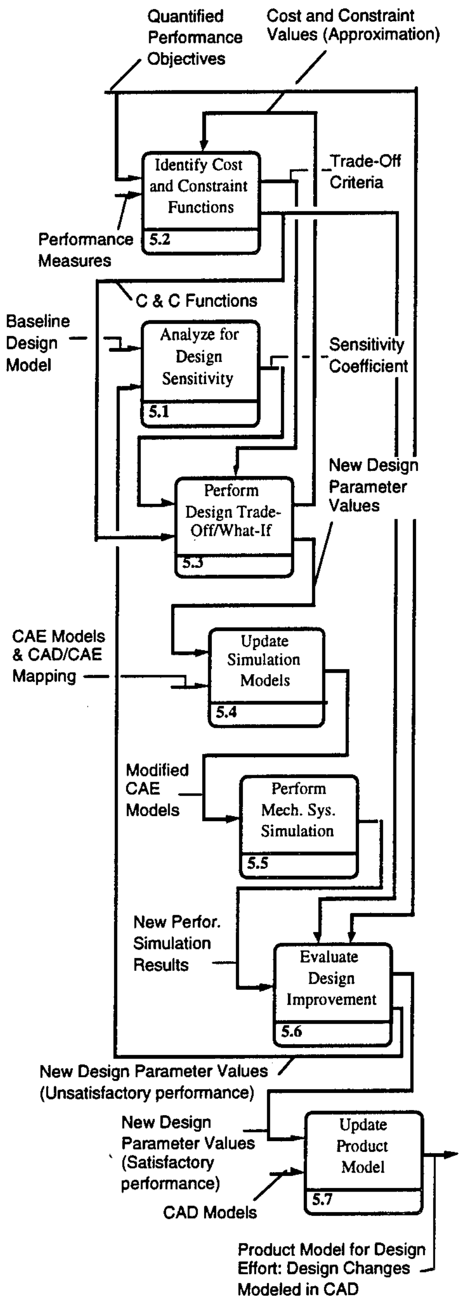
The objective of this activity is to define cost functions (the function to be minimized) and constraint functions (the functions to stay within specified bounds) to support design trade-off ana-lyses and design iterations. Cost and constraint functions are selected from performance measures defined in activity 4.1. Only one cost function can be defined for the mechanical system. Multiple constraints can be defined for the mechanical system by selecting existing performance measures. Also, upper or lower bounds must be defined for each constraint.

Perform Design Trade-Off/ What-If (5.3)

The objective of this activity is to perform design trade-off using numerical algorithms to obtain design directions that will improve designs and use these directions to support what-if studies. After the design direction is determined, the analyst can provide a step size to perturb the design to carry out the what-if study. By performing the what-if study, cost and constraint function values for the perturbed design will be approximated using the design perturbation and design sensitivity coefficients, without the need for regenerating the CAE models and analysis processes.

Update Simulation Models (5.4)

When potential design improvements have been identified from preceding trade-off and what-if studies, the CAE analysis models will be updated via assignment of new parameter values in preparation for re-analysis of system performance.



Perform Mechanical System Simulation (5.5)

Once the CAE analysis models have been update to reflect potential design improvements, the updated models and existing simulations scenarios will be assembled and launched for analysis of new design performance. The resulting analysis data will be employed in the next activity to assess performance improvement.

Evaluate Design Improvement (5.6)

Evaluation of performance simulation results of potential design improvements will proceed much as in activity 4.1, using quantified performance objectives to determine the success of the new design. In addition, new simulation results will be assessed against the existing product simulation results to assist in determining relative design improvement and facilitate determination of design change direction in successive design iterations. Should design improvement simulation results not achieve the objectives of the design effort, Activi-ties 5.1, and 5.3 through 5.6 will be iterated until a successful design improvement is achieved. Design improvement analysis in each successive iteration will employ the analysis model configur-ation, i.e., parameter values, from the preceding iteration, rather than returning to the original parameter set values representing the existing design.

Update Product Model (5.7)

When an acceptable level of mechanical system performance has been achieved, the CAE parameter values representing the successful design improve-ment are propagated to the CAD product model via the CAD/CAE mapping scheme defined in Activity 4.3. The CAE SD process will then terminate with a yield of a product model for the design effort with design changes modeled in CAD form.

(1)

