NISTIR 6996

A Survey of Design–Analysis Integration Issues

Gregory M. Mocko
Steven J. Fenves

NIST
National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

NISTIR 6996

A Survey of Design – Analysis Integration Issues

Gregory M. Mocko[1]

Steven J. Fenves[2]

May 2003

U.S. Department of Commerce
Donald L. Evans, Secretary

Technology Administration
Phillip J. Bond, Under Secretary for Technology

National Institute of Standards and Technology
Arden L. Bement, Jr., Director

---

[1] NIST Gaithersburg MD 20899

[2] NIST (Guest Researcher) Gaithersburg MD 20899 and Carnegie Mellon University Pittsburgh PA 15213

Table of Contents

Table of Contents i

List of Figures ii

List of Tables iii

Abstract 1

Keywords 1

1 Design Process Issues 2

  1.1 Design Process Overview 2

  1.2 The Role of Design and Analysis Models 5

2 Information/Data Issues 5

  2.1 Information Flow in the Design Process 6

  2.2 Modeling the Flow of Information in the Design Process 6

  2.3 Product Data Structures for the Support of Design and Analysis 7

3 Capturing Product Data during the Design Process 9

  3.1 Activities and Tasks in the Pahl and Beitz Design Process 9

  3.2 Deliverables in the Pahl and Beitz Design Process 13

  3.3 Information Containers in the Core Product Model 15

  3.4 Mapping Design Process Information to the Core Product Model 17

4 Bridging the Gap between Engineering Design and Analysis 18

  4.1 Object-Oriented Modeling 19

  4.2 Computer Aided Design and Finite Element Analysis Integration 27

  4.3 Multi-Aspect Information Structures 38

5 Summary 42

6 Areas for Future Work 42

7 References 43

List of Figures

Figure **Page**

Figure 1. Steps in the Planning and Design Process (Pahl & Beitz, 1996)........................ 3
Figure 2. Design Process Roadmap (Recreated from Tate and Nordlund, 1996).............. 4
Figure 3. Design Information Flow Model (Shooter et al., 2000). 7
Figure 4. Steps in Product Planning (Pahl & Beitz, 1997). 10
Figure 5. Steps in Conceptual Design (Pahl & Beitz, 1997). 11
Figure 6. Steps in Embodiment Design (Pahl & Beitz, 1997). 12
Figure 7. Steps in Detailed Design (Pahl & Beitz, 1997). 13
Figure 8. Core Product Model Class Diagram (Fenves, 2001b). 17
Figure 9. Interactions of Enabling Technologies 19
Figure 10. Model of an Engineering System - Port-Based Modeling (Diaz-Calderon et al., 2000). 21
Figure 11. A Reconfigurable System Model (Diaz-Calderon et al., 2000). 22
Figure 12. The Multi-Representation Architecture (Peak, et al., 1994). 23
Figure 13. Categories of General Purpose ABBs (Peak, et al., 1994). 24
Figure 14. Activity Chart, Decomposition of Design Verification (Andersson, 1999). ... 26
Figure 15. Dimensional Reductions (Armstrong, 1994). 28
Figure 16. A 2D Region, its Medial Axis and an Equivalent Beam Model (Armstrong, 1994) 28
Figure 17. Merging of Faces and Edges (Armstrong et al., 1996). 29
Figure 18. Morphological Analysis Tool Components (Belaziz, 2000). 31
Figure 19. A Systems Model and its Relationships (Sellgren and Drogou, 1998). 32
Figure 20. Behavior Feature and Mating Face Associated to Design Shape (Sellgren and Drogou, 1998). 33
Figure 21. Implicit Master-Slave Selection (Sellgren and Drogou, 1998). 33
Figure 22. Thermal Interface Couplings at Different Levels of Abstraction (Sellgren and Drogou, 1998). 33
Figure 23. Analysis Geometry same as Design Geometry (Gordon, 2001). 35
Figure 24. Analysis Geometry Different from Design Geometry (Gordon, 2001). 35
Figure 25. Simulation Specific Geometry Drives Both Simulation And Design (Gordon, 2001). 36
Figure 26. Design–Analysis Shape Association (recreated from Hunten, 2000a). 37
Figure 27. Architecture of EACM (Leal, 1999). 38
Figure 28. Master Model Architecture with Client Views (Hoffman and Joan-Arinyo, 1998). 39

List of Tables

Table **Page**
Table 1. Deliverables in the Pahl and Beitz Design Process 14
Table 2. Nature of the Deliverables Generated in the Design Process 15
Table 3. Correlation of Pahl and Beitz Deliverables to the Core Product Model............ 18

Abstract

The report constitutes a survey of work directed towards the emergence of next-generation product development tools, specifically in the area of design-analysis integration. The report is organized into two main areas: (1) an examination of issues pertaining to design processes and information generation and capture; and (2) a literature survey of current design-analysis integration efforts.

First, engineering design is presented and discussed in terms of the product data that are generated during the design process. Based on analyses of the Pahl and Beitz systematic design process and the National Institute of Standards and Technology (NIST) Core Product Model (CPM), a correspondence between deliverables in the design process and the information that can be captured in the CPM is generated. The primary objectives of this task are to identify the basic information-generating activities in the design processes and to verify the comprehensiveness and completeness of the CPM in its ability to support the storage and retrieval of the information generated.

The second area covered is a comprehensive literature survey of current design-analysis integration research thrusts and efforts. It was found convenient to classify current research into three general categories: (1) object-oriented modeling paradigms of mechanical systems; (2) efforts in the area of Computer-Aided Design (CAD) and Computer-Aided Engineering (CAE) integration; and (3) multi-aspect information structures.

The literature survey suggests that there is a strong need for a common vocabulary, framework, and roadmap for the improved integration of design and analysis tools to be used in next-generation product development systems. By developing a standard vocabulary and framework, synergies in design-analysis integration and product modeling can be leveraged to decrease the gaps and increase product development efficiency.

Keywords

product modeling, information modeling, data modeling, technical artifacts, Pahl and Beitz, design process, hierarchical modeling, object-oriented modeling, CAD, CAE, FEA, interoperability, simulation

1 Design Process Issues

In order to set the context for engineering design-analysis integration, two models of the product design process are first presented. The Pahl and Beitz (Pahl & Beitz, 1996) systematic design process and the design process roadmap developed by Tate and Nordlund (Tate and Nordlund, 1996) describe the basic process for designing technical artifacts. Additionally, the synthesis and analysis tasks associated with engineering design are presented

1.1 Design Process Overview

The Pahl and Beitz design process, accepted by many engineers and educators as *the* process for engineering design, is a phase-based process that progresses from the abstract (qualitative) to the concrete (quantitative) through a series of analysis and synthesis tasks. The phases in the Pahl and Beitz process are Planning and Clarification of Task, Conceptual Design, Embodiment Design, and Detailed Design (see Figure 1).

The analysis and synthesis tasks, to be discussed later, may be iterated until a satisfactory result is achieved and a decision is made to progress to the next design phase. Iteration in the design process is almost always required and occurs continuously within the steps and between the steps because of the coupling of design information. These iteration steps should be kept as small as possible to lessen the risk of oversights and mistakes in the development process (Pahl & Beitz, 1996).

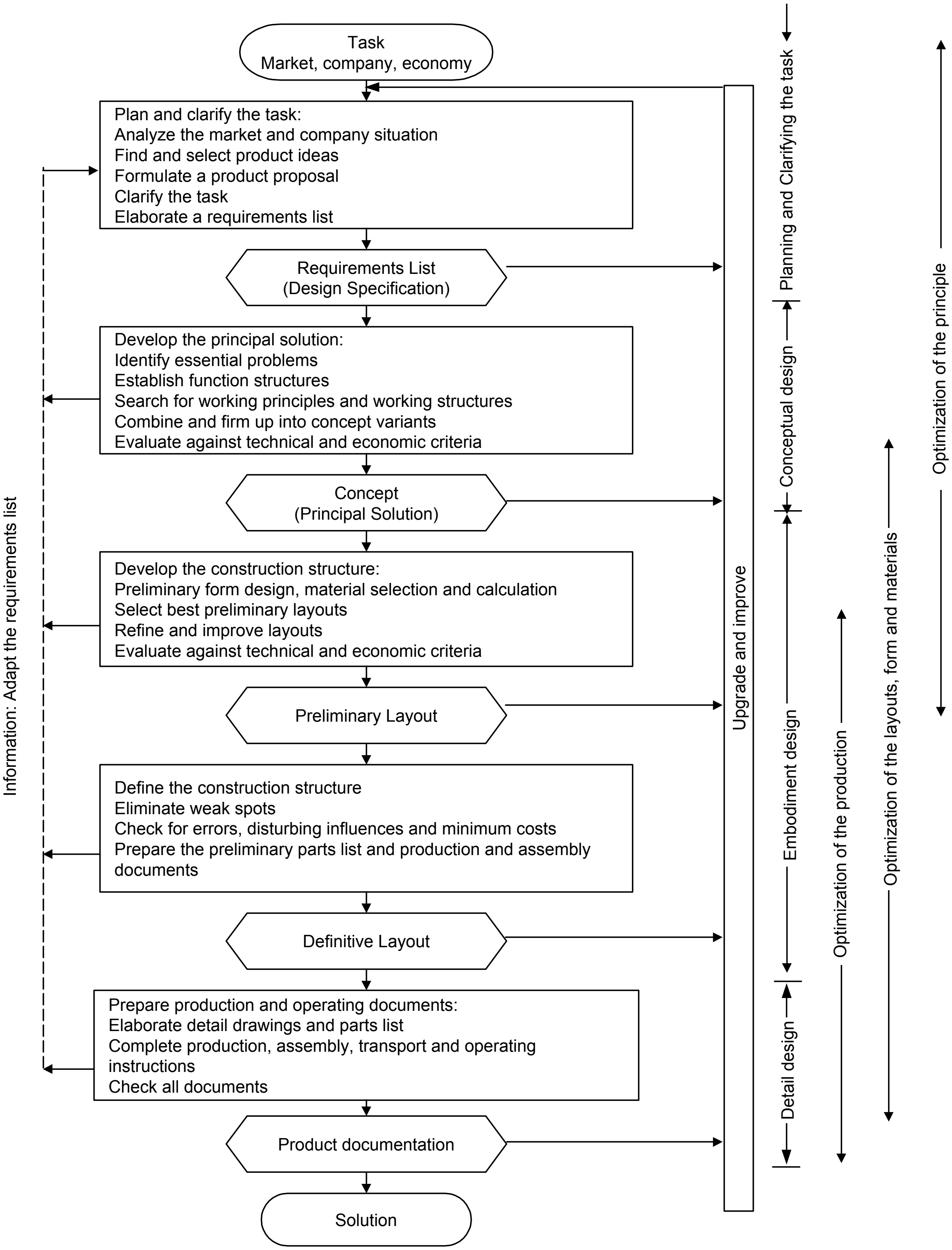


Figure 1. Steps in the Planning and Design Process (Pahl & Beitz, 1996).

The second design process model, developed by Tate and Nordlund (Tate & Nordlund, 1996), was motivated by the authors' perception that current design process models, such as the Pahl and Beitz model, were unable to accurately represent "real-world" design processes. The design roadmap combines the strength of both phase-based and activity-based models (see Figure 2).

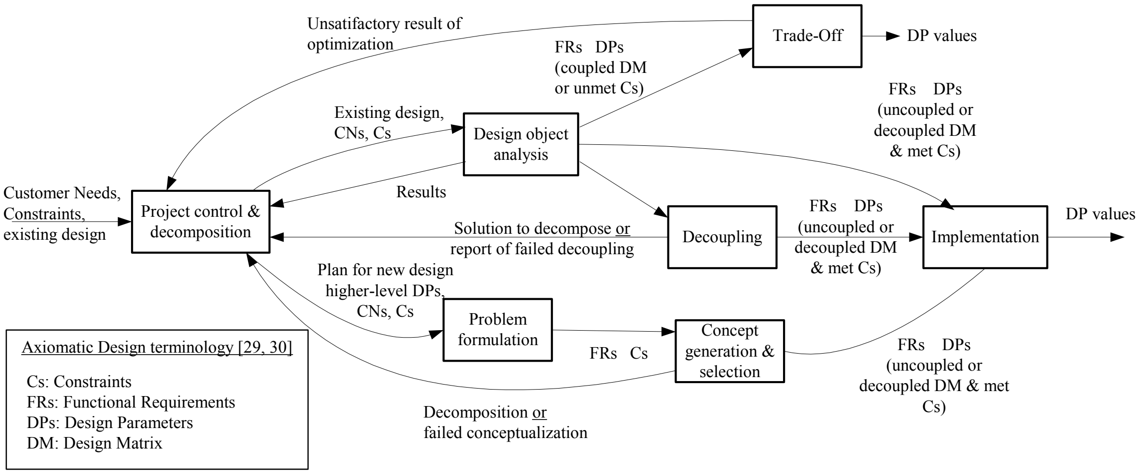


Figure 2. Design Process Roadmap (Recreated from Tate and Nordlund, 1996).

The resulting design roadmap is a collection of distinct activities with clear starting and ending points. The activities may be sequenced in many different ways, depending on the scope and goal of the project.

The design activities proposed by Tate and Nordlund are:

- *Project Control and Decomposition*
- *Analysis of Existing Solutions*
- *Problem Formulation*
- *Decoupling*
- *Concept Generation and Selection*
- *Tradeoff*
- *Implementation*

Design synthesis and analysis tools support the *Concept Generation and Selection* activity.

Both processes presented follow a sequence of activities, in which a designer, or team of designers, systematically transform an abstract problem to a concrete design solution (Sellgren, 2001). Synthesis and analysis are two of the three major contributing activities in the design process. Evaluation is the third of the activities. According to Ullman, evaluation involves two activities, comparison and decision making. Alternative concepts, generated during synthesis, are compared and chosen, based on analysis results, in accordance with design requirements.

1.1.1 Synthesis in the Design Process

Typically, engineering synthesis is performed with the aid of formal ideation techniques such as brainstorming, literature search, and analysis of existing technical systems. Computer Aided Design (CAD), most commonly used for design synthesis in the embodiment phase of design, enables the realization of shape, structure, and form of solution concepts in a computer-based environment. CAD applications support interactive computer-based graphics for the creation, modification, and visualization of engineering artifacts.

1.1.2 Analysis in the Design Process

Engineering analysis deals with understanding the design problem and verifying that the design fulfills the requirements and constraints. Engineering analysis in many functional domains is typically accomplished with the aid of formal analysis tools and methodologies. Analysis, like synthesis, is iterated at various levels of detail at the different phases in the design process.

Computer-Aided Engineering (CAE) is an often-used term to describe all computer-based engineering analysis tools and methodologies (Sellgren, 2001). Such tools include computational fluid dynamics (CFD), finite element analysis (FEA) and factory simulation, to name a few.

1.2 The Role of Design and Analysis Models

In the design processes currently used by a large number of enterprises, product form is first determined in full detail with the aid of CAD tools. The geometric form descriptions are then used to drive engineering analysis tools to validate the design against functional criteria to predict the physical behavior of the product (Tamburini, 1999). If it is determined that the artifact's observed behavior varies significantly from the desired behavior, the geometric form of the artifact is modified and the analysis models are recreated.

The generation of the analysis model is a combination of creativity and scientific reasoning that requires experience and insight in the assumptions, limitations, and applicability of the tools. Much of the time and effort of creating analysis models is a result of not using the information from design models or past analysis models. The process of creating an analysis model may result for 80% of the total analysis time (Hsiung, 1998).

