Capturing Design Process Information and Rationale to Support Knowledge-Based Design and Analysis Integration

Final Report
GaTech Project #B-01-691

August 3, 2004

Submitted to:
National Institute of Standards and Technology
Gaithersburg, MD

Technical Point of Contact:
Steven R. Ray
ray@nist.gov
+1-301-975-3524

Prepared by:
[1]Manufacturing Research Center
[2]G.W. Woodruff School of Mechanical Engineering

Georgia Institute of Technology (GIT)
Atlanta, Georgia USA
http://www.marc.gatech.edu/

Technical Points of Contact:
Russell S. Peak [1] (PI)
russell.peak@marc.gatech.edu
+1-404-894-7572

Gregory M. Mocko[2] (Graduate Research Assistant)
gtg888d@prism.gatech.edu
Manas Bajaj[2] (Graduate Research Assistant)
manas.bajaj@eislab.gatech.edu
Injoong Kim[2] (Graduate Research Assistant)
Injoong.kim@eislab.gatech.edu
+1-404-385-1674

Georgia Tech Contracting Organization:
Office of Sponsored Programs
Georgia Tech Research Corporation (GTRC)
http://www.osp.gatech.edu/

ABSTRACT

With the increase in complexity of technical products and a trend towards multi-locational product development teams, an efficient and effective framework to support the design of technical artifacts must be developed. Despite the advances in computing and computer supported engineering tools, significant gaps still exist between formal tools that aid engineering design and analysis activities throughout the product development process. Advances in product modeling, design process modeling, and knowledge-based engineering offer opportunities to develop design support systems to bridge the gaps associated with design-analysis integration.

In this effort the National Institute of Standards and Technology (NIST) and Georgia Tech (GIT) collaborated to apply and develop technologies to support computer-based design-analysis integration support systems. Initial focus was on the development and refinement of product models and design process models. These models should support a high level of associativity between the design process and the corresponding product information generated. Additional work will be on the integration of product and process models to support knowledge-based engineering frameworks. The knowledge captured can be used to aid design engineers in the selection of or usage of appropriate analysis models or the steps that must be followed to create an analysis model. Long-range goals support the development of computer-based design environments that incorporates both product and process knowledge in collaboration with design engineers to aid in the entire product development process.

Benefits include capturing and reusing product knowledge throughout the development process. Such techniques will reduce the time and effort to create appropriate analysis models. Additional benefits will arise from a consistent framework by which to capture product development information in a distributed design environment. These capabilities will lead to decreasing or eliminating gaps associated with design-analysis integration, a major research effort at NIST.

KEYWORDS

Design, analysis, product modeling, artificial intelligence, knowledge-based engineering

TABLE OF CONTENTS

NOMENCLATURE AND TERMINOLOGY

The following is a list of relevant terminology utilized in this report. The definitions presented below are consistent with the DAI lexicon [1]

Abstraction Abstraction is the simplification of detail from a system for a particular view, while ignoring or suppressing those details that are not important.

Analysis Verification of product behavior using mathematical simulation tools.

Analysis Model An idealized representation of the design model used in analysis. Analysis models include behavior models and the mapping of product design parameters.

Analysis Template An established analysis model that is repeatedly used for a specific type of product in a particular context. Variables and parameters and the relations between design parameters and analysis model parameters are fairly well known.

Behavior Represents how the artifact implements its function.

Behavior Model A model that captures the mathematical description of the physical behavior. Behavior models often capture engineering first principles and may be tied to a variety of products.

Behavior Model Meta-Knowledge The implicit assumptions, and limitations used by the behavior modeling experts in developing the behavior model. The meta-knowledge is the expertise employed by behavioral experts in developing the models

Design Model The specification of the artifact as it should be manufactured. The design model is an idealized version of the “real” or “physical” part.

Fidelity Used to convey the notion of different levels of detail of the models. Higher fidelity models capture more detail than lower fidelity models.

Form Represents physical characteristics and includes aspects such as geometry and material properties.

Function Represents the artifact’s intended behavior. An artifact satisfies engineering requirements through its function.

Idealization To construct an abstracted model of the real system that will admit some form of mathematical analysis. Most frequently, idealization refers specifically to the transformations that are applied to the design representation.

Model A representation of something, as an abstraction of reality.

Simulation Model *see analysis model*

1 PROBLEM CONTEXT & MOTIVATION

The increased complexity in modern engineered products has forced a change in the way in which products are developed. Product development is increasingly becoming a collaborative set of tasks among multidisciplinary, distributed design teams [2]. While the advantages of multidisciplinary, distributed product development include increased quality and decreased time, disadvantages arise in the communication of knowledge and expertise across domains by specialists. These problems are not only present across various domains, such as electrical and mechanical, but also between design and analysis activities throughout the entirety of the product development process.

Product development is iterative by nature and requires consideration of knowledge and expertise from several different domains. Designers, for example, devise product specifications based on required functions. Similarly, analysts simulate the behavior of the resulting product specifications. In a generic product development process (PDP), designs emanating from the creative minds of design engineers are usually sent to analysts for validation. In many cases, analysis models are created based on design specifications and an appropriate set of idealizations or simplifications to make simulation possible and decrease computational expenses. If analytic results indicate unacceptable behavior, the design is sent back to the designer for modification, resulting in an often costly, iterative loop between design and analysis that repeats itself throughout the product realization cycle. Depending on the changes required, such iterations may result in a partial or a complete redesign of the product. In this research, our efforts are focused on a subset of activities in product development, namely *design* and *analysis* activities. In this context, design is the specification of product related parameters and features through synthesi*s* and analysis is the verification of product behavior. Fenves et al. [3] state that designers are responsible for the shape of the artifact and analysts are responsible for ensuring that the behavior of the artifact satisfies the functional needs.

Computer-based design and analysis tools are becoming invaluable assets for product development. Computer-based tools and networks make distributed, collaborative product development possible by enabling product development team members to create and share digital product models from different perspectives and at varying levels of abstraction. Collaboration can take place at different levels ranging from communication of product models via email to real-time collaboration of CAD files. Computer-based design tools, such as Computer-Aided Design (CAD) and systems engineering tools [4] enable engineers to synthesize and create product specification at varying levels of detail and abstraction during product development. Similarly, computer-based analysis tools, such as Finite Element Analysis (FEA) and Computational Fluid Dynamics (CFD), provide increased information and knowledge about the behavior of the systems, enabling engineers to make better decisions throughout product development without having to construct expensive physical prototypes.

While advances in computing performance and functionality have decreased computational cost and in a large part enabled distributed collaborative design, they are still inadequate for supporting the development of complex products by distributed design teams in an efficient and effective manner. Szykman et al. [5] state that sharing Computer-Aided Design (CAD) models across distributed networks is not adequate to support knowledge-based product development because a CAD model can only provide a small subset of the total product-related knowledge. Distributed product development and collaboration has been the focus of research and software development from various aspects.

Integration issues exist between engineering design and analysis activities and associated support tools, thus retarding product development. Specifically, knowledge gaps exist between engineering design and analysis models. The challenges in integrating design and analysis in modern product development are twofold. The first issue is a syntactic issue between engineering design and analysis tools. The disparity in heterogeneous software applications and reliance on proprietary data formats limit sharing product knowledge across tools and organizations. For example, product models created in computer-aided design (CAD) applications cannot be directly used in analysis applications, such as finite element analysis (FEA).

The second issue is a matter of semantics in the domains of engineering design and analysis. Designers and analysts do not share the same knowledge or expertise, thus resulting in knowledge integration issues between the domains. Finn [6] captures the underlying challenge in design-analysis integration as follows:

"It is usually neither feasible nor desirable to analyze all aspects of a physical system. This is because most problems contain complexities that render numerical simulation difficult and redundancies that are unnecessary to analyze. Thus, in practice, certain complexities can be simplified, thereby facilitating more efficient computation, whereas redundancies can be ignored without loss to the integrity of the physical system. The essence of physical modeling is the ability to effect simplifications and remove redundancies without affecting the integrity of the problem or solution. Thus, in a physical modeling task, the major challenges to the engineer are, identifying the various complexities and redundancies in a physical system, applying the appropriate modeling strategies and techniques to simplify or reduce these features."

The research in this report is presented in the context of three areas, (1) the notion of form, function, and behavior of a product; (2) design-analysis integration efforts at the National Institute of Standards and Technology; and (3) synergistic efforts at Georgia Tech from the Multi-Representation Architecture for Design-Analysis Integration. Each of these areas is discussed in further detail in appropriate sections to establish a deeper understanding of design-analysis integration in product development.

2 OBJECTIVES

The primary objective of the research presented in this report is to address integration gaps between engineering design and analysis. The domain of design-analysis integration in the context of product development is an enormous and diverse research area. As such, the focus of this research is the integration of design and analysis models and the complex relationships between these models. The overarching research issue is reducing gaps in design-analysis integration through the development of knowledge-intensive methods and frameworks. The following areas are explored in achieving this goal:

- Formal characterization of behavioral models to facilitate reuse and reduce misuse

- Capturing analysis models and the associated analysis idealizations that relate design models and analysis models

- Development of a repository for storing analysis models of varying complexity and scope to facilitate efficacy and effective reuse during product development

- Embodiment of computer-based frameworks for integrating design and analysis models

The contributions in this work are expected to be refined and enhanced and eventually lead to the development of computer-based frameworks to support design-analysis integration.

3 Frame of Reference

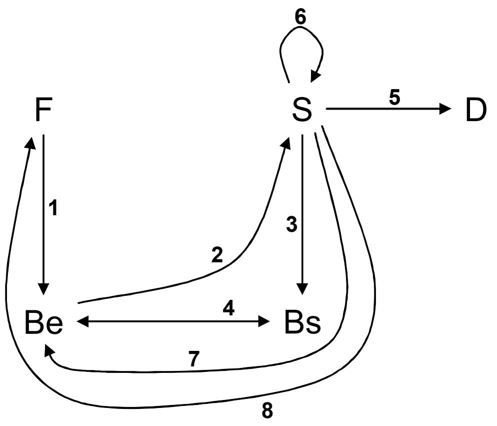
As previously stated, problems associated with design analysis integration include (1) syntactic issues - the disparity and interoperability in design and analysis tools and (2) semantic issues - context-dependent product representation, idealizations and simplifications between engineering design and analysis models [3]. Several approaches for addressing these design and analysis problems include standards-based product representations [7-10], automatic mesh generation and shape modification [11-15], attribution and feature recognition of product models [16-18], and model idealization and simplification to name a few [6, 19-24]. While there has been more than a decade of research effort and technology development, there are still many opportunistic areas for advancement. In this section we highlight and review the current state of design-analysis integration from a tool and application perspective and from a research perspective.

3.1 A Form, Function, Behavior Perspective

Shooter, et al. [25] state that design is a set of activities that operate on the information that describes the product being designed. The result of the design effort is a description, or specification, or what the product looks like, what is it made of, how it functions, etc. [25]. The artifact that is designed can be described in terms of form, function, and behavior. Several researchers have approached product representation from the perspective of form, function, and behavior (FFB), resulting in slightly different definitions of these terms. The FFB framework provides a consistent manner for describing products as several researchers have developed or proposed formal information models for products in this framework. In this research we adhere to the definitions established by Shooter, et al. [25]:

Form	The physical characteristics of the artifact being designed. This includes, among others, its topology, geometry and material properties.
Function	What the artifact is supposed to do. Function is often used synonymously with intended behavior, although some make a distinction between the two.
Behavior	How the artifact implements its function. The behavior of physical systems is governed by engineering principles and is often incorporated into a causal or behavioral model. The behavior model allows designers to explore the satisfaction of function with form.

The definitions from Shooter et al. establish the underlying scope on which the FFB framework is developed. Gero and Kannengiesser [26] state that design can be characterized through three classes, namely: function, behavior, and structure[1]. These variable classes can be linked together through several processes that transform the classes into one another (see Figure 1).



1. Formulation: F→Be
2. Synthesis: Be→S via Bs
3. Analysis: S→Bs
4. Evaluation: Bs↔Be
5. Documentation: S→D
6. Reformulation -1: S→S’
7. Reformulation - 2: S→Be’
8. Reformulation - 3: S→F’ via Be

Where: Be – expected behavior; Bs – behavior derived from structure; D – design description; F – function; S – structure; → - transformation; ↔ - comparison

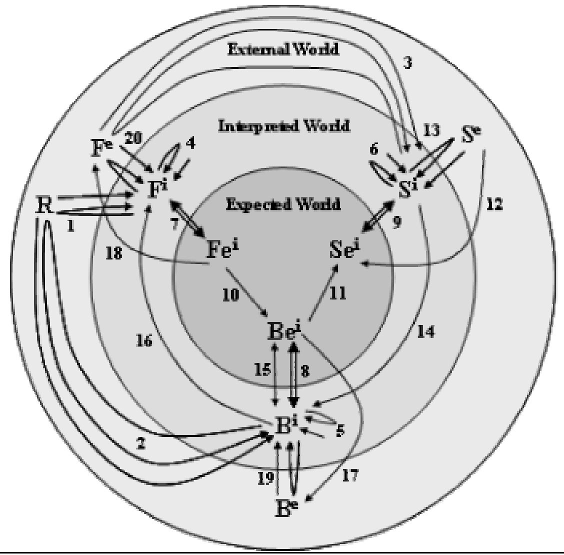
Figure 1. FBS Framework, recreated from [27].

The original FFB Framework proposed by Gero [27] is the extended to represent the idea of context through the inclusion of three “worlds” as follows:

- External World – the physical world in which the product exists. It includes objects external to us.
- Interpreted World – the world that models are built up in minds. It is the interpretation of the part
- Expected World – the world that is used to predict and decide upon

The augmented FFB framework is illustrated in Figure 2.

[1] Gero and Kannengiesser’s definition of structure is equivalent to the definition of form in this work



1. Formulation: 1, (2), (3), 4, 5, 6, 7, 8, 9, 10
2. Synthesis: 11,12
3. Analysis: 14
4. Evaluation: 15
5. Documentation: 12, (17), (18)
6. Reformulation-1: 9, {6}, {13}
7. Reformulation -2: 8, {5}, {14}, {19}
8. Reformulation – 3: 7, {4}, {16}. {20}

Figure 2. The situated FFB Framework [26].

The variable states in Figure 2 are associated through several processes or transformations. The relationships between design variables and processes are the following:

Of particular interest in the FFB framework developed by [26, 27] are the synthesis, analysis, and evaluation processes from Figure 1 and Figure 2. These processes are defined in the context of integrating design and analysis models in the design process.

Synthesis – creation of design related information. The desired function and expected behavior of the design is transformed into product form (structure) through anticipated behavior based on product structure.

Analysis - simulating the behavior of the product based on the form. In the context of this research, behavior is determined using behavior and analysis models. Tools used include equation-based models and finite element analysis based numerical models.

Evaluation – the comparison between expected behavior and actual product behavior.

Chang et al. [28] establish the notion of viewing the designed artifact from different perspectives. A perspective is defined as *a model of a design that must be built, reconciled with new information, and revised throughout the design process.* A perspective is created through a complex mapping of syntax, semantics, and parameter set from the master design representation.

Balazs and Brown[29] discuss form, function and behavior in the context of different types of knowledge in product development. They classify several types of knowledge including:

Structural knowledge – knowledge about components which comprise the object their relationships

Behavioral knowledge – knowledge about the behavior of the object, i.e. about ways the object responds to changes in environment or state

Functional knowledge – knowledge about how the behavior of an object is used to accomplish its intended use

Iwasaki and Chandrasekaran [30] propose a framework that used product form knowledge and functional knowledge to verify the design. They state that simulation of product behavior is important in determining whether the desired function of the product is achieved. They focus on answering the question when the predicted behavior of the system achieves the desired functionality.

In this research, we build on contributions of FFB in the context of design-analysis integration. The FFB framework proposed includes the notion of design and analysis form. The FFB framework proposed in this research is illustrated in Figure 3.

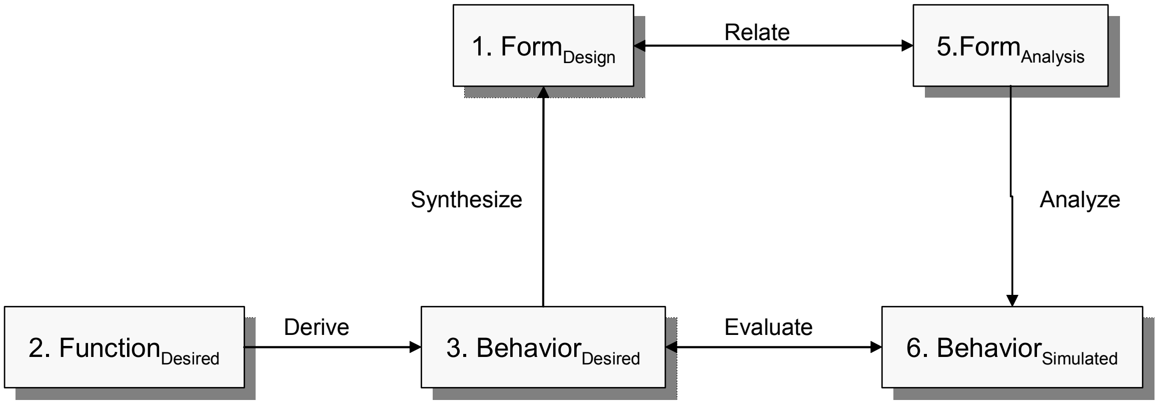


Figure 3. FFB Framework in the context of Design-Analysis Integration.

The FFB framework presented in Figure 5 is comprise of six (6) product artifact representation and seven relationships between these representation.

Form$_{Design}$ The design form captures the geometry, topology, and material that represents the artifact that is produced. The physical product is produced from the specification of the design form. Manufacturing, assembly, and production processes can be driven from the design form. The design form is the primary representation used for manufacturing the product.

Form$_{Analysis}$ The analysis form is the form utilized to complete analysis of the product. This form may be identical to the design form or may be a simplified representation of the design form. A single design form can have multiple analysis forms related to it. Analysis form is a specialized aspect view of design form. Analysis form is for predicting and simulating the behavior the product. Analysis form and design form must be kept consistent.

Function$_{Desired}$ This indicates what the product is supposed to do. It represents the artifact's intended behavior. An artifact satisfies engineering requirements through its function.

$\text{Behavior}_{\text{Desired}}$ The desired behavior is derived from the function, based on engineering expertise. This relationship is beyond the scope of the research. This is identical to Gero's intended behavior.

$\text{Behavior}_{\text{Simulated}}$ The behavior of the product as simulated, based on $\text{Form}_{\text{Analysis}}$

The form, function, and behavior of the product are related through the following processes:

Derive Establish a relationship between product functionality and desired behavior. A behavior can have multiple functionalities.

Synthesize Create a $\text{Form}_{\text{Design}}$ based on the $\text{Behavior}_{\text{desired}}$ of the product. This process is completed during product development and design

Evaluate Determine whether the $\text{Behavior}_{\text{Simulated}}$ agrees with the $\text{Behavior}_{\text{Deisred}}$ for the product.

Relate The $\text{Form}_{\text{Design}}$ and $\text{Form}_{\text{Analysis}}$ must be related. The product is ultimately produced based on the design form, but decisions about the design form are based on analysis driven by $\text{Form}_{\text{Analysis}}$.

Analyze Execute the analysis model in order to gain information and knowledge about the behavior of the object based on $\text{Form}_{\text{Analysis}}$.

A key difference in the FFB framework proposed in this work with the previous frameworks is the explicit representation of multiple analysis forms for a single design form and the verification between simulated behavior and desired behavior based on analysis form. This small detail is important in addressing integration issues between design and analysis because several forms, used in different domains during product development, represent the same underlying product.

Numerous researchers have addressed the notion of multiple design perspectives throughout the product development process. A comprehensive survey is presented by Mocko et al. [31]. In the following sections, several research areas are presented in the area of design-analysis integration.

3.2 Design-Analysis Integration Efforts at NIST

Fenves et al. identify that a core issue associated with product development is the gap between engineering designers and analysts [3]. To illustrate their point several interaction scenarios between design and analysis activities are presented. A simplified model, based on the interaction scenarios is presented in Figure 4. Figure 4 illustrates how engineering designers interact with engineering analysts through knowledge-intensive *idealization* and *mapping* process. In this process designers specify how the design parameters are related to analysis parameters and what simulation models are used. Similarly, after the simulation is completed analysts map the parameters back to design for appropriate changes to be made. Designers and analysts are forced to work closely together to ensure that design and analysis parameters are appropriately related between design and analysis activities and that the appropriate analysis models are use for the design situations.

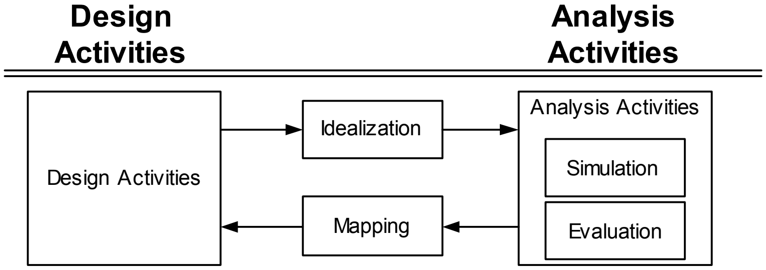


Figure 4. Integration of design and analysis activities.

While the interaction between design and analysis is presented in a simplified representation in Figure 4, there are many complex problems associated with integrating design and analysis. The interaction scenarios developed in [3] are described below. The scenarios depict the typical interaction between design and analysis activities (see Figure 4, Figure 5 and Figure 6).

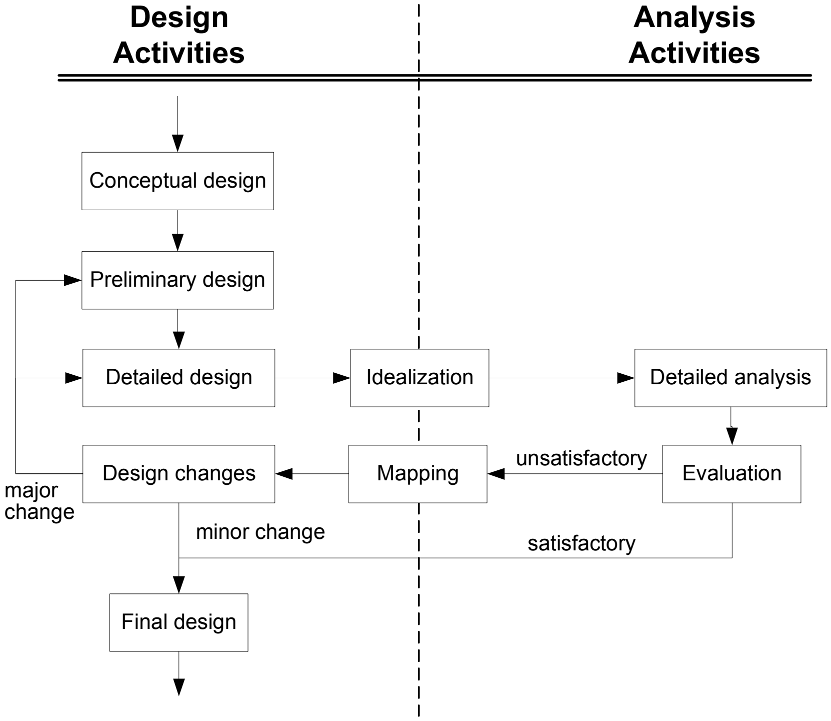


Figure 5. Retroactive analysis scenario [3].

The interaction between engineering designers and analysts in Figure 5 is referred to as *Retroactive Analysis*. The artifact is designed based on the designer's knowledge. The design is then validated at the completion of detailed analysis by analysts. Often times, Retroactive Analysis results in over design and long realization cycle times.

The second scenario is termed *Integrated Spatial-Functional Design* because design decisions are supported by analysis knowledge throughout the major phases of the realization process. The design is analyzed at the completion of each phase during product development. The artifact is analyzed at varying levels of detail from conceptual design to detail design to support engineering decisions [3]. The integrated approach relies on analysis to be performed at various levels of detail throughout the design process (see Figure 4)

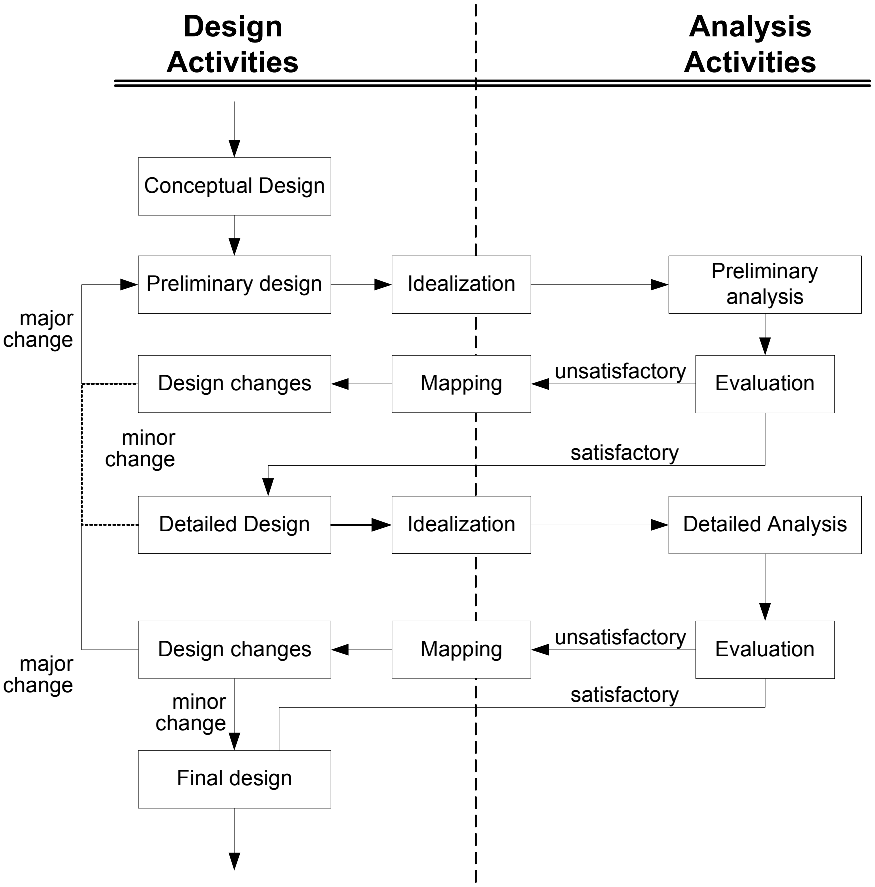


Figure 6. Integrated spatial and functional design scenario [3].