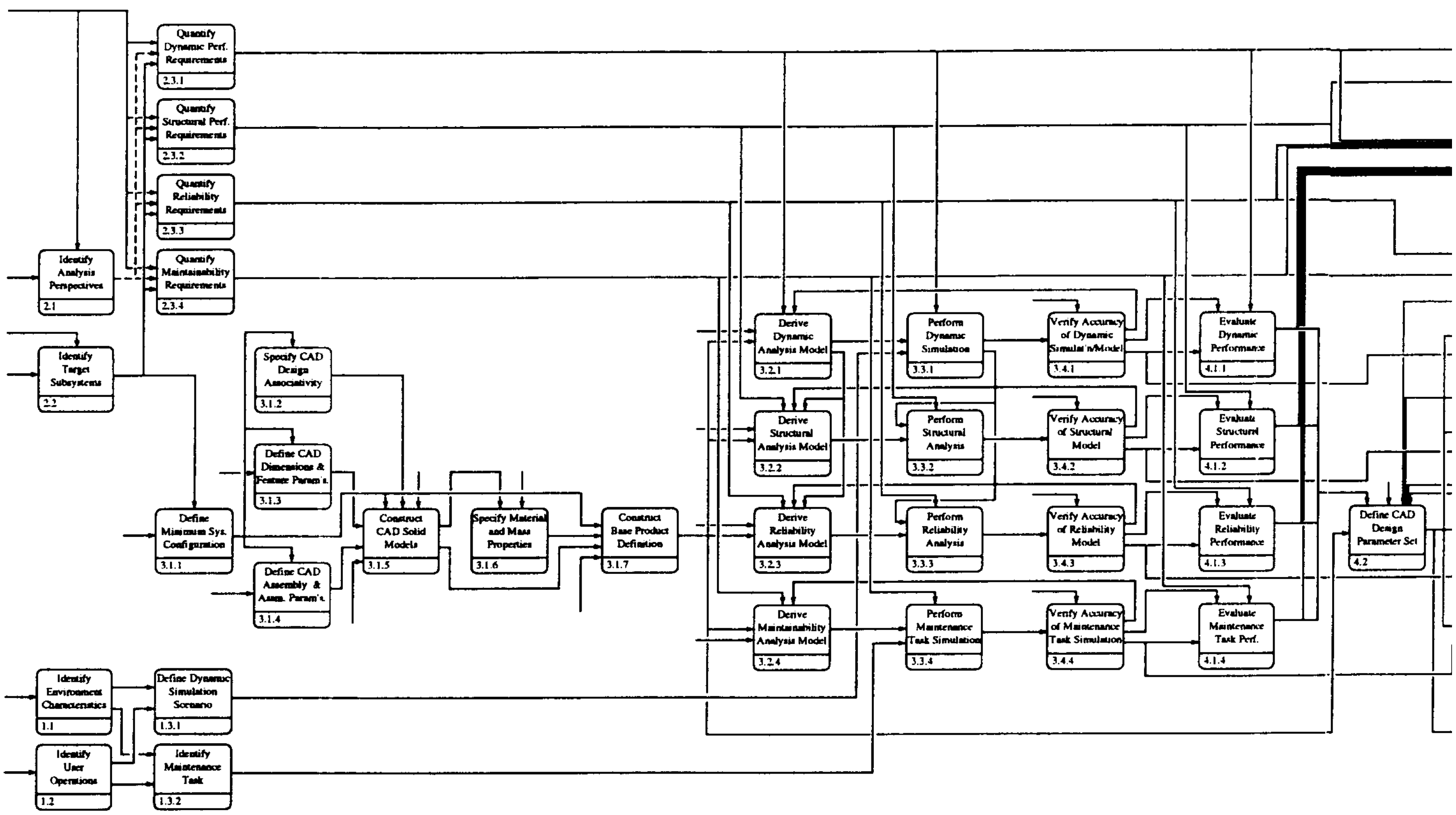
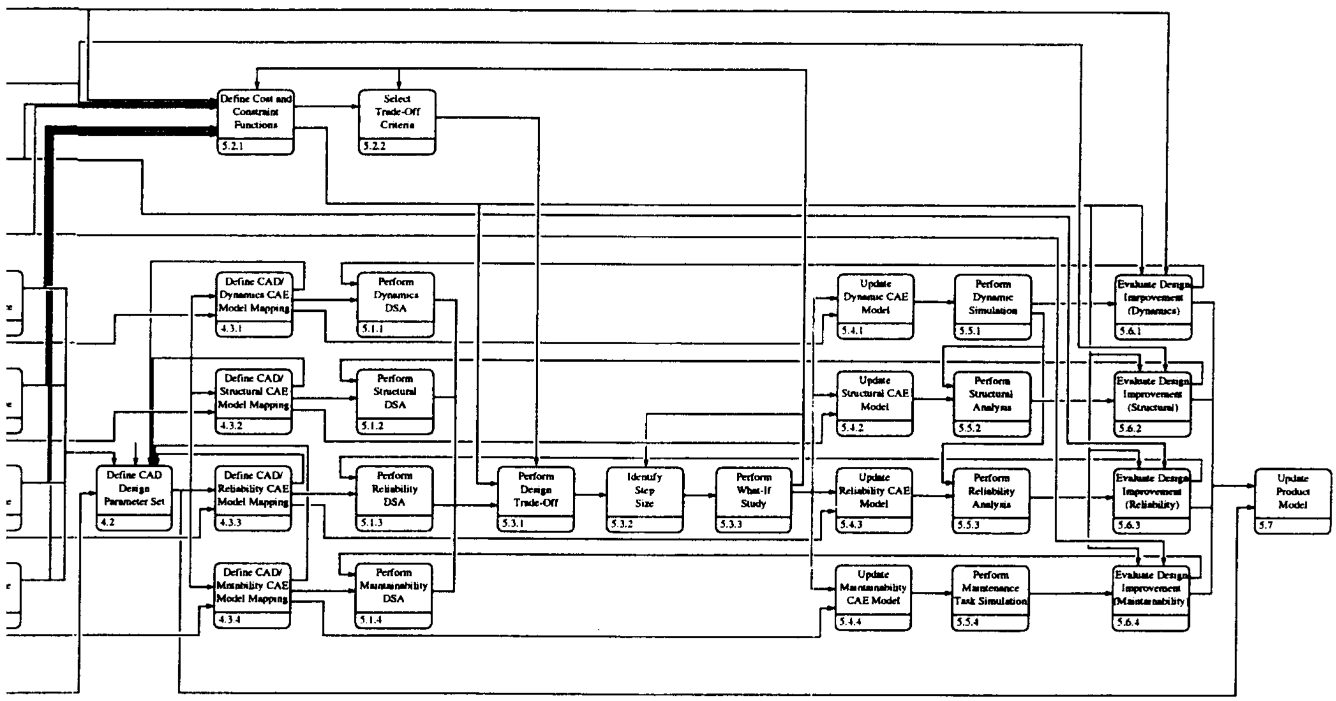