Although the design and analysis models are views of the same product, the semantic content of the models varies significantly. Additionally, the relationship between the design model and the analysis model is a one-to-many relationship. For these reasons design-analysis integration is often difficult to accomplish (Tamburini, 1999).

2 Information/Data Issues

In the following sections issues associated with the capturing, sharing and accessing of product data in the design process are discussed. The multi-location nature of current product development introduces obstacles into the efficient sharing of information generated in the course of product design. Next, a research effort to model the generation of information in the design process, so as to serve as a basis for the development of data structures to support distributed collaborative design, is discussed. Finally, a discussion of formal product data structures to better support product development in the future is presented. Two research projects that introduce formal structures for capturing the information created during the design process are presented.

2.1 Information Flow in the Design Process

In the past, product development was often completed by designers or teams of designers at a single company. This enabled the sharing of information and collaboration. Currently, the design of complex engineering systems is increasingly becoming a set of collaborative tasks among designers or teams of designers that are physically, geographically, and temporally distributed (Szykman, et al., 1998).

The advantage of a distributed, collaborative product development process is that of leveraging the expertise of other design firms or companies. The disadvantage is the burden of sharing and exchanging product information. Additionally, product information is not always created on the same software platform, and must be exchanged across heterogeneous systems (Szykman et al., 2002).

The current trend in product development is pushing the envelope of available technology for information management. A product model does not currently exist that captures all aspects of the product development process so as to support the seamless sharing of information. Many of the gaps in design-analysis integration are caused by incomplete translations of product information from the design domain to the analysis domain and vice versa. The information lost during the translation process is a key motivation for developing more robust product data structures. To recreate lost or unused data resulting from the translation process requires a substantial manual effort and involves large expenses. A recent study for NIST reports that the U.S. automotive industry spends billions of dollars annually as a result of inefficient interoperability, of which the lack of efficient design-analysis integration forms a substantial part (Brunnermeier and Martin, 1999).

2.2 Modeling the Flow of Information in the Design Process

A recent effort at NIST has modeled the flow of information in the product development process. A summary of the model for the flow of design information developed by Shooter, et al (Shooter et al., 2000) is presented. The model characterizes the flow of information independent of any particular design process (Shooter et al., 2000). The general model of the design process followed in the research is presented in Figure 3.

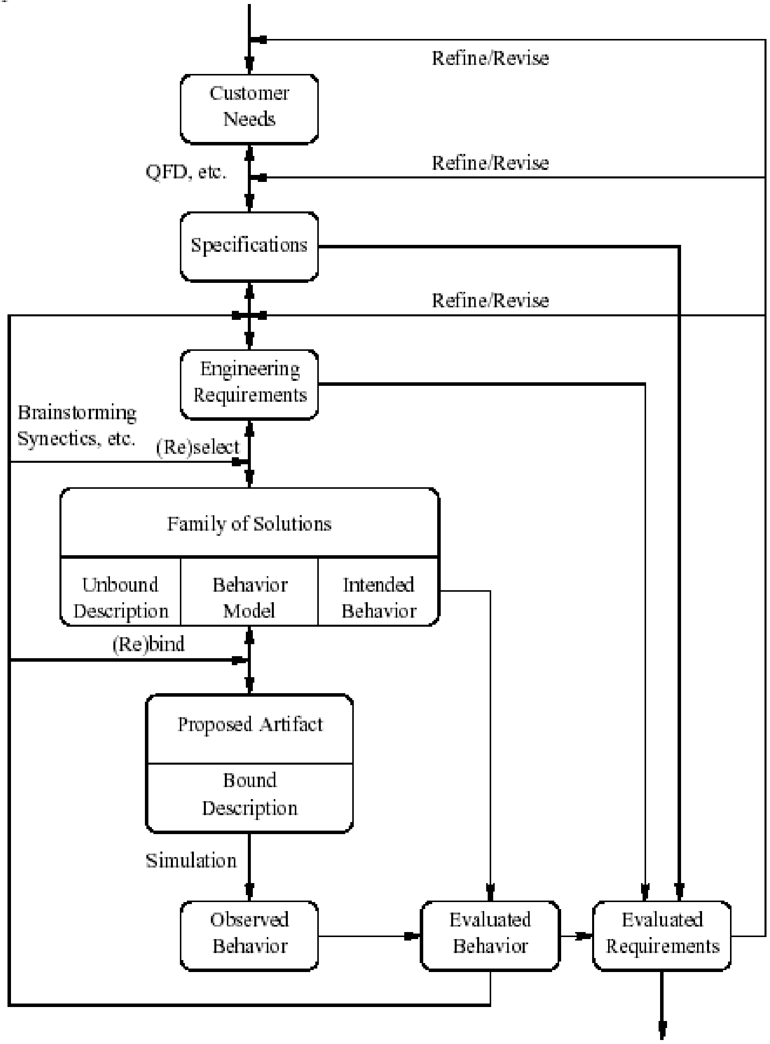


Figure 3. Design Information Flow Model (Shooter et al., 2000).

The information first captured are the *Customer Needs,* that are then formalized into *Specifications*. The *Specifications* are further formalized into *Engineering Requirements*. A *Family of Solutions* based on the engineering requirements provides a partial description of the proposed design. Finally, a *Proposed Artifact* is created when enough information is obtained at the desired level of abstraction.

Next, the *Observed Behavior* includes information about the behavior of the artifact, that is evaluated and then compared to the *Intended Behavior*. Finally, the *Evaluated Requirements* provide the basis for making the decision whether the artifact must be refined/revised or whether the development process can continue at a more detailed level of abstraction.

2.3 Product Data Structures for the Support of Design and Analysis

Software tools that effectively support the formal representation, capture and exchange of product information are vital to an efficient and effective product development process.

The lack of technologies for sharing product information creates a major barrier in the design process (Szykman et al, 2002).

Engineering systems that support seamless interoperability allow the sharing of information generated and used by various product development activities. (Shooter et al., 2000) develop representations of information that are currently unavailable in traditional CAD/CAE and Computer Aided Manufacturing (CAM) systems to support the exchange of information in a distributed product development environment.

The exchange of engineering information over the full design life of the product is further supported by the research of Fenves and Szykman, et al. through the development of product information representation schemes (Fenves, 2001b), (Szykman et al., 1998). These technologies are intended to serve as the basis for next-generation computer-based engineering tools and allow information to be shared throughout the design process. Both the Core Product Model presented by Fenves and the NIST Design Repository by Szykman are beginning efforts towards the development of a representation that will enable engineering interoperability issues to be eliminated, or at least decreased, in the future.

The NIST Design Repository Project

Szykman, et al. view design repositories as a natural progression from traditional engineering databases to capture evolutionary design information in the product development process (Szykman et al., 1998). Design repositories differ from traditional databases, in that databases are archives of completed designs, analogous to design catalogs. Design repositories, on the other hand, capture the evolutionary information generated during the design process.

The research intent of the NIST design repository project is to develop a framework for the support of the implementation and use of design repositories. The research is driven by the needs to support knowledge-based design by sharing, capturing, and reusing design information.

Core Product Model for Representing Design Information

Fenves (Fenves, 2001b), developed the Core Product Model (CPM) for representing design information over the design cycle of technical artifacts. The core model was developed based on the synthesis of several independent projects. The objective is to provide a base-level product model that is independent of any specific software vendor, is simple, open, and expandable to account for a wide variety of products and processes. The core model must be able to capture an extensive amount of information for different products at different phases in the design process. The aim of the model it to serve as a means of information exchange in the early, conceptual stages of design, before the information can be exchanged using the STEP (Standard for the Exchange of Product Model Data) standard.

3 Capturing Product Data during the Design Process

Because of the preeminence of the Pahl and Beitz systematic design process model (Pahl & Beitz, 1996) and the dependence of the current design-analysis integration research activities at NIST on the CPM (Fenves, 2001b), these two models are examined in further detail and a mapping is presented between the information generated in the phases and steps of Pahl and Beitz process and the classes in the CPM. The objective of this mapping is to ascertain whether the CPM is capable of supporting all the types of information generated in the design process.

3.1 Activities and Tasks in the Pahl and Beitz Design Process

A detailed examination of the Pahl and Beitz reveals the specific tasks, activities, and deliverables associated with each phase towards the development of a technical product. The Pahl and Beitz diagram, as presented in Figure 1, comprises specific tasks and associated deliverables. The steps associated with each phase of the design process are presented in detail in the following sections.

Product Planning

Long before a product can be designed there has to be a concept for it. The search for product concepts occurs in the product planning phase. Product planning is the systematic search for and selection and development of a promising product concept. The steps of the Product Planning phase are shown in Figure 4.

Many of the activities, tasks, and information generated during the Product Planning phase are considered to be out of the scope of engineering design. For this study, we are only interested in the product proposal as a deliverable from this phase. The product proposal, often referred to as product definition, serves as the starting point of the design process. The product definition describes the intended functions, contains preliminary requirements (customer needs), and provides an indication of the cost target.

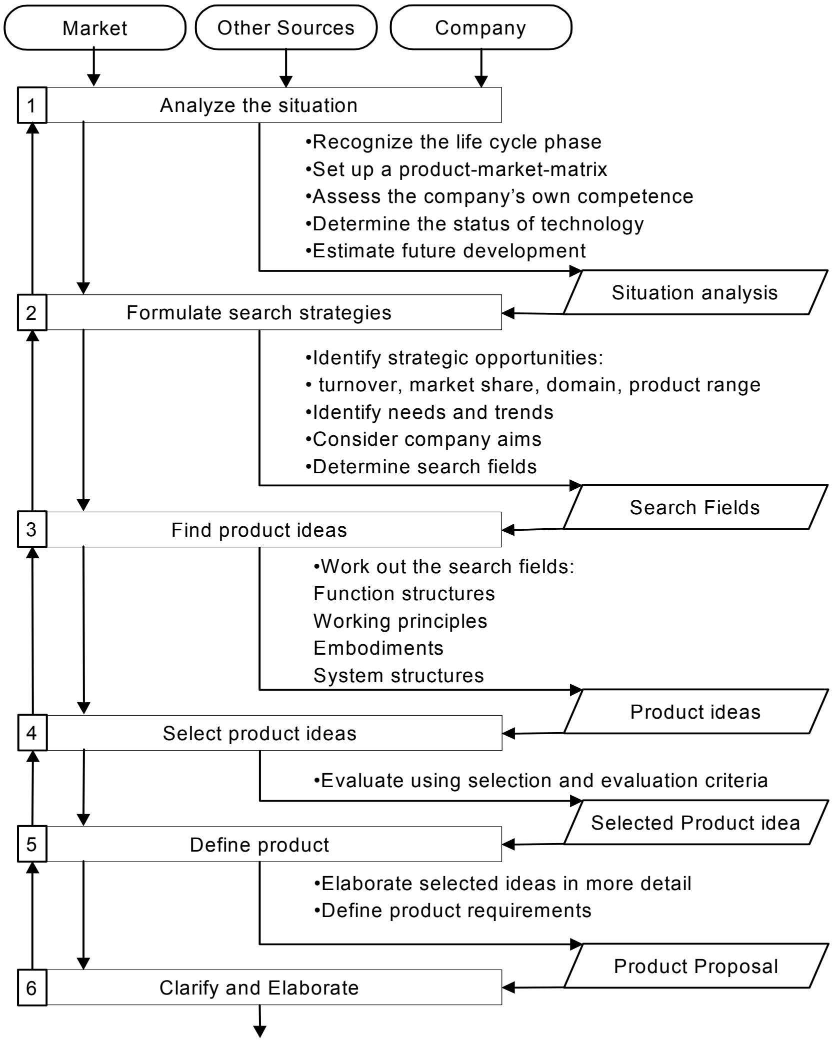


Figure 4. Steps in Product Planning (Pahl & Beitz, 1997).

Clarification of Task

The Clarification of Task phase develops a basic understanding of the problem. Information is collected about the requirements and constraints that must be fulfilled by the artifact. This is accomplished through background research and previous design experience.

Ultimately, a requirements list for the technical artifact is developed. The requirements list is an important deliverable in the design process because it frames the problem that must be solved. The requirements list is a living document that is modified and refined as more knowledge is gained.

Conceptual Design

In the conceptual design phase, essential problems are identified through abstraction, establishing function structures, and search for working principles and their appropriate combination to meet the functional demands of the requirements list. The steps in the conceptual design phase are shown in Figure 5.

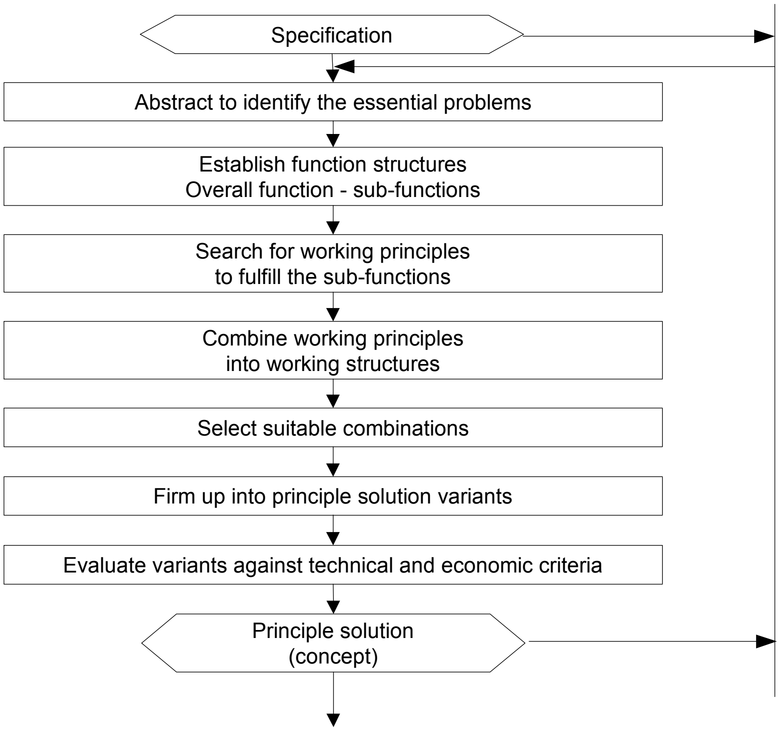


Figure 5. Steps in Conceptual Design (Pahl & Beitz, 1997).

Essential requirements are abstracted from the completed requirements list. Next, the crux of the problem is formulated. The overall function and sub-function hierarchy is determined. Function structures are built based on the flow of material, energy, and signal through and between the system sub-functions.

Based on the function structure, working principles are found for each of the sub-functions. Working principles are then combined together into working structures. Working structures are design independent. Finally, working structures are firmed up into concept variants. Before proceeding to the next design phase, the most promising concept variants are chosen based on technical and economic criteria.