In both scenarios, engineering analysis is performed to support design decisions, most notably decisions that affect the product form. A key difference between the retroactive and integrated scenarios is the frequency at which analysis is completed during product development. In the Integrated Spatial and Functional design scenario, the behavior of the product is simulated more frequently, thus enabling designers to explore the design space and verify the behavior of the product more readily. However, a similar problem is present in both scenarios, namely the knowledge shared between product design and analysis from diverse domains which is often not shared efficiently between designers and analysts. This problem is elevated when considering the computer-based tools used in design and analysis. The disparity in commercially available tools contributes to integration problem between designers and analysts.

The product specifications created by designers using design tools are often not directly used by analysts using analysis tools. For example, the geometry of the product as specified in CAD tools can often not be used in FEA applications. Add to the fact that product geometry is often idealized from design to analysis resulting in additional integration problems.

To address these needs and problems, several information models have been proposed. The NIST Core Product Model(CPM) [4] serves as a conceptual product model for capturing form, function, and behavior about complex product (see Figure 7 and Figure 8).

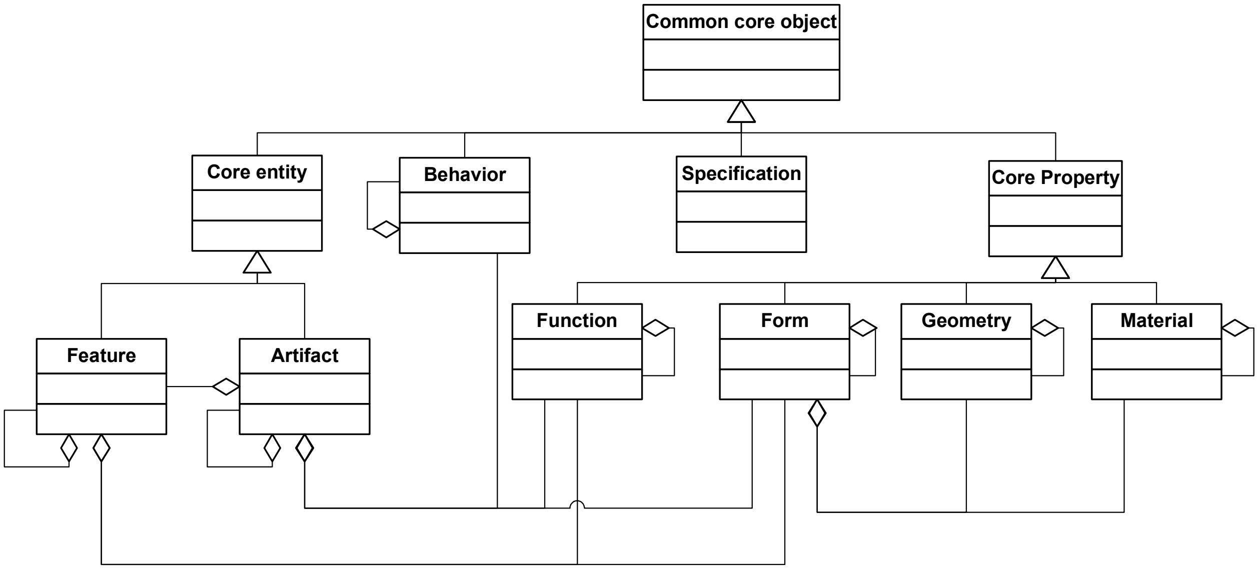


Figure 7. Entity Classes in the Core Product Model [4].

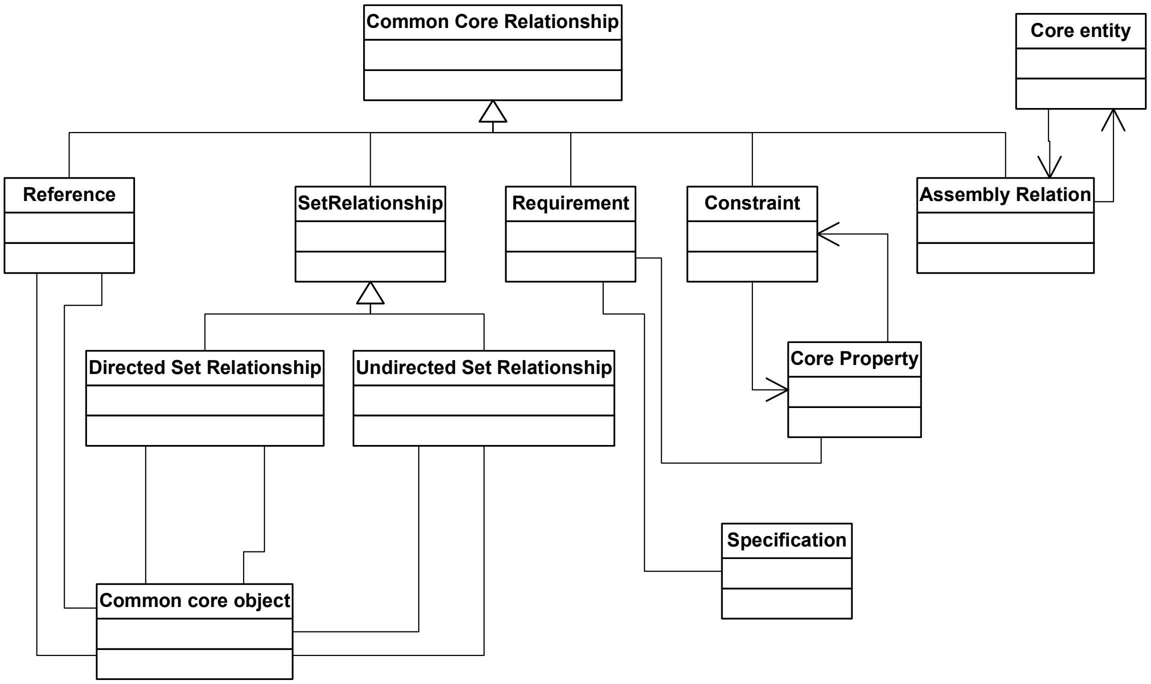


Figure 8. Relationship Classes in the Core Product Model [4].

The CPM serves as a conceptual model that is generic and able to capture many aspects of product development. Several complimentary product models have been developed to support specific aspects of product development including: Open Assembly Model (OAM), Product Family Evolution Model (PFEM), and the Design-Analysis Integration Model (DAIM) [32].

A more recent effort at NIST, involving information and knowledge capture over product lifecycle to support knowledge-intensive design, is the development of Product Lifecycle Management (PLM) technologies. PLM enabling technologies are next generation technologies aimed in part at extending the core functionality of PDM systems. Fenves et al. [32] present a conceptual framework for capturing product knowledge over the life of the product. The authors assert that two limitations are present in current PLM technologies. The first limitation deals with design changes and rationale throughout the product development process. This limitation is not within the scope of this work. The second limitation is that most PLM systems and enabling technologies are primarily focused on the form representation of the product.

Of particular interest in this work is DAIM for capturing tighter integration between design and analysis [32]

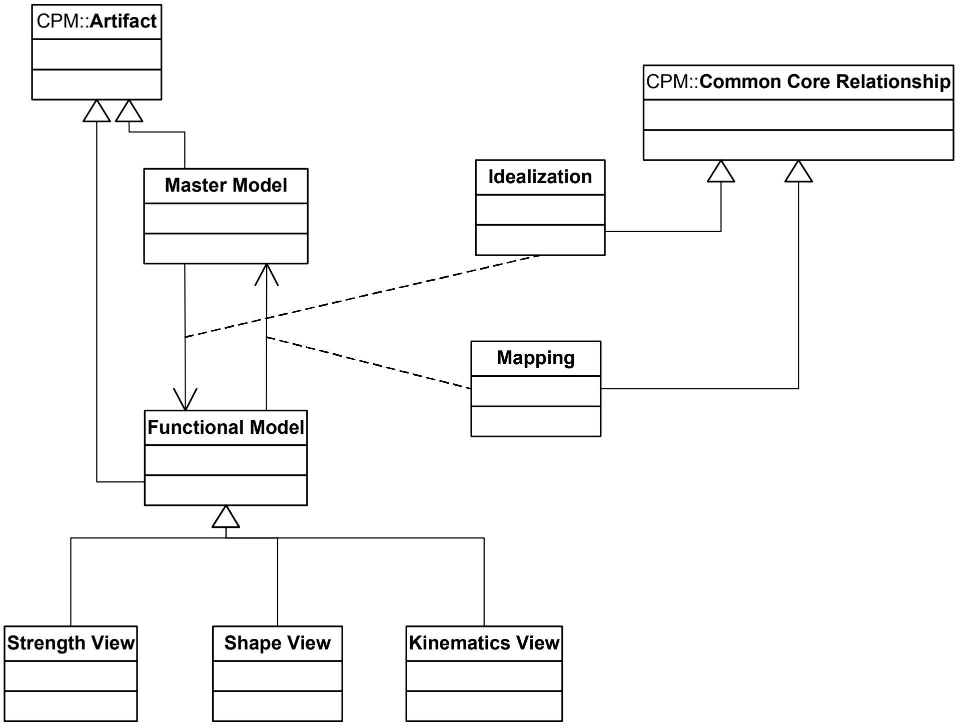


Figure 9. Design Analysis Integration Model (DAIM) [32].

3.3 The Multi-Representation Architecture for Design-Analysis Integration

The multi-representation architecture (MRA) is presented in the context of the information gaps between traditional design (CAD) and analysis (CAE) tools. The MRA is aimed at satisfying the needs in the links between CAD and CAE: (1) automation of ubiquitous analyses; (2) representation of design and analysis associativity and of the relationships among models; and (3) provision of various analysis models throughout the life cycle of the product [33]. The initial focus of the MRA has been on ubiquitous analysis. Ubiquitous analyses are those analyses that are regularly used to support the design of a product [34]. The MRA supports capturing knowledge and expertise for routine analyses through semantically-rich information models and the explicit associations between design and analysis models. While the MRA captures routine analysis and the mapping between design parameters and analysis parameters, there is still the opportunity for misuse of the analysis templates. The assumptions, variable definitions, and application context are not explicitly captured. Reliance on appropriate use of the model is based on designers' understanding of the model or communication with the analysis model developer. The MRA supports capturing knowledge and expertise for routine analyses through semantically-rich information models and the explicit associations between design and analysis models. The MRA attempts to bridge the gap between design and analysis based on four building blocks (see Figure 10).

The focus of this research does not directly extend current research efforts, but rather builds on the contributions of Composable Simulation and MRA. These research efforts establish the motivation and set the tone for creating and capturing reusable analysis and behavioral models to support simulation in engineering design.

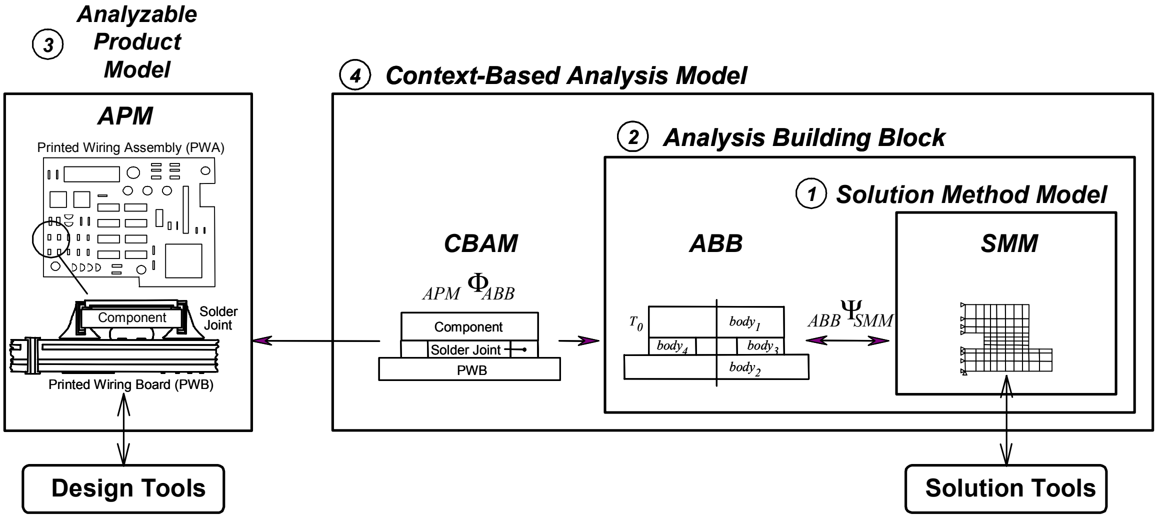


Figure 10. The Multi-Representation Architecture [35].

The Solution Method Models (SMM) represent low-level, solution-specific methods. SMMs combine inputs, output, and control for a single type of analysis solution. The SMM is a wrapper of the necessary information to complete an analysis solution. The SMMs serve as tool agents to provide what solution tool to use, the inputs to the tool, the control for each tool, and to retrieve the results from each 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. The ABB structure represents the information template to define relations.

Analyzable Product Models (APM) represent all data associated with the product over its life cycle. The APM model, as used in the MRA, represents analysis-oriented information. Items such as geometry, loading conditions, and boundary conditions are included in the APM. The

MRA extends the APM beyond the detailed manufacturing description of the product. Analysis models are created, often times, based on idealizations and simplifications of the model. Such idealizations are captured in a reusable sense for the APM.

Context-Based Analysis Models (CBAMs) contain the linkages between the APM and the ABBs. CBAMs connect APM to product independent ABBs to solve specific analysis problems. A major focus of the CBAM analysis models is routine analysis, or regular use of specific analysis in the product design.

The MRA is realized through the development of the constrained objects (COBs) knowledge representation [36]. The COB representation is based on objects and constraint graph concepts. Constrained Objects are used to represent ABBs, APMs, and CBAMs. COBs support the following:

- various modeling forms- computer and human interpretable lexical and graphical forms
- object constructs – sub/supertypes, inheritance, multi-fidelity
- multi-directional inputs and outputs
- adaptability, reusability, modularity, and semantic richness

3.4 Standards-Based Approaches for Design-Analysis Integration

Standards-based product representation development efforts, such as eXtensible Markup Language (XML) and ISO10303-STEP, have enabled product models to be shared amongst heterogeneous software applications in engineering design and analysis through the development of common product models [7-10, 37]. For example, STEP AP209 addresses interoperability of product models between CAD and FEA applications, thus enabling closer integration of design models and analysis models. While AP209 addresses interoperability issues between diverse software tools, it does not capture the knowledge-intensive approach of idealizations between models. Lubell et al [38] survey the complimentary technology of XML, STEP, and UML. In their paper, they develop the need for open standards and the use of this complimentary technology. A discussion on each standard and the need to integrate them and leverage from them to support interoperability in product development is presented.

3.5 Artificial Intelligence in Design-Analysis Integration

There has been a substantial research effort in design-analysis integration based in artificial intelligence (AI) techniques for generating simplified analysis models. Finn et al. [23, 24, 39] asserts that while advances in numerical simulation have become an invaluable asset in engineering analysis, the task of creating appropriate simplified physical models of the product remains a key problem. Finn uses artificial intelligence (AI) techniques, such as rule-based systems and model-based reasoning, to capture idealizations and simplifications employed in creating physical models. These idealizations and simplification are often based on first principles, approximations, and heuristics.

Similarly, Armstrong, et al. [14, 15] present the idea of *a priori* knowledge and *a posteriori* analysis of the simulation results to make the appropriate idealizations. Operations, such as medial-axis transform, dimensional reduction, and feature removal are used to create analysis models. An AI-based framework is developed to support the automatic creation of analysis models using idealizations.

Finally, Shephard et al. [19-21] propose a method and software framework for automating the idealization process in developing simulation models. The proposed method is based on “goals and strategies” using a series of knowledge-based systems and analysis applications to generate idealized analysis models.

3.6 Design Repositories to Support Product Realization

Design repositories are a knowledge-based approach for supporting engineering design. Design repositories not only capture *what* is designed, but also *how* and *why* the product is designed. Design repositories enable engineers to capture the evolutionary nature of product knowledge and information throughout the design process [5]. There are several examples of design repository research to support product development. Grosse [40] presents an ontological design repository for capturing knowledge about behavioral models in engineering. Similarly, research in [41, 42] propose the development of design repositories for formally characterizing analysis model in design.

Szykman, et al. [5] make a clear distinction between design repositories and traditional engineering databases. The authors state that design repositories capture additional information and knowledge about the design representation, may contain heterogeneous information including formal data models, text, design rules, and additional types of information, and may include additional functionality for simulation of behavior, composition of systems. We believe the actual difference between design repositories and databases is absent at the implementation level.

Design repositories are not prolific in engineering design. In contrast, engineering tools and technologies such as computer aided design (CAD), finite element analysis (FEA), and product data management (PDM) are in widespread usage. However, only a subset of the entire product knowledge is captured by current tools. For example, the knowledge captured by traditional CAD systems is typically limited to geometric shape and related knowledge such as constraints and parameterization [5].

4 PRIMARY RESEARCH QUESTIONS, TASKS, HYPOTHESES, AND APPROACH

The research questions addressed in this research have been refined since the initial proposal of the research. As additional knowledge was gained and exploratory studies were realized, research questions were augmented to reflect a more focused research contribution. While the supporting research questions have been refined, the primary research question has remained consistent throughout the entirety of the research effort:

Primary Research Question*: How can designers' and analysts' knowledge be captured and reused throughout product development to support multi-disciplinary engineering design-analysis model associativity?*

Overarching Focus*: There has been a large effort to capture knowledge within a particular product domain. For example, mature technologies and method exists in domains of behavior* *models and establishing product form. However, research gaps exists in integrating the domains. Sufficient knowledge is not captured between domain, such as design and analysis, to meet the needs of modern product development processes.*

Answering the primary research question is a monumental task and is not fully addressed in this work. As such several supporting research questions are identified that focus on specific issues within the larger picture of design-analysis integration. Like most research, these supporting research questions have been refined as work has progressed and specific problems have been identified. The evolution of the supporting research questions from the initial proposal to final report are summarized in Table 1.

Table 1. Research question from proposal and status report.

Research Questions from Initial Proposal (September 23, 2002)
• How can analysis models be generated/accessed/created based on previous design knowledge? • What types of information, at various levels of detail, are available to aid the engineer in the design process? • Can design repositories and design databases be used to aid the functional representation of technical products in order to recommend appropriate working principles/conceptual design configurations? • How can different aspect/domain models be linked to the main product model in a consistent manner? • What types of formal representation and methods can be leveraged to aid in knowledge-based design? • How can standards be developed, used, or leveraged to support product and design process modeling?
Research Questions from Status Report (March 10, 2004)
• How do we capture the rationale and intent for creation of analysis models? • Can we develop a hierarchy that trickles the rational and intent from the model level to the level of individual associativities and mappings amongst models? • Can we capture the rationale for idealization and then verify if the subject problem needs to be idealized? In other words, how do we decide what to idealize and what not? • How can the idealizations in a particular domain be formally represented and support the creation of appropriate analysis models? • How can design models and behavior models (capturing intended functionality) be used to form appropriate analysis models?
Final Research Questions
• How can behavioral models be captured to support reuse in product development? • How can the idealizations employed by analysis experts be formally captured to facilitate automated creation of simplified analysis models based on engineering design specifications? • How can simplified analysis models used throughout product development be archived for reuse? • How can analysis models be efficiently retrieved to support model-based engineering decision making in product development?

The *Final Research Questions* summarized in Table 1 are expanded upon in the following text. The final questions are presented in the vein of the original research questions, but reflect a more refined understanding and focus on the problems associated with integrating design and analysis models. The supporting research questions follow:

Research Question 1: How can behavioral models be captured to support reuse in product development?

Hypothesis 1: Engineering behavioral models serve as the models for simulating and predicting the behavior of a product. Behavior models must be characterized to facilitate usage across product domains. Additionally, because behavioral models are developed based on engineering first principles, and empirical data, the behavior of many different types of products can be predicted with the same behavioral model using alternate product mappings.

Research Question 2: How can the idealizations employed by analysis experts be formally captured to facilitate automated creation of simplified analysis models based on engineering design specifications?

Hypothesis 2: The idealizations and simplifications used by analysis experts in mapping design features and characteristics to the analysis representation models can be formally captured using information and knowledge modeling representations. Once formally described and captured, the idealization can be leveraged using knowledge-based approaches, such as expert systems and ontologies, to facilitate closer relationships between design and analysis representations.

Research Question 3: How can simplified analysis models used throughout product development be archived for reuse?

Hypothesis 3: Engineering design is a model-driven activity in which various analysis models are used to support engineering decision making through validation of product behavior. Analysis models can be characterized based on metrics such as complexity, time for execution, accuracy of results, underlying modeling assumptions, and formalized idealizations from Research Question 1 (RQ1). Formal characterization of analysis models will enable rapid and efficient reuse.

Research Question 4: How can analysis models be efficiently retrieved to support model-based engineering decision making in product development?

Hypothesis 4: Analysis models can be characterized and archived based on developments in RQ1 – these analysis models can be searched and retrieved using a hyperspace repository approach. A multi-dimensional model repository will enable engineers to access and reuse engineering analysis models during product development.

Towards developing answers to the above-highlighted research questions, several research tasks are proposed. The tasks proposed this effort are as follows:

Table 2. Research tasks proposed in this work (September 23, 2002).

Task 3	Identify analysis activities in the design process and what type of information is needed to complete each type of analysis. At first glance, these types of analysis should be considered: cost analysis, finite element analysis, and equation-based physical analysis.
Task 4	Complete background research on case-based reasoning and knowledge-based design support. The information captured in the CPM should be used to aid in the generation of future designs. The design environment should support smart queries of completed designs at various stages of the design process. The queries should support search criteria in the areas of function, form, analysis type, etc.
Task 5	Develop a formulation of how the CPM supports the associativity between design models and analysis models for knowledge based and case-based design.

The tasks are answered in varying degrees of completion based on a literature survey and the following case studies. The focus of this research is on knowledge capture in engineering design and analysis. The core product model (CPM) and subsequent information models set the stage for this research. In the following sections, several example cases and research areas help to complete the above tasks and address the research questions.

5 Example Cases and Test Bed Approach in DAI Research

In the course of answering the proposed research questions, several examples and research developments are completed. In the sections that follow an overview of each example case/research area is presented. The examples help us to explore and develop answers to the proposed research questions, but also help us to formulate additional research directions. A summary of the examples and the relevance in achieving a tighter integration between design and analysis is provided in Table 3.

Table 3. Test case and research areas explored in this research.

Example Case/Thrust Area	Relevance in DAI Research
Printed wiring board warpage analysis	An overall picture of DAI is presented in a computer-based framework. The warpage analysis uses standard product models (STEP) and analysis idealizations to integrate design and analysis models.
Behavioral model repository	Established a formalized information model for archiving behavioral models. The knowledge representation enables the behavioral knowledge and meta-knowledge to be captured and utilized throughout product development. The proposed behavioral model knowledge representation serves as the precursor to capturing analysis models
Linear cellular alloy analysis model idealization	Provides an example of analysis idealization and analysis models in design. The research area highlights a conceptual architecture in which analysis models and associated analysis idealizations are archived for reuse in product development.
Electronic chip package analysis	An integration framework for electronic chip package design and analysis is presented. The example case illustrates the process followed for integrating design and analysis models and demonstrates a software framework for integrating geometric models and finite element analysis models.

The example cases and research areas fall within the scope of three main product models in design-analysis integration, namely.

Analysis Model A model that is often an idealized representation of the design model used in analysis. Analysis models include behavior models and the mapping of product design parameters, similar to the CBAM in the MRA.

Behavior Model A model that captures the mathematical description of the physical behavior. Behavior model often capture engineering first principles and may be tied to a variety of products, similar to the ABB in the MRA.

Design Model The specification of the artifact as it should be manufactured. The design model is an idealized version of the "real" or "physical" part, similar to the APM in the MRA.

These models are related through the following processes/ information transformations.

Search/query for behavioral models

Behavioral models are selected that represent and capture the physical behavior of the system. This is a complex task that may also include composing a system of behavioral models to represent the object. In the context of MRA, these behavioral models are analogous to ABB models.

Identify product parameters

For each different type of behavior simulated, the product model may contain different parameters. For example, a structural analysis will require a different set of parameters than thermal analysis. The product model is processed and these parameters are identified. It is important to note the parameters identified are coupled to the behavioral models in the repository. This is analogous to creating the APM in the MRA

Select behavioral models

Behavioral models are selected that *best* describe the interest. Behavioral models may be selected based analysis domain of interest, accuracy of results desired, or constraint on computation effort to name a few.

Map product parameters and behavioral parameters

The product parameters must be mapped to the behavioral models selected. This process may be completed using complex engineering idealization or simplification.

A graphical illustration of research scope is shown in Figure 11.

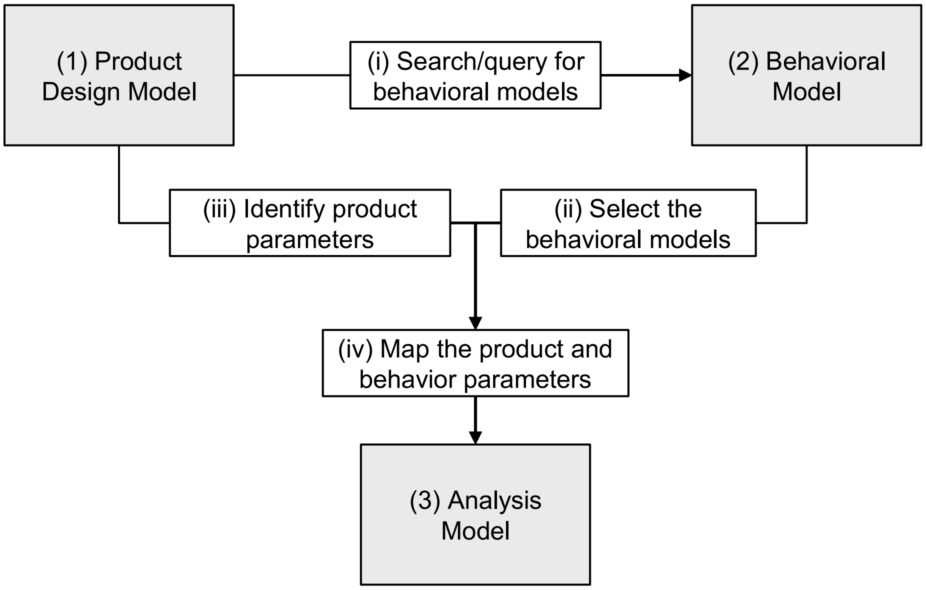

(1) Product Design Model
(i) Search/query for behavioral models
(2) Behavioral Model
(iii) Identify product parameters
(ii) Select the behavioral models
(iv) Map the product and behavior parameters
(3) Analysis Model

Figure 11. Illustrative map of DAI research presented in this report.

In the following sections, a detailed description of each example is presented.