Figure D-2 Level 3 of the Design Evaluation and (



Evaluation and Optimization Process

(1)

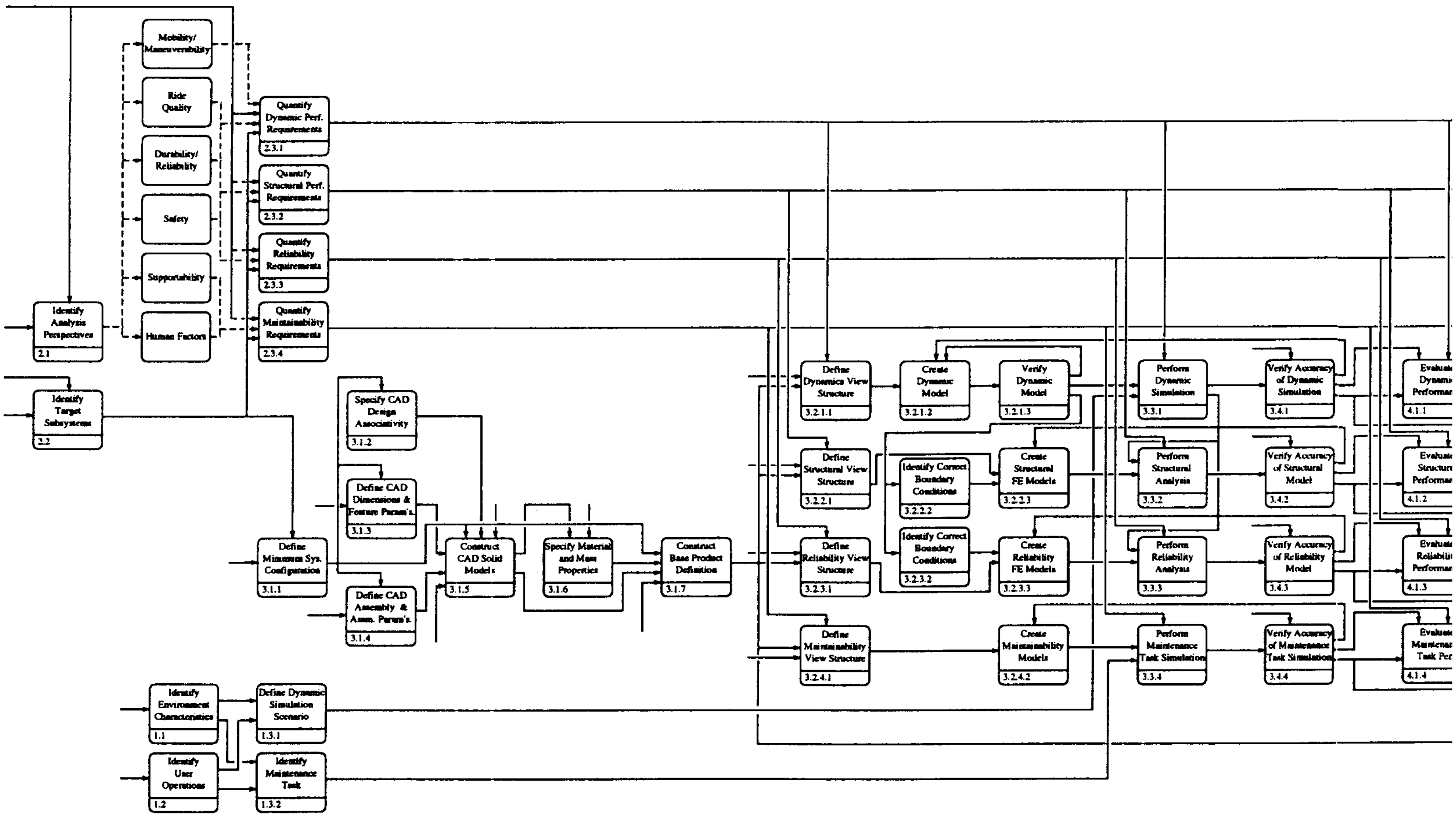
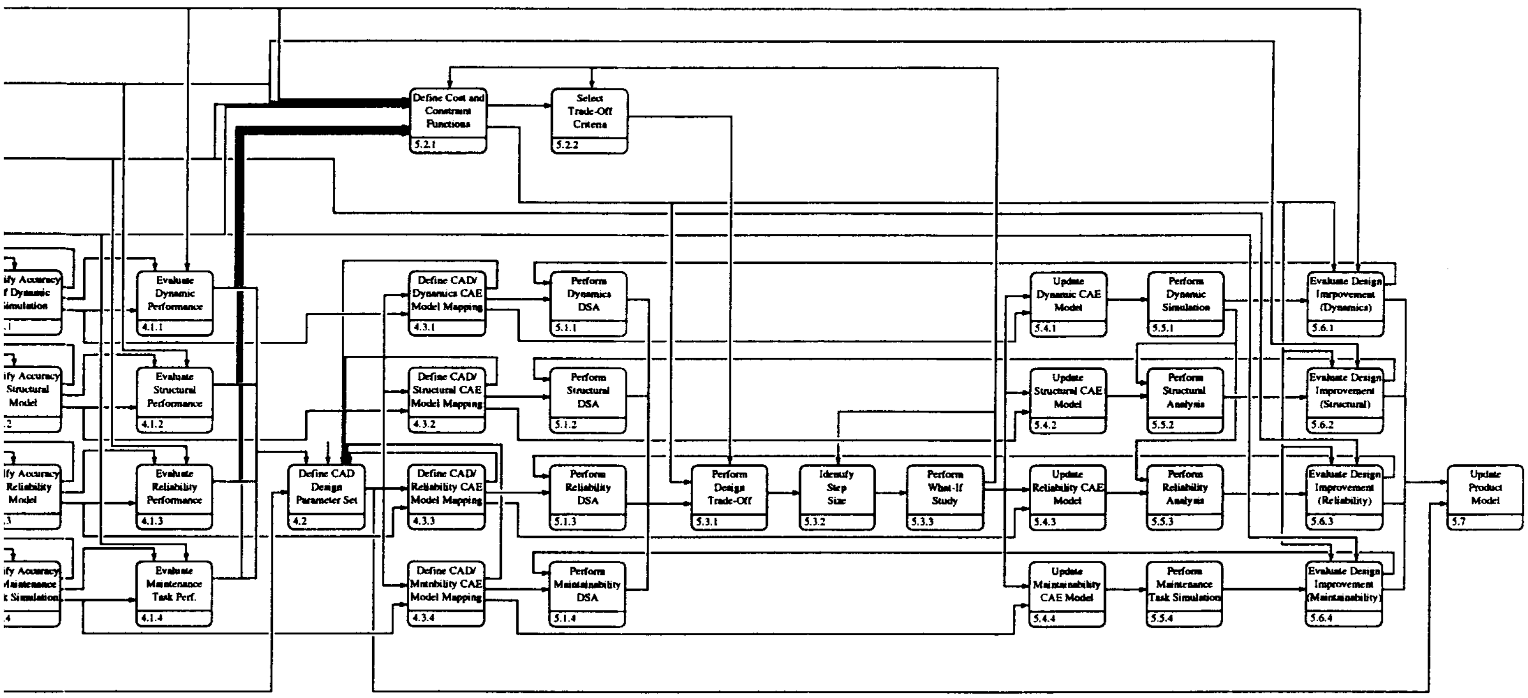