Embodiment Design

In the embodiment phase, designers begin to firm up the concepts generated in the conceptual design phase. This is often achieved through the creation of scale drawings. Normally, several layouts are generated. These alternative layouts are evaluated against technical and economic criteria. A preliminary layout is chosen and is further embodied in the detail design phase. The definitive layout, developed in this phase, provides a check of the function, behavior, and spatial constraints prior to starting the detail design phase (see Figure 6)

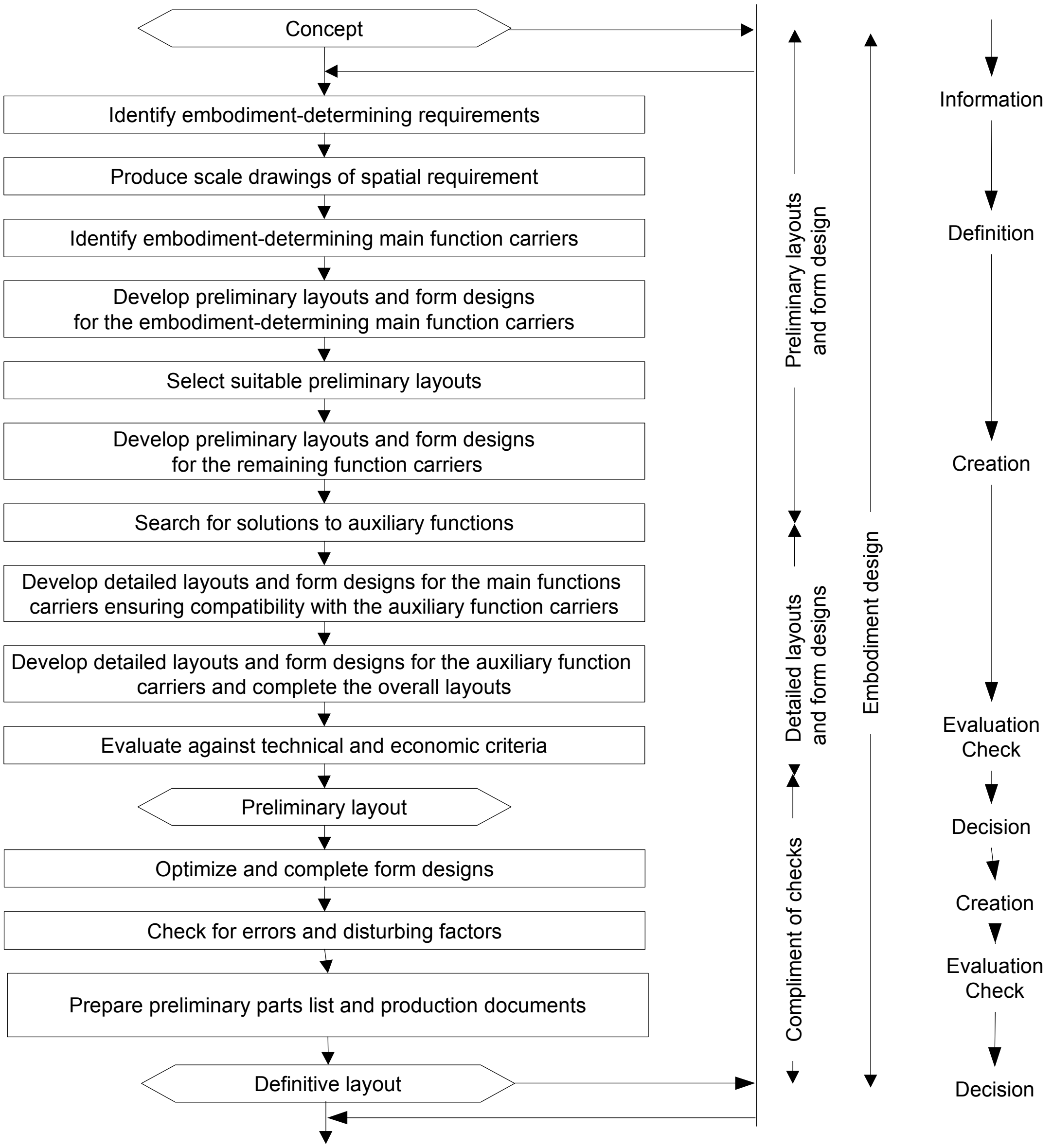


Figure 6. Steps in Embodiment Design (Pahl & Beitz, 1997).

The embodiment phase involves a large number of iterations and corrective steps. Design synthesis and analysis alternate and complement each other, requiring modification and refinement of the design.

Detailed Design

In the detailed design phase, the final arrangement, form, dimensions, and surface properties of all parts in the design are determined. The detailed design phase involves finalizing the definitive layout by completing the detailed drawings of all components and fasteners. The shape, material and tolerance specifications and cost estimates must be completed for each part. The integration of the individual parts into assemblies must be determined, again with the use of detailed plans and drawings.

Additionally, the documentation of assembly, operation, and transportation must be developed. Lastly, checks of all the plans, processes, and drawings must be completed for all parts and assemblies.

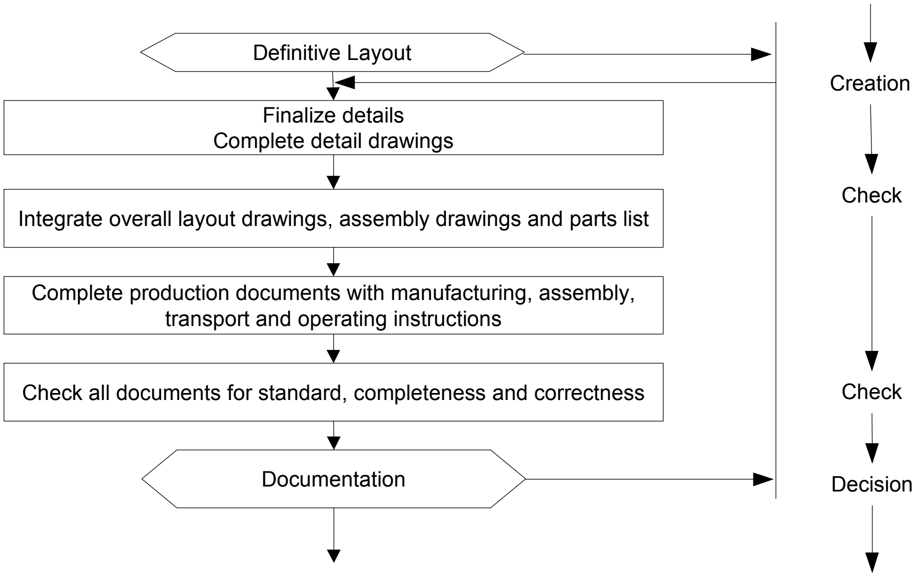


Figure 7. Steps in Detailed Design (Pahl & Beitz, 1997).

3.2 Deliverables in the Pahl and Beitz Design Process

Based on the detailed descriptions of the steps in each design phase, the following deliverables are identified. A deliverable is defined as knowledge, data, or information that is generated over the course of the product development process. Deliverables can be in the form of electronic or physical documentation or intermediate documentation that supports the final design of the technical artifact. The deliverables are classified into the following categories:

- Key Deliverable - the result of a specific design phase. Keys deliverables are the culmination of tasks that are passed on to subsequent design phases. Key deliverables are milestones in the design process and used in downstream processes.
- Intermediate Deliverable - used in the creation of the key deliverables. They can be in the form of hard copy documentation but may or may not be used in subsequent design phases.
- Evolving Design Documentation - confined to one phase of the design process. The evolving design documentation contains a high level of detailed information.

The deliverables generated in the design process are tabulated in Table 1 arranged in the general order in which they are created. The nature of the deliverables is summarized in Table 2.

Table 1. Deliverables in the Pahl and Beitz Design Process

Phase	**Key Deliverables**	**Intermediate Deliverables**	**Evolving Design Documentation**
Product Planning	Product Definition		
Clarification of Task	Requirements list		
Conceptual Design		Abstract requirements list	
		Function structure	
		Morphological matrix	
		Solution evaluation (select concept)	
	Concept variants		
Embodiment Design		General layout requirements	
			Spatial, material, arrangement requirements list
			Scale drawing of spatial constraints
			Preliminary form and arrangement diagram
			Detailed form and arrangement diagram
		Preliminary layouts	
		Solution evaluation (choose layout)	
		Preliminary Diagram	
			Definitive layout (various levels of abstraction)
			Preliminary parts list
			Preliminary production/assembly documentation
	Definitive layout		
Detail Design			Detailed design (various levels of abstraction)
			Detailed component drawings
			Parts list
			Assembly drawings
			Layout drawings
			Transport documentation
			Assembly documentation
			Manufacturing documentation
			Operation documentation
	Design documentation		

Table 2. Nature of the Deliverables Generated in the Design Process

Deliverable	Nature
Product Definition	Sets the context for the product design
Requirements list	Identifies customers want for the product
Abstract requirements list	Identifies the function-specific requirements to form solution-neutral problem statements
Function structure	Identifies the functions and organized the flow of energy, matter and signal
Morphological matrix	Catalogs the working principles and organizes them to meet the functions
Solution evaluation (select concept)	Provides systematic evaluation of concepts
Concept variants	Provides high level embodiment of working structures
General layout requirements	Identifies requirements dealing with the layout of the design
Spatial, material, arrangement requirements list	Forces engineers/designer to focus on essential requirements
Scale drawing of spatial constraints	Provides a realized diagram of spatial constraints that the design must meet, does not require allocation of space to subsystems
Preliminary form and arrangement diagram	Forms the general layout of the product still in a solution neutral format
Detailed form and arrangement diagram	Provides natural progression from preliminary to the more detailed description of layout
Preliminary layouts	Presents preliminary concepts for product layout
Solution evaluation (choose layout)	Provides systematic evaluation, in more detail
Preliminary Diagram	Envisions the layout of the design in with minimal knowledge about details.
Definitive layout (various levels of abstraction)	Presents arrangement of system parts (components, etc.) in systematic progression from abstract to detailed
Preliminary parts list	Provides a list of needed parts for procurement or manufacture
Preliminary production/assembly documentation	Allows production and assembly concerns into the design side of product development – provides room for feedback and refinement of these activities
Definitive layout	Provides final layout of the technical system
Design documentation	Provides detailed information on design process and decisions made
Detailed design (various levels of abstraction)	Enables the product to be manufactured, assembled, , operated, maintained, etc.
Detailed component drawings	
Parts list	
Assembly drawings	
Layout drawings	
Transport documentation	
Assembly documentation	
Manufacturing documentation	
Operation documentation	

3.3 Information Containers in the Core Product Model

The CPM consists of two sets of classes, objects and relationships. The objects store generic types of information, and the relationships describe the association between them. A brief explanation of the classes in the Core Model, including the general representations and the semantics are presented. Finally, the complete class diagram is shown in Figure 8. For additional information about the Core Product Model, see (Fenves, 2001b).

Semantics of the Core Model

Object Classes

Common Core Object - no instances, highest abstract level

Core Entity - abstract class, Artifact and Feature are specializations

Core Property - abstract class, Function, Flow, Geometry, Form and Material are specializations. Constraint and Requirement relationships can be applies to Properties

Artifact – a distinct entity; may be a component, product, or assembly

Feature – a subset of the form of an object, such as a design feature, analysis feature, or interface feature (port)

Specification – container of the customer needs or engineering requirements that the form, function, and geometry must satisfy

Function - represents what the Artifact is supposed to do, its *intended behavior*

Transfer Function - specialized form of function, changes the input flow to an output flow

Flow - the medium (material, energy, signal) being transferred

Behavior – represents how the Artifact's form implements the function, the *observed behavior* based on engineering principles (simulation or analysis)

Form - geometry and material

Geometry - spatial description

Material - material description

Relationship Objects

Common Core Relationship - abstract class

Requirement – links Specification to a specific property of the Artifact

Constraint - shared property that must hold in all cases

Reference - direct linking between two entities

Assembly - relationship between artifacts

Set-Relationship - abstract class, specializes further

Undirected Set-Relationship - simple relationship among entities

Directed Set-Relationship - subsets have two different roles, e. g., "controlled", "controlled-by"

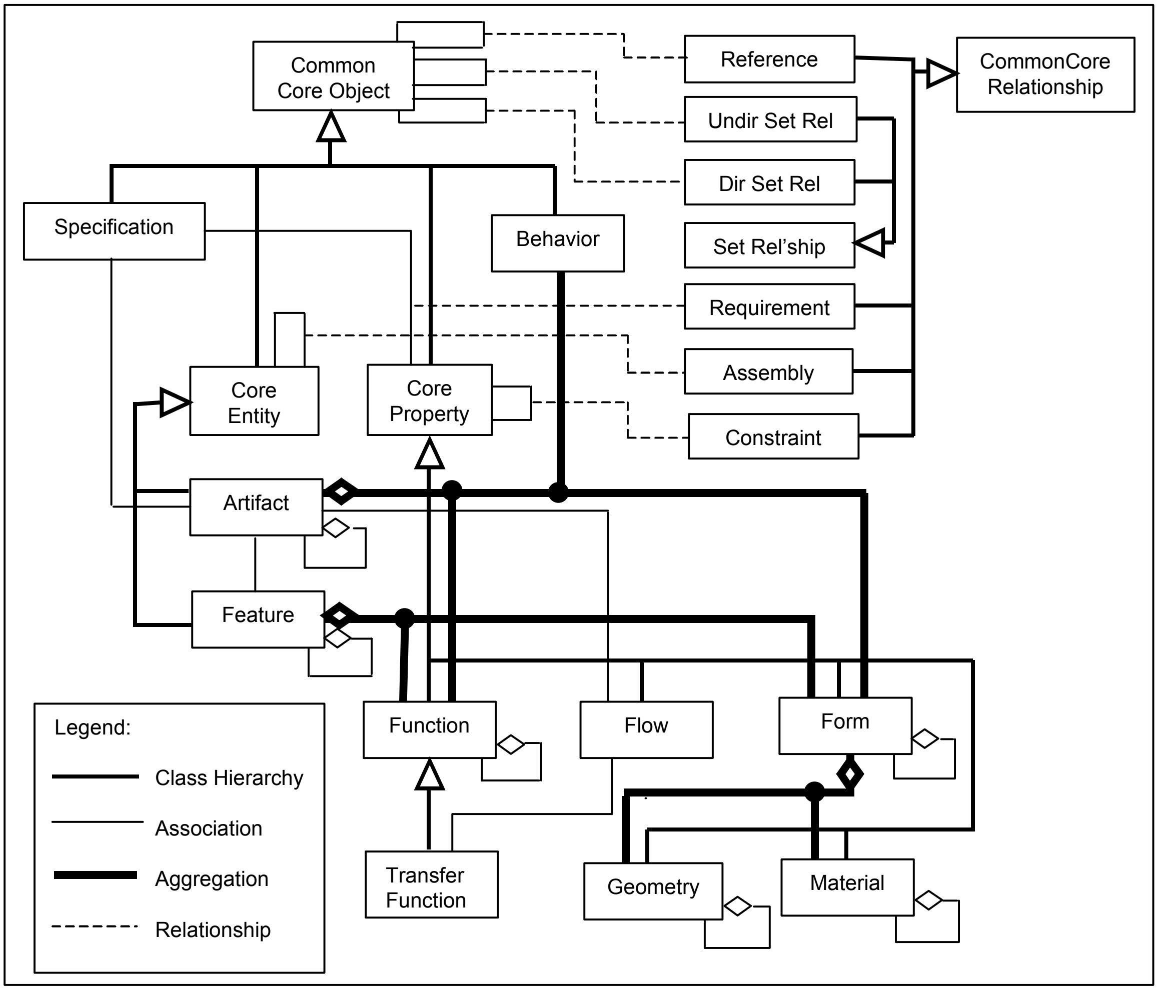


Figure 8. Core Product Model Class Diagram (Fenves, 2001b).

3.4 Mapping Design Process Information to the Core Product Model

The mapping between deliverables in the Pahl and Beitz process and the Core Product Model is presented in Table 3.