5.1 Example Case 1: Printed Wiring Board Warpage [43]

In this example, we attempt to formulate design analysis integration issues in a complex analysis problem. We use the example of warpage analysis of printed circuit boards. With reference to Figure 12, we traverse the complete scope of design analysis integration activities. Warpage worthiness evaluation of PWBs commences with a standards-based product design model which in this case is the STEP AP210-based circuit board design. In this example, the printed circuit board is idealized to behave as a layered shell. Further, the printed circuit assembly is idealized as a grid outlined by the shape of the board and where each cell in the grid has effective material properties of that particular region in the assembly. Based on this, a suitable analysis model is created (Figure 12).

The activities in the design-analysis integration bridge are embedded in an automation framework, known as the *PCB Warpage Analysis Framework* that we shall explain later in more details.

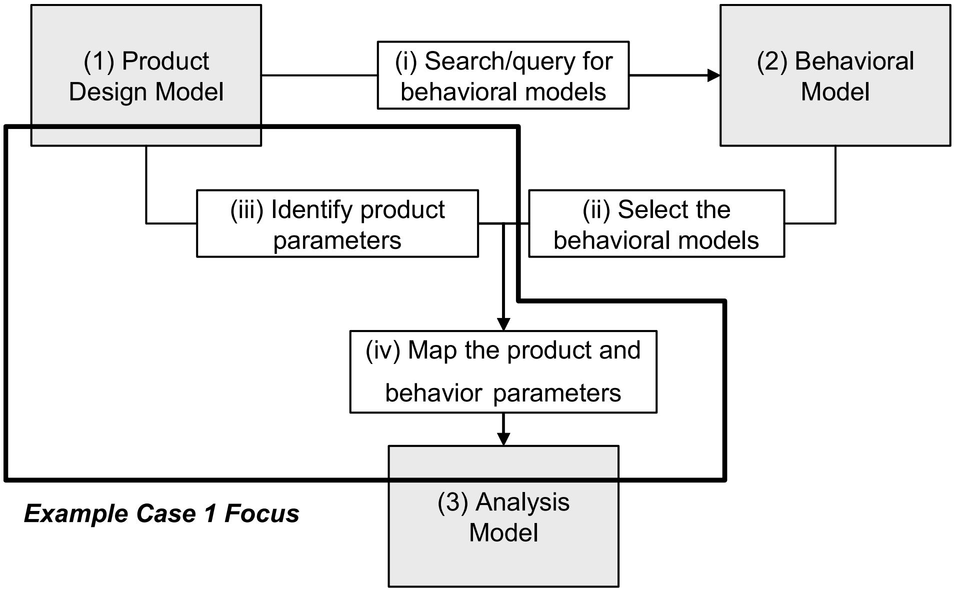

(1) Product Design Model
(i) Search/query for behavioral models
(2) Behavioral Model
(iii) Identify product parameters
(ii) Select the behavioral models
(iv) Map the product and behavior parameters
(3) Analysis Model
Example Case 1 Focus

Figure 12. Focus of Example Case 1 - PWB warpage analysis in DAI effort.

5.1.1 Introduction – Warpage Basics

Warpage is the out-of-plane deformation of an artifact. Thermally induced warpage is the warpage of the artifact when it is subjected to thermal loading, caused by differential thermo-mechanical properties of elements composing the artifact. The subject of our current analysis is thermally induced warpage of printed wiring board / assembly (PWA/B), which is critical for managing the manufacturing yield and reliability of electronic devices. Figure 13 shows warpage measurement scenarios for one and two dimensional multi-material stackup, subjected to a uniform temperature change.

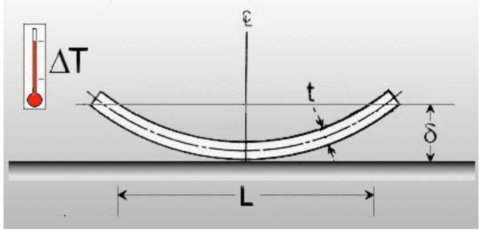
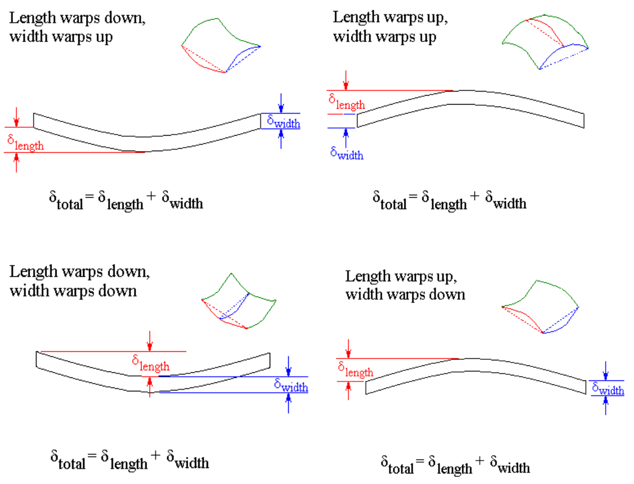
 $\delta = (\alpha_b \text{ L2 } \Delta T) / t$ where **L**: Undeformed Length; **t**: Undeformed Thickness; **ΔT**: Temperature Change; α_b: Specific Coefficient of Thermal Bending	 Length warps down, width warps up δ_{length} δ_{width} $\delta_{total} = \delta_{length} + \delta_{width}$ Length warps up, width warps up δ_{length} δ_{width} $\delta_{total} = \delta_{length} + \delta_{width}$ Length warps down, width warps down δ_{length} δ_{width} $\delta_{total} = \delta_{length} + \delta_{width}$ Length warps up, width warps down δ_{length} δ_{width} $\delta_{total} = \delta_{length} + \delta_{width}$
a. Warpage of a 1D linear element due to differential thermal expansion of multi-metallic strips	**b. Warpage of planar elements**

Figure 13. Warpage (δ) – Basic measurement scenarios for linear and planar elements [44].

Warpage is a critical concern for electronic manufacturing industries. During the assembly of electronic components, the circuit board is exposed to different degrees of thermal treatment. Also, with each step in the assembly process, the response of the board to the thermal treatment changes due to additional material that has been added. Warped board surfaces are unfavorable for seating components on the board. This results in weakening of component attachments to the board and in most severe cases detachment of components from the board.

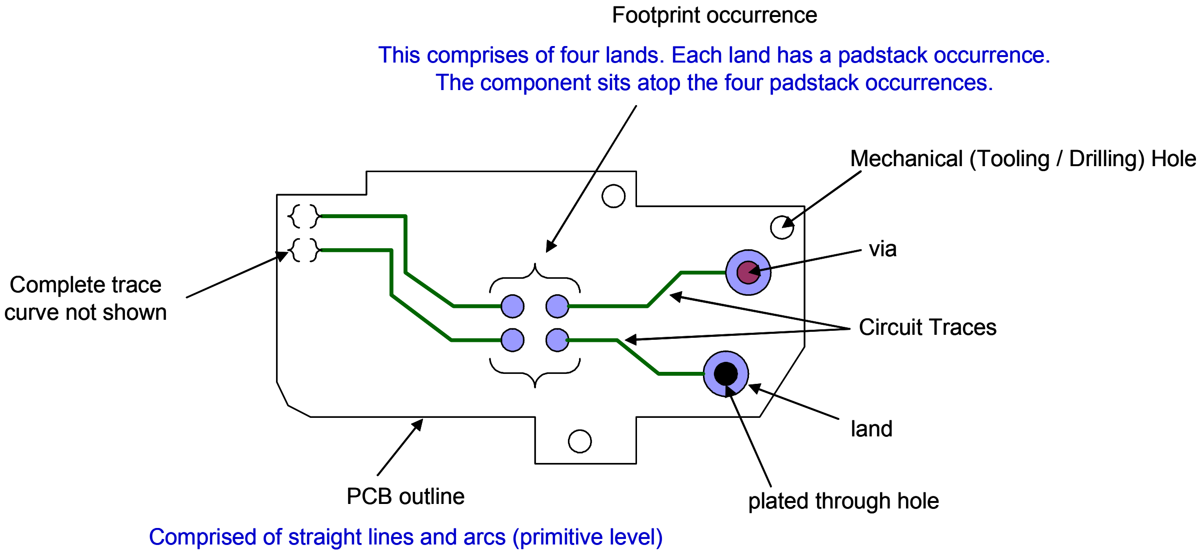
There have been widespread empirical efforts to limit warpage of boards during the assembly process. But a unified methodology, dealing with the diversity of electronic products and thermal treatments during manufacturing, to predict warpage worthiness during the design phase is absent. As a result, designers are often limited to manual, back-of-the-envelope calculations which are not sufficient for the purpose.

Hence, there is a need for a disciplined and well-organized effort to evaluate the warpage worthiness of circuit boards. Further, the evaluation needs to be an ongoing process parallel to the design of the electronic component. Results from the evaluation would then designers to take decisions on corrective measure like relocating components, rerouting traces etc

5.1.2 Measuring Warpage – Can we predict warpage worthiness?

In this section, we shall focus on measuring warpage. Advanced and robust techniques for experimental measurement of warpage have evolved in the past [44-46]. However, a comprehensive effort to predict warpage of in-design electronic components is absent. In light of this need, NIST and Georgia Tech in collaboration with LKSoft and Rockwell Collins initiated efforts to develop a methodology and embed it in a warpage prediction tool to aid PCA designers.

Figure 14 shows features on a printed circuit assembly that are of interest to warpage prediction.


Footprint occurrence
This comprises of four lands. Each land has a padstack occurrence.
The component sits atop the four padstack occurrences.
Mechanical (Tooling / Drilling) Hole
via
Complete trace
curve not shown
Circuit Traces
land
PCB outline
plated through hole
Comprised of straight lines and arcs (primitive level)

This figure shows the top view of a PCA without the components ➔ bare PCB with features (mechanical and electrical).

This figure captures the top two layers of the PCB. The top layer with the metallization (traces, lands, etc.) and the adjacent dielectric layer (area show above within the PCB outline and not occupied by any metallization)

Figure 14. A typical PCA with metallization features that affect warpage.

A typical PWA consists of metallization features such as lands (for electrical connection), traces (for intra-layer circuitry), plated through holes and vias (for inter-layer connection), etc. Also, a circuit board is a stackup of alternating layers of dielectric material (like FR4, providing insulation) and metallization layers.

Increasingly, local warpage, e.g. warpage in the region of a critical component footprint, is a more critical issue than global warpage, the warpage of the PWB as a whole [44]. Changes in the contour of the component footprint can create shorts or opens in the PWB-component solder joints during reflow soldering or build stresses into the assembly that appear later as reliability problems.

Simulation of local warpage must consider key local features such as conductive traces, vias, tooling holes, etc. yet simulation models cannot effectively include these details fully for typical PWA/Bs. Thus, efficient transfer and appropriate idealization of design data from proprietary electrical and mechanical CAD formats to model-soluble form is required.

We now reflect on the nature of models and the methodology to build an architecture that would enable us to achieve the aforementioned objectives.

5.1.3 Conceptual Architecture of PWB Warpage Analysis Framework (PWAF)

We perceive the need for a conceptual architecture, as an aggregation of well connected and high fidelity information models, for the physics-based representation of PWB designs and the analysis of their warpage worthiness. Figure 15 shows the multi-representation architecture [33, 47] view of the subject framework. The PWAF is comprised of five model types. First, we describe the need and the nature of each of these model types.

Manufacturable Product Model (MPM)

The manufacturable product model, as the name implies, consists of detailed information concerning the PWA and the associated PWB at the level of richness at which they can be manufactured. Design information is the basic need for any analysis that can be performed at different stages of the life cycle of an electronics product (a PWB in this case). Traditional ECAD tools can represent only a share of this complete information pool for a certain range of electronics products and with varying semantics across them. To answer the needs for intelligent representation of electronics product design information and its usage, we use the STEP AP210 standard (ISO 10303-210) [44]. The manufacturable product model is based on AP210 and shall be referred here on as a 210-based MPM. The 210-based MPM is a higher fidelity product model as compared to the set of information sub-pools that are used to populate it (ECAD tools, auxiliary sources like component databases, etc.). The term "fidelity" here refers to the extent of information coverage and the ability to exchange information with both downstream and upstream information models in the PWAF. Being a standards-based model, the MPM has a greater neutrality with respect to an increasing number of ECAD tools that can generate models conforming to the standard [44].

Analyzable Product Model (APM)

The APM [47] is an analyzable view of an MPM. Only a subset of the information contained in an MPM might be needed for the contexts in which the electronic product needs to be analyzed. For this reason, a view (model in this case) containing the relevant information is generated from the MPM. The APM shall be explained in more depth while discussing the development of the PWAF.

Analysis Building Block (ABB) System Model

The ABB [47] system model is an information model that contains the idealized view of the APM for the purpose of the subject analysis. ABBs are product independent analytical objects (e.g. representing physics-based concepts like continuum mechanics bodies and idealized material behavior properties) and ABB systems are made up of ABBs. As a reusable concept, an ABB system may itself serve as an ABB in the hierarchy.

Context-based Analysis Model (CBAM)

A CBAM [47] enables us to map the behavior to the ABB system model for a particlar kind of analysis and fidelity (e.g. 2D PWB warpage analysis). The physical structure of the actual electronic product is usually complex. It might be sufficient to analyze (within the limits of significance) a much simpler structure by eliminating non-contributing features, calculating effective material properties for some constituents, etc. This simpler structure is the idealized behavior model for the analysis template as represented by the ABB system model. CBAMs capture the knowledge and the decision to derive a particular ABB system model, for analyzing the APM in a given context. Rightly so, the "context" is the identifying entity in a CBAM. As an example, for thermo-mechanical warpage analysis of a given PWB, we might select a set of plane-stress ABBs or 3D continuum ABBs, depending upon the nature of the loading and the structure of the product itself, to build the ABB system. In this case, the thermal loading and the structure of the electronic product define the "context". The CBAM captures the actual idealizations that are needed to generate the ABB system model from the APM, in this context. A given product can have multiple CBAMs (product-specific analysis templates) for other fidelities and behaviors.

Solution Method Model (SMM)

The SMM [47] contains information pertaining to the specific solution strategy applied to the ABB system model. For example, an ABB system model can be developed into a finite element-based SMM or a finite difference-based SMM for solution purposes. The information contained in the SMM is in a ready-for-interpretation state by the traditional analysis tools (e.g. ANSYS [44] for finite element analysis, etc.).

5.1.4 PWAF Architecture Overview

Now, we discuss the specific architecture of the PWAF, as comprised of the above model types. The navigation map for the conceptual architecture is comprised of a series of model transformations, as shown in Figure 15. The basis for developing this stepping-stone architecture (multiple transformations as opposed to one) is to be able to provide greater flexibility in the framework, hence addressing a wider range of designs and different analysis theories. The transformation function (as a conglomerate of parameter-based associativity) [33, 47] is derived based on the decision made by an analyst (an experienced engineer) in dealing with the particular design. The same design might be idealized differently based upon the context of the problem. Hence, the possibilities and the range for a particular transformation function expand as we move from the left to the right in Figure 3.

Information captured by traditional ECAD tools and enhanced by gap-filling tools [48] is integrated to develop the 210-based MPM. Thereafter the APM is derived from the MPM by extracting an information pool relevant to the particular analysis at hand. Further, based upon the context of the analysis (physical and functional structure of the product, boundary conditions, loading etc.) the corresponding CBAM is generated for a particular analysis. The CBAM then enables the derivation of a ABB system model from the APM. Lastly, the SMM is generated from the ABB system model in a ready-to-solve state and is interpreted by traditional analysis tools (example: ANSYS for finite-element analysis etc.). Overall this approach increases modularity and resuability as well as enhancing knowledge capture and tool independence.

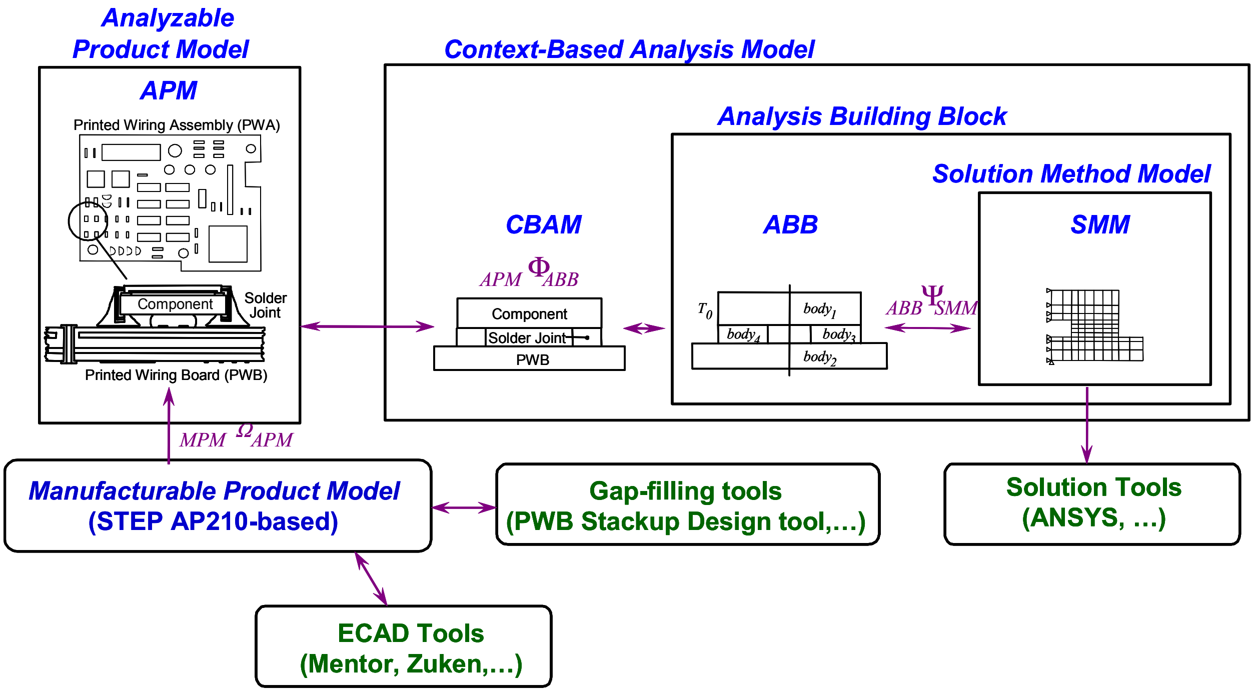

Analyzable Product Model
APM
Printed Wiring Assembly (PWA)
Component
Solder Joint
Printed Wiring Board (PWB)
Context-Based Analysis Model
Analysis Building Block
Solution Method Model
CBAM
ABB
SMM
APM Φ ABB
Component
Solder Joint
PWB
T_0
$body_1$
$body_4$
$body_3$
$body_2$
ABB Ψ SMM
MPM Ω APM
Manufacturable Product Model (STEP AP210-based)
Gap-filling tools (PWB Stackup Design tool,...)
Solution Tools (ANSYS, ...)
ECAD Tools (Mentor, Zuken,...)

Figure 15.Multi-representation architecture (MRA) view of PWAF [33, 47].

5.1.5 A Framework to predict warpage worthiness of PWBs

Realization of PWB Warpage Analysis Framework (PWAF)

The PWB Warpage Analysis Framework is a realization of the conceptual architecture proposed in the previous section. We now elucidate the navigation map for this architecture, the detailed contents of the stepping stone models, and the associated transformations.

Creating the Manufacturable Product Model – a normalization and enrichment process

The first step in the PWAF navigation map is the normalization of information captured in disparate sources to a unified and rich AP210-based MPM. In a typical engineering environment (as in an electronics design enterprise), information about the PWA and the PWB is contained in the ECAD tools and other auxiliary data sources (e.g. component databases). However, the extent of ECAD information coverage concerning the PWB is usually insufficient for the evaluation of the warpage worthiness of the subject design. But, before we can fill these information gaps, this design information is normalized to create an AP210-based model using the *LKSoft Design Integrator* (commercially, also referred to as an AP210 converter) [44, 49, 50 Kim, Thurman, Benda]. The normalized AP210-based model is then imported into an AP210 standards-based *PWB Stackup Design Tool* [48, 51]. In this environment, information specific to the PWB stackup, such as layer thickness, material, layer constituents etc. can be captured and communicated. This provides a platform for the design engineer to add missing pieces of information concerning the PWB stackup details that are not supported by the traditional ECAD tools. Thereafter, an enriched AP210 based design model is generated from this environment. This enriched model is the *Manufacturable Product Model*.

Figure 16 shows a snapshot (with different examples) of the state of information captured in a typical ECAD tool in (a). It also shows a view of the board stackup information as viewed from the *PWB Stackup Design Tool* environment in (b). Snapshots (c) and (d) show the 2D and 3D views of an enriched and normalized AP210-based MPM.

Extracting an Analyzable Product Model from the MPM

For the purpose of thermo-mechanical warpage analysis of the subject design, we need a subset of the information pool captured in the MPM. The next navigation step in the PWAF is to generate an APM from the MPM. Figure 14 shows the specific design objects that are of key concern for warpage analysis. We are interested in certain key features on each stratum (layer) of the PWB stackup and the shape of the PWB itself (outline, mechanical tooling holes, etc.). These specific features on a stratum are the conductive traces and islands of metallization (example: lands around vias and plated through holes and constituting component footprints, etc.), mostly providing electrical connectivity between different points. The APM is comprised of these design objects and is generated from the MPM using a Java [44]-based tool communicating with the MPM using STEP standard JSDAI libraries [44].

Generating the Analysis Building Block system model from the APM and the CBAM

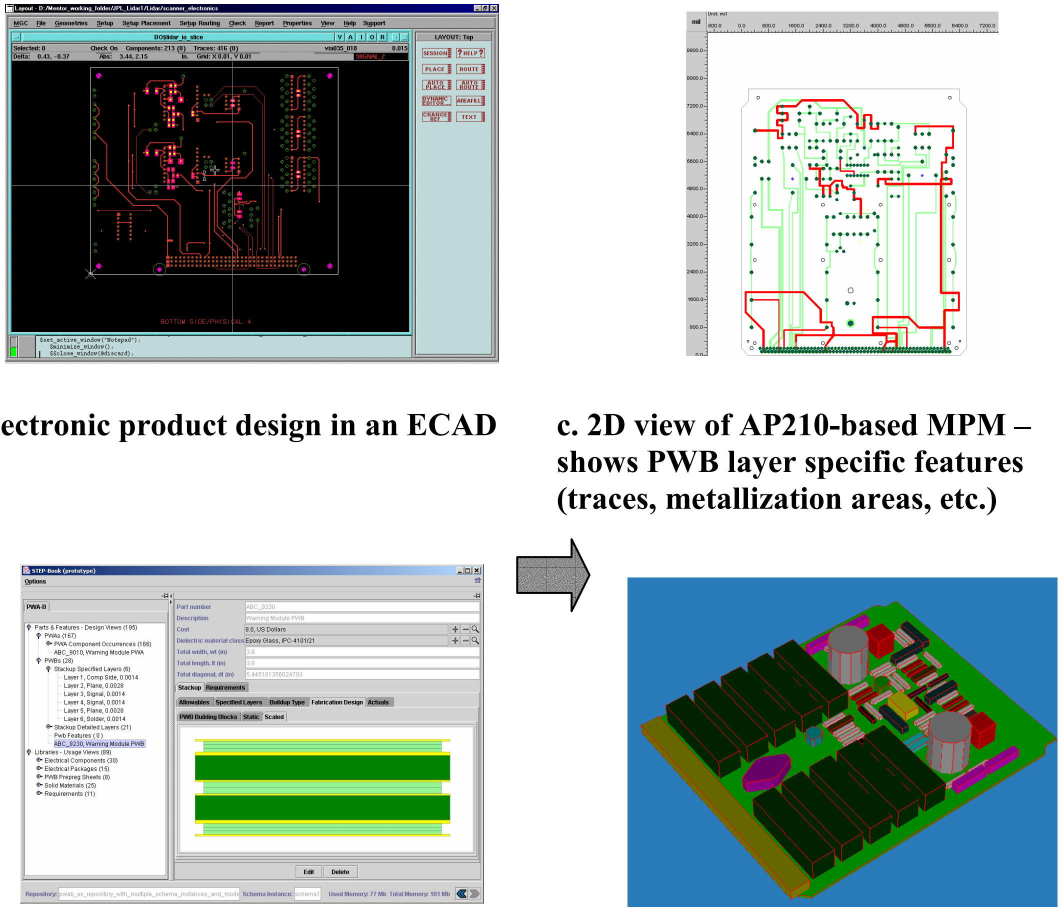
Navigating further on the PWAF architecture map, the next step is the generation of an Analysis Building Block (ABB) system model. However, the context of the problem needs to be highlighted before we proceed. For the subject analysis, the thermal loading profile and the boundary conditions are provided for the problem. Also, the thickness of the PWB is very small compared to its length and breadth and we consider it to behave structurally as a layered shell. This defines the context of the given problem and is captured as a CBAM.

From the expertise of design engineers and analysts, for the given context, it is within the limits of reasonable significance to analyze the PWB as a grid of elements with effective material properties as opposed to dealing with the exact layout of metallization on each stratum. We intend to capture this expertise in our architecture. The ABB system model specifically does this. The ABB system model in the PWAF is a grid of elements with effective material properties, as computed from the APM. Figure 17 shows the nature of the ABB system model as generated from the APM in the context represented by the CBAM. The effective material properties are computed using geometric algorithms developed by Georgia Tech which utilize Java 2D [44] and LKSoft's geometric processing libraries.

Deriving the Solution Method Model from the ABB system model and its interpretation in a finite element solver

The last navigation step in the PWAF map is the creation and interpretation of a Solution Method Model. As highlighted while describing the conceptual architecture, there may be multiple methods used to solve a physics-based representation of the ABB system model. The SMM captures the application of a particular method to the subject problem. In the scenario of PWAF, we use the finite element method to evaluate the warpage and associated thermal stresses in the PWB subjected to the given boundary conditions and thermal loading. In essence, the physics of the problem is captured in the ABB system model and the SMM is a solution-specific wrapper to the same. Figure 18 shows an example lexical view of the finite element-based SMM, as generated from the ABB system model. In this scenario, the SMM is described using APDL (ANSYS Parametric Design Language) [44].

Thereafter, the subject SMM is interpreted and solved using ANSYS [44].Figure 19.a shows an example view of the meshed finite-element model of a PWB with a single point constraint (locking all degrees of freedom) at bottom left. Figure 19.b shows the warpage profile (out of plane deflection) for the same PWB (with homogenous material properties) when subjected to a linear temperature increase from 25 deg. C to 150 deg. C. As evident, the deflection increases with the radial distance from left-lowermost corner, which is fixed.