Figure D-3 Level 4 of the Design Evaluation and (

2



Evaluation and Optimization Process

Appendix E: Design Process Management Software Tool Requirements

Table E-1 Design Process Management Software Tool Requirements

Activity	Tool Description	Requirement
Define Areas of Concern Determine Project Team Assignments	Development Team Organization Modeler	• Be capable of graphically modeling the design development team organizational structure. • Be capable of specifying design development team personnel assignments in the distributed, networked computer environment. • Be capable of establishing network communications links with development team personnel. • Operate in a UNIX environment.
Define High Level Process Plan Define Detailed Process Plan	Design Development Process Modeler	• Be capable of graphically modeling process activities, relationships between activities, and information flow in the IDEF standard format. • Be capable of modeling the iterative design development/analysis/change cycles charac-teristic of the Concurrent Engineering paradigm. • Allow modifications to process models. • Be capable of providing design process model data (activity names, numbers, transitions, duration) sufficient for GT process analysis algorithms. • Be capable of providing design process model data (activity duration, number of cycle iterations, resources, schedule dates) sufficient for project status representations. • Be capable of representing process activity personnel assignments. • Be capable of providing design process model data (activity names, numbers, and transitions) sufficient for communications framework establishment. • Operate in a UNIX environment.
Analyze Process Plan for Concurrency Explore Process Alternatives	Group Technology Analysis	• Be capable of consistent, reproducible analysis to determine degree of concurrency in the design process. • Be capable of identifying process activities and activity relationships that are potential bottlenecks to concurrency. • Be capable of permitting modifications to process plan characteristics, i.e., activities and relationships (activity addition/deletion, transition addition/deletion), for rapid re-analysis to explore process options to enhance concurrency. • Operate in a UNIX environment.
Refine Process Plan	Design Development Process Modeler	• Permit modifications to defined process models.
Establish Process Framework for Communications	Communications Framework Modeler	• Be capable of relating process activity personnel assignments with personnel locations in the distributed network environment. • Permit graphical representation of communications framework according to process plan. • Operate in a UNIX environment.
Disseminate Communication Framework	Communication Utility Workspace Wrappers	• Permit ICEE users to access graphical representation of communication framework to establish communications links.