Table 3. Correlation of Pahl and Beitz Deliverables to the Core Product Model

Pahl and Beitz Deliverables	Core Product Model
Product Definition	Artifact + Specification
Requirements list	Specification +Requirements
Abstract requirements list	Specification +Requirements
Function structure	Function + Behavior
Morphological matrix	Function + Behavior
Solution evaluation (select concept)	Function +Behavior + Requirements
Concept variants	Form +Function + Behavior +Assembly
General layout requirements	Specifications + Requirements
Spatial, material, arrangement requirements list	Specifications + Requirements
Scale drawing of spatial constraints	Form +Requirements
Preliminary form and arrangement diagram	Form + Requirements +Constraints
Detailed form and arrangement diagram	Form + Requirements +Constraints
Preliminary layouts	Form +Requirements
Solution evaluation (choose layout)	Form + Requirements +Constraints
Preliminary Diagram	Form
Definitive layout (various levels of abstraction)	Form
Preliminary parts list	Artifact
Preliminary production and assembly documentation	Artifact +Assembly
Definitive layout	Form
Detailed design documentation	Artifact +Assembly +Requirements
Detailed design (abstract level 1)	Artifact +Assembly +Requirements
Detailed component drawings	Artifact +Assembly +Requirements
Parts list	Artifact
Assembly drawings	Artifact +Assembly +Requirements
Layout drawings	Artifact +Assembly +Requirements
Transport documentation	Artifact +Requirements
Assembly documentation	Artifact +Assembly +Requirements
Manufacturing documentation	Artifact +Requirements
Operation documentation	Artifact

It can be seen that the classes in the CPM adequately capture the information generated in the design process. Additional effort is needed to provide for the representation of decisions and design rationale in the CPM. Although not identified as deliverables, progress in the design process comes about through a series of comparisons and decisions. While the Core Product Model is capable of capturing the information on which the decisions are based, support for capturing the complete decisions must be further explored.

4 Bridging the Gap between Engineering Design and Analysis

The current state of research leading to tighter design-analysis integration is discussed in three categories: (1) Object-Oriented Modeling; (2) Computer Aided Design and Finite Element Analysis (CAD-FEA) Integration; and (3) multi-aspect information structures.

First, the object-oriented modeling paradigm is presented as a method for modeling physical systems, with benefits of reusability and modularity, that provides an intuitive way to model real-world, physical objects.

Next, design-analysis integration issues are presented specifically related to the integration of CAD and finite element analysis (FEA). Issues such as dimensional reduction, geometric model simplification, design model and analysis model associativity, and model idealization are addressed. Not presented are specific tools, technologies, or software addressing the above issues.

Lastly, an overview of multi-aspect information structure research is presented. While the implementations vary widely, all the models reviewed share the property that the information models representing the various discipline-specific aspects of the artifact are linked, physically or virtually, to a global model that contains all information about the artifact.

The object-oriented modeling paradigm lends support for modular and reusable models with associativity between design and analysis models, while CAD-FEA integration leads to the development of technologies and techniques; CAD-FEA applications based on object-oriented technology further support modularity. Finally, the multi-aspect modeling paradigm supports the development of a product model for the lifecycle of the product, containing information and knowledge associated with the product during the entire development process (see Figure 9).

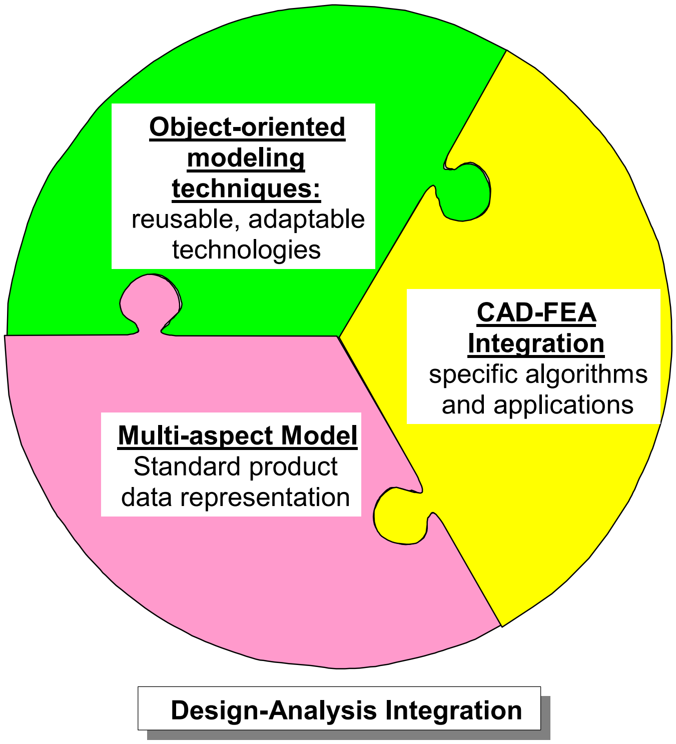


Figure 9. Interactions of Enabling Technologies

4.1 Object-Oriented Modeling

Sinha, et al., (Sinha et al., 2002) propose that the object-oriented programming design methodology can be applied to mechanical systems modeling. The object-oriented modeling approach, as leveraged from the software development domain, is a step in the natural progression of modeling mechanical systems. Tamburini summarizes that object-oriented modeling makes it possible to create physically relevant and easy-to-use components that support hierarchical structuring, reuse, and evolution of large and complex models covering multiple technology domains (Tamburini, 1999).

The benefits of constructing models using objects are: (1) objects are only accessed through established ports, therefore hiding the underlying implementation methods; (2) through the inheritance principle, child objects will inherit the interface and data members from parent objects; and (3) object-oriented models simplify the reuse, maintenance, and extension of models.

Three research projects based on the object-oriented modeling paradigm are presented in detail here:

1. The Composable Simulation Project (Diaz-Calderon et al., 2000; Diaz-Calderon et al. 2002; Paredis, 2001; Sinha et al., 2000; Sinha et al., 2001).
2. The Multi-Representation Architecture (Peak et al., 1994; Peak et al., 1993a; Peak et al., 1993b; Peak et al., 1998; Peak et al., 1996; Peak et al., 1999; Scholand et al., 1997a; Scholand et al., 1997b; Tamburini, 1999).
3. MOSAIC - Integrated modeling and simulation of physical behavior of complex systems (Andersson, 1999a; Andersson and Sellgren, 1998; Andersson, 1999b; Andersson et al., 1995; Andersson 1996; Sellgren and Drogou; 1998; Sellgren 2000, Sellgren, 2001).

The Composable Simulation Project

The Composable Simulation Project, developed at the Institute for Complex Engineered Systems at Carnegie Mellon University, is based on the idea that system level simulations can be automatically generated from individual components from a CAD system. This allows for systems to be simultaneously designed and simulated (Sinha et al., 2001). The technology is based on the following characteristics:

- Simulation models are composed in an object-oriented, hierarchical fashion from model fragments;
- Multiple models are associated with a single system component. These models are organized so that model fragments can be easily reconfigured (through composition and instantiation) to suit a particular simulation experiment; and
- Model parameters are automatically extracted from the CAD geometry and material properties.

The ultimate goal of composable simulation is to develop a modeling methodology that allows the designer to quickly and easily verify the behavior of the system being designed (Paredis, 2001). Toward this goal, the modeling technique must support the reuse of models and be integrated with the design environment (Sinha et al., 2001). Composable simulation will allow designers to quickly select the most appropriate model for the current phase in the design process. The simulation models developed allow for easy refinement and modification, necessary to support the evolutionary nature of the design process.

The Composable Simulation Project is supported by *Reconfigurable Models* and *Component Libraries*. A *Reconfigurable Model* is a system representation based on interface and implementation. Interface is used to describe the interaction through ports and implementation described the internal behavior of a system. The *Component Library* is a set of reconfigurable models for use by the designer/analyst (Diaz-Calderon et al., 2002).

Port-Based Modeling. Port-based modeling is based on two concepts: ports and connections (Diaz-Calderon et al., 2002). Ports represent the localized points of interaction on the boundary of a system. Ports allow for energy flow in and out of the system to interact with other systems. A port exists for each interaction in the system. The energy flow through a port is represented by across and through variables. A connection between two ports on different components represents the exchange of energy between the components. In addition to energy, physical interactions and signals can also be captured through ports. Physical interactions have no direction and are modeled as non-causal relations. Signals have a predefined input and output. Figure 10 is a high level representation of the port-based modeling paradigm.

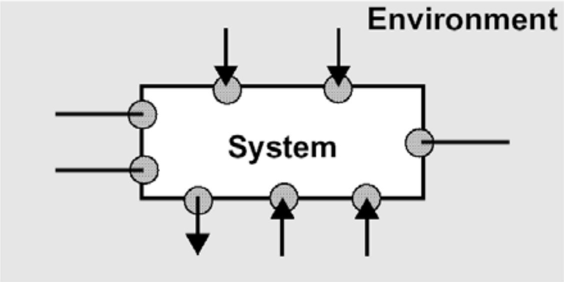


Figure 10. Model of an Engineering System - Port-Based Modeling (Diaz-Calderon et al., 2000).

The ports of a system are grouped into the interface that describes the interaction with the environment. Systems are self-contained entities, the internal behavior of which is independent of the implementation. A limitation of the port-based modeling is only lumped-parameter interactions can be modeled (Diaz-Calderon et al., 2000).

Reconfigurable Models. A reconfigurable model is a representation of a system based on interface and implementation (Diaz-Calderon et al., 2002). Interfaces between component are described through ports, and implementation defines the internal behavior of the system. It is possible to achieve different configurations of the same system by altering the different implementations while keeping the interfaces constant. Reconfigurable component models provide a mechanism for describing system changes to both structure and parameters (see Figure 11*).* Multiple configurations and simulation instances can be achieved by changing parameters and reconfiguring system components, known as *composition* and *instantiation* (Diaz-Calderon et al., 2000).

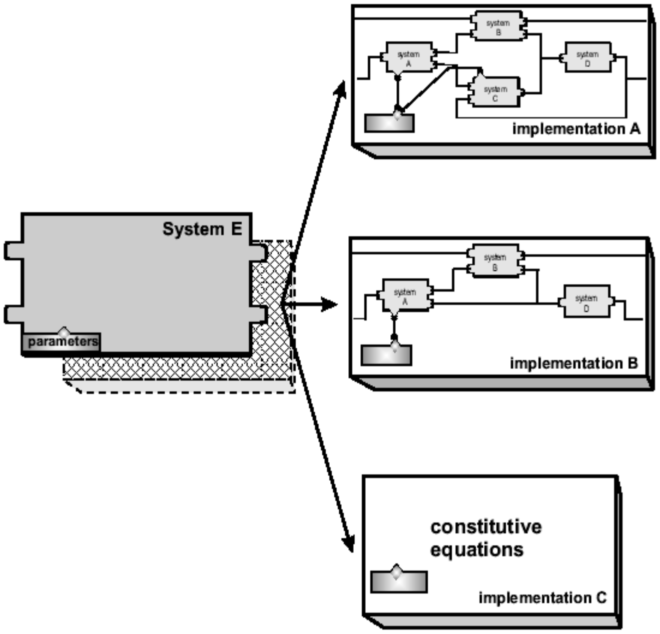


Figure 11. A Reconfigurable System Model (Diaz-Calderon et al., 2000).

Component Libraries. The component library presents to the users a set of reconfigurable models. Two kinds of models are present in the library: system component models and component interaction models. Components include motors, gears, resistors, micro-controllers, etc.. Interaction models include the dynamics of two components without being tied to particular component types. The models are organized and classified in an intuitive manner, such that the designer can easily develop the appropriate analysis model. Additionally, a component may be an abstract simulation model, where the structure is defined, but the system parameters are not yet instantiated. The organization of the component library supports simulation-based design by associating function, behavior, and form. Designers can progress from highly abstract representations for first-run simulation to detailed components for final design.

Integrating Behavior and Form Models. As a natural extension, tight integration of design and simulation environments is achieved. System behavior can be simulated at a cost less than that of physical models. The observed system behavior, based on the simulation of the virtual prototype, is compared to the desired behavior, resulting in either iteration and redesign of the system or progression to subsequent design phases (Diaz-Calderon et al, 2002; Sinha et al., 2000; Sinha et al., 2001). To achieve tight integration of design and analysis, design models should support the creation of composable simulation models. Just as importantly, simulation models should also support design model views. Sinha, et al. develop a design environment that tightly integrates the design and analysis. Using the component library, the simulation model and the design model can be created simultaneously. The designer can simulate the behavior of the system and evaluate the geometric form of the system. This type of design and analysis integrating is common in electrical CAD (ECAD). However, most commercial mechanical CAD applications do not support this type of integration.

The Multi-Representation Architecture

The multi-view representation architecture (MRA), developed at the Engineering Information System Laboratory (EISLab) at the Georgia Institute of Technology, is addressing the gaps between traditional design (CAD) and analysis (CAE) tools. The strategy presented is based on information-intensive mapping between design and analysis models. The MRA is aimed at satisfying the following needs in linking CAD and CAE: (1) automation of routine analyses; (2) representation of design and analysis associativity and of the relationships among the models; and (3) provision of analysis models throughout the life cycle of the product (Peak et al., 1998).

The MRA attempts to bridge the gap between design and analysis based on four building block constructs (see Figure 12).

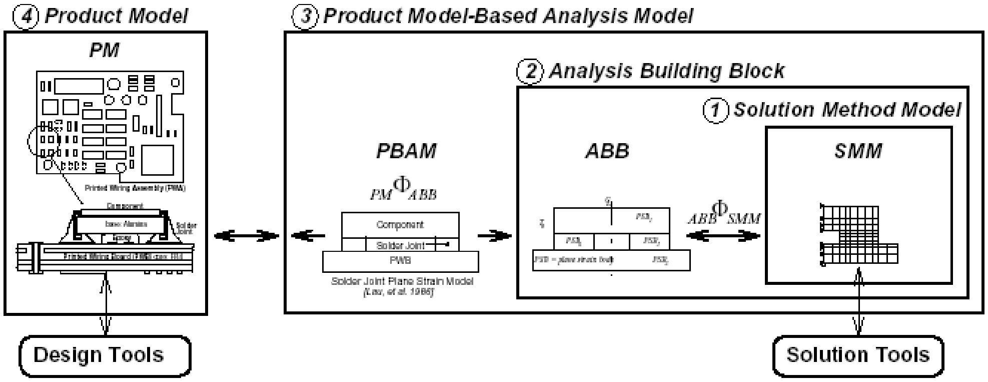


Figure 12. The Multi-Representation Architecture (Peak, et al., 1994).

The Solution Method Models (SMM) represent low-level, solution-specific methods combining inputs, output, and control for a single type of analysis solution. A SMM is a wrapper of the necessary information to complete an analysis solution. It serves as tool agent to provide information on what solution tool to use, the inputs to the tool, the control for the tool, and how to retrieve results from the tool. SMMs are created for diverse solution methods and for various vendor-specific tools.

Analysis Building Blocks (ABB) represent engineering concepts that include engineering semantics and are independent of the SMM. Analysis systems are assemblies of ABBs to represent a particular model. ABBs are constructed utilizing constraint graphs and object-oriented techniques. The ABB structure represents the information template to define relationships. ABBs are represented in both graphical and lexical views (see Figure 13). ABBs uses transformation operators to associate them with SMMs. The SMM instance is created from inputs based on the ABB. The nature of ABBs allows for different solution methods to be used.