Layout - D:/Mentor_working_folder/JPL_Lidar1/Lidar/scanner_electronics

BO$lidar_io_slice

Selected: 0 Check On Components: 213 (0) Traces: 416 (0) via035_018 0.015
Delta: 0.43, -0.37 Abs: 3.44, 2.15 In. Grid: X 0.01, Y 0.01 SIGNAL_2

```
$set_active_window("Notepad");
   $minimize_window();
   $$close_window(@discard);
```

STEP-Book (prototype)

- Parts & Features - Design Views (195)
  - PWAs (167)
    - PWA Component Occurrences (166)
    - ABC_9010, Warning Module PWA
  - PWBs (28)
    - Stackup Specified Layers (6)
      - Layer 1, Comp Side, 0.0014
      - Layer 2, Plane, 0.0028
      - Layer 3, Signal, 0.0014
      - Layer 4, Signal, 0.0014
      - Layer 5, Plane, 0.0028
      - Layer 6, Solder, 0.0014
    - Stackup Detailed Layers (21)
    - Pwb Features (0)
    - ABC_9230, Warning Module PWB
- Libraries - Usage Views (89)
  - Electrical Components (30)
  - Electrical Packages (15)
  - PWB Prepreg Sheets (8)
  - Solid Materials (25)
  - Requirements (11)

Part number	ABC_9230
Description	Warning Module PWB
Cost	8.0, US Dollars
Dielectric material class	Epoxy Glass, IPC-4101/21
Total width, wt (in)	3.9
Total length, lt (in)	3.8
Total diagonal, dt (in)	5.445181356024793

Repository: pwab_ex_repository_with_multiple_schema_instances_and_mode Schema Instance: schema1 Used Memory: 77 Mb Total Memory: 101 Mb

a. Electronic product design in an ECAD tool

c. 2D view of AP210-based MPM – shows PWB layer specific features (traces, metallization areas, etc.)

b. Gap-filling tool for PWB stackup design

d. 3D view of AP210-based MPM – shows assembly level information (components and their layout, etc.)

Figure 16. Electronic product design information in traditional ECAD tools and gap-filling tools normalized into an AP210-based manufacturable product model (MPM).

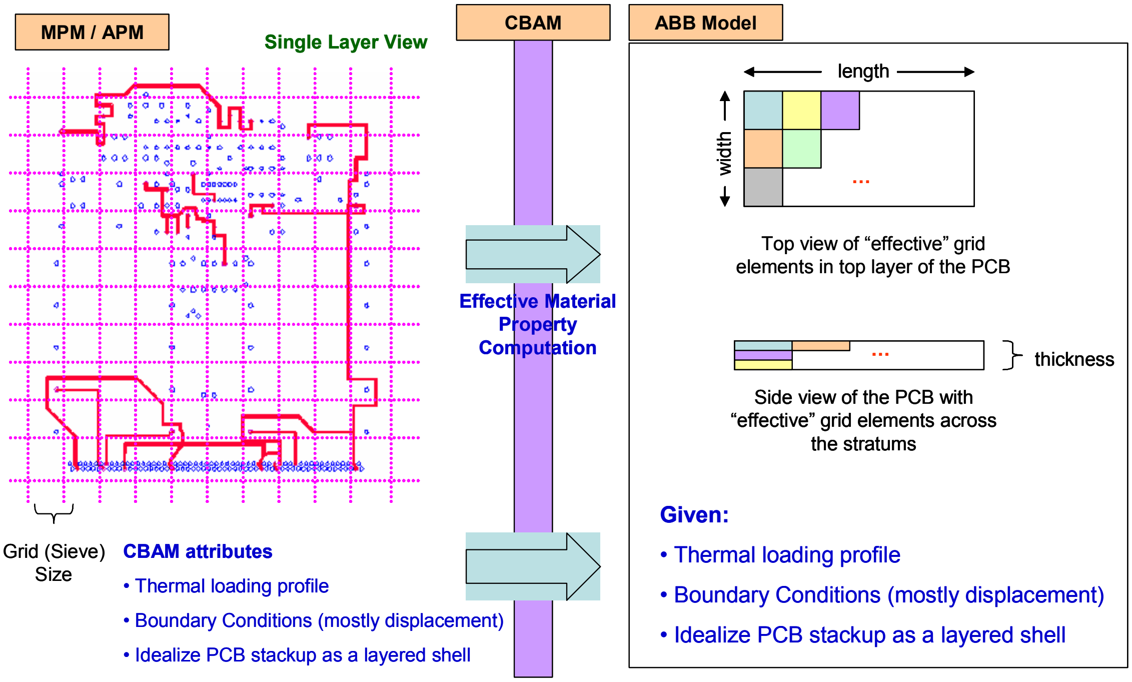

MPM / APM
Single Layer View
Grid (Sieve) Size
CBAM attributes
• Thermal loading profile
• Boundary Conditions (mostly displacement)
• Idealize PCB stackup as a layered shell
CBAM
Effective Material Property Computation
ABB Model
length
width
...
Top view of "effective" grid elements in top layer of the PCB
...
thickness
Side view of the PCB with "effective" grid elements across the stratums
Given:
• Thermal loading profile
• Boundary Conditions (mostly displacement)
• Idealize PCB stackup as a layered shell

Figure 17. Key aspects of a warpage CBAM (product specific analysis template) and its associated ABB system model (generic analysis model).

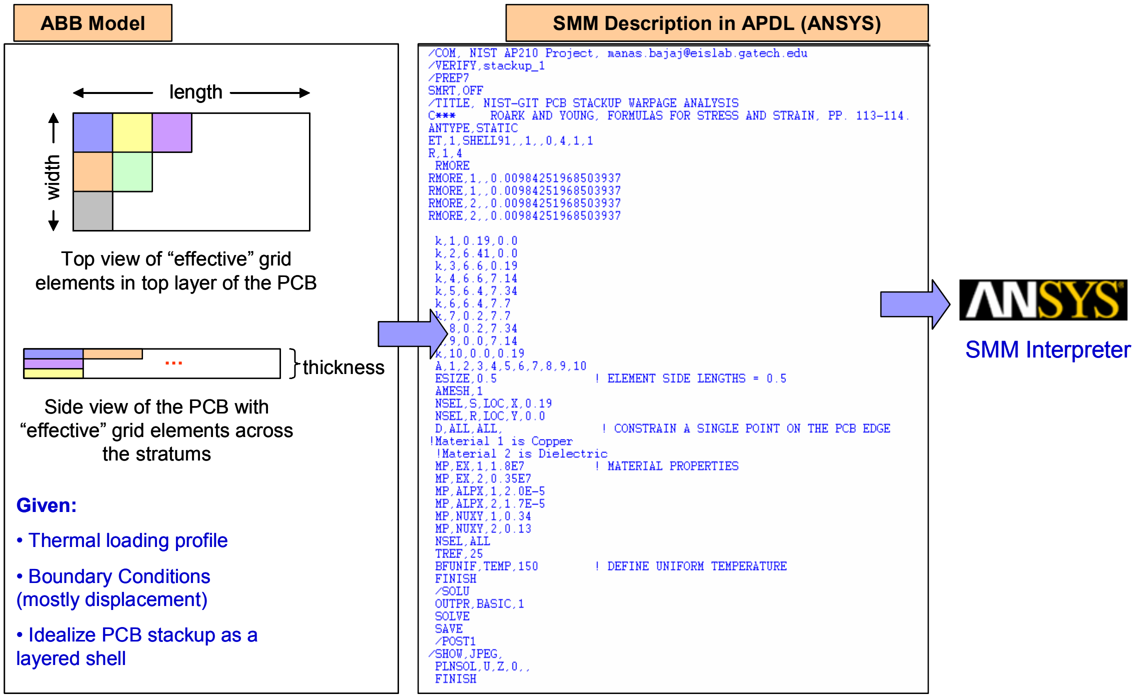
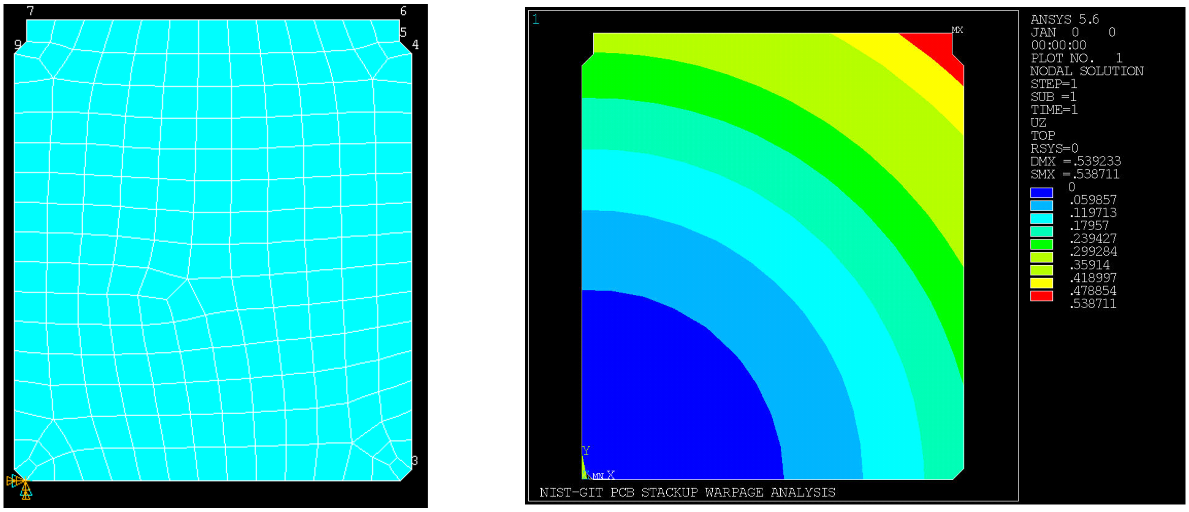


ABB Model
length
width
Top view of "effective" grid elements in top layer of the PCB
...
thickness
Side view of the PCB with "effective" grid elements across the stratums
Given:
• Thermal loading profile
• Boundary Conditions (mostly displacement)
• Idealize PCB stackup as a layered shell
SMM Description in APDL (ANSYS)
ANSYS
SMM Interpreter

```
/COM, NIST AP210 Project, manas.bajaj@eislab.gatech.edu
/VERIFY,stackup_1
/PREP7
SMRT,OFF
/TITLE, NIST-GIT PCB STACKUP WARPAGE ANALYSIS
C***     ROARK AND YOUNG, FORMULAS FOR STRESS AND STRAIN, PP. 113-114.
ANTYPE,STATIC
ET,1,SHELL91,,1,,0,4,1,1
R,1,4
 RMORE
RMORE,1,,0.00984251968503937
RMORE,1,,0.00984251968503937
RMORE,2,,0.00984251968503937
RMORE,2,,0.00984251968503937

 k,1,0.19,0.0
 k,2,6.41,0.0
 k,3,6.6,0.19
 k,4,6.6,7.14
 k,5,6.4,7.34
 k,6,6.4,7.7
 k,7,0.2,7.7
   8,0.2,7.34
  ,9,0.0,7.14
 k,10,0.0,0.19
 A,1,2,3,4,5,6,7,8,9,10
 ESIZE,0.5              ! ELEMENT SIDE LENGTHS = 0.5
 AMESH,1
 NSEL,S,LOC,X,0.19
 NSEL,R,LOC,Y,0.0
 D,ALL,ALL,              ! CONSTRAIN A SINGLE POINT ON THE PCB EDGE
!Material 1 is Copper
 !Material 2 is Dielectric
 MP,EX,1,1.8E7          ! MATERIAL PROPERTIES
 MP,EX,2,0.35E7
 MP,ALPX,1,2.0E-5
 MP,ALPX,2,1.7E-5
 MP,NUXY,1,0.34
 MP,NUXY,2,0.13
 NSEL,ALL
 TREF,25
 BFUNIF,TEMP,150        ! DEFINE UNIFORM TEMPERATURE
 FINISH
 /SOLU
 OUTPR,BASIC,1
 SOLVE
 SAVE
 /POST1
/SHOW,JPEG,
 PLNSOL,U,Z,0,,
 FINISH
```

Figure 18. Lexical view of the SMM model generated from the ABB system model.



a. Meshed finite element model of an example PWB

b. Warpage profile results for the example PWB

Figure 19. PWB finite element model processed in ANSYS.

5.1.6 Value to DAI Research – How do we improve our efforts?

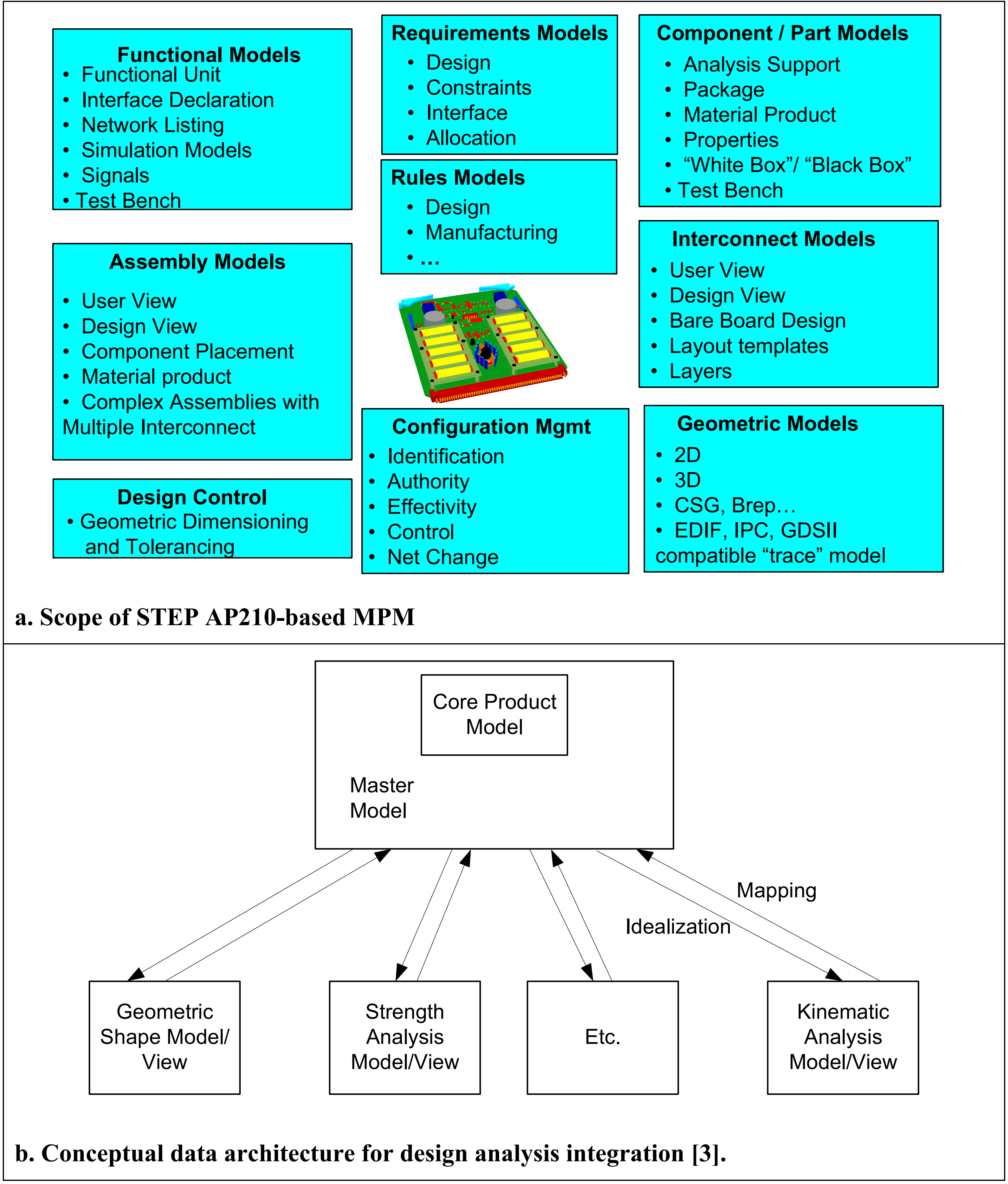
Standard-based smart product model modularized into product model views

The manufacturable product model is the starting point of the analysis in this case. In a typical design process, pieces of information that a designer refers or generates are drawn from or committed to disparate sources. The semantics and the structure of information locked in these sources differ, making the process time consuming for the designer. Hence, more time is actually spent on resolving conflicting information than brainstorming on possible design alternatives. Also, the design process can be viewed as the conceptualization, development and use of the manufacturable product model or in other words an enrichment process. It is imperative that there be a standard-based representation of the manufacturable product model.

The use of STEP application protocol 210 for electronic assembly design is thus critical to this architecture. A 210-based MPM provides not only a translation channel amongst sources of information but also provides a semantically rich structure on which assembly design could be instantiated.

In a similar vein, we would also like to refer to the *core product model* that Fenves et al. had proposed. The 210-based MPM provides a similar structure as the core product model. Figure 20 reflects the similarity. The core product reflects the idea of an integrated design model which is evident in the scope of STEP AP210 standard itself. Also, within the 210-based implementation of the MPM, there is a need to organize the views (requirements, functional, assembly, etc.) of the product model in implementation. This has also been highlighted in the concept of master product model and the model views

There are ongoing efforts in the AP210 development community (of which we are a part) to modularize the instantiated design model.



a. Scope of STEP AP210-based MPM

b. Conceptual data architecture for design analysis integration [3].

Figure 20. Content coverage of AP210-based MPM and Core Product Model

Rationale for Idealizations

Idealizations encompass the simplification measures that reduce the complexity of the manufacturable product model and create a context specific analyzable model. We have highlighted the role of idealizations in the PWAF architecture. They form a web that links the stepping stone models in the MRA view (Figure 15).

It is important to realize that idealizations are supported by knowledge, especially expertise. They represent decisions that an analyst or a designer takes for evaluating the design. Hence, it is critical that there is a comprehensive computer-based representation to support them. This would enable their efficient development, archival and updating thus reducing time taken to utilize the range of expertise. At the basic level, analytical equations with bi-directional interpreters, IF-THEN rules with inference engines, etc. provide a mode for implementing idealizations. Some of these forms have also been used in the PWAF. However, this is not sufficient. These representations need to be rewritten each time there is even a slight variation in the expertise. Thus, we need to capture the rationale for these idealizations in a sensible computer-based form. If we can back idealizations with rationale, we could automate the generation of idealizations by using the rationale.

Multi-representation architecture at various levels of abstraction

The multi-representation architecture [47] view of the PWAF in Figure 15 provides a valuable insight into design-analysis integration. It has been well established that stepping stone-based model architectures provides a much-needed formalism when linking design and analysis efforts. The nature and scope of models at each step in this architecture are key to the reusability of any framework that is based on the architecture. However, past and recent efforts have used this architecture for evaluating designs at a very detailed level. Design and analysis are parallel activities that begin with the conceptualization of the product. It is important that this effort is extended to more abstract levels in the design process. Modifications may be made to the terminology and scope of the models in the stepped architecture but the underlying idea would be consistent. We perceive that the MRA-based approach would provide an efficient framework to converge on design alternatives during the early design stages that are abound with greater freedom.

5.2 Example Case 2: A Knowledge Repository for Behavioral Models in Engineering Design [41]

The goal in this research effort is the development of a repository to reduce the knowledge gap between engineering design and analysis by facilitating reuse of behavioral models. In this context, a behavior model captures the underlying behavior, but is not tied to a particular product model. To achieve a higher level of reuse in the product design process, we propose a meta-data representation for formally characterizing behavioral models. The meta-data representation captures the assumptions, limitations, accuracy, and context of engineering behavioral models. Based on this knowledge representation, a proof-of-concept repository is implemented for archiving and exchanging reusable behavioral models. The knowledge representation and implementation is illustrated with a simple cantilever beam example.

The development of the behavioral model repository attributes to overarching goals of close design-analysis integration in the following ways: (see Figure 21).

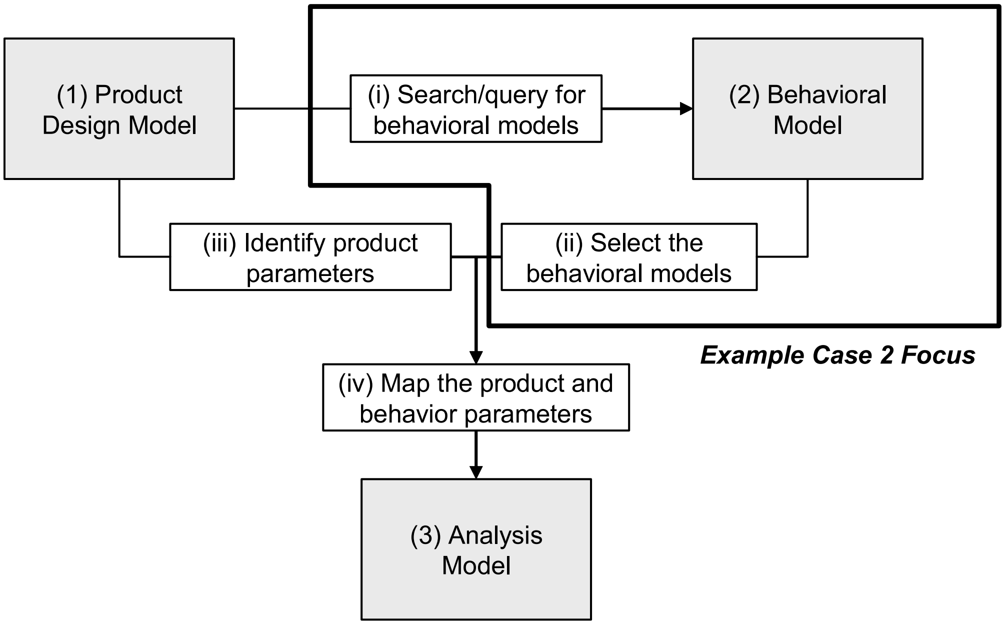

(1) Product Design Model
(i) Search/query for behavioral models
(2) Behavioral Model
(iii) Identify product parameters
(ii) Select the behavioral models
Example Case 2 Focus
(iv) Map the product and behavior parameters
(3) Analysis Model

Figure 21. Focus of behavioral model repository research area in DAI effort.

5.2.1 Description of Behavioral Model Repository

The focus of this research is reducing the semantic knowledge gaps between engineering design and analysis by formally describing behavioral models. In this context, behavioral models are models that capture the mathematical description of the physical behavior of a product. Examples of behavior models include, but are not limited to: stress-deflection of a beam or the current-torque relationship in DC motors. Behavior models can vary in complexity. For example, the behavior of a cantilever beam may be modeled as a simple equation-based model or a complex numerically-based finite element analysis model. Engineering behavior models can be developed across multiple domains and for various stage of design. For example, at the conceptual stages of design engineers may use a simple beam equation model to obtain a rough estimate of design constraints. Engineering designers must decide what fidelity of model is appropriate for the design phase.

We believe that two primary types of knowledge exist in developing behavioral models, namely: (1) the knowledge captured in the behavioral model and (2) the meta-knowledge that describes the behavioral model. The first type is the explicit knowledge represented as a behavioral model. In the case of computer-based behavioral models, this knowledge may include, but is not limited to, the geometric representation, parameterization, constraints, first principles, and the underlying behavior representation of the product encoded in a particular modeling or programming language. The second type of knowledge is meta-knowledge that describes the behavioral model. The meta-knowledge includes the underlying assumptions, limitations, and context in which the behavior model is applicable and can be used with confidence. The knowledge captured in the behavioral model and the meta-knowledge that describes the behavioral model are not independent. Rather, the behavioral model and the characterization of the model are developed simultaneously.

For example, modeling experts may use engineering first principles, stress-strain relationships, and Theory of Flexure to develop a bending model of a beam that conforms to basic assumptions and limitations. While the behavioral model is captured in a computer-sensible format, the meta-data, including the assumptions and limitations of behavior models are often not. The behavioral model knowledge representation supports both meta-information and meta-knowledge. Meta-information is captured for configuration control (i.e. versioning and tracking) and meta-knowledge is captured for increased reuse. In essence, we are developing a wrapper for describing the assumptions, limitations, accuracy, and validity of the behavior model (see Figure 22).

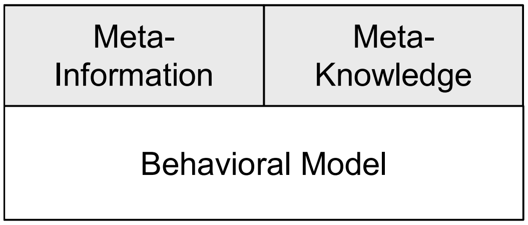


Figure 22. Behavioral model and meta-knowledge for describing formalized behavioral models.

The development and subsequent usage of reusable behavior models is beneficial to engineering design by capturing design-analysis knowledge in a format that can be leveraged throughout the product development process. Reusable models that capture both the explicit and implicit product knowledge will reduce the integration gaps between engineering design and analysis activities, thus reducing the cost and time of product development.

5.2.2 Motivation for Capturing Behavioral Model Knowledge

As previously stated, Fenves, et al. [3] assert that problems with product development can be associated with the knowledge gaps between engineering design and analysis domains. The integration between designers and analysts is limited, in a large part due to the diverse expertise and knowledge between the domains. The overarching problem in Figure 4 is that the knowledge interface between design activities and analysis activities. Designers do not often understand the underlying mathematical model, limitations, assumptions, and context in which the behavior model is relevant. Thus reuse of the model is limited at best or designers use behavioral models in improper situations.

5.2.3 Creating Reusable Behavior Models

The process associated with behavior simulation in product development has two components (1) creation of reusable behavioral models and (2) use of these models in engineering design problems by engineering designers. Reuse of behavior models requires:

- Capturing the meaning of a model so that a human can reuse the model

- Facilitating automated creation of behavioral model instances based on design representations

Reusable behavioral models are created by behavioral models experts, most often engineering analysts. The models are then used by engineering designers, by instantiating the model for a particular product development process. Reuse of engineering behavioral models will increase the use of simulation in design by enabling engineers to select and use behavioral models appropriately and map the design form parameters to the behavioral parameters clearly.

The process of developing reusable behavioral models is a complex activity composed of two activities, namely: *Model Development* and *Model Characterization*. Behavioral modeling experts must develop and characterize the models to enable reuse of the models by engineering designers. The process presented in Figure 23 reflects the complex activity of developing reusable models. For example, a behavioral modeling expert simultaneously develops the model and the characterization of the model. The interaction between these activities is represented by arrows between *Model Development* and *Model Characterization.*

Once the expert fully characterizes and develops the model, the behavioral model is published to the repository. The executable behavioral model and the formalized meta-knowledge are stored in the repository, thus enabling designers to access a complete description of the model (see Figure 23).