Table E-1 (Con.) Design Process Management Software Tool Requirements

Activity	Tool Description	Requirement
Establish Project Status Chart Disseminate Project Status Chart	Project Status Utility	• Be capable of importing process data (activity names, duration, schedule dates) sufficient to establish GANTT project status chart rep-resenting duration and schedule for all activities in the process plan. • Be capable of graphically representing changes in individual activity initiation, termination, and degree of completion. • Be capable of reading input data and automatically updating activity status for changes in activity initiation, termination, and degree of completion. • Be capable of representing iterative process cycles in schedule form. • Operate in a UNIX environment.
Update Project Status Chart	Workspace Wrapper Utility	• Permit workspace users to input changes in activity initiation, termination, and degree of completion.
Assess Project Status	Project Status Utility	• Be capable of displaying GANTT project status chart representing duration and schedule for all activities in the process plan. • Be capable of graphically representing changes in individual activity initiation, termination, and degree of completion.
Modify Process Plan	Design Development Process Modeler	• Permit modifications to defined process models

Description	Requirement
User Interface	• Employ graphical user interface to launch development team organization modeler, process modeler, process analysis, communication framework, and project status utilities. • Employ listings of personnel, activity assignments, and activity data as appropriate to promote ease of operation of workspace utilities. • Employ appropriate import/export functions for access to local database and communications board in support of process archiving, and communications framework/ project status dissemination. • Employ graphical representation of process management methodology to assist in utilization of workspace functionalities. • Operate in a UNIX environment.

Appendix F: Communications Utility Requirements

Table F-1 Communication Utility Software Requirements

Activity	Requirement
Access Communication Utility	• Employ graphical user interface to access Communication Utility through CAE workspace wrappers. • Incorporate Communication Utility as an integral element of the ICEE. • Provide appropriate Communication Utility user interface menu to launch/enable all Commmunication Utility functionalities.
Employ Organization and Communication Framework Displays to Establish Network Links	• Be capable of displaying process based communication framework. • Be capable of displaying development team organizational diagram. • Utilize communication framework display as interactive user interface for establishment of communication links. • Utilize development team organizational diagram as interactive user interface for establishment of communication links. • Permit user to select activities displayed in process communication framework as transmission/reception points. • Automatically confirm user selection of transmission point of origin corresponds to user location. • Permit user to select team members represented in organizational diagram as transmission/reception points. • Provide user interface function to enable communication link between transmission and reception points. • Employ existing communication channel used in ICEE integration architecture to support user communications.
Create Text and Import Graphics	• Provide user with a capability to create and edit text. • Be capable of importing and pasting graphical design representations into text document. • Provide graphics editing capability to identify or otherwise highlight areas of interest. • Automatically create document header including To, From, To Network Address, From Network Address, Date, Time, and Subject. • Automatically retrieve To, From, To Network Address, From Network Address, information from communication link transmission/recep-tion points selected from communication framework or organizational diagram. • Automatically retrieve Date and Time infor-mation from system server. • Permit user to manually input subject des-cription for text document
Transmit Text and Graphics	• Automatically copy message to user local data-base. • Automatically copy message to CE environment global database. • Provide capability for user to view list of user generated messages copied to local and global databases. • Provide capability to view user generated messages copied to local and global databases. • Provide capability for user to delete user generated messages from local database.
Receive Text and Graphics	• Provide audible and graphical notification of incoming messages. • Copy incoming messages to user local database. • Provide capability for user to view incoming messages. • Provide capability for user to view list of received messages copied to local and global databases. • Provide capability for user to view messages copied to local and global databases.

Table F-1 (Con.) Communication Utility Software Requirements

Activity	Requirement
Receive Text and Graphics (Con.)	• Provide capability for user to delete received messages from local database. • Provide functionality for user to reply to received messages.
Update Project Status in Communication Framework	• Provide color coding for representing activity status (not started, ongoing, completed) in communication framework display. • Change activity color in communication framework display to represent ongoing activity status upon CAE workspace user input of actual start date when corresponding to current date. • Change activity color in communication framework display to represent completed activity status upon CAE workspace user input of 100% activity completion.