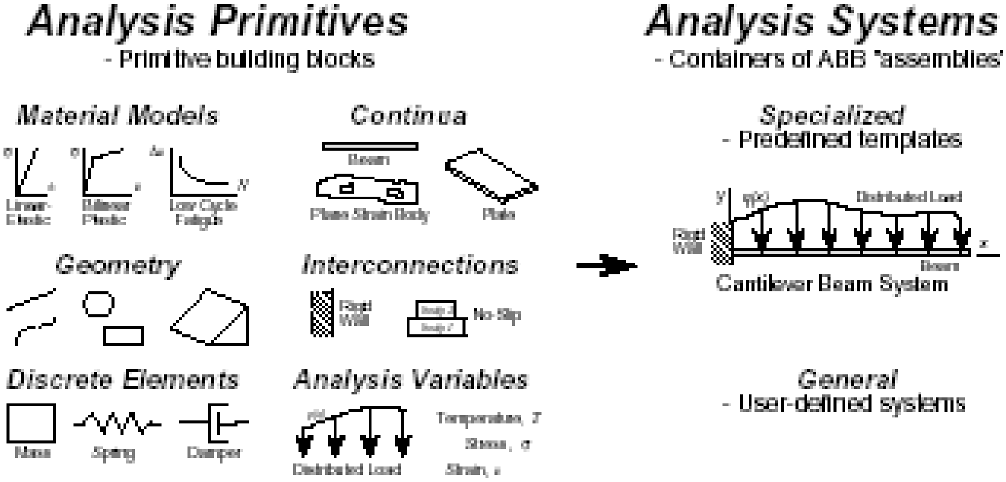


Figure 13. Categories of General Purpose ABBs (Peak, et al., 1994).

Product Models (PM) represent all data associated with the product over its lifecycle. Traditionally, CAD and CAE focus was on geometric representation of the product. However, additional information is vital to the completion of analysis over the product's lifecycle including design, analysis, manufacturing, marketing, installation, and repair. The PM model represents design-oriented information. Items such as geometry, loading conditions, and boundary conditions are included in the PM. The MRA extends the PM beyond the detailed manufacturing description of the product. Analysis models are created based on idealizations and simplifications of the model. Such idealizations are captured in a reusable sense for the PM.

Product Model-Based Analysis Models (PBAMs) contain the linkages between the PM and the ABBs. PBAMs connect the PM to product-independent ABBs to solve specific analysis problems.

A major focus of the PBAM analysis models is routine analysis. Peak, et al., define routine analysis as analyses that are regularly used to support the design of a product (Peak et al., 1999). In the process of designing a physical artifact, checks need to be performed in functional areas such as performance, reliability or manufacturability. The same types of analyses may need to be performed at several stages of the design process. A routine analysis model can be regularly used throughout the design process. Routine analysis modules are created based on the MRA structure. PBAMs represent analysis activity models that associate the analysis model (ABBs) with the design model (PM). Using PBAMs, catalogs of ready-to-use analysis modules are created and available for use in later analysis activities (Peak, et al., 1994). An implementation of routine analysis for printed wiring board (PWB) warpage analysis is presented in (Peak et al., 1996).

The following observations pertain to the routinization of analysis in the design process:

1. Knowledge capture - the MRA method captures the knowledge and expertise of analysts for use by designers.

2. Synergy - development of PBAMs can identify gaps in the analysis domains and can encourage analysts to address different analysis domains.

3. Encourage additional analysis - by identifying the need for routine analysis models, designers are forced to think about what questions should be answered.

4. Ensure proper usage - currently there are no checks for proper usage of the modules. Guideline should be included to ensure the modules are being used properly.

5. Usage by non-designers - all engineers can benefit from routine analysis. For example, manufacturing engineers can greatly benefit from modules associated with manufacturing of the product.

Finally, the MRA paradigm is implemented in a Java-based prototype toolkit, XiaTools. XiaTools uses constrained object (COBs) to address the integration of design and analysis activities and is integrated with analysis tools such as ANSYS (FEA) and equations-based tools such as Mathematica (Peak, 2000).

The benefits of MRA are that it: (1) addresses the information-intensive nature of CAD-CAE integration; (2) breaks design-analysis gaps into smaller problems; (3) supports the use of different tools; (4) is modular and reusable; and (5) defines a method for creating routine analysis tools.

MOSAIC - Integrated modeling and simulation of physical behavior of complex systems

The aim of the MOSAIC project, based on research by K. Andersson and U. Sellgren at the Royal Institute of Technology in Sweden, is to improve and increase the integration of modeling and simulation of products during the product development process. The main goal of the project is the development of an object-oriented model of behavior of the product. Toward this end, the researchers develop a general product model applicable to the entire product development process, and develop a prototype system to support design and simulation of complex products (Andersson, 1999).

The approach in the MOSAIC system is to treat the product development process and the engineering data that are created as *technical systems*. This approach enables the product to be divided into a number of subsystems to be analyzed. Each system can be characterized by what is within its boundaries and how it interacts with other systems. Interfaces between systems are described by mating features and interface features. Mating features are used to characterize the position of the connected systems. Interface features characterize the connection between the mating features. In other word, mating features are *what* is connected between two systems and interface features are *how* the two systems are connected. Because connections consist of both mating classification and interface classification, the systems are easy to modify. Multiple design alternatives can be developed by changes in the interface connections (Andersson and Sellgren, 1998).

The authors aim to increase the efficiency of the modeling process from conception of product to verification. Designer, given better tools, can develop models of systems and subsystems that can be reused and reconfigured during the development process.

An IDEF0 diagram that represents the activities associated with design and analysis is presented in Figure 14.

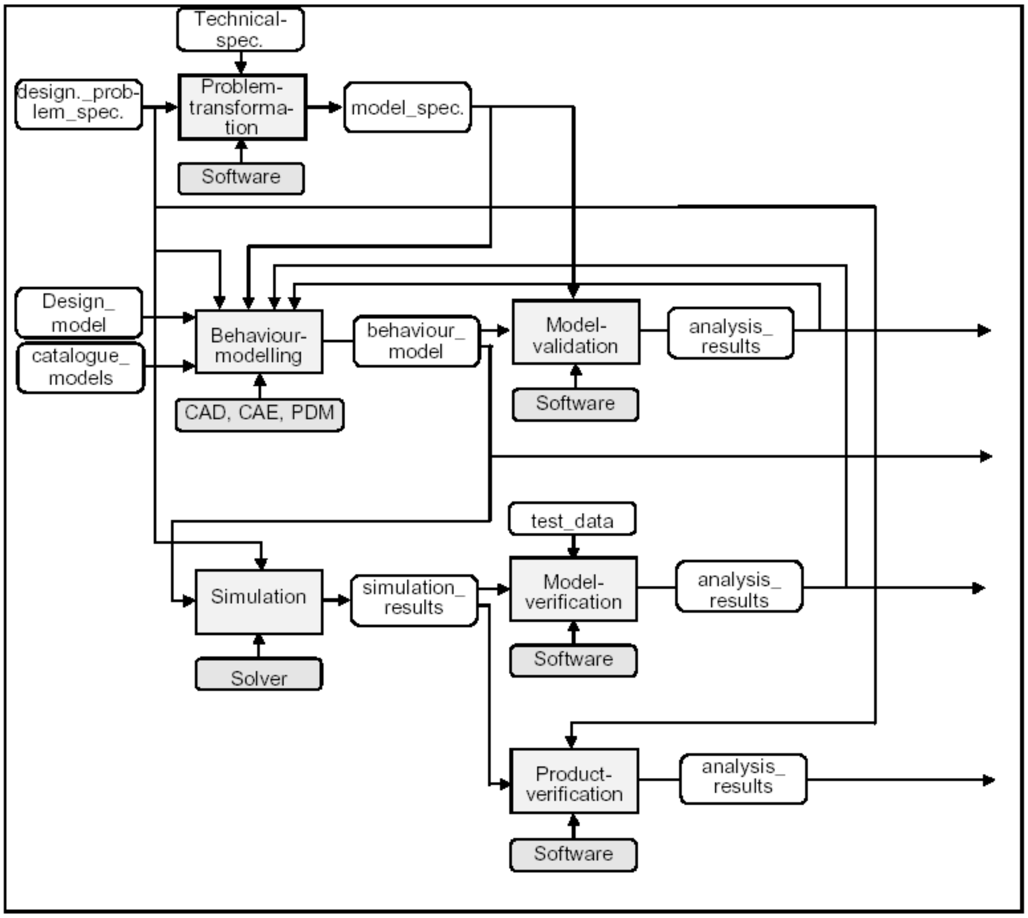


Figure 14. Activity Chart, Decomposition of Design Verification (Andersson, 1999).

The model of complex systems is constructed from the following subsystems:

1. requirements specification - determine the technical demands on the product;
2. environmental models - models of variables and attributes of the simulation environment;
3. geometric models - CAD models forming the basis of multi-body systems (MBS) and FEA models;
4. MBS models - the model of the system; and
5. Finite Element models - strength analysis models.

The prototype MOSAIC system consists of a process model, object model, catalogs of requirements and analysis models, system models, methods for validation, and methods for translating requirements to technical specifications.

The MOSAIC project contributes to engineering design-analysis integration through a prototype system that provides: (1) a generalized model of the product development process: (2) an object model of mechanical systems; (3) guidelines for configurable and interchangeable systems; and (4) the capability of configuring systems for a variety of simulations.

4.2 Computer Aided Design and Finite Element Analysis Integration

Research on CAD-CAE integration has largely concentrated on one CAE analysis tool, Finite Element Analysis (FEA). The area of CAD and FEA integration has been the focus of research for many years. The following literature review spans CAD-FEA integration from the early 1990's to the present. CAD-CAE integration research can be categorized into two focus groups. The *microscopic* view deals with specific issues, such as automatic mesh generation, model simplification, loading and boundary condition idealizations, etc., required for creating the finite element models. The *macroscopic* view is concerned with the overall product data structuring and with the sharing and reuse of product data among applications.

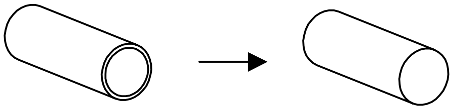
The complete categorization of technologies and tools for the seamless integration of CAD and FEA applications is a colossal undertaking, one that is out of the scope of this study. The aim of this section is to shed light on the problems encountered and the research issues that must be addressed to achieve more seamless integration between CAD and FEA.

Microscopic Approaches

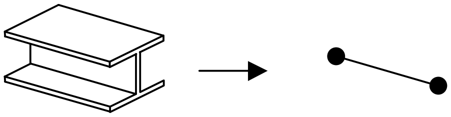
Microscopic approaches to CAD-CAE integration concentrate on issues such as automatic mesh generation, face clustering, geometry simplification, and idealizations. The technologies presented here are only a small sampling of the current and past research.

Substantial simplification of the design geometry is required to create a usable analysis model for the FEA. Analysis models are generated based on simplification and idealization.

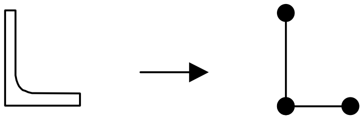
To automate the creation of analysis models, the operations must have rationale. In other words, the operations must use knowledge of the design to automatically create the analysis model. Armstrong, et al. use the idea of *a priori* knowledge and *a posteriori* analysis of the results to make appropriate idealizations. Saint-Venant's principle is one basis to achieve appropriate idealizations. Additional operations, such as medial-axis transform, dimensional reduction, and feature removal are used to create the analysis model. Figure 15 and Figure 16 summarize the transformations needed for creating analysis models based on design models (Armstrong, 1994).



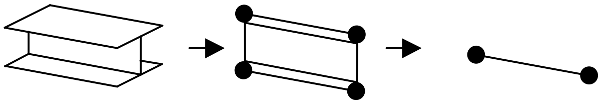
(a) 3D solid - shell



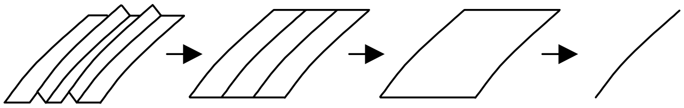
(b) 3D solid - beam



(c) 2D solid - beam

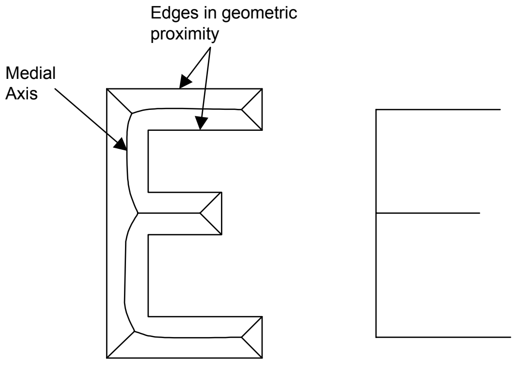


(c) 3D shell - plate with stiffeners - beam



(d) 3D shell - shell with stiffeners - shell with orthotropic material (smeared stiffness) - beam

Figure 15. Dimensional Reductions (Armstrong, 1994).



2D Region Beam Model

Figure 16. A 2D Region, its Medial Axis and an Equivalent Beam Model (Armstrong, 1994)

Armstrong, et al. (Armstrong et al., 1996) expand on the concepts described in (Armstrong, 1994) by describing the operations that allow analysts to suppress details and reduce the dimensionality of the part. Detail suppression is used to remove the geometric features that cause disturbances in the stress field. By suppressing details, a simpler mesh and faster finite element analysis results. The merging of adjacent features and collapsing portions of the model to lower dimensions simplify the model. Procedures for merging faces and edges are shown in Figure 17.

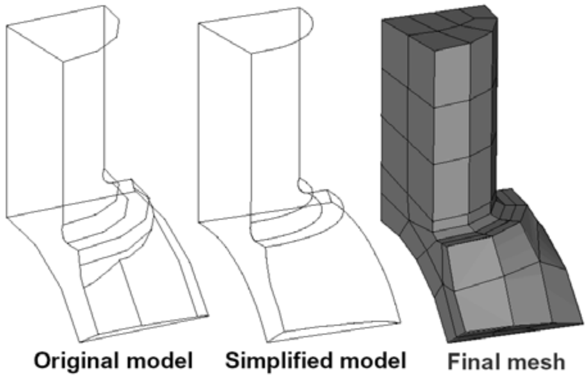


Figure 17. Merging of Faces and Edges (Armstrong et al., 1996).

Finally, the idealization operations, as presented in Figure 15, Figure 16, and Figure 17, are automated by use of command files. Considerations such as stress concentration and multiple regions are applied to appropriately idealize the model and increase the accuracy of the results. The operations presented have been successfully implemented to create an idealized analysis model from a design model. While the operations are important to automatically create analysis models based on the design model, this is achieved through active intervention and not by associativity of the models.

Macroscopic Approaches

A review is provided of current and past research efforts that support the integration of CAD and CAE on a macroscopic level. The macroscopic view of integration focuses on the formal description of products and the development and usage of standards to support integration of product information in many engineering domains. The research spans the development of frameworks for integrating CAD and FEA to standards, such as STEP, to support sharing of product information throughout product life cycle.

Frameworks for CAD and CAE Integration. In the early 1990's, Arabshahi, et al. presented a vision of CAD and FEA based design-analysis integration (Arabshahi et al., 1991). The paper, although dated, provides valuable insight into analysis modeling. The authors develop an IDEF0 process model of the FEA process.

The steps associated with building a finite element model based on a solid model are: (1) select geometric model; (2) abstract to reduce the size of the model; (3) subdivide into mappable mesh regions; and (4) generate analysis input (mesh creation).

As a natural extension to the work in (Arabshahi et al., 1991), Arabshahi, et al. present the activities of an automated CAD-FEA transformation (Arabshahi et al., 1993). The aim of the work is to enable the analysis of the product to respond to design changes and allow seamless integration between design and analysis. The problems in CAD-CAE integration identified in this work still exist.