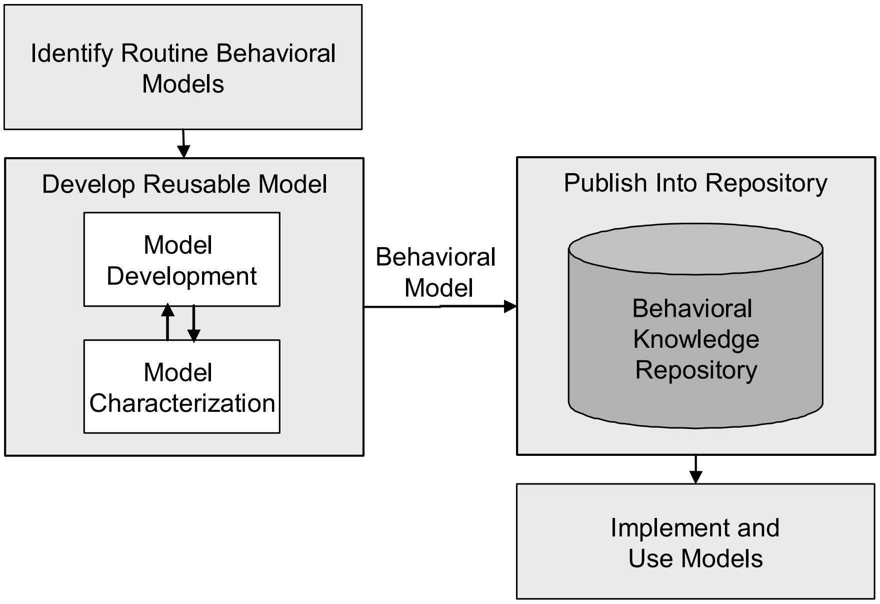


Figure 23. Information flow for creating and characterizing models.

Behavior modeling experts work with designers to continually identify and develop behavioral models. These behavioral models are published to the repository to facilitate reuse by engineering designers.

Engineering designers can then query the behavioral model repository, select the most appropriate behavioral models given the objectives of the behavior simulation. The simulation process and information flow is presented in Figure 24. The process consists of Evaluation

Problem Formulation, Behavioral Model Development, Model Execution, and Results Evaluation.

In the *Evaluation Problem Formulation* activity the overall objectives of the simulation are established. *Behavioral Model Selection & Instantiation* consists of querying and selecting the most appropriate model from the repository and populating the behavioral model with product design parameters. Creating the behavior model instance is complex because it requires a complex mapping between the design-specific parameters and analysis-specific parameters needed for the behavior model.

Once an instance of the behavioral model is created for the product parameters, *Model Execution* is completed and simulation results are obtained. In *Results Evaluation,* the simulation results are evaluated against the formal evaluation problem objectives. If the system-level behavior model indicates the design will not meet the intended behavior, then the design parameters are altered and the process is completed.

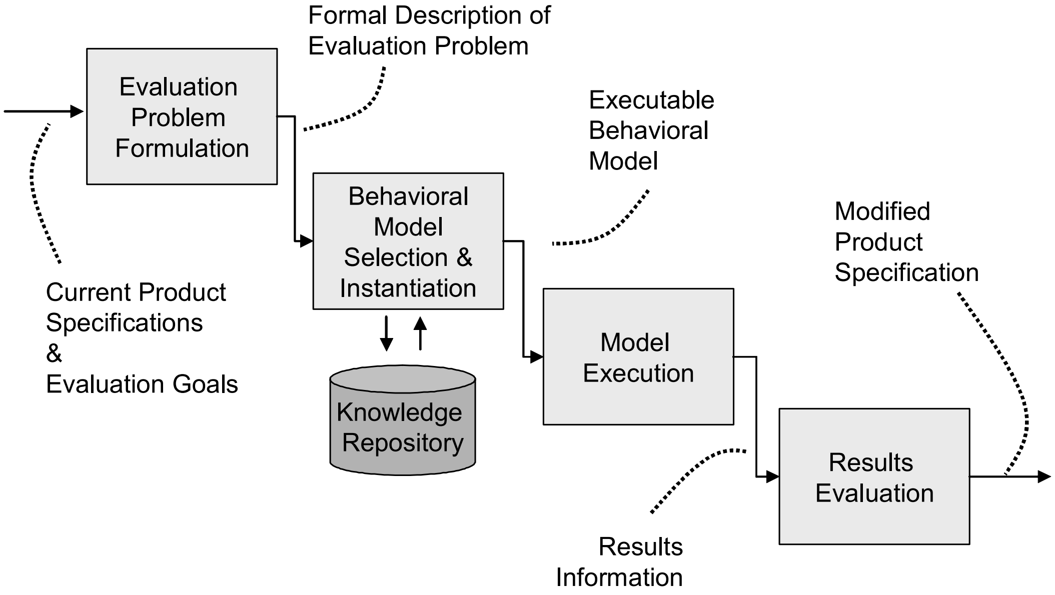

Formal Description of Evaluation Problem
Evaluation Problem Formulation
Executable Behavioral Model
Behavioral Model Selection & Instantiation
Modified Product Specification
Current Product Specifications & Evaluation Goals
Model Execution
Knowledge Repository
Results Evaluation
Results Information

Figure 24. Knowledge flow for using behavior models in design.

5.2.4 Characterizing Behavioral Models

The behavior model repository will contain engineering behavior models accompanied by formal descriptions that facilitate easy characterization and reuse while reducing the change for improper usage. This implies that knowledge, usually left unspoken by the developer of a behavior model, such as its accuracy and the range of conditions over which it can operate effectively must be captured. While this expert knowledge is used in the development of a behavior model, it is seldom captured to characterize the models.

The approach of this research effort is twofold. First, we propose a knowledge representation for formally capturing and characterizing behavior and analysis models by capturing meta-knowledge about the models. Second, we present the development and implementation of a repository for facilitating the reuse of these behavior models.

Scope and Type of Behavioral Models

The behavioral model knowledge representation supports various types of models across all phases of engineering design and product development. Gross [40] states that engineering analysis models include lumped parameter models, continuum parameter models, and statistical models derived from empirical observations, such as response surface models. The behavioral models stored in the repository can cover a range of domains including structural analysis, dynamics, thermal, and thermo-mechanical to name a few. Additionally, the models can vary in complexity from simple model to highly detailed models. The complexity of the behavioral models reflects the phase of product development. For example, a simple model may be adequate to support a design decision at the early stages of design (i.e. Conceptual Design Phase [52]) or may be highly detailed finite element model to support decision made at the latter stages of design (i.e. Detailed Design Phase [52]). An open issue in behavioral model knowledge representation is *How can the right model be chosen?* This issue raised by Brooks [53] addresses the idea of selecting a model based on complexity, level of detail, and performance.

Concepts for Describing Behavior Models

In this research we are primarily concerned with publishing characterized behavioral models into the repository. Thus, the concepts of interest exist in the *Model Characterization* box of Figure 24. The concepts for characterizing behavioral model proposed in [42] are expanded upon in this research. Additionally, an information model is developed to enable a behavioral model repository to be developed. The concepts for describing behavior models are included in Table 4.

Table 4. Concepts for characterizing behavioral models.

Configuration Control	Serves as a method to track the history of the behavioral model in the repository. It is developed as a superclass to enable inheritance of such attributes as author, date created, version, etc. The configuration class enables behavioral modeling experts to track the history of models in the repository
Characterized Behavioral Model	The focus of the repository development. Characterized behavioral models entries are published to the repository for reuse. A published instance to the repository must contain an Executable Behavioral Model, Assumptions, Context, Inaccuracy, and Interface Description to the model.
Assumption	A human interpretable description of the behavioral model. Assumptions characterize the behavioral model. Assumptions make clear the implicit considerations in the model, such as operational conditions (steady state, etc.) as well as any restrictions on variable values (i.e., bounds). An Assumption instance is a string and not computer-processible during model execution. Assumptions and Context are related to each other in the development process

Executable Behavioral Model	A model that captures the mathematical description of the physical product (or one or more of its subsystems) that is being designed. The Executable Model is captured in a modeling or programming language such as Mathematica or Modelica. The Executable Model is described by Model Name, Model Type, and Model File Pointer
Interface Description	Describes how the model is used by the designer and/or the environment. The interface has the subclasses of Causal of Non-Causal. The Quantities associated with the model are published to the user through the interface
Causal Interface	The causality is imposed in the model. The assignments of variables in the models are expressed through assignment operations. The input and output variables, parameters, and constants are described in a Causal interface
Non-Causal Interface	These interfaces do not impose mathematical causality on the model. The model is described by a set of equations are solved by a constraint solver. A non-causal interface is described by Ports, Constants, and Parameters
Port	Port represents how exchange takes place between the model and the user or environment
Quantity	Quantity is a superclass of Variable, Parameter, and Constant. A Quantity is described by the Name, Units, and Value
Variable	Serves as the input and output to a model. A Variable can vary during the course of a behavioral evaluation (for time-varying models). A Variable often conveys the simulation result to the user
Input Variable	An input variable is what the user of the model specifies.
Output Variable	An output variable is what is produced from the model execution
Parameter	A parameter is similar to a variable, but parameters do not change during the course of a behavioral evaluation. Within the scope of this research, a parameter conveys the information from the design form to the behavioral model
Constant	A quantity that is assumed to have a fixed value in a particular context. Examples of constants include the Gravitational Constant (G) the speed of light. Behavioral model users should be aware of the units and associated constants in the model

Context	Context can be represented as a set in the space of physical quantity values [42]. Context may include quantities that are not present in the corresponding model. This can happen when a model creator abstracts away the effects of some quantity. For example, it is common to formulate the deflection equation for a cantilever beam with the assumption that the mass of the beam is negligible. In general, Context can be any set of quantity value restrictions. In the current implementation, context is represented as a set of value bounds on problem quantities. All quantities are presumed unbounded unless explicit bounds are stated
Quantity Restriction	The quantity restriction specifies how a quantity is bounded. In this model, a Quantity can only be bound by upper and lower limits
Inaccuracy	Inaccuracy reflects how well a model corresponds to the physical system that it represents. In the current implementation inaccuracy represents the upper bound on the magnitude of the difference between a model and the real system over the stated context. Inaccuracy and Context are related, for example shrinking the Context can result in decrease on model Inaccuracy [42]

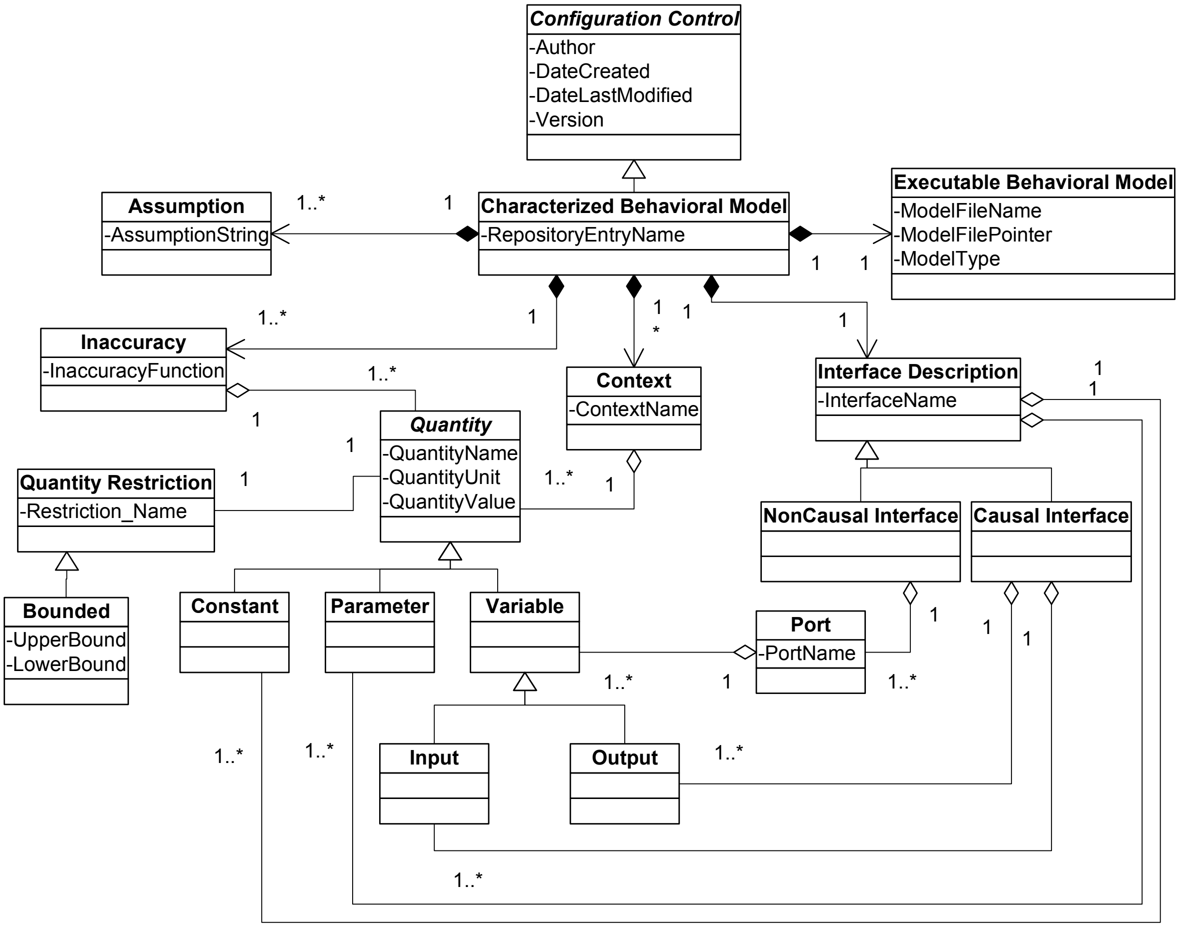


Figure 25. Model for characterizing behavior models.

Behavior Model Knowledge Representation

Based on the previously described concepts, a conceptual knowledge model is developed. The Unified Modeling Language (UML) is used for conceptualization of behavior model knowledge. The behavioral model knowledge representation presented in Figure 5, captures the executable behavioral model and the knowledge that described the model. A behavioral model in the repository is described by (1) the executable model and (2) the meta-knowledge about the model including *Assumptions, Inaccuracy, Context,* and *Interface Description*. By publishing this information, the likelihood of abuse of the model can be reduced. The model interface description provides users with the quantities and associated units in the model. Secondly, by publishing the Context and Inaccuracy, a quantitative understanding of the simulation result and validity can be obtained. In the following section, an illustrative example is presented to demonstrate the formal characterization of a behavior model.

Behavior Model Knowledge Representation Discussion

The research presented in this paper is an ongoing effort towards addressing the knowledge gaps between engineering design and analysis. While the behavior model knowledge representation illustrated in Figure 25 is stable, research is ongoing to refine the model.

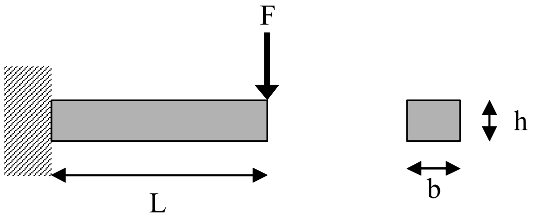
The benefits of the behavior knowledge model include the ability to publish the behavior model and associated meta-knowledge, thus enabling model users (designers) to gain a better understanding of the model. However, there are several considerations that must be addressed. First, the model must be refined to better represent Assumption and Inaccuracy class variables. In the current state, Assumption is represented as a human-interpretable string. A long term goal is to represent Assumptions as computer interpretable knowledge representations to facilitate automated checks for model appropriateness. Ongoing research is focused on representing inaccuracy of behavior models [42]. Next, the behavior model knowledge representation is taken independent from design models. Behavior models are used by populating the models with design parameters (outside of scope of model). Finally, how can consistency be ensured across models? In this research we are not trying to standardize the vocabulary for behavior models, rather we are providing a mechanism to share and publish behavior model knowledge.

Behavior Model Example

To illustrate the behavioral models knowledge representation proposed in this research, a simple cantilever beam example is presented (see

Table 5). The actual behavior models published in the repository can vary from simple, such as the cantilever beam, to highly complex FEA-based behavior models.

Table 5. Deflection models for cantilever beams.

Beam and Loading	Maximum Deflection
	$\delta_{Max} = -\frac{F \cdot L^3}{3 \cdot E \cdot I}$

The equations for computing the maximum deflection of the beam are presented in

Table 5. The deflection behavior model are developed using the fundamentals of Mechanics of Materials.

The meta-knowledge associated with development and implementation of the cantilever beam behavior models is summarized in Table 6.

Table 6. Meta-knowledge for cantilever beam models [54]

- The fixed end of the beam has zero rotation
- The weight of the beam is negligible and not considered
- A beam is bent with a concave and a convex side. The beam is subjected to compression and tension
- The intersection of the neutral plane with the face of the beam is the neutral axis or elastic curve
- The beam is prismatic
- The length of the beam is 10 times the depth
- Externally applied forces remain at right angles to the axis of the beam and in a plane of symmetry,
- Flexure of the beam is slight (i.e. The angles of rotation are small with respect to cross-section and remains linear)
- The material of the beam is homogeneous and obeys Hooke's Law
- The stresses in the beam are within the elastic limit
- Every layer of the material is free to expand and contract longitudinally and laterally under stress as if separate layers
- The tensile and compressive moduli of elasticity are equal
- The cross-sections of the beam remains plane (shear is constant or zero over the cross section)

The seemingly simple cantilever beam example illustrates the need to capture both the behavioral model and the meta-knowledge that characterizes the behavioral model. The behavioral model meta-knowledge must be made explicit for to facilitate the appropriate behavioral model selection and to enable engineering designers to make the right decision based on simulation results from the model.

The description of the cantilever beam model in Table 5 and Table 6 are mapped to the concepts for describing behavioral model (see Table 7).

Table 7. Illustrative example mapped in terms of knowledge representation.

Concept	**Cantilever Beam Example**
Assumption	*See Table 2*
Executable Behavioral Model	Cantilever Beam Deflection
Model Name	cantilever_beam.c
Model File Pointer	ANSI C Programming Language
Model Type	
Interface Description	Causal
Variable	
Input Variable	F, load, pounds force
Output Variable	δ, deflection, inch
Parameter	L, beam length, in
	E, modulus of elasticity, psi
	I, Moment of Area, in^4
	W, Beam Weight, lb
	b, Beam Depth, in
Constant	N/A
Context	
Quantity Restriction	W = 0
	L ≥ 10* b
Accuracy	-

Behavior Model Repository Implementation

A behavior model repository has been implemented as a proof-of-concept system using the following implementation technologies: (1) static HTML pages, (2) HTML forms, (3) Perl CGI/DBI scripts, and (4) a MySQL relational database. The knowledge repository presented in Figure 25 is partially implemented in the system. Refinement and implementation of the complete knowledge representation is ongoing.

The repository is accessed through a variety of static and dynamic HTML pages that communicate with the database via common gateway interface (CGI) and DBI scripts. HTML is chosen because it offers an easily accessible platform on which to deploy applications. The combination of Perl and HTML forms make an elegant and quick solution for deploying a proof-of-concept behavior model repository (see Figure 26).

Behavioral modeling developers publish models to the repository through web-based HTML forms. The process of creating reusable model, illustrated in Figure 26, is complex and requires further research and development of a method that support behavioral model development and characterization.

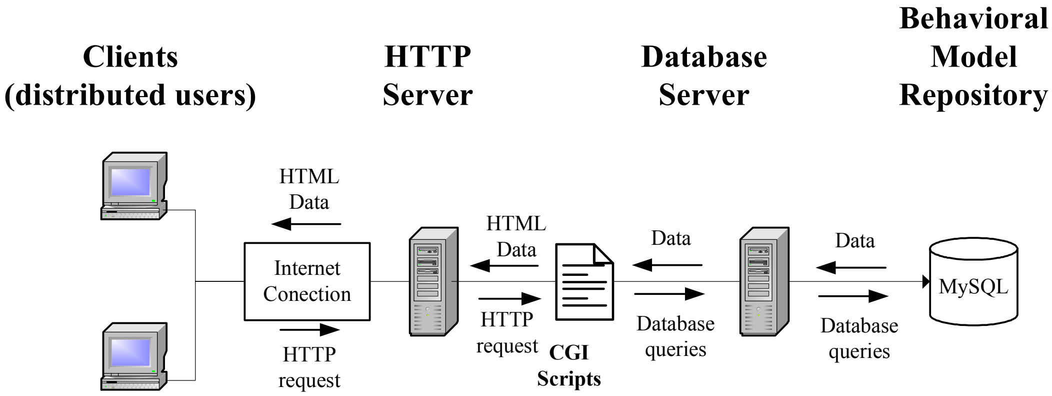
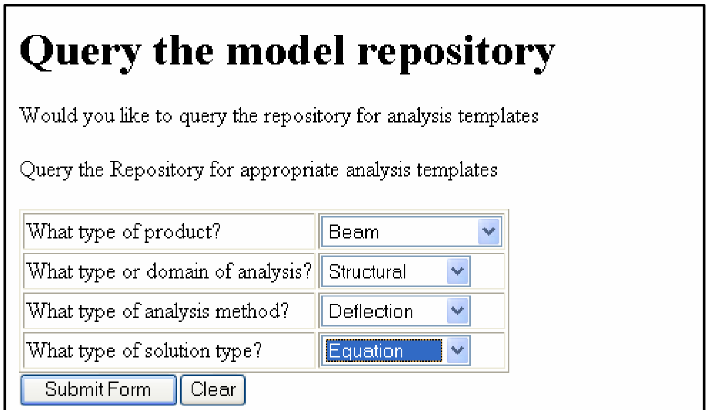


Figure 26. Behavioral model repository system architecture.

Once published, designers are able to access the repository and search and select the behavior models that most appropriately match their design activity. The behavioral models are described by product domain, analysis method to name a few (see Figure 27).



Query the model repository

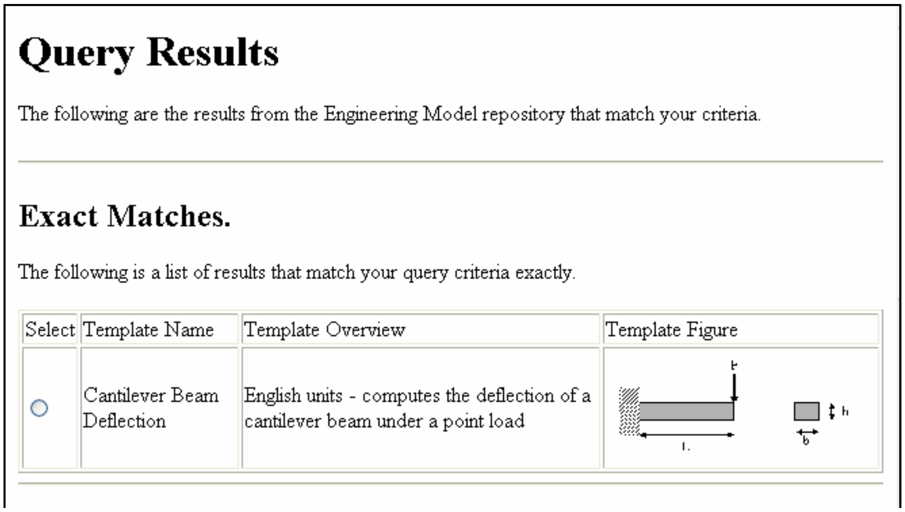
Would you like to query the repository for analysis templates

Query the Repository for appropriate analysis templates

What type of product?	Beam
What type or domain of analysis?	Structural
What type of analysis method?	Deflection
What type of solution type?	Equation

Figure 27. Behavioral model repository query form.

Once the repository search is completed, the designer can then select the appropriate behavioral model from those models that are stored in the repository. Behavioral models, published to the repository, that match the search criteria are displayed. Designers then choose the *best* model from a subset of models in the repository. A compact view of the model is displayed to the designer including information such as Template_Name, Overview of the Template, and a figure of the describing the model (see Figure 28).



Query Results

The following are the results from the Engineering Model repository that match your criteria.

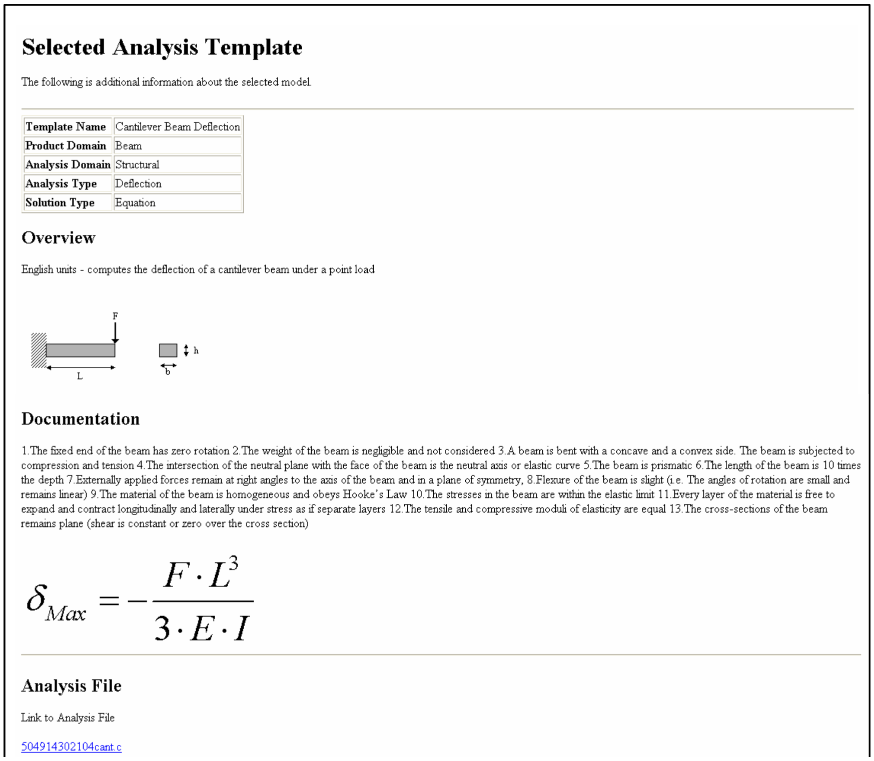
Exact Matches.

The following is a list of results that match your query criteria exactly.

Select	Template Name	Template Overview	Template Figure
○	Cantilever Beam Deflection	English units - computes the deflection of a cantilever beam under a point load	

Figure 28. Results from the query form for behavioral models that match the search criteria.

Designers can select a behavior model and the meta-knowledge about the behavioral model, such as assumptions, and limitations. Figure 29 illustrates the cantilever beam behavioral models with the associated executable model and the model description. A diagram of the system setup, documentation, and mathematical relationship can be accessed by designers. The extended view of the chosen model includes the query criteria, Overview, detailed Documentation, the mathematical relationships in the mode, and the executable model.



Selected Analysis Template

The following is additional information about the selected model.