①

Appendix G: HMMWV Application Process Model

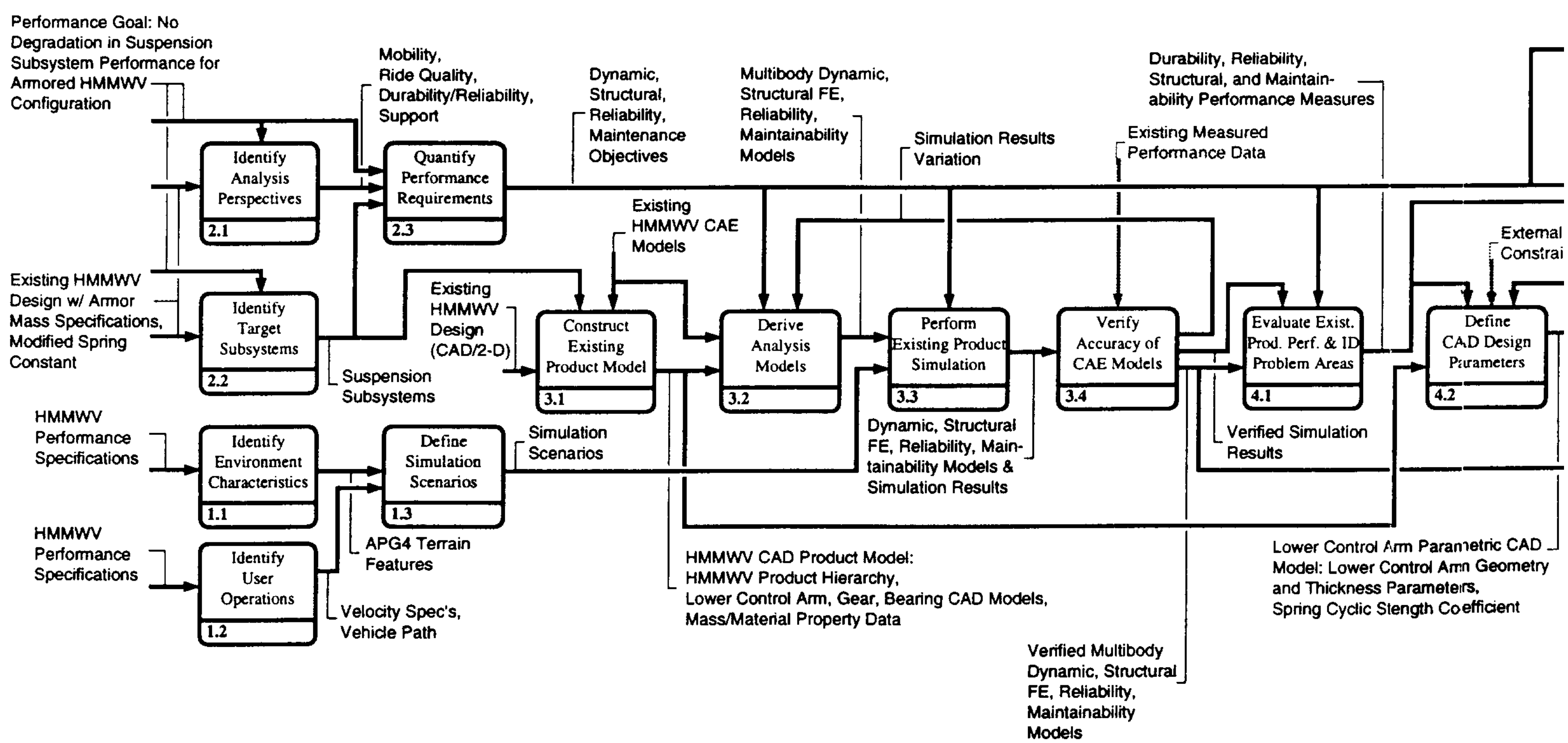
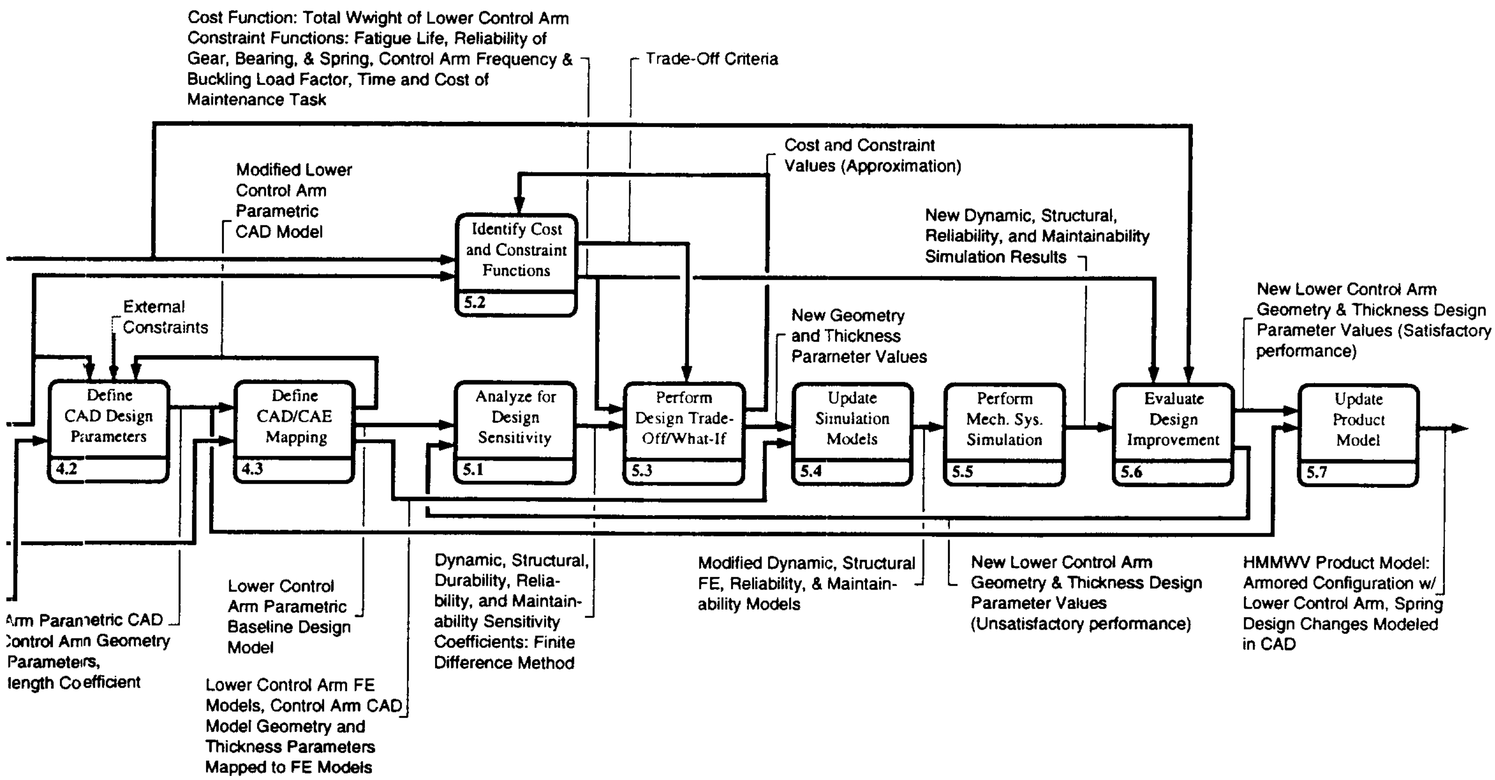