A large portion of the analyst's time is spent preparing an idealized model, even though complete geometric information already exists, because most analysts reconstruct the model from scratch in the FEA application. This process is time-consuming and error prone, and can result in a model that is substantially different from the design model.

An overview of the future system of CAD-CAE integration is presented. The system would enable the automated created of analyses models from the design model. The system consists of the following:

1. A Product Description System (PDS) to hold the geometric data and non-geometric data associated with the product;
2. A semi-automatic means for transforming the geometric and non-geometric data to an analysis model that can be meshed;
3. Intelligent meshing routines to provide varying degrees of meshing and feedback on the meshed geometry;
4. A series of finite element solvers for a range of solutions; and
5. A post-processing capability to associate results from the idealized model to the design model and allow for modifications.

Additionally, the authors present the vision of components in the CAD-FEA transformation algorithms. These components enable the use of solid model data to drive the analysis model. The functional components the authors anticipate as part of the CAD-FEA schemes are the Attribute Editor, with the ability to apply attributes such as loads and constraints and the Detail Editor that allows for the modification of detail to be simplified with little effect on the analysis results. Additionally, the detail must be kept track of in order to reverse the process. For adaptive idealization, results from previous analyses are used to identify the appropriate analysis defeaturization. The Dimensional Reduction Aid is used when appropriate to reduce the dimensionality of the solid model. In some instances, it is appropriate to reduce the dimensions to decrease the cost of the analysis. The Macro-Feature Builder aids the analyst by creating several larger features that are easier to mesh. The Cutting Surface Facility further “cuts” the features into parts that can be more easily meshed.

In summary, Arabshahi, et al. develop a vision for integrating CAD and CAE. Although technology has increased greatly, the problems that existed almost 10 years ago are still present in the current state of engineering analysis. The research presents a good view of how integration should be approached from the CAD and FEA perspective.

Morphological Analysis. Belaziz, et al. (Belaziz, 2000), develop a feature-based tool to aid in the integration of analysis during design. The tool is based on the morphological analysis of solid models, and a simplification and idealization process. Modifications can be made to the design features based on parameterization. It is possible to walk back from the idealized model to a new modified solid model based on analysis results.

The morphological analysis concept is based on the idea that an object is created from a solid "stock" through a progression of modification steps. The morphological features are classified into elementary features, composite, interacting, and characteristic relationships. The morphological analysis is completed in three steps: (1) detect all characteristic modifications; (2) re-constitute the previous model based on the modifications; and (3) code the modifiers. The structure of the morphological analysis is included in Figure 18.

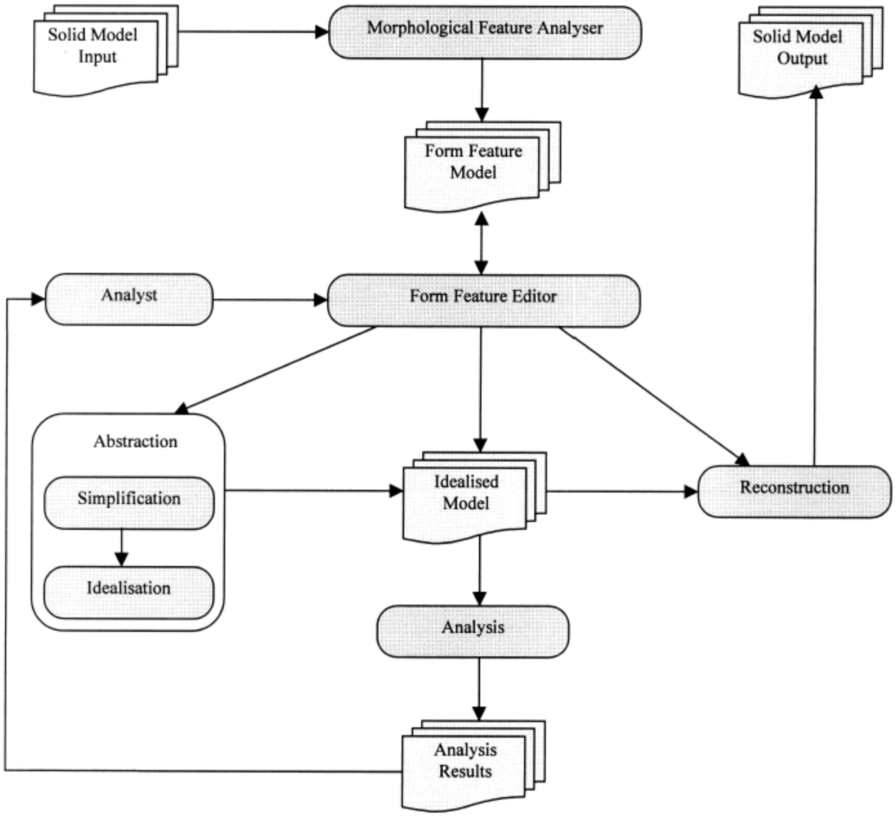


Figure 18. Morphological Analysis Tool Components (Belaziz, 2000).

The form feature model can be obtained in two ways. If the geometry exists, the features can be mapped to the geometric model. This allows each feature to be associated with a particular function. If the geometry does not exist, designers can create a feature description. Next, the analysis model is generated in a two-phase process of simplification and idealization. In the simplification phase, irrelevant information is cleaned out of the model. In the idealization phase, the geometry is constructed to ideal shapes. The analysis is then completed on the idealized model. Finally, reconstruction enables the recreation of a solid model based on analysis results. This idea supports the bi-directionality needed for design-analysis integration. The reconstruction allows modification made during the analysis phase to propagate to the design phase.

The morphological analysis method is applicable to the design process because the level of detail of geometry progresses in a manner similar to the level of detail associated with the various phases of the design process. Additionally, designers iterate between design and analysis. In preliminary design, functional design is of most concern with little focus on detailed geometry. In this phase the simplification step is irrelevant, but abstract modeling techniques by functional analysis are needed.

The morphological analysis techniques provide a tool for linking CAD and analysis systems. The analysis models are based on the net design shape of the part and the reconstruction operators make it possible to link the analysis model back to the design model.

STEP and Express-based models. Sellgren (Sellgren and Drogou, 1998) develops an object-oriented approach to FEA modeling. The modeling paradigm separates models, submodels, interfaces and orientations. A systems approach is used to model products. Systems are described by recursive subsystems and related through interfaces. A system model is an idealized representation of a system at a level of complexity and detail to complete analysis.

System models are aggregated models of subsystems connected by interfaces. The interfaces are aggregates of mating faces. The modeling is expressed in EXPRESS-G format (see Figure 19).

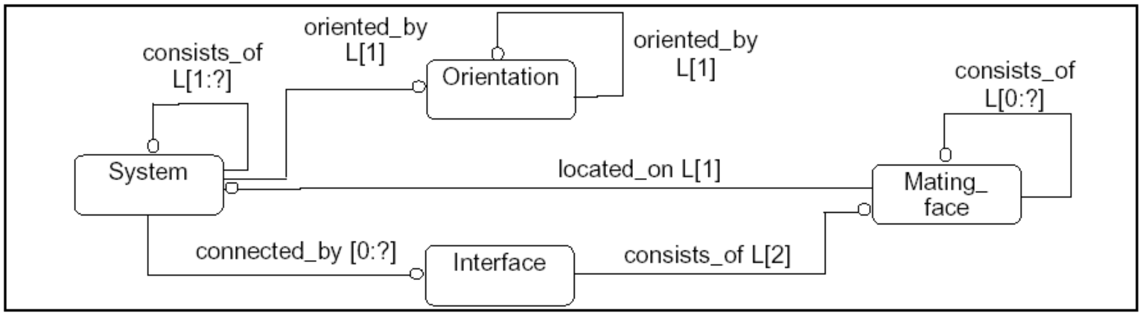


Figure 19. A Systems Model and its Relationships (Sellgren and Drogou, 1998).

Behavior features represent the form features at a particular level of detail. The form features can be represented by a number of different behavior features for different fidelity models. The behavior features describe the physical properties of the form feature and how they relate to other features (see Figure 20). (Note that this is a different definition of "behavior" than that used in the Core Product Model.)

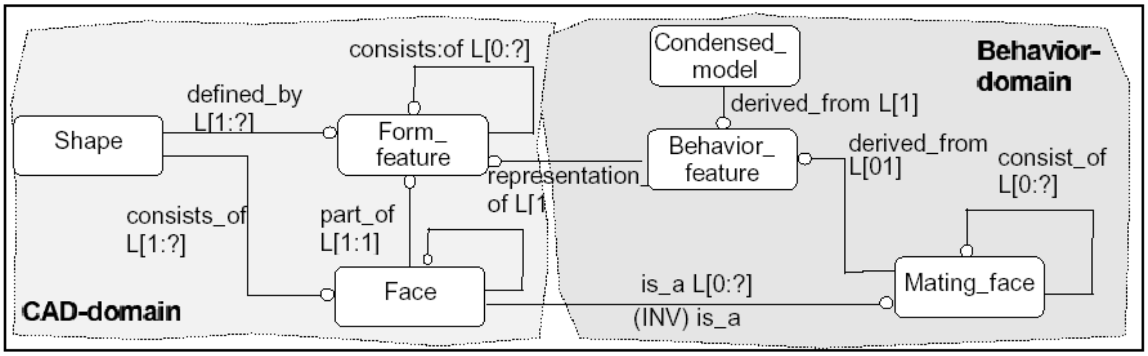


Figure 20. Behavior Feature and Mating Face Associated to Design Shape (Sellgren and Drogou, 1998).

The FEA mating relations are treated as relations between nodal degree of freedoms (DOFs) in two different bodies. The submodels may be nodally compatible or incompatible. Incompatible submodels make it very difficult to mesh transitions. The interface between models can be specified as contact or attachment. To deal with incompatible bodies, the nodal relations are based on the master and slave node concept (see Figure 21).

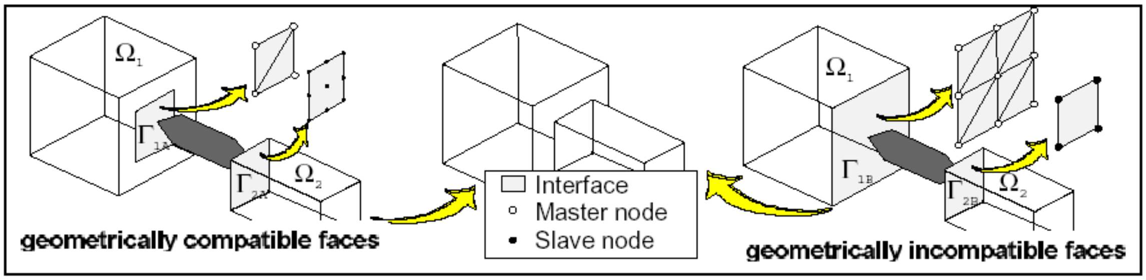


Figure 21. Implicit Master-Slave Selection (Sellgren and Drogou, 1998).

An example of order based design (OBD) is presented in the design of a high-speed grinding wheel. Figure 22 depicts how the separate components are related through relationships and interface features.

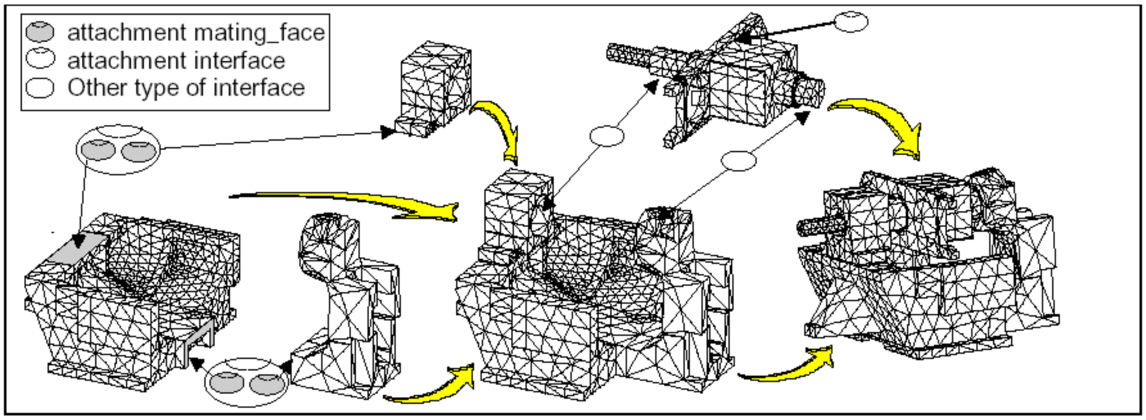


Figure 22. Thermal Interface Couplings at Different Levels of Abstraction (Sellgren and Drogou, 1998).

Sellgren states that "the strong relationship between shape and behavior drives the need for CAD and FE domains to be integrated." An integrated environment allows behavior modeling and simulation to be completed for various life cycle phases in the development process. Sellgren provides a good method for relating components to systems. The relationships can be taken to any level based on assemblies or single components.

Implementing STEP to address Design and Analysis Integration

As previously stated, trends in current research are toward the standardization of product information. A large effort has been put into the development of ISO10303 standards to support data exchange (ISO, 1999). ISO10303, commonly known as STEP, aims to eliminate many of the problems associated with integration by providing a method to exchange and share a rich range of product information. The standard allows for platform-independent sharing and exchange of product data information. The STEP standard includes the focus of this review, AP209. AP209 deals with Composite and Metallic Structural Analysis and Related Design.

The scope of AP209 is the product definitions of the analysis and design disciplines. The analysis discipline of AP209 primarily focuses on FEA models and analysis controls, results and reports. The analysis reports capture and document design and analysis decisions, such as geometric and material idealizations, as well as reference documents containing textual and graphical descriptions of the model, controls and results.

The design discipline of AP209 is concerned with shape representation of components and assemblies. The shape representation captured in AP209 is interoperable with that in AP203, which is currently implemented in many commercially available CAD applications.

AP209 provides an important mechanism for sharing information between analysis design models. Shape information is shared between nodes, point, surfaces and shapes. Additionally, the models share composite material information, and the same product structure. AP209 provides a standards-base solution to iterative design-analysis integration problems (Hunten, 1997).

Although the focus of design-analysis integration cannot be limited to form alone, it does play a major role in each domain. Additional concerns are the appropriate material model, mesh size, and the application of boundary and loading conditions. However, the major roadblock in design-analysis integration is the generation of the appropriate analysis geometry. The cost to change or defeature the CAD model is sometimes greater than that of directly creating the idealized geometry.

Gordon (Gordon, 2001) summarizes CAD and CAE integration issues, as well as identifies three categories of design geometry to analysis geometry translations:

> Today's bottleneck in CAD-CAE integration is not automated mesh (grid) generation, it lies with efficient creation of appropriate simulation-specific geometry."

> "The key to understanding CAD-CAE Integration, is related to the scale, scope and purpose of the required engineering analysis - e.g. Finite Element Analysis (FEA)."
>
> "It is not simply related to the existence of captured CAD geometry, a perception unwittingly left during product model 'walk-throughs'."
>
> "The closer the scale, scope and purpose of an engineering analysis is to the type and detail of the existing CAD product model geometry, the greater the likelihood that a closely-coupled, automated, or even seamless integrated CAD-CAE process can be implemented (Gordon, 2001)."

The categories of geometry translation, as developed by Gordon (Gordon, 2001) are the following:

- Category I - the CAD geometry and analysis geometry are identical.