Template Name	Cantilever Beam Deflection
Product Domain	Beam
Analysis Domain	Structural
Analysis Type	Deflection
Solution Type	Equation

Overview

English units - computes the deflection of a cantilever beam under a point load

Documentation

1.The fixed end of the beam has zero rotation 2.The weight of the beam is negligible and not considered 3.A beam is bent with a concave and a convex side. The beam is subjected to compression and tension 4.The intersection of the neutral plane with the face of the beam is the neutral axis or elastic curve 5.The beam is prismatic 6.The length of the beam is 10 times the depth 7.Externally applied forces remain at right angles to the axis of the beam and in a plane of symmetry, 8.Flexure of the beam is slight (i.e. The angles of rotation are small and remains linear) 9.The material of the beam is homogeneous and obeys Hooke's Law 10.The stresses in the beam are within the elastic limit 11.Every layer of the material is free to expand and contract longitudinally and laterally under stress as if separate layers 12.The tensile and compressive moduli of elasticity are equal 13.The cross-sections of the beam remains plane (shear is constant or zero over the cross section)

$$\delta_{Max} = -\frac{F \cdot L^3}{3 \cdot E \cdot I}$$

Analysis File

Link to Analysis File

504914302104cant.c

Figure 29. Meta-knowledge is supplied to the designer that characterizes the model.

Finally, designers can access executable behavioral models and instantiate the model with design model values. The intent is to enable engineering designers to populate behavioral models, thus creating instances, that can be executed through a service or existing solver.

To extend the current capability, implementation efforts are focused on model-driven dynamic user interfaces for creating behavioral model instances. The eXtensible Markup Language (XML) and the Extensible Stylesheet Language Family Transformations (XSLT) are being investigated as a technology for achieving dynamic creation of user interfaces and execution of behavior models.

Limitations and Future Consideration of Repository Implementation

The current generation repository is implemented as a form-based lookup system. The form-based provides a simple means to prototype the behavioral model repository. As the number of models in the repository increases, the lookup implementation becomes less useful. However, the current generation implementation illustrates the viability and usefulness of a behavior model repository. Future implementation consideration is ontological representation of the behavioral models. Current research is being conducted on the use of the Protégé ontology development tool [55]. Long-term visions of the repository include a software application or framework embedded in existing design tools, such as CAD software.

5.2.5 Behavioral Model Repository Closure & Discussion

A knowledge representation and proof-of-concept repository implementation for capturing and sharing engineering behavioral models is proposed to capture behavioral knowledge. The behavioral models repository reduces the integration gaps between engineering designers and analysts by providing designers with an increased understanding and availability of behavioral models. This is achieved by capturing expertise possessed by analysis experts and publishing both the executable behavior model and the meta-knowledge about the model to a repository that can be accessed by engineering designers and analysts. Designers are able to select behavioral models that are (1) appropriate for their desired simulation context and (2) understand the underlying assumptions and limitation of the model. The reuse of behavior models can be increased while reducing the risk of misuse because validated behavioral models and the associated application context are published to the repository. Additionally, engineering designers can rely on behavior simulations to further explore the design space. Higher level reuse of behavioral models is achieved by making corporate knowledge available across the extended enterprise. Engineering designers and analysts can access the repository and access the executable model and the knowledge and expertise employed in creating the model. The gap between design and analysis is decreased by providing engineering designers with increased knowledge and understanding about behavioral simulation. Future work and ongoing research includes the following

(1) Further instantiating of the behavioral model repository,

(2) Refinement of the knowledge representing using ontology languages - the use of semantic languages and ontologies (e.g. OWL, etc.) will enable behavioral models to be related and achieve a higher level of semantic richness

(3) Next-generation implementation to support instantiation with design parameters for execution.

(4) Development of a refined behavioral model knowledge representation and

(5) Knowledge-based mappings between design model parameters and behavioral model parameters, thus linking the product design specification and behavioral model

5.3 Example Case 3: Integrating Analysis Models in the Design of Linear Cellular Alloys [56]

The overarching goal in this research area is to develop a framework for capturing idealized analysis models and the associated idealizations that map product parameters and design models. By capturing both the idealization and the analysis models additional product development knowledge can be captured to support knowledge-based design analysis integration. The focus of this research area is put into context of design analysis integration (see Figure 30).

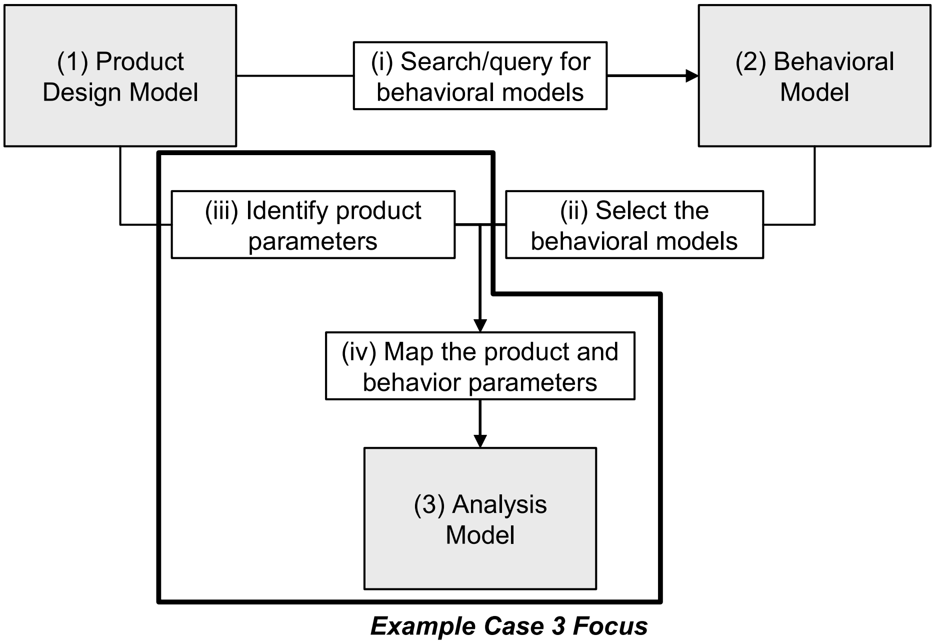

(1) Product Design Model
(i) Search/query for behavioral models
(2) Behavioral Model
(iii) Identify product parameters
(ii) Select the behavioral models
(iv) Map the product and behavior parameters
(3) Analysis Model
Example Case 3 Focus

Figure 30. Focus of analysis model idealization hyperspace research in DAI effort.

In Section 5.2, a repository for characterizing and capturing behavioral models is proposed. Behavioral models, based on engineering first principles or empirical data, may be used in a variety of design situations for different products. For example, a beam equation can be used to compute the deflection or stress in an I-beam, square shaped beam, or even a complex Linear Cellular alloy material (see Figure 31).

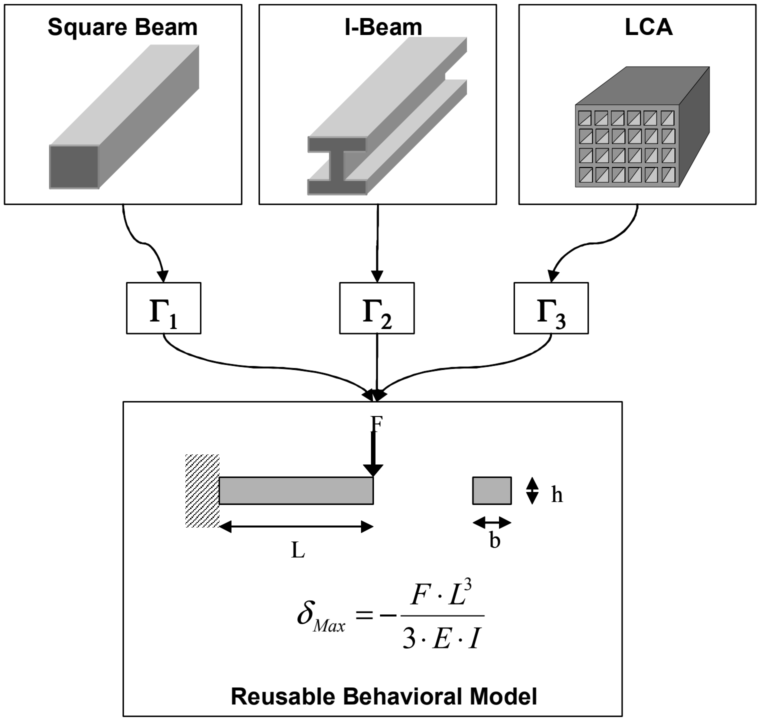

Square Beam
I-Beam
LCA
Γ_1
Γ_2
Γ_3
F
h
b
L
$\delta_{Max} = -\frac{F \cdot L^3}{3 \cdot E \cdot I}$
Reusable Behavioral Model

Figure 31. Mapping design parameters to behavior model through analysis idealizations.

While the same behavioral model is employed to determine design parameter, different idealizations are used to map design parameters into the beam behavioral model. Fenves [3] illustrate this in Figure 4 from a conceptual level. The focus of this research thrust is on capturing the resulting behavioral models and the idealizations used in creating the analysis model.

5.3.1 Description of Linear Cellular Alloys

For demonstrating the requirements of a knowledge-based design analysis integration framework, we present an example scenario related to the design of Linear Cellular Alloys (LCAs). Linear Cellular Alloys are metallic cellular materials with a constant cross section, fabricated through a process developed by the Lightweight Structures Group at Georgia Tech. The process combines extrusion of ceramic slurry, composed of metal oxides and water through a die, allowing for the achievement of quasi-arbitrary two-dimensional cellular topologies. Extrusion of the ceramic is followed by exposure to thermal and chemical treatments that cure the composites. The inherent advantage in producing materials using this process is the ability to tailor properties of the resulting structure such as the effective moduli of elasticity and conductivity by altering the cell topologies.

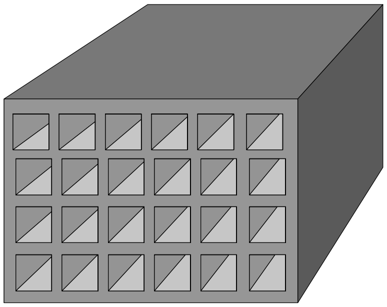


Figure 32. The shape of Linear Cellular Alloys [57].

LCAs are cellular materials with extended prismatic cells (see Figure 32). Structures may be composed of either periodically repeating unit cells or functionally-graded, non-uniform cells of various topologies. LCAs can be manufactured with arbitrary cross-sections (see Figure 33 for representative examples). LCA wall thicknesses are generally in the range of a few hundred microns.

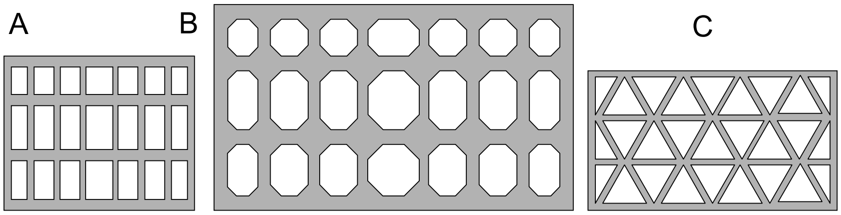


Figure 33. Examples cross-section of LCAs [57].

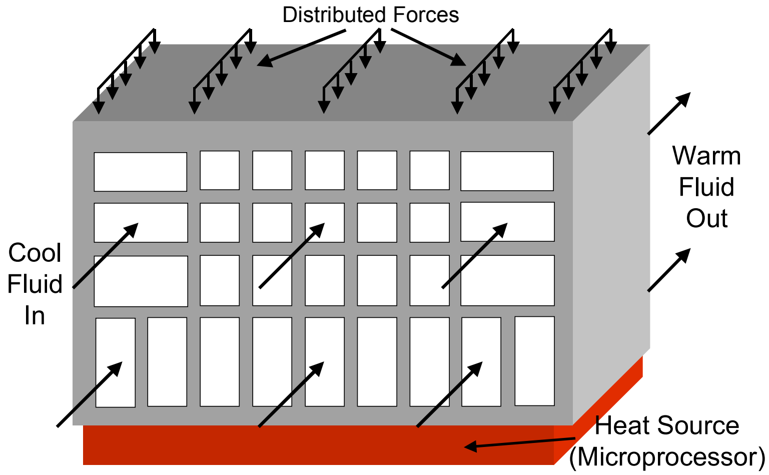


Figure 34. A conceptual illustration of an LCA as a structural heat transfer device for an electronic cooling application.

LCAs are suitable for multi-functional applications that involve not only structural but also thermal considerations (see Figure 34). One of the main advantages of using LCAs is that desired material properties can be obtained by design. Potential applications of LCAs include heat sinks for microprocessors and combustor liners for aircraft turbines, among others.

5.3.2 LCA Design Specifications and Considerations

The models used for LCA design include information about *form, function*, and *behavior*, the definitions of which, used in this paper, are taken from [25]:

Form Represents physical characteristics and includes aspects such as geometry and material properties.

Function Represents the artifact's intended behavior. An artifact satisfies engineering requirements through its function.

Behavior Represents how the artifact implements its function.

Analysis models, used in LCA design, map *form* to *behavior* in order to evaluate the satisfaction of *functional* requirements. Relevant considerations for the design of LCAs, relating to form, function, and behavior are discussed next.

Form Characteristics. *Unit cell topology*: This includes the shape of each unit cell, which can be triangular, rectangular, hexagonal etc. (see Figure 33)

Arrangement of cells: These repeating cells can be arranged in a number of different configurations, captured in the form.

Unit cell dimensions (and possible ranges): The dimensions of each cell can vary, resulting in graded structures. In the product model, we thus need to represent each dimension separately.

Geometric Constraints: Limits on overall dimensions like length, width, rib dimensions, and aspect ratio must be specified.

Dimensional uncertainty: Dimensions are not exact due to variations in the manufacturing process. This uncertainty must be represented in the model.

Bulk material properties: Constituent solid material must be represented in terms of properties such as thermal conductivity, porosity, Poisson's ratio, density, etc.

Function and Behavior Characteristics. *Thermal requirements*: Amount of heat to be transferred from the surface per unit time or maintaining a certain temperature at a given point.

Structural requirements: The strength of the LCA is predominantly a function of the form. This strength is quantified in terms of the effective elastic stiffness, buckling strength, and compliance. The inherently complex data structures (e.g., tensors) must be captured effectively.

Manufacturing requirements: The manufacturing process greatly influences the design considerations of LCAs. There is a limit to the accuracy that the manufacturing process can achieve. For example, porosity plays a role in the behavior of the material and needs to be taken into account during the design process. Other factors related to manufacturing process include defects in cell walls and joints as well as tolerances.

Pressure drop: Generally, fluid is forced through the channels of the LCA to achieve convective cooling. In the case of a CPU heat sink, forced convection is achieved through the use of a fan, the capacity of which is limited by the pressure drop. Similar considerations apply to combustor liners.

Other behavioral information: The design models must capture behavioral information that includes boundary conditions, structural responses (for e.g., stresses, strains, etc.) and thermal responses (for e.g., heat transfer, temperature at notes, etc.). Information regarding uncertainties in behavior evaluation is also important.

LCA product specifications are determined through a well-defined development process. Designers systematically determine the product form specifications through a development process that relies heavily on design-analysis. Engineering designers and analysts work collaboratively to specify the LCA form based on functional requirements and simulate the behavior of the LCA in an iterative manner toward the final design specification. An overview of the LCA development process is presented in the following section in order to more clearly illustrate this point.

5.3.3 Process for Designing LCAs

The LCA development process is shown in Figure 35. The process consists of six steps starting with gathering customer requirement and formulating the desired behavioral aspects of the LCA and culminating with optimization of the LCA form.

1) Capture customer requirements and determine behavior - customer requirements are captured and formalized into engineering specifications. Based on these requirements, functional and performance characteristics are expressed in terms of LCA behavior.

2) Specify the LCA Design form – the LCA design form is embodied based on expected requirement and designer's knowledge and experience using CAD tools.

3) Numerical Simulation – numerical simulations are performed to determine the simulated behavior of the LCA. Numerical simulation consists of two primary steps (1) simulation model generation and (2) mathematical modeling. Thermal and structural simulation models are developed from design models and a set of idealizations. Mathematical modeling, then maps the simulation model into the appropriate mathematical formulation. In LCA simulation, mathematical modeling is the finite element method of computational fluid dynamics.

4) Evaluate Simulated Behavior – the simulated behavior of the LCA is compared against the desired behavior (function). If the two do not match, appropriate changes are made to the form parameters to obtain the desired performance.

5) Optimization Decision and Optimize LCA Design - optimization techniques are employed in the form of the compromise Decision Support Problem (cDSP) technique [57] to determine the final geometry of the LCA to best meet behavioral performance requirements.

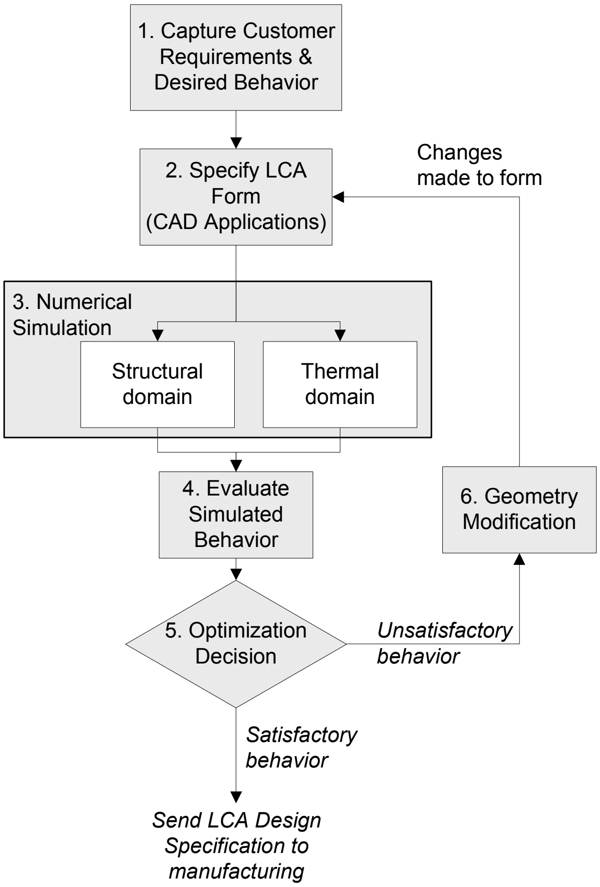


Figure 35. The process for designing LCAs.

To enable the LCA development process to be completed efficiently and effectively, the following is needed:

- LCA design models must be formalized and include relevant design information, such as function and form

- LCA analysis idealizations must be encapsulated and characterized to promote efficient reuse for generating analysis models used in numerical simulation

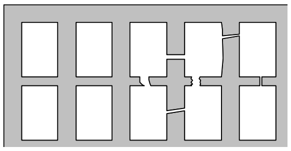
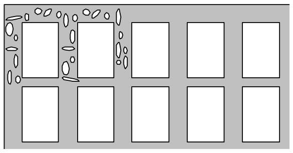
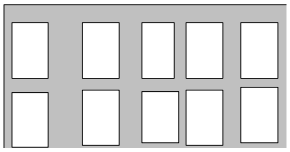
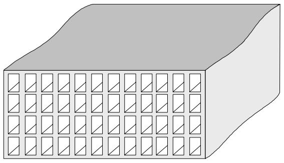
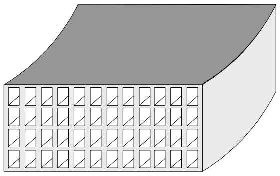
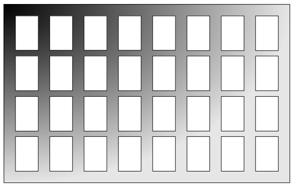
- The resulting simplified LCA analysis models must be archived to enable knowledge-based retrieval to reduce the design-analysis cycle time

The overarching goal in the LCA development process is the specification of the LCA form such that it can be manufactured. In contrast, the goal of analysis is to simulate the behavior of what is actually built. As a result, engineering designers often idealize the design form to enable simulation. In the following section, we identify several idealizations commonly employed by engineering analysts when creating simplified LCA analysis models.

5.3.4 Idealization in LCA Design and Analysis

Ultimately, analysis models are generated based on the design form and an appropriate set of idealizations. The LCA idealizations are roughly decomposed into *Form Idealization* and *Behavior/Functional Idealization*. Form idealizations are used by designers to create a simulation model that operates on the form of the LCA model, including geometry, material and topology considerations (see Table 8)

Table 8. Representative set of form idealizations used in creating LCA simulation models.

Form Idealization	Description
	Truss members in the LCA structure may be imperfectly connected or breaks or fractures may be present in the structure
	Voids in the material continuum may be present due to manufacturing
	Manufacturing variability may result in wall thickness variations and dimensional variations of cell in the LCA
	Sides of overall LCA and cells within LCA are not parallel
	LCA form is warped due to manufacturing – internal stresses or imperfect heat transfer may result because of shape
	Inhomogeneous material properties and variations in density, thermal conductivity, strength of the LCA due to manufacturing may affect behavior

Behavior/Function idealizations operate on boundary and loading conditions and underlying behavioral models (see Table 9)

Table 9. Representative set of behavioral and functional idealization used in creating LCA simulation models.

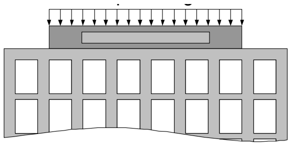
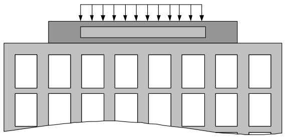
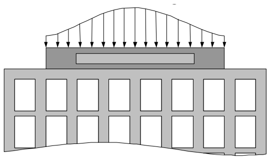
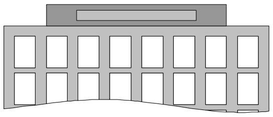
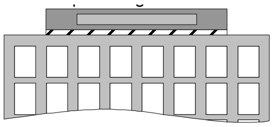
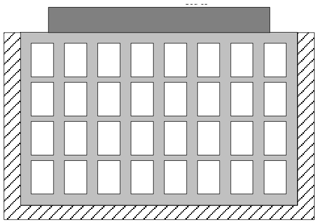
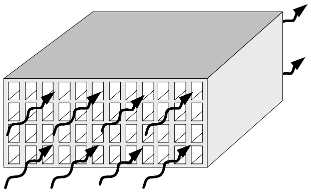
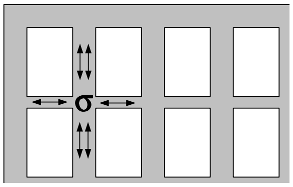
Behavior/Function Idealization	Description
	Uniform heat flow from the entire chip package into the LCA is assumed
	Uniform heat flow from die in the chip package into the LCA is assumed
	Non-uniform heat flow from the entire chip package into the LCA is assumed
	No contact resistance between microprocessor and LCA
	Contact resistance considered between microprocessor and LCA
	Perfect insulation is considered on three sides of LCA.
	Uniform flow and no pressure loss of fluid through cells
	Internal stresses in LCA structure may effect overall structural performance

Table 8 and Table 9 illustrate several common idealizations used in the design of LCAs. The list is, however, by no means comprehensive. These idealizations represent a set of operations that engineering designers can use to develop analysis models of varying fidelities. For example, an analyst may determine that fractures in the LCA structure will complicate and increase the simulation time, but the increase in accuracy of the result may provide insight to make key design decisions. Essentially, thus, engineering analysts construct an analysis model based on a design model and a set of appropriate idealizations.

5.3.5 A Strategy for Integrating Design and Analysis through Knowledge Based Idealizations

Considering the numerous challenges associated with integrating design and analysis activities, we propose a knowledge-based approach to decrease the overall product development cycle time. Specifically, we present the conceptualization and initial development of a knowledge-based framework for capturing analysis knowledge and expertise to reduce overall product realization time. The overarching goal is to enable the modular reuse of analysis idealizations, thus reducing the knowledge gap between engineering design and analysis activities. Design-analysis integration is supported by synthesizing domain expertise as follows:

- Capturing knowledge about analysis models, in a robust fashion that stands in contrast to the brittle nature of expert systems. Idealizations are defined at various levels of abstraction, thereby providing greater reusability. This provides increased context about the analysis models that are used to support design decisions.
- Capturing and formally characterizing idealizations in progressing from design to analysis; thereby increasing the knowledge gained throughout product development. To enable analysis reuse, idealizations are captured using storable building blocks.
- Capturing the domain specificity of idealizations which is important for determining the scope of applicability. Much the same is true for capturing analysis context.
- Selecting appropriate sets of idealizations depending on the current phase in the design process.

These four tasks can be broadly divided between two main research thrusts, namely capturing knowledge and retrieving it, described in Sections 4.1 and 4.2, respectively.

5.3.6 Developing a Knowledge-Based Repository of Analysis Models

Capturing Knowledge Characterizing Analysis Models

The first step towards creating a knowledge-base for analysis models is capturing all the relevant knowledge that can affect the applicability of a model. Such factors include context, scope, simplifying assumptions, domain, results, information requirements (inputs) and contributions (outputs), accuracy, level of detail, fidelity, complexity, and scalability of a given model.

Structuring Knowledge by Mapping Design Models to the Appropriate Analysis Models using a Hyperspace: A Conceptual Architecture for an Analysis Model Repository

The conceptual architecture of an analysis model repository can be viewed as an n-dimensional hyperspace, with each dimension pertaining to a different type of idealization performed while designing artifacts. In Figure 36, we illustrate a three-dimensional analysis model hyperspace where the three dimensions are geometric idealization, behavioral idealization, and boundary condition idealization. Each point in this hyperspace represents a different analysis model. As we proceed in the positive direction along each axis, models become more and more simplified. For example, Model B constitutes a simplification of Model A with regard to artifact geometry. All other characteristics of the model like boundary conditions and behavioral model remain the same. This is apparent from Figure 1, since there is only a change in the geometry dimension. Similarly, Model D makes use of idealized boundary conditions when compared to model B.

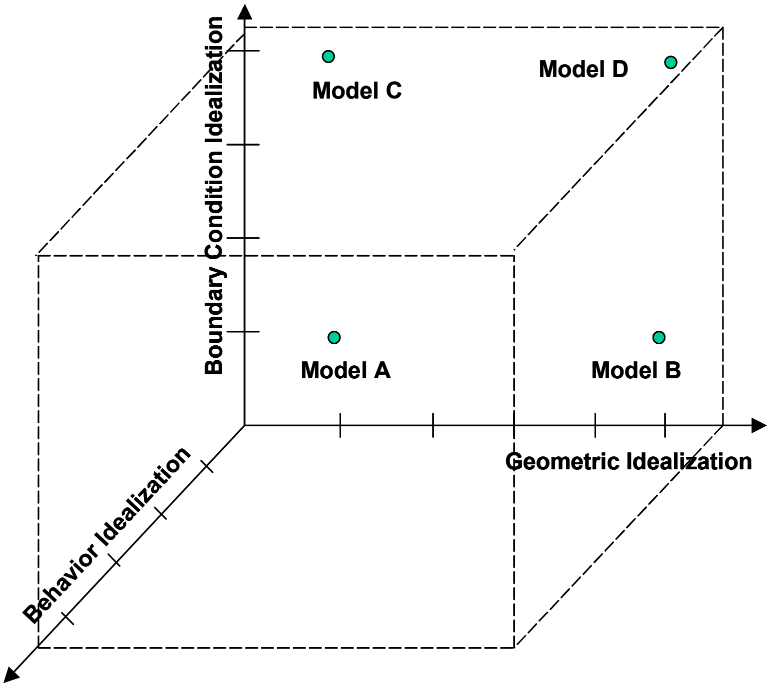


Figure 36. The Analysis Model Hyperspace.