Figure G-1 HMMWV Application Proces



plication Process Model (Level 2)

1

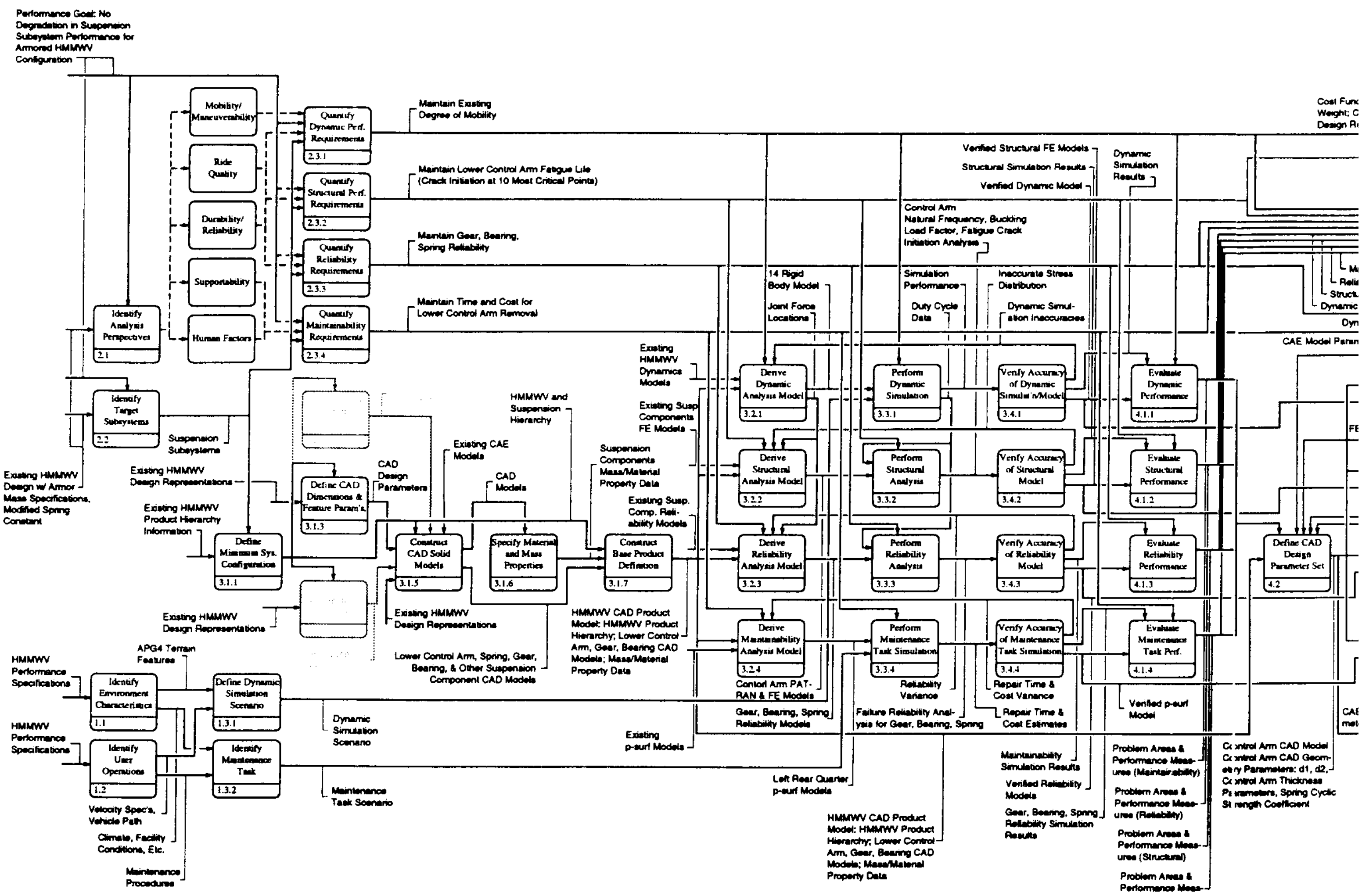