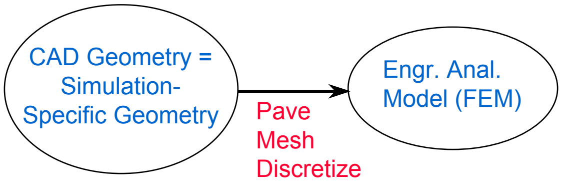


Figure 23. Analysis Geometry same as Design Geometry (Gordon, 2001).

- Category II - the CAD geometry has too many features for the analysis model. Unnecessary detail is removed and/or the type is modified to create the analysis model.

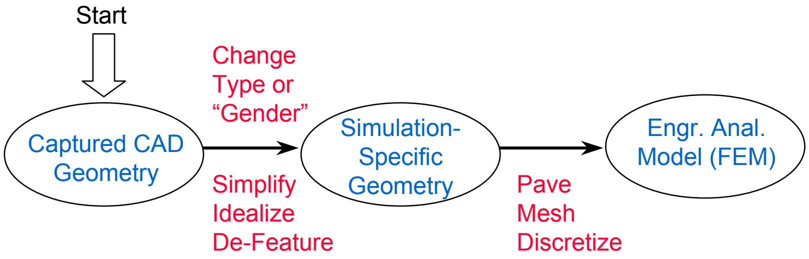


Figure 24. Analysis Geometry Different from Design Geometry (Gordon, 2001).

- Category III - the geometry is CAE centered. Analysis is performed to define and refine a concept. The detailed geometry is derived from the analysis model.

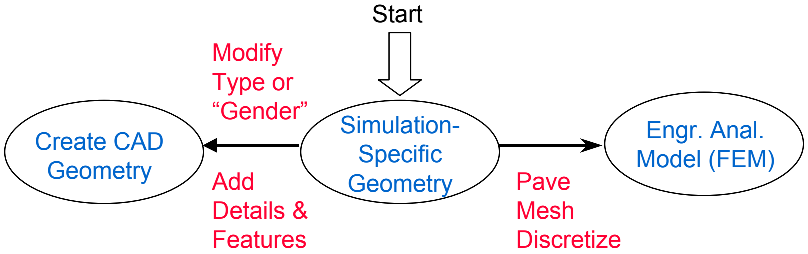


Figure 25. Simulation Specific Geometry Drives Both Simulation And Design (Gordon, 2001).

In many systems, a point-to-point translator is used to facilitate the connection between design and analysis. However, point-to-point translation does not solve all CAD-CAE integration issues, because it does not contain the full richness of the product information originally associated with the solid model. This poses a problem in large-scale design processes, where the integration of many life-cycle processes is necessary. During product development, there may be dozens of applications that must be integrated. Without a common representation, point-to-point translators are used and information is lost (Hunten, 1997).

Toward this end, Hunten discusses the geometric transformation available within STEP AP209 (Hunten, 2001b). Figures 8, 9, 10, and 11 in (Hunten, 2001b) depict the relations between the idealized analysis shape and the design shape. Hunten emphasizes the ability to link the idealized shape to the actual design shape in the product development process.

Additionally, Hunten discusses the capability of AP209 to deal with the new concepts of idealized analysis shape (IAS) and node shape (NS) (Hunten, 2001a). These shapes add enhanced capability to the AP in terms of relating idealized shape to the appropriate design shape. The design specification can be used to specify intent of the designer in order to create the correct idealized shape. However, it is not clear in the presentation how the idealized shapes are created. AP209 provides a mechanism to relate the shapes, but does not provide associativity between the shapes. Figure 26 is extracted from Hunten (Hunten, 2000a).

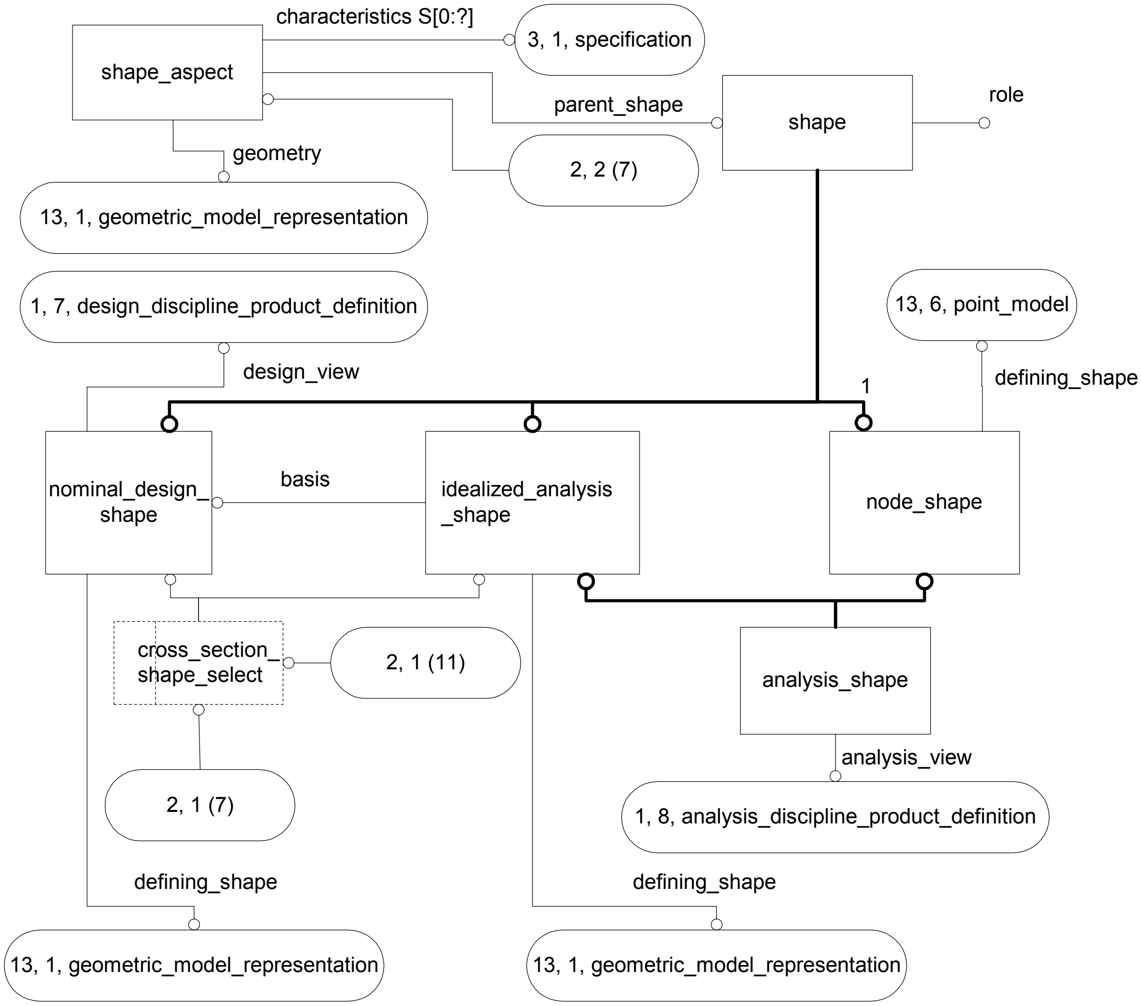


Figure 26. Design–Analysis Shape Association (recreated from Hunten, 2000a).

The aim of STEP 209 is to capture and associate design and analysis models in the product development process, not by simple data exchange, but by close relationship between the design and analysis information.

Research in the development and improvement of the STEP standards is continuing. The Engineering Analysis Core Model (EACM) is part of the STEP standard suite. The EACM describes the way that engineering analysis data are stored and the way that engineering analysis information is exchanged. According to Leal, AP203 and 209 are useful but limited in scope (Leal, 1999). The goal of EACM is to increase the scope by capturing all engineering information to support business practices.

The EACM deals with three key aspects in engineering information management:

1. The management of engineering analysis and design information;
2. The linking of engineering information to activities, decisions, and analyses; and
3. The storage of information about a product in time and space.

These three characteristics of the EACM bridge the gap between CAD and PDM, between workflow and project management, and analyses. The architecture of EACM is included in Figure 27.

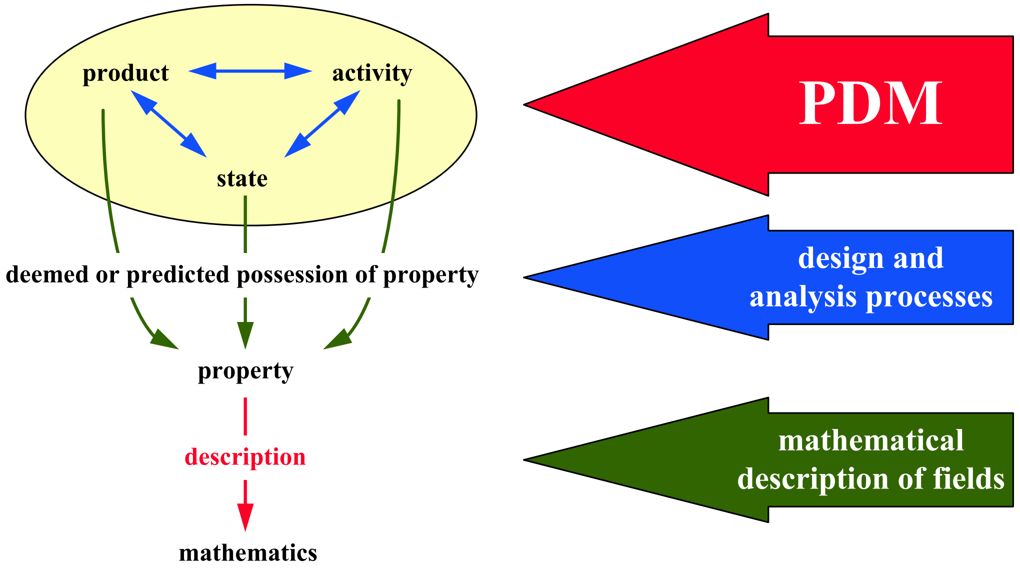


Figure 27. Architecture of EACM (Leal, 1999).

To integrate the three areas, modules will provide data management information, the definition of properties, audit trails for product information, and mathematical techniques (Leal, 1999).

4.3 Multi-Aspect Information Structures

Several research groups are exploring design-analysis integration through a multi-aspect modeling paradigm. The concept is named differently depending on the researcher, but the premise is the same. Product data for the entire lifecycle of a product is stored in a design database. The data can be viewed through different domain-specific aspects or views by applications to generate, display, or modify the product information. For example, a design view of the product may be a solid model, whereas the analysis view of the same product may be a FEA model or a simple Excel spreadsheet to estimate and calculate costs.

CAD and the Product Master Model

Hoffman and Joan-Arinyo present one organization for a product master model (Hoffman and Joan-Arinyo, 1998). The authors develop an architecture that keeps consistent associations between CAD and downstream applications. The architecture accounts for associating product data between various applications, while maintaining the proprietary data of the applications. The authors raise crucial issues regarding the master model:

> "…the data in the master model originate from different domain-specific programs, how can this information be kept consistent and how is it maintained under design changes? In our view, the CAD system is one of

> the clients of the master model, with the primary charge of creating and maintaining the net shape information. …
>
> How can we establish and maintain a persistent association between the geometry data contributed by the CAD system and data originating from other application programs?“ (Hoffman and Joan-Arinyo, 1998)

The design-analysis association is predicated on the master model being an object-oriented repository that has mechanisms for maintaining the integrity and consistency of the information structures for the various engineering domains. Additionally, the master model has several clients, one of which is the CAD application responsible for creating the initial net shape and also for modifying the net shape.

Additional clients associated with the master model may deal with manufacturing process planning, geometric dimensioning and tolerancing, cost estimation, etc. For each of these clients, there is a corresponding view of the product. Each client application can deposit product information it processes to the master model, as well as keep a private repository of information relevant to itself. When a change is made to the model, a protocol is followed to ensure the most up-to-date product data is available to all clients. The change information is posted and it is up to the clients to reassociate with the new information. The overall architecture of the master model is an object server in charge of coordinating the information to all the clients (see Figure 28).

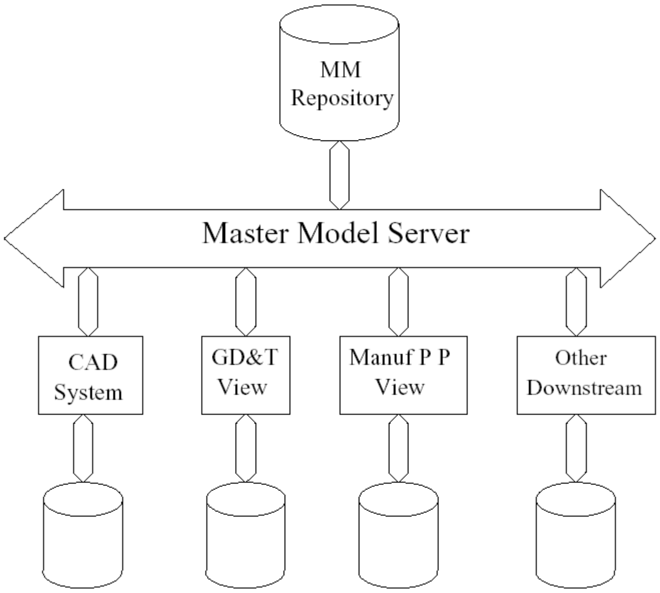


Figure 28. Master Model Architecture with Client Views (Hoffman and Joan-Arinyo, 1998).

The net shape associativity mechanism is developed to create and maintain associativity between elements in the net shape of the product. The mechanism is based on the premises that: (1) clients do not need to communicate directly with CAD; (2) the CAD system creates a simple information structure that allows each client to reassociate the information; and (3) common methods are used to support the CAD systems.

Each net shape element is identified with a unique id, a topological type, and a characteristic point, defined as the *geometric certificate*. This allows the CAD model to be re-evaluated and new geometry to be created. The CAD model, however, must not be edited so as to keep the associativity. The CAD system must provide a time stamp and identification of the CAD file to assure synchronization with the net shape.

Once a net shape is deposited in the master model, each client application is allowed to associate information with that shape. For each net shape, the master model creates an inventory of geometry certificates and of the applications that have made an association to it. If the net shape is changed in the CAD system, the master model calls on each of the applications that are associated with the element. The information, although accessible by all clients, is assigned to a primary client. The primary client is in charge of doing the primary editing of the data. For example, the net shape is assigned to a CAD system. The information that is owned and modified by one client affects all other client applications with association to the product information. The coupling of information between views requires rules to ensure the data remains consistent and orderly.

Hoffman and Joan-Arinyo summarize the difficulties with the product master models. The biggest problem is the ability to establish associations between and with net shapes and to keep the information consistent in distributed network. However, the authors believe that a change protocol provides a realistic mechanism for associating downstream applications to the CAD model. Additionally, the change protocol eliminates the burden of creating custom associations for each different CAD system. The change protocol provides a realistic approach for model association without compromising proprietary data. The architecture is modular and extensible.

The Pluggable Metamodel Mechanism

Yoshioka and Tomiyama (Yoshioka and Tomiyama, 1999) present a mechanism for integrating various aspect models, such as geometric, kinematic and finite element models for knowledge intensive engineering. The KIEF (Knowledge Intensive Engineering Framework) is constructed using multiple objects (i.e., aspect models) expressed through a metamodel mechanism. The metamodel represents the relationships between the concepts in the aspect models.