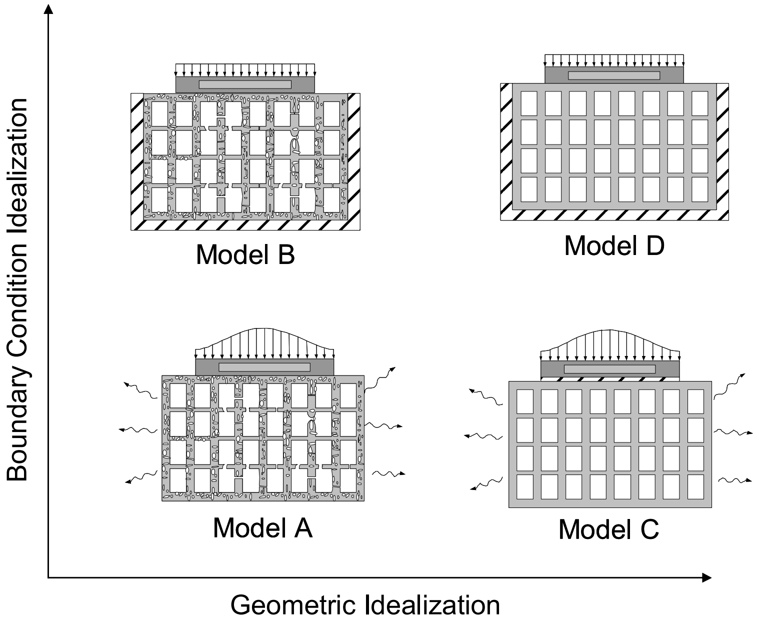


Figure 37. LCA analysis models hyperspace.

An example of the analysis model hyperspace for LCAs using the idealizations from Table 1 and Table 2 is depicted in Figure 37. The hyperspace is presented in two dimensions: Geometric Idealizations and Boundary Condition Idealizations.

Table 10. Description of LCA analysis models in hyperspace.

Model A: • Void and fractures are considered in the LCA geometry • Radiation heat transfer from the LCA to the environment • Non uniform heat from the microprocessor • Contact resistance between the heat source and the LCA
Model B: • The simulation geometry is identical to the design geometry • Radiation heat transfer from the LCA to the environment • Non-uniform heat from the microprocessor • Contact resistance between the heat source and the LCA
Model C: • Void and fractures are considered in the LCA geometry, • Radiation heat transfer from the LCA to the environment • Perfectly insulated on three-sides • Uniform heat from the microprocessor • Contact resistance between the heat source and the LCA is not considered
Model D: • The simulation geometry is identical to the design geometry • Perfectly insulated on three-sides • Uniform heat from the microprocessor • Contact resistance between the heat source and the LCA is not considered

In general, there are no restrictions on the number of dimensions considered in the analysis model hyperspace. Dimensions must have noticeable impacts on the analysis results. It is evident that the n-dimensional hyperspace offers a convenient way to isolate and organize idealization effects.

Directions in the model space may further be viewed as subspaces. For example, the boundary conditions may be subdivided into loads and supports. The loading may further be split into structural loading, thermal loading, etc. Hence, the analysis model hyperspace is inherently based on a hierarchical structure that can be captured using a tree-like construct as commonly employed in the design of repositories.

The approach suggested here relies on the use of a "destruction tree" for systematically capture simplifications in analysis models. This "destruction tree" is akin to the construction tree in constructive solid geometric modeling, where detailed models are created from basic geometric shapes and Boolean operations. Hayes and Regli [58] present the model process history as a similar extension to the traditional construction tree to address how the design changes throughout product development. The idealizations in the "destruction tree" are captured in a knowledge base for reuse in different design scenarios. Additionally, capturing idealization knowledge facilitates the selection and creation of the most appropriate analysis models. Associativities between design and analysis models along with the impact of idealization on the predicted behavior are also captured. The hierarchical relationships between simulation models (SM) and a design model (DM) are illustrated in Figure 38. The design model is related to simulation models of varying complexity through a set of idealizations (□). As depicted in Figure 38, the idealization level of the simulation models are inversely related to the idealization set.

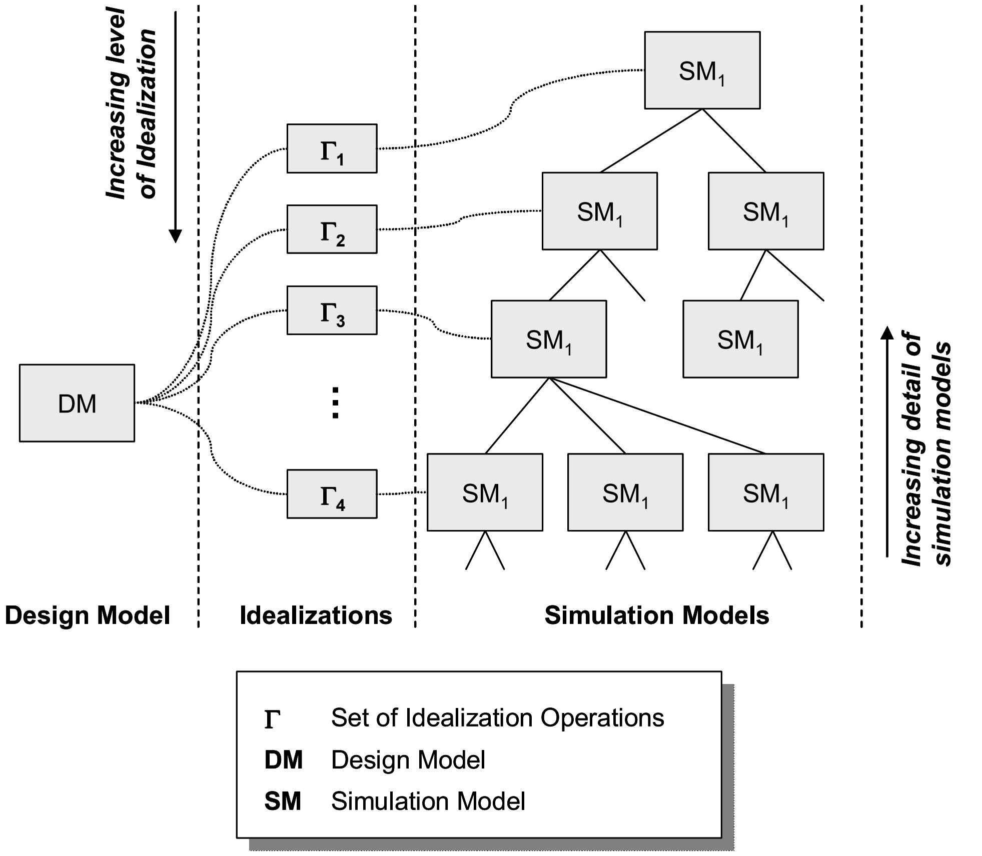


Figure 38. Hierarchical decomposition and idealization relationships between analysis models.

It is important to realize that the hierarchical structure of the analysis model repository can further be taken advantage of in evaluating the effect of different idealizations on the analysis results and computation time required. Generally, idealization of a model leads to increased error and reduced analysis time. A tradeoff between analysis time and accuracy can be obtained by selecting an appropriate model from the analysis model hyperspace. If prior knowledge about the analysis models and associated error is captured in the knowledge repository, the designers can appropriately select a model by moving accordingly throughout the analysis space illustrated in Figure 36.

Implementing Flexible, Hierarchical Design-Analysis Templates

An important prerequisite for effectively using the analysis model hyperspace discussed previously is the ability to integrate the design model with various fidelities of analysis models. The fundamental constructs for doing so are *flexible, hierarchical idealization templates* that model the associativity between these two kinds of models. We thus propose a hierarchical, object oriented model for *idealization templates*. This is in contrast to the static mapping templates generally used for this purpose.

5.3.7 Knowledge-Based Retrieval of Analysis Models

Moving Along the Analysis Model Hyperspace

The analysis model hyperspace can be viewed as a design space and the response is a combination of accuracy and cost. The key assumption here is that the knowledge base contains information about all the models and their impact on accuracy, time, cost, etc. Assuming such information is readily available from the knowledge base, the destruction tree can be used to map out a strategy for attaining improvements with regard to any of the dimensions considered in the model hyperspace.

Progression along a design timeline may be illustrated via movement throughout the analysis model hyperspace. This is due to the fact that the appropriateness of a model is greatly dependent upon the current stage in the design process, information requirements, and a designer's knowledge. In the early stages of the design process, model accuracy is usually not of great concern. In fact, the detailed models are not even available. Consequently, designers must rely on simplified models for quick exploration and evaluation of artifact behavior. Towards the latter stages of a design, however, more detailed models are available and can be used to evaluate product performance. The strategy for moving along the analysis hyperspace is determined by careful consideration of tradeoffs between metrics.

Metrics for Assessing Model Applicability in a Given Scenario

In order to select the model most appropriate for obtaining the desired behavioral information about the design, there is a need for characterizing the models with regard to a set of quantifiable metrics. Some of these metrics are: *computational time* required for execution of the model, *accuracy* in results of analysis, value of information obtained with respect to designer requirements, *relevance* of the model to the analysis situation in terms of the context and application domain, *uncertainty* associated with the underlying behavioral model, *validity* of the assumptions made in constructing the model, the *level of detail* captured in the model, the *level of abstraction* that is of interest to the designer, *complexity* of operations in terms of executing the model, *adaptability* of the model to different situations, *modularity* of the model with regard to interfacing with other models, and *robustness* of the model. Multiple metrics can be of interest to designers while selecting a model. Relying on the right set of metrics in choosing among available analysis models will force a designer to remain conscious of opportunity costs and carefully consider the tradeoffs involved. This approach will help reducing the design-analysis iterations and promises a closer integration between the two domains.

5.3.8 Analysis Model Hyperspace and Idealizations Closure & Future Work

In this research, we highlight the need for integrating design and analysis activities and propose a knowledge-based framework for archiving and accessing analysis models and associated idealization. This framework has the potential to reduce design-analysis iterations significantly. Specific research issues associated with developing a knowledge-base and extracting appropriate analysis models are discussed. The knowledge-base proposed in this work is based on a conceptual analysis model hyperspace which can be used to model different fidelities of analysis models in a hierarchical fashion. Metrics are proposed for characterizing analysis models to facilitate selection of the models during product development. The enhanced integration between design and analysis models is achieved through flexible object-oriented idealization associations. These templates form a critical part of the design-analysis knowledge base. Future work includes

1) Instantiating analysis model hyperspaces using object-oriented information modeling standards like STEP EXPRESS,

2) Developing object-oriented models for a hierarchy of idealizations, and

3) Using multi-objective selection methods for selecting the right model from a model hyperspace.

Finally, immediate research bringing together the contribution of the Behavioral Model repository (see Section 5.2) and the Analysis Model Hyperspace to reduce the integration gaps between design and analysis activities.

5.4 Example Case 4: Chip Package Design-Analysis Integration Framework [59, 60]

For demonstrating the requirements for realizing a computer-based design analysis integration framework, we present an example of scenario related to the design of an electronic chip package. The overarching goal in this research area is to develop a computer-based framework for integrating design and analysis models in the domain of electronic chip package design. The focus of this research area is put into context of design analysis integration (see Figure 39).

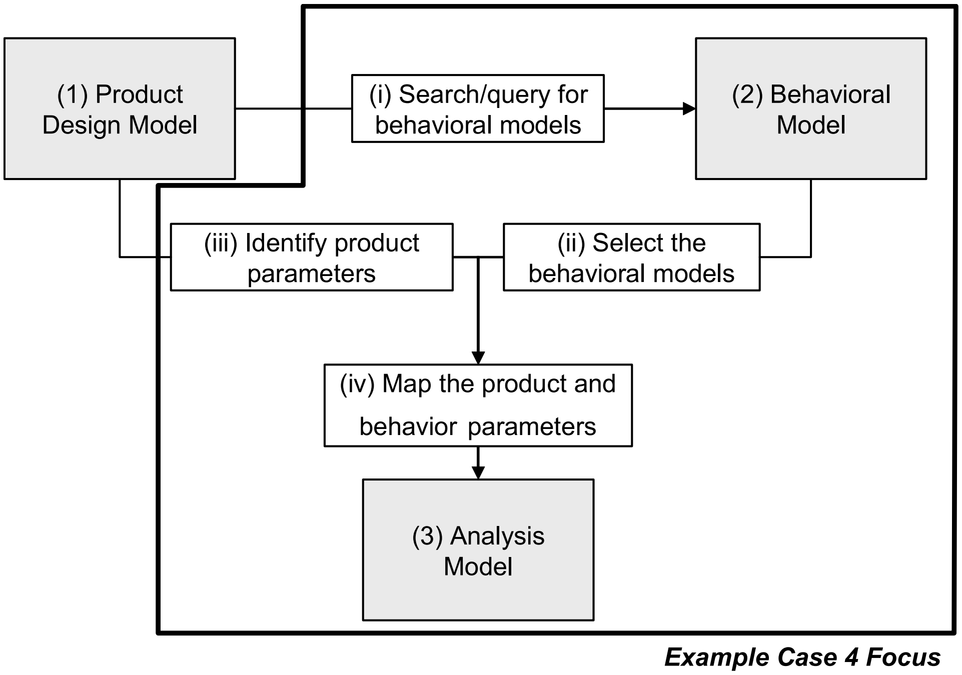
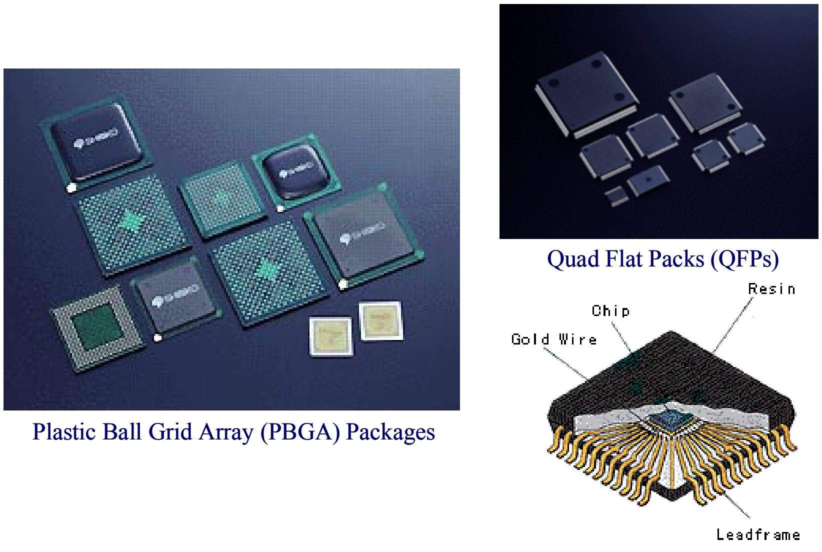

(1) Product Design Model
(i) Search/query for behavioral models
(2) Behavioral Model
(iii) Identify product parameters
(ii) Select the behavioral models
(iv) Map the product and behavior parameters
(3) Analysis Model
Example Case 4 Focus

Figure 39. Focus of chip package design-analysis integration framework.

Electronic chip packages are complex systems that often require design considerations from multiple disciplines (see Figure 40). These disciplines may include, but are not limited to thermal management and temperature distribution within the chip and stress and strain in the chip due to thermo-mechanical behavior. To verify the behavior of the electronic chip package design meets the requirements, behavioral simulations are required from multiple disciplines. Computer-based simulations are become increasingly important during product development to meet the changing needs of the market including higher power dissipation, higher speed, increased electronic pin counts, and smaller chip footprints. Multidisciplinary computer-based simulations enable designers to quickly validate the behavior of the chip from many perspectives. In the design of chip packages, FEA is used to simulate the temperature distribution and deflection due to thermal loads.

As previously stated, the integration gaps between engineering design and analysis tools and the associated models hinder the ease in which the behavior of electronic chip packages can be simulated. Additionally, there is a bottleneck in creating discipline-specific simulation models. This bottleneck occurs because the chips are composed of multiple bodies that are highly coupled, numerous materials, and many complex shapes.


Quad Flat Packs (QFPs)

Plastic Ball Grid Array (PBGA) Packages

Figure 40. Examples of chip package products.

In the following sections, we present the development of a computer-based framework to capture knowledge between design and analysis and automate the simulation process. Much of the work presented in this study is an embodiment of the MRA for integrating design and analysis models. This case serves as an example of a computer-based integration framework as well as highlighting the problems that occur in an actual design situation.

5.4.1 Chip Package Design Specifications and Considerations

The design models of electronic chip packages include product information pertaining to *form, function*, and *behavior* as follows (see Figure 41and Figure 42:

Form Characteristics

Chip: This is an electrical component. The chip is idealized as a hexahedron.

Substrate: The substrate extends the connection areas from the chip to the boardThis is for extending connection area from a chip to a board, which is idealized as multi-layered plates.

Die Attach: The die attach is used to connect the chip to the substrate material. The die attach is idealized as hexahedron.

Mold: The mold protects and wraps the chip for electrical insulation and for heat dissipation.

Solder Ball: Solder balls make the physical connection between the chip and the substration and the board. The solder balls are idealized as cubes

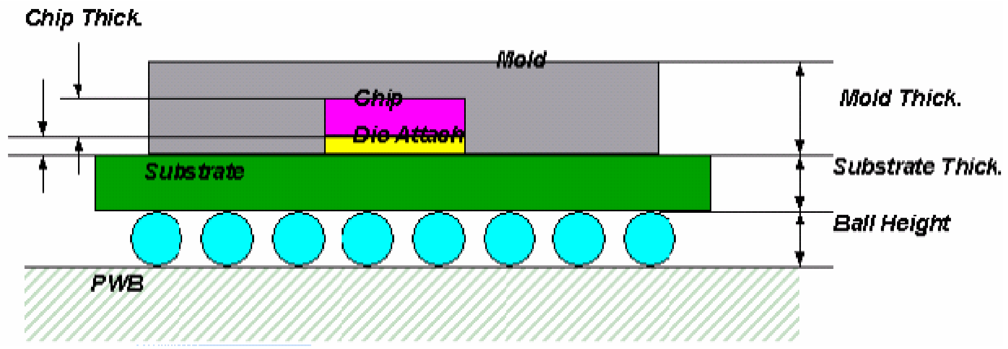


Figure 41 Side view of electronic chip package (Plastic ball grid array type is shown)

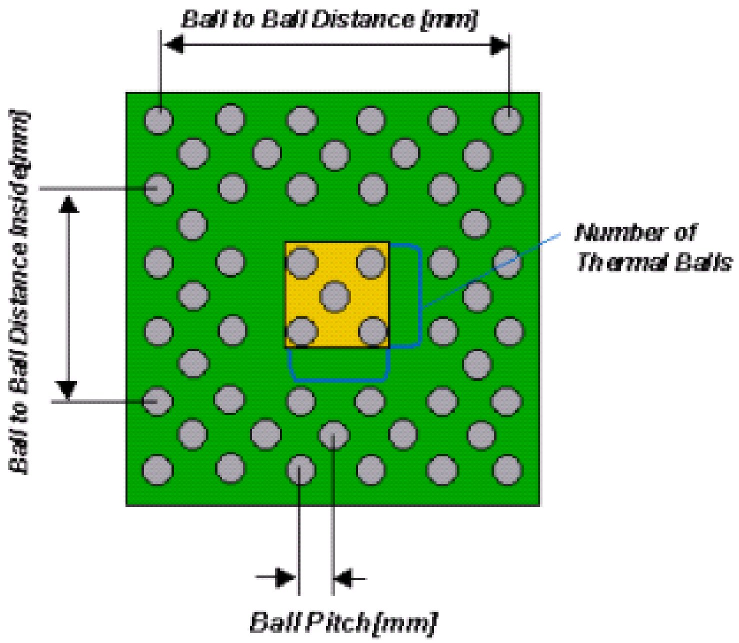


Figure 42. Top view of electronic chip package (Plastic ball grid array shown)

Function and Behavior Characteristics

Thermal requirements: Amount of heat to be transferred from the surface per unit time.

Structural requirements (1): The maximum stress of the solder ball should be less enough compared to its yield strength.

Structural requirements (2): The maximum z directional deformation of substrate should be less than the required amount to guarantee the electrical performance.

5.4.2 Research Questions of Chip Package design and analysis

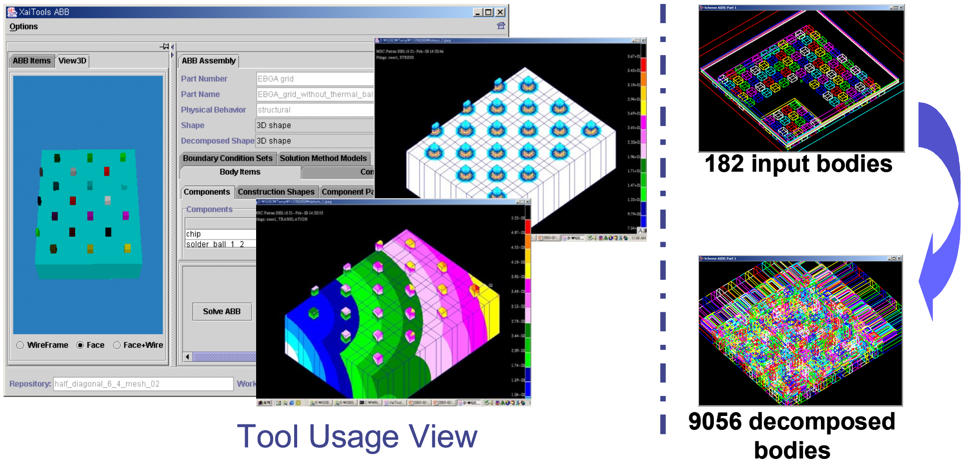
The following four research questions are proposed in design-analysis integration. While these questions are presented in the context of electronic chip package design, they are applicable across many engineering product development domains.

(1) How is the bottleneck of design model manipulation and manual processing be removed or reduced to achieve a higher level of automation?
(2) How is an interoperable design and analysis integration framework achieved?
(3) How is a distributed design and analysis environment supported?
(4) How can existing standards be used, or leveraged to support design and analysis modeling?

These questions and challenge are answered or solved through the following Chip Package Analysis Automation project sponsored by Shinko.

5.4.3 Electronic Chip Package Analysis Automation

The framework developed in this project is aimed at automating thermal and thermo-mechanical analysis of electronic chip packages. Figure 43 illustrates the resulting computer application, titled XaiTools Chip Package (XCP). A major contribution of this work is the automation of the labor-intensive process of “chopping” the product geometry in preparation for meshing. The geometry of the chip package is chopped into bodies to product a high quality mesh for FEA. Finally, several technologies, including STEP and SOAP technology, are employed to realize the MRA-based framework. The detail components of the XCP framework are explained at the following subsections.



XaiTools ABB

Options

ABB Items | View3D

○ WireFrame ● Face ○ Face+Wire

ABB Assembly

Part Number: EBGA grid

Part Name: EBGA_grid_without_thermal_bal

Physical Behavior: structural

Shape: 3D shape

Decomposed Shape: 3D shape

Boundary Condition Sets | Solution Method Models

Body Items

Components | Construction Shapes | Component Pa

Components: chip, solder_ball_1_2

Solve ABB

Repository: half_diagonal_6_4_mesh_02

Figure 43 XaiTools Chip Package (XCP): An automated system for chip package analysis

Knowledge Based system

A knowledge intensive approach is developed to eliminate the human intensive chopping process. This approach is constructed by of two parts: (1) the information model (data structure) and (2) the algorithm (functions). The information part is for capturing the connectivity of objects and the algorithm is for executing decomposition activities. Figure 44 shows a simple example of decomposition. The connectivity between the original two blocks and the decomposition of the blocks are shown. At the implementation, EXPRESS is used information modeling language and ACIS is used for constructing algorithm.

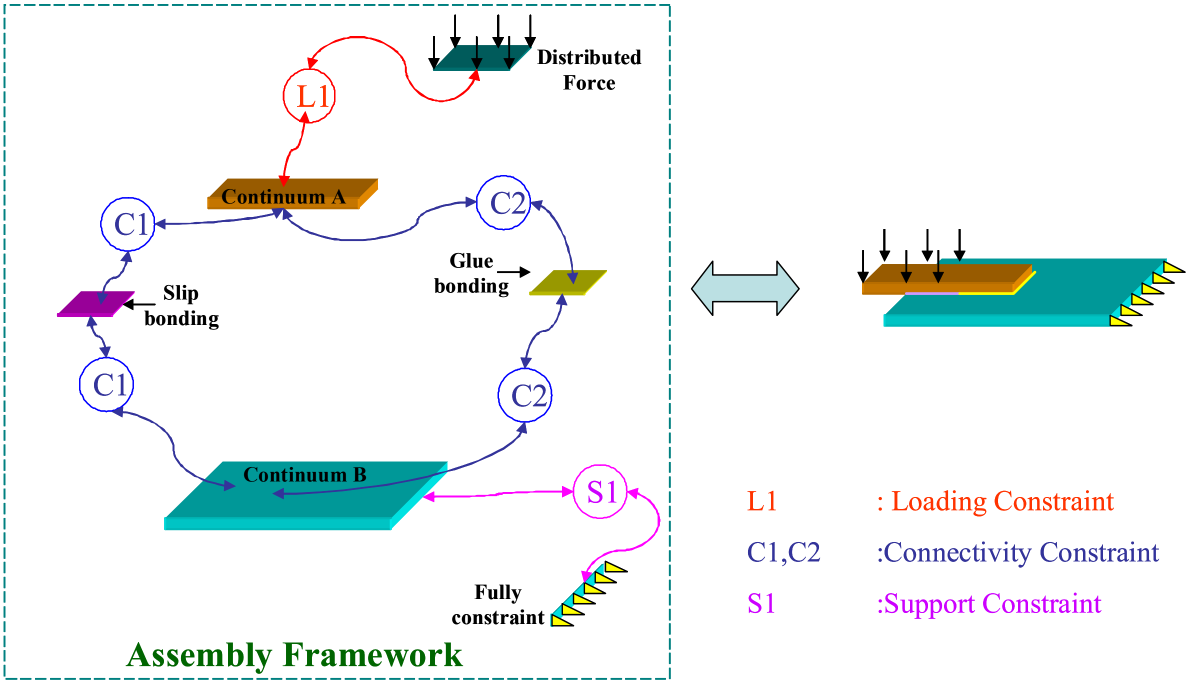


Figure 44 Conceptual illustration of object connectivity in the chopping method.