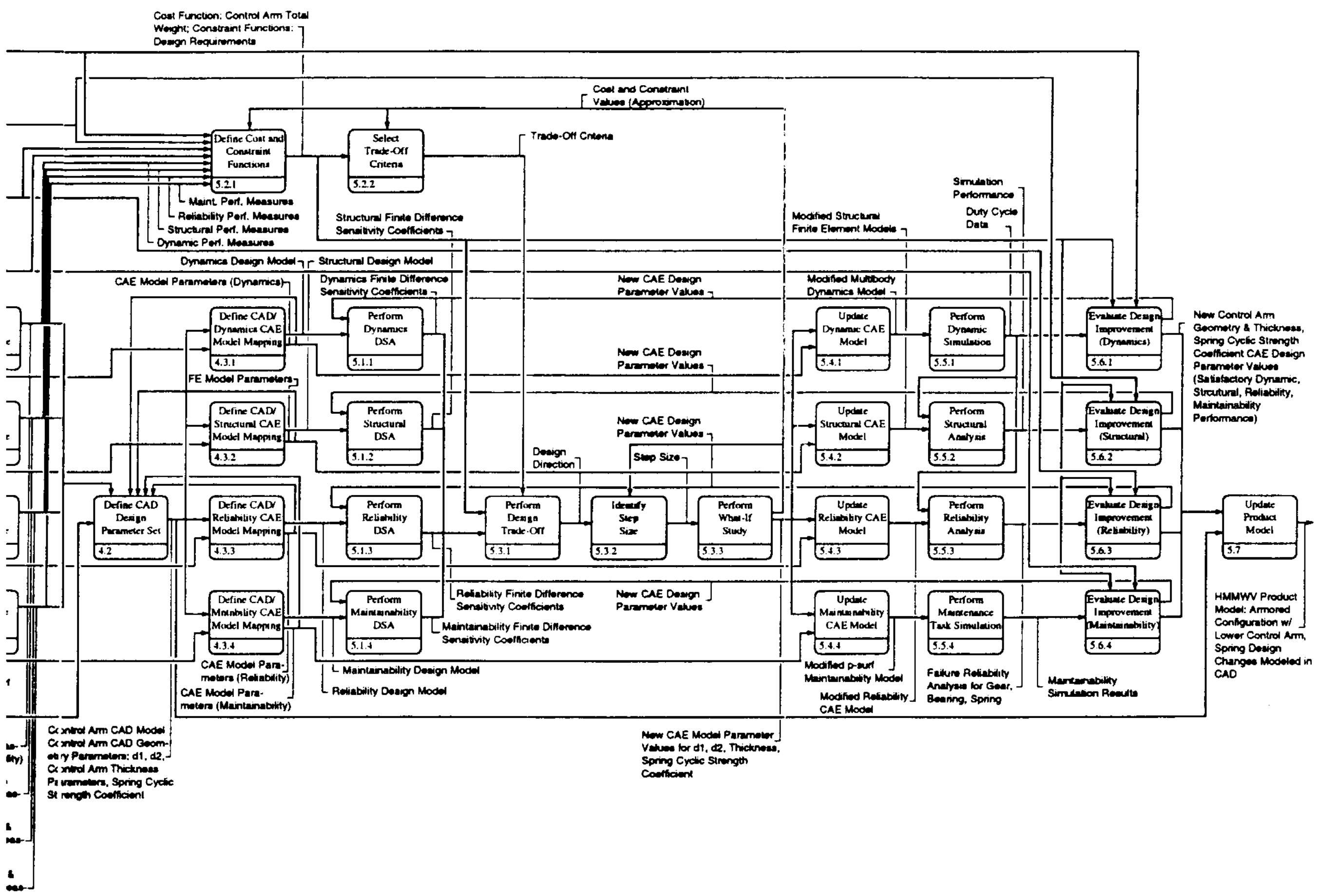


Figure G-2 HMMWV Application Process Mod



ition Process Model (Level 3)