Knowledge intensive engineering assists engineering activities in various stages of the product lifecycle using various kinds of engineering knowledge. The framework proposed integrates and maintains the consistency of the various models. The KIEF framework integrates commercially available software tools through a "Pluggable Metamodel Mechanism" (Yoshioka and Tomiyama, 1999).

An aspect model is a model of a designed artifact from a particular point of view. For example, a FEA aspect model may be completely different from the geometric shape aspect model. Aspect models are built by first constructing relationships between models. Model simplification and abstraction are common tasks during this step. Next, data is transferred from existing models to complete the new aspect model.

The metamodel mechanism provides the framework for integrating the many aspect models associated with a technical artifact. Designers initially build a primary model that represents the intended physical behavior of the object. Aspect models are built by determining the level of abstraction desired, determining the appropriate simplification needed, and finally by the exchange of data between aspect models. The metamodel mechanism describes how information is exchanged among the aspect models. However, it is not always easy to extract all the necessary parameters to complete the aspect model. For this reason, the ability to plug in existing modelers is presented.

The technology presented by Yoshioka, et al. contribute to the master model paradigm of product design significantly. However, it is not clear how the various modelers share product information to support the various aspect models.

The Multiple View Intermediate Modeler

De Martino, et al. present an approach to CAD-CAE integration based on design-by-features and feature recognition (De Martino et al., 1998). Feature-based modeling allows for the representation of semantic information of the product and for more direct communication between engineering processes. However, the sharing of semantic information across engineering applications and domains is not currently supported. To achieve integration between design and engineering processes, a common model must exist. The shared model provides different views for different analysis domains. Toward this end, the intermediate model (IM) is developed. The IM is shared between applications and provides them with context-specific feature-based views. Initially, the designer creates an object using design features from a library. The design is evaluated and stored in the IM to maintain the semantics of the features. The intermediate model links the parametric model and the shape model. Additional semantic information allows for application-specific views for various engineering contexts.

The kernel activities consist of design-by-features, solid modeling, feature recognition, and feature matching. Design by features is based on the parametric instantiation of features retrieved from a library of features. In the solid modeling process, the geometric evaluation and any needed change of the geometry is performed. During the shape feature recognition process, the algorithm iterates to recognize the shape based on context-independent information and returns a neutral file. Finally, in the application feature matching process, the generic shape features extracted from the shape recognition process are interpreted to context-dependent features. The process is based on a teach-by-example technique that uses information stored in a feature library. Users can decide, based on the results of the library search, the appropriate features.

The intermediate modeling process provides the main source of information for the various processes. The process maintains the global state and information content of the intermediate model. An important problem with representing features from different points of the view is that of feature interaction and shape sharing. Additionally, the features may have different representations in different domain views. The intermediate model supports the existence of different representation and non-homogenous data.

The intermediate model provides a mechanism for supporting different types of information representations and allows for the coexistence of application specific views.

5 Summary

The value to be derived from this report is categorized into two areas, similar to the overall organization. The first portion of the report (Sections 1 – 3) is intended to establish the role of synthesis and analysis in engineering design so as to expand the traditional thought of "design-analysis integration" to "design synthesis-analysis integration". Design is not separate from synthesis and vice versa. The deliverables generated in the Pahl and Beitz process are general enough to be identified in any design process. The Core Product Model may be used to store the information as class objects. The correlation between the Pahl and Beitz process and the Core Product Model helps to clarify the classes in the model. Some initial reactions to the Core Product Model questioned its applicability to the entire design process. This exercise provides a clear mapping between an established design process and the objects in the Core Product Model.

The second portion (Section 4) presents a literature survey that charts recent efforts of industrial, academic, and governmental research institutions. It shows that significant effort has been made toward decreasing the barriers in design synthesis and analysis integration. The research reviewed identifies the major issues in integration. These issues are: product information and database management; use of standards (e.g. STEP); development of domain-specific tools and technologies; and paradigms in mechanical system modeling.

6 Areas for Future Work

There are several areas for future research to bridge the gap in engineering design-analysis integration. The literature survey strongly indicates that a general framework, a development roadmap, and a shared vocabulary should be developed. A common vocabulary must be established to provide synergy and exchange of knowledge between engineers and researchers. In conducting the literature survey, cross-references between contemporary research efforts were rarely found and a common terminology to describe the same constructs and methods was not present.

Additional effort must be devoted to the development and implementation of the Core Product Model. In order to refine, prove, and improve the capabilities of the core model, test cases must be implemented. The product model does not have to be complete, or implemented in a software application. This effort will lead to a better understanding of the needed design and analysis models, as well as the associativity between them. Currently, the idea of design model is limited to CAD geometry and the analysis model is limited to a FEA model. These preconceptions must be broken down and more robust meaning of design and analysis models must be developed. The Core Product Model must be verified against a variety of technical artifacts, but also for a variety of design processes and practices.

7 References

Andersson, K. (1999), "A design process model for behavior simulations of complex products", in *Proceedings of DETC 1999, Computers in Engineering Conference*, paper no. DETC1999/ CIE-9008, Las Vegas Nevada, USA. September 12-15.

Andersson K., and U. Sellgren (1998), "MOSAIC – Integrated modeling and simulation of physical behavior of complex systems", *Proceedings of NordDesign ´98*, Stockholm, Sweden.

Andersson, K. (1999), "A Modularized Approach to Modeling Behavior," presented at the *International Conference on Engineering Design, ICED99*, Munich, Germany, August 24-26, 1999.

Andersson K., P. Makkonen, and J-G. Persson (1995), "A proposal to a product modeling language to support conceptual design," *CIRP Annals,* Volume 44, Number 1, pp 129-132.

Andersson, K. (1996), "Computer Support for Early Design Work," appears in *OST ´96 Oulu-Stockholm -Tallinn Symposium of Machine Design*, KTH, Stockholm, May 1996.

Arabshahi, S., D.C. Barton, and N.K. Shaw (1991), "Towards Integrated Design and Analysis." *Finite Elements in Analysis and Design*, Volume 9, Number 4, pp.271-293.

Arabshahi, S., D.C. Barton, and N.K. Shaw (1993), "Steps Towards CAD-FEA Integration." *Engineering with Computers*, Volume 9, Number 1, pp.17-26.

Armstrong, C.G. (1994), "Modeling Requirements for Finite-Element Analysis." *Computer-Aided Design*, Volume 26, Number 7, pp. 573-578.

Armstrong, C.G., R.J. Donaghy, and S.J. Bridgett (1996), "Derivation of appropriate Idealizations in Finite Element Modeling"', the *Third International Conference on Computational Structures Technology*, Budapest.

Belaziz, M., A. Bouras, and J.M. Brun (2000), "Morphological Analysis for Product Design," *Computer Aided Design*, Volume 5-6, Number 32, pp. 377-388.

Brunnermeier, S.B., and S.A. Martin (1999), *Interoperability Cost Analysis of the U.S. Automotive Supply Chain*, Research Triangle Institute, Project Number 7007-03, March 1999. Research Triangle Park, NC

De Martino, T., B. Falcidieno, and S. Habinger (1998), "Design and Engineering Process Integration Through a Multiple View Intermediate Modeler in a Distributed Object-oriented System Environment," *Computer Aided Design*, Volume 30, Number 6, pp. 437-452.

Diaz-Calderon, A., Paredis, C. J. J., and Khosla, P. K., "Reconfigurable Models: A Modeling Paradigm to Support Simulation-Based Design," *2000 Summer Computer Simulation Conference*, Vancouver, Canada, 2000.

Diaz-Calderon, A., C.J.J. Paredis, and P.K. Khosla (2000), "Organization and Selection of Reconfigurable Models", in *Proceedings of the Winter Simulation Conference 2000*, Orlando, Florida, December 10-13.

Fenves, S.J. (2001b), "A Core Product Model for Representing Design Information," *NISTIR 6736.* National Institute of Standards and Technology (NIST), Gaithersburg, MD, 38 pp.

Gordon, S. (2001), "An Analyst's View: Step-Enabled CAD-CAE Integration," presented at *NASA's STEP for Aerospace Workshop,* JPL, Pasadena, CA, January 17.

Hoffman, C. M. and R. Joan-Arinyo (1998), "CAD and the Product Master Model," *Computer Aided Design,* Volume 30, Number 11, pp. 905-918.

Hsiung, C.H. (1998), *The Reusable Engineering Analysis Preprocessing Methodology to Support Design-Analysis Integration,* Doctoral Thesis Proposal, Georgia Institute of Technology, Atlanta.

Hunten, K.A. (1997),"CAD/FEA Integration with STEP AP209 Technology and Implementation," Lockheed Martin Corporation.

Hunten, K.A. (2001a), "AP209 ARM Overview," Available at http://pdesinc.aticorp.org/graphics/aps/ap209_tutorial.ppt. [Date visited 7-12-2002].

Hunten, K.A. (2001b), "Geometric Founding and Associativity in ISO 10303-209 – Revision C 3-14.2001," Lockheed Martin Corporation.

ISO 10303-209 (1999), *Product data representation and exchange: Application protocol: Composite and metallic structural analysis and related design.*

Leal, D. (1999), *The Engineering Analysis Core Model: A 'plain man's guide'*, Version 5.0, PDES, Inc.

Pahl, G., and W. Beitz (1996), *Engineering Design: A Systematic Approach*, 2nd ed., Springer-Verlag, London, UK.

Paredis, C.J.J. (2001), "Composable Simulation and Design," Available at http://www-2.cs.cmu.edu/~compsim/. [Date visited 7-19-2002].

Peak, R.S. (2000), *X-Analysis Integration Technology*, Georgia Tech Engineering Information Systems Lab Technical Report EL002-2000A.

Peak, R. S., and R.E. Fulton (1994), “A Multi-Representation Approach to CAD/CAE Integration: Research Overview,” *Rapid Thermomechanical Design of Electronic Products in a Flexible Integrated Enterprise, Interim Report,* Fulton, R. E.; Ume, C.; et al., Advanced Electronic Packaging Lab., Prj. MS-93-03, Manufacturing Research Center, Georgia Tech, Atlanta, 22-27.

Peak, R. S., and R.E. Fulton (1993a), “Automating Routine Analysis in Electronic Packaging Using Product Model-Based Analytical Models (PBAMs), Part I: PBAM Overview,” presented at *ASME Winter Annual Meeting*, paper no. 93-WA/EEP-23, New Orleans, LA, November 28 – December 3.

Peak, R. S., and R.E. Fulton (1993b), “Automating Routine Analysis in Electronic Packaging Using Product Model-Based Analytical Models (PBAMs), Part II: Solder Joint Fatigue Case Studies,” presented at *ASME Winter Annual Meeting,* paper no. 93-WA/EEP-24, , New Orleans, LA, November 28 – December 3.

Peak, R.S.; R.E. Fulton, I. Nishigaki; and N. Okamoto (1998), “Integrating Engineering Design and Analysis Using a Multi-Representation Approach”, *Engineering with Computers*, Volume 14, Number 2., pp. 93-114.

Peak, R.S.; A.J. Scholand, and R.E. Fulton (1996), “On the Routinization of Analysis for Physical Design,” 1996 ASME International Mechanical Engineering. Congress & Expo, Application of CAE/CAD to Electronic Systems, EEP-Vol.18, D. Agonafer, et al., Atlanta, pp. 73-82.

Peak, R.S.; Scholand, A.J.; Tamburini, D.R.; Fulton, R.E. (1999) "Towards the Routinization of Engineering Analysis to Support Product Design", Invited Paper for Special Issue: Advanced Product Data Management Supporting Product Life-Cycle Activities, *International Journal of Computer Applications in Technology*, Vol. 12, No. 1, pp. 1-15.

Scholand, A.J.; R.S. Peak, and R.E. Fulton (1997a), “Improving Electronic Packaging Manufacturing Through Product and Process-Driven Analysis: A PWB Case Study,” *ASME Intl. Mech. Engr. Congress & Expo, CAE/CAD and Thermal Management Issues in Electronic Systems,* EEP-Vol. 23/HTD-Vol. 356, Agonafer, D.et al., Dallas, 23-31.

Scholand, A.J.; R.S. Peak, R.S.; and R.E. Fulton (1997b), “The Engineering Service Bureau - Empowering SMEs to Improve Collaboratively Developed Products,” *CALS Expo USA*, Orlando, Track 2, Session 4.

Sellgren, U. and R. Drogou (1998), "A systems and process approach to behavior modeling in mechanical engineering", in *Proceedings of the Integrated design and Manufacturing in Mechanical Engineering*, IDMME´98, May 27-29, 1998, Campiégne, France.

Sellgren, U. (2000),"An approach to behavior modeling of complex systems in mechanical engineering", *The International ANSYS Conference*, Pittsburgh, PA, USA, August 2000.

Sellgren, U. (2001),"Aspects of simulations in product development", *Endrea-17, Simulation in product development*, *The Swedish Engineering Design Research and Education Agenda*, Editor. S. Andersson, pp. 1-20.

Sellgren U. (1997), "Requirements on a FEA Assisted Engineering Process", in *Proceedings of the NAFEMS World Congress 1997*, April 9-11, Stuttgart, Germany.

Shooter, S.B., W. Keirouz, S. Szykman, and S.J. Fenves (2000), “A Model for Information Flow in Design,” in *Proceedings of DETC 2000, Design Theory and Methodology*, paper no. DETC2000/DTM-14550, Baltimore, MD, September 10-13, 2000.

R. Sinha, V.C. Liang, C.J.J. Paredis, and P.K. Khosla (2001) "Modeling and Simulation Methods for Design of Engineering Systems," to appear in *Journal of Computing and Information Science in Engineering*. Vol. 1, pp. 84-91.

Sinha, R., C.J.J. Paredis, and P.K. Khosla (2000), "Integration of Mechanical CAD and Behavioral Modeling," in *Proceedings of the International Workshop on Behavioral Modeling and Simulation*, Orlando, Florida, November 18-21, 2000.

Sinha, R., C.J.J. Paredis, and P.K. Khosla (2001), "Interaction Modeling in Systems Design," in *Proceedings of DETC 2001, Computers in Engineering Conference*, paper no. DETC2001/CIE-21285, Pittsburgh, PA, September 9-12, 2001.

Szykman, S., S.J. Fenves, S. B. Shooter, and W. Keirouz (2001), "A Foundation for Interoperability in Next-generation Product Development Systems," *Computer-Aided Design*, Vol. 33, No. 7, pp. 545-559.

Szykman, S., R. D. Sriram, C. Bochenek and J. W. Racz (1999), "The NIST Design Repository Project," *Advances in Soft Computing - Engineering Design and Manufacturing*, Roy, R., T. Furuhashi and P. K. Chawdhry (eds.), Springer-Verlag, London, pp 5-19 (invited keynote paper; not peer reviewed).

Tamburini, D.R. (1999), *The Analyzable Product Model Representation to Support Design-Analysis Integration,* Doctoral Thesis, Georgia Institute of Technology, Atlanta.

Tate D., and M. Nordlund (1996), "A Design Process Roadmap as a General Tool for Structuring and Supporting Design Activities", in *Proceedings of the Second World Conference on Integrated Design and Process Technology (IDPT-Vol. 3)*, Society for Design and Process Science, Austin, TX, , Dec. 1-4, 1996. pp. 97-104

Yoshioka, M. and T. Tomiyama (1997), "Pluggable Metamodel Mechanism: A framework of an Integrated Design Object Modeling Environment," *Computer Aided Conceptual Design `97, Proceedings of the 1997 Lancaster International Workshop on Engineering Design {CACD}'97*, pp. 57-70.