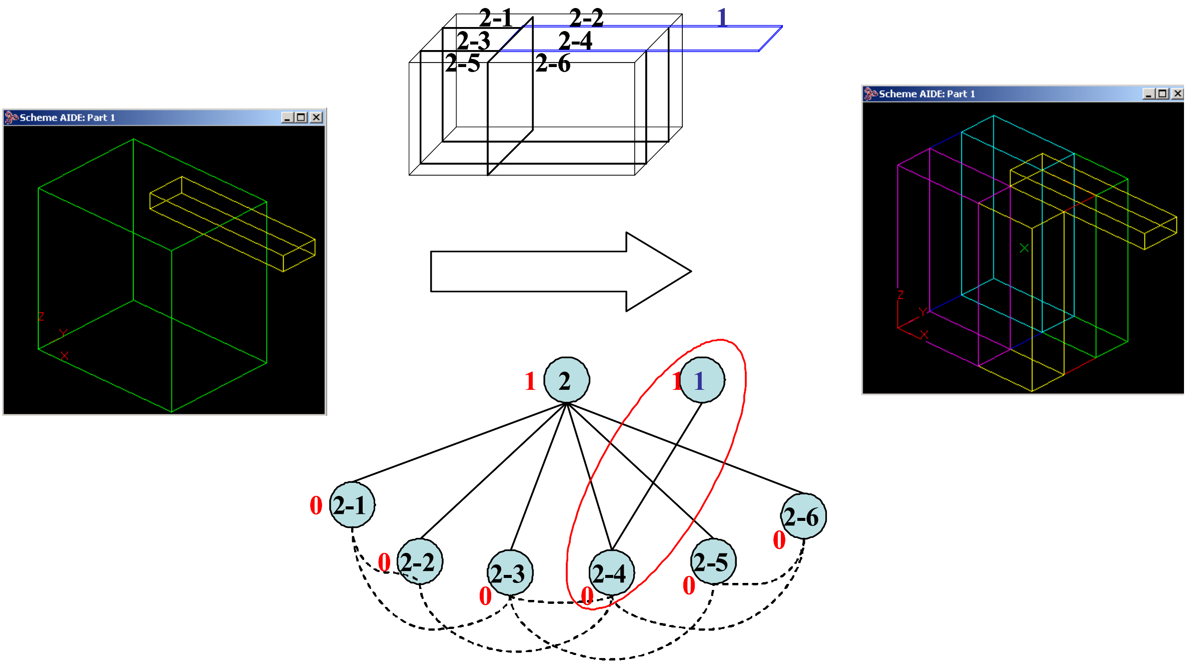


Figure 45. Illustration of the chopping algorithm.

Multi-Representative Architecture (MRA)

The MRA serves as the basis for model integration and system interoperability [61]. The MRA represents multiple models in the design and analysis process. Each model in the MRA represents a *state* of the product (see Figure 46).

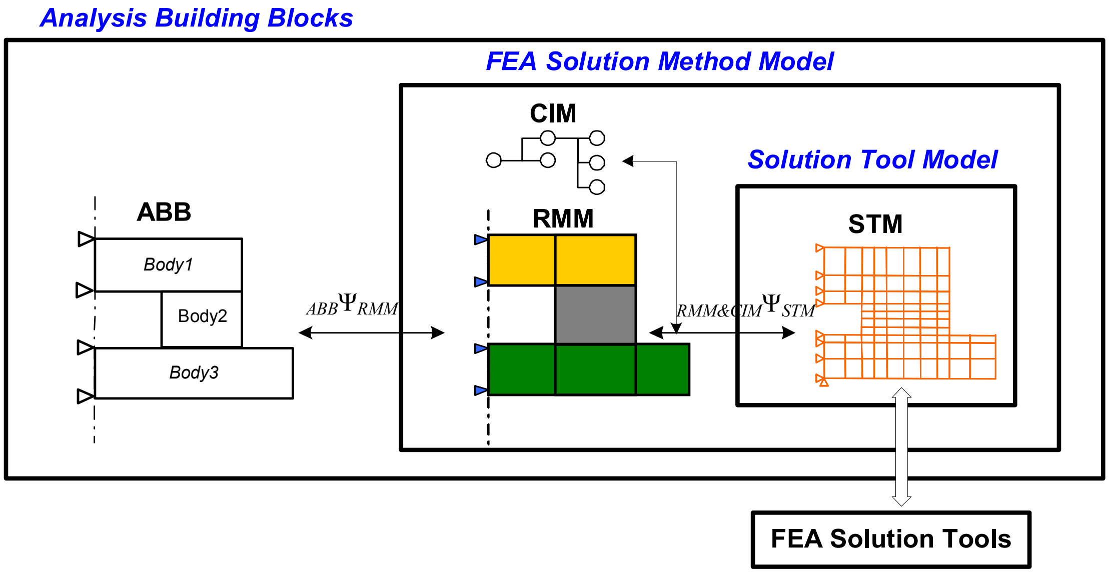


Figure 46 Various levels of analysis

Distributed Design and Analysis Environment

To support distributed design and analysis environment, XCP is integrated with the Internet standard Simple Object Access Protocol (SOAP). SOAP enables worldwide access to solvers located on remote servers. Data exchange and remote procedure call are done between client and server. Therefore efficient system operation is achieved to distribute the light load object like XCP on client and heavy loads like analysis solver and decomposition module on server (see Figure 6). The operation has succeeded between Shinko (Nagano, Japan) and Georgia Tech (Atlanta GA USA) [31].

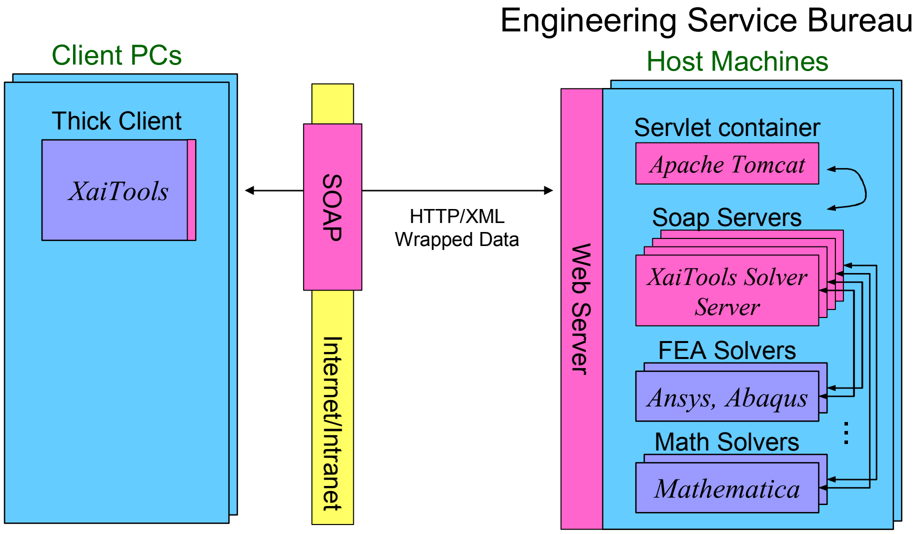
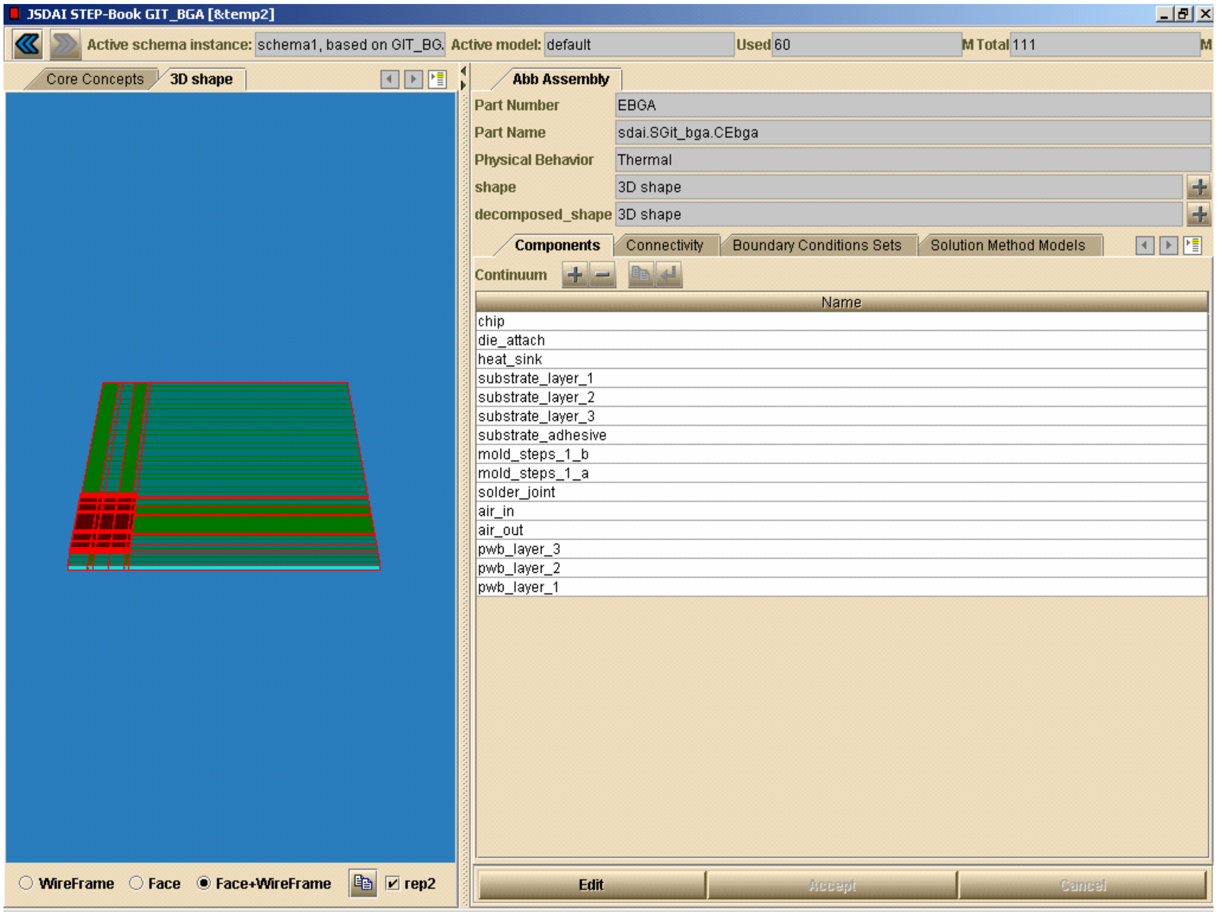


Figure 47. The XCP framework supporting SOAP

Usage of STEP standard

XCP is developed based on STEP technologies such as EXPRESS, SDAI and XML so that it provides a good environment to incorporate STEP APs and modules. To visualize chopped chip package models, the system use a modularized STEP schema (aic514_advanced_brep) and a standard based application module (LKSoft 3D viewer). Figure 7 shows the 3D view of chopped chip package which is realized by the STEP schema and LKSoft 3D viewer.



Active schema instance: schema1, based on GIT_BG. Active model: default Used 60 M Total 111 M

Part Number: EBGA

Part Name: sdai.SGit_bga.CEbga

Physical Behavior: Thermal

shape: 3D shape

decomposed_shape: 3D shape

Continuum

Name
chip
die_attach
heat_sink
substrate_layer_1
substrate_layer_2
substrate_layer_3
substrate_adhesive
mold_steps_1_b
mold_steps_1_a
solder_joint
air_in
air_out
pwb_layer_3
pwb_layer_2
pwb_layer_1

○ WireFrame ○ Face ● Face+WireFrame ☑ rep2

Figure 48 Usage of STEP Standard (aic514_advanced_brep)

6 Summary of Accomplishments and Research Contributions

As a result of the work completed in this research several contributions to design-analysis integration are made. The contributions are tied to research questions and task proposed in Section 4. Additionally, specific research contributions are discussed in subsequent sections. The example research areas are mapped to research question in Table 11.

Table 11. Mapping of research questions to case studies.

	Example / Research Thrust			
Research Questions	**PWB Warpage**	**LCA – Analysis Models**	**Behavioral Models**	**Chip Package Design**
How can behavioral models be captured to support reuse in product development?			✓	
How can the idealizations employed by analysis experts be formally captured to facilitate automated creation of simplified analysis models based on engineering design specifications?	✓	✓		✓
How can simplified analysis models used throughout product development be archived for reuse?	✓	✓		✓
How can analysis models be efficiently retrieved to support model-based engineering decision making in product development?		✓		

The research thrusts and example cases help to address the task proposed in this work in the following ways (see Table 12).

Table 12. Research task mapped to case studies.

	Example / Research Thrust			
Research Tasks	**PWB Warpage**	**LCA – Analysis Models**	**Behavioral Models**	**Chip Package Design**
Identify analysis activities in the design process and what type of information is needed to complete each type of analysis.	✓	✓		✓
Complete background research on case-based reasoning and knowledge-based design support.	✓	✓		✓
Develop a formulation of how the CPM model supports the associativity between design models and analysis models for knowledge based and cased based design.		✓	✓	

6.1 Computer-based Framework for Automating PWB Warpage

The PWB warpage framework provides an example of integrating design and analysis models and associated engineering tools. The framework is based on a semantically-rich information model, STEP AP210. The product information captured in AP210 is idealized using analysis idealizations for creating a simplified analysis model. The computer-based framework provides the infrastructure for integrating design and analysis models developed in diverse engineering tools. In the warpage example, design models created in ECAD tools and stored in STEP Part 21 files are integrated with ANSYS for analysis.

6.2 Knowledge Based Behavior Models Repository

A knowledge representation for characterizing and repository archiving engineering behavioral models is developed. This knowledge representation enables engineering behavioral models and the meta-knowledge employed in developing the models to be captured in a formalized manner, thus increasing reuse by engineering designers and reducing misuse of the model in incorrect situations. This knowledge representation supports explicit capturing of assumptions, context, and validity of the behavioral models. The behavioral model repository provides engineering designers with an increased understanding related to the model limitations and context of applicability, thus increasing the reuse of the model while decreasing misuse.

6.3 Framework for Archival of Analysis Models and Associated Idealizations

A description and method for associating design, behavioral and analysis models is proposed. In this work, LCAs serve as an example for characterizing and describing idealizations used in product development. A *catalog* of idealizations is proposed for creating simplified analysis models based on design specification and analysis context. Additionally, a conceptual framework is proposed for organizing and archiving analysis idealization and the associated analysis models. The framework enables complex idealization to be systematically captured for associating design and analysis models. The hyperspace and idealization decomposition (see Figure 36, Figure 37, and Figure 38) provide a framework in which multi-disciplinary analysis models of varying complexity can be characterized and retrieved for reuse.

6.4 Electronic Chip Package Design Analysis Framework

The Chip Package Framework provides an additional example of a computer-based design analysis integration framework. While the scope is similar to the warpage framework, the idealizations and models employed are different, thus demonstrating the need to capture many different types of engineering idealizations.

The framework developed in this project is aimed at automating thermal and thermo-mechanical analysis of electronic chip packages. The framework is proposed in the context of the MRA and uses STEP and SOAP technologies for integrating design models and analysis models in product development. Additionally, a geometric chopping algorithm is embedded in the framework to automate and reduce human errors during the creation of analysis models.

6.5 Lexicon for Design Analysis Integration

As an effort towards close design-analysis integration technologies and related research, a lexicon is developed. This work is motivated, in part, by the disparity and significant differences in terminology utilized across the domain of engineering design-analysis integration and simulation-based design. While we do not assert that a unified *ontology* will ever be established, it is necessary as a community to use terminology in a consistent manner. As such, we have developed the DAI lexicon. The lexicon is a web-based dictionary of terms from several researchers in the general umbrella of design-analysis integration. The lexicon is developed in the context of a generalized MRA structure. The illustrative MRA figure (see Figure 10) is illustrated using the UML notation (see Figure 49).

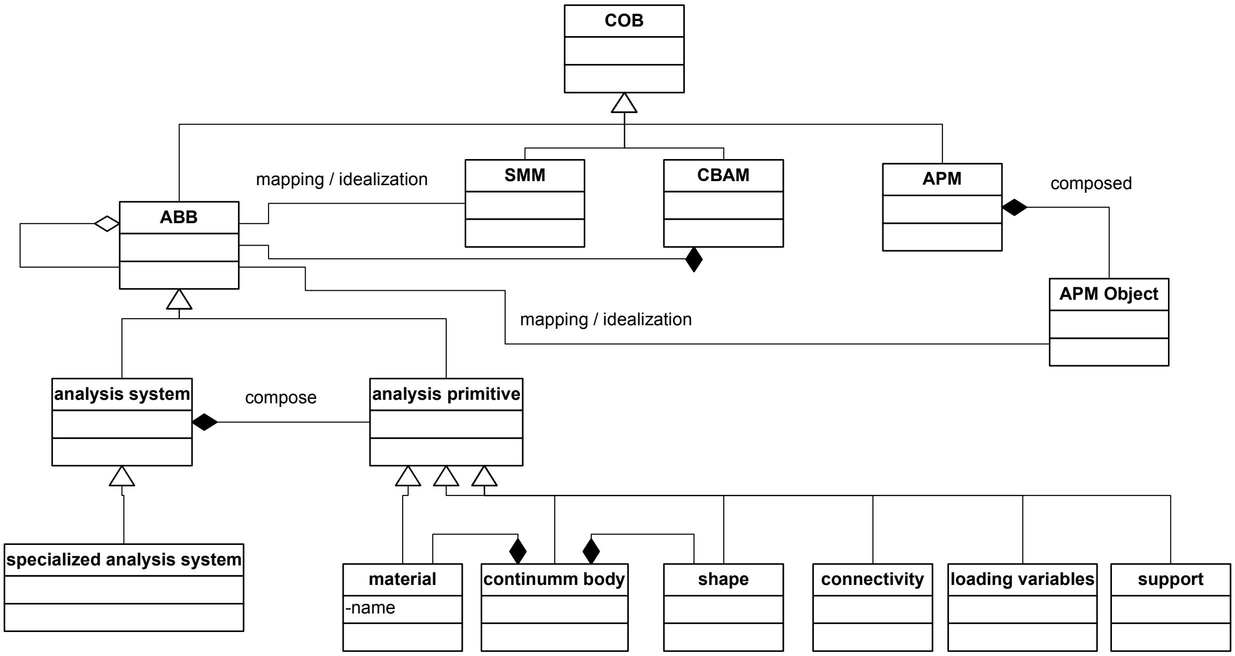


Figure 49. UML Notation of MRA - Design Analysis Integration Models.

The UML model illustrates the integration of design and analysis models as independent objects through a series of complex mapping and idealizations. The model presented in Figure 49 is similar to those developed at NIST (CPM and DAIM – see Figure 7, Figure 8, and Figure 9). The formalized DAI lexicon is included in the following report:

Capturing Design Process Information and Rationale to Support Knowledge-based Design and Analysis Integration: DAI Lexicon; GaTech Project #B-01-691

The report is located at the following URL:

http://www.eislab.gatech.edu/projects/nist-dai/

7 RESULTING PUBLICATIONS

Several publications have resulted in completing design-analysis integration research

- Mocko, G., R. Malak, C. Paredis, and R.S. Peak. *A Knowledge Repository for Behavioral Models in Engineering Design*. in *ASME 2004 International Design Engineering Technical Conferences and the Computers and Information in Engineering Conference*. 2004. Salt Lake City, UT.

- Bajaj, M., Peak, R., Wilson, M., Kim, I., Thurman, T., Benda, M., Jothishankar, M.C., Ferreira, P., Stori J. (July 16-18, 2003) *Towards Next-Generation Design-for-Manufacturability (DFM) Frameworks for Electronics Product Realization. Best Paper Award for Session 210*, IEMT, Semicon West 2003, San Jose, California.

- Zwemer, D., M. Bajaj, R.S. Peak, T. Thurman, K. Brady, S. McCarron, A. Spradling, M. Dickerson, L. Klein, G. Liutkus, and J. Messina. *PWB Warpage Analysis and Verification Using an AP210 Standards-based Engineering Framework and Shadow Moiré*. in *EuroSimE 2004*. 2004. Brussels, Belgium

- Mocko, G., J. Panchal, M. Fernandez, C.J.J. Paredis, and R.S. Peak. *Towards Reusable Knowledge-Based Idealizations for Rapid Design and Analysis*. in *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. 2004. Palm Springs, CA.

Much of the work presented in this paper has been the basis for the afore-mentioned publications.

8 FUTURE WORK

The work presented in this report addresses the integration of design and analysis from many perspectives. While we believe the contributions in this work are substantial at reducing the integration efforts between the domains of design and analysis, there are many areas for future work. These areas of future work address the development and refinement of methodologies for integrating design and analysis and also include implementation of computer-based design environment. Future work is identified in the following areas:

- Development and realization of a method and technology for mapping design parameters to behavioral models – a method is needed for creating associations between engineering design models and behavioral models. The method should leverage past knowledge and expertise, but must also enable analysis experts to create new relationships between models.

- Formal ontology representation of engineering idealizations – a conceptual framework and taxonomy for engineering idealization is presented. However, additional work is needed in the formal representation of engineering idealizations and the implementation of these idealizations in a computer-sensible format

- Metrics for assessing and characterizing analysis models – the models (design, analysis, and behavior) and the relationships (idealization) must be characterized for reuse. They must be characterized to enable querying and selecting these models from a repository of product development knowledge. Questions may arise as to what idealizations should be used or what model is best? The models and idealizations must be characterized to support knowledge intensive design-analysis integration

9 REFERENCES

[1] Mocko, G.M., *Capturing Design Process Information and Rationale to Support Knowledge-based Design and Analysis Integration: DAI Lexicon*. 2004, Georgia Institute of Technolgy: Atlanta.

[2] Szykman, S., R.D. Sriram, C. Bochenek, and J. Senfaute, *Design Repositories: Next-Generation Engineering Design Databases.* IEEE Intelligent Systems and Their Applications, 2000.

[3] Fenves, S., Y. Choi, B. Gurumoorthy, G. Mocko, and R.D. Sriram, *Master Product Model for the Support of Tighter Integration of Spatial and Functional Design*. 2003, National Institute of Standards and Technology.

[4] Fenves, S.J., *A Core Product Model for Representing Design Information, NISTIR 6736*. 2001, National Institute of Standards and Technology (NIST): Gaithersburg, Maryland.

[5] Szykman, S., R.D. Sriram, C. Bochenek, and J. Racz, *The NIST Design Repository Project.* Advances in Soft Computing - Engineering Design and Manufacturing, 1998.

[6] Finn, D.P., *A Physical Modeling Assistant for the Preliminary Stages of Finite Element Analysis.* Artificial Intelligence in Engineering Design Manufacturing and Analysis, 1993. **7**(4): p. 231-237.

[7] ISO, *ISO 10303-209*, International Standards Organization (ISO).

[8] Hunten, K.A., *CAD/FEA Integration with STEP AP290 Technology and Implementation*. 1997, Lockheed Martin Corporation.

[9] Hunten, K.A., *AP209 EA Pilot Scenario - Presentation*, Lockheed Martin Tactical Aircraft Systems.

[10] Gordon, S., *An Analyst's View: STEP-Enabled CAD-CAE Integration*. 2001, NASA's STEP for Aerospace Workshop: Pasadena, CA.

[11] Sheffer, A., *Model Simplification for Meshing Using Face Clustering.* Computer Aided Design, 2001. **33**: p. 925-934.

[12] Tautges, T.J., *The Generation of Hexahedral Meshes for Assembly Geometry: Survey and Progress.* International Journal for Numerical Methods in Engineering, 2001. **50**(12): p. 2617-2642.

[13] Tautges, T.J., T. Blacker, and S.A. Mitchell, *The Whisker Weaving Algorithm: A Connectivity-Based Method for Constructing All-Hexahedral Finite Element Meshes.* International Journal for Numerical Methods in Engineering, 1996. **39**: p. 3327-3349.

[14] Sheehy, D.J., C.G. Armstrong, and R. D.J. *Computing the Medial Suface of a Solid from A Domain Delauney Triangulation*. in *To appear in Proc. ACM/IEEE Symp. on Solid Modeling and Applications,*. 1995. Salt Lake City, Utah.

[15] Armstrong, C.G., R.J. Donaghy, and S.J. Bridgett. *Derivation of Appropriate Idealisations in Finite Element Modelling*. in *Submitted to The Third International Conference on Computational Structures Technology*. 1996. Budapest, Hungary.

[16] Arabshahi, S., D.C. Barton, and N.K. Shaw, *Towards Integrated Design and Analysis.* Finite Elements in Analysis and Design, 1991. **9**(4): p. 271-293.

[17] Arabshahi, S., D.C. Barton, and N.K. Shaw, *Steps Towards CAD-FEA Integration.* Engineering with Computers, 1993. **9**(1): p. 17-26.

[18] Jha, K. and B. Gurumoorthy, *Automatic Propogation of Feature Modification Across Domains.* Computer Aided Design, 2000. **32**: p. 691-706.

[19] Wentorf, R. and M.S. Shephard, *User Interface Functions for an Analysis Idealization System.* Microcomputers in Civil Engineering, 1993. **8**: p. 85-95.

[20] Shephard, M.S., P.L. Baehmann, M.K. Georges, and E.V. Korngold, *Framework for the Reliable Generation and Control of Analysis Idealizations.* Computer Methods in Applied Mechanics and Engineering, 1990. **82**: p. 257-280.

[21] Shephard, M.S. and R. Wentorf, *Toward the Implementation of Automated Analysis Idealization Control.* Applied Numerical Mathematics, 1994. **14**: p. 105-124.

[22] Prabhakar, V. and S.D. Sheppard, *A Knowledge-Based Approach to Model Idealization in FEM.* 1994.

[23] Finn, D.P., *Introduction: Preliminary Stages of Engineering Analysis and Modeling.* Artificial Intelligence in Engineering Design Manufacturing and Analysis, 1993. **7**(4): p. 231-277.

[24] Finn, D.P., J.B. Grimson, and N.M. Harty. *An Intelligent Modelling Assistant for Preliminary Analysis in Design*. in *2nd International Conference on AI in Design*. 1992.

[25] Shooter, S.B., W. Keirouz, S. Szykman, and S.F. Fenves. *A Model for Information Flow in Design, paper no. DETC2000/DTM-14550*. in *in Proceedings of DETC 2000, Design Theory and Methodology*. 2000. Baltimore, Maryland.

[26] Gero, J.S. and U. Kannengiesser. *Towards a Situated Function-Behaviour-Structure Framework as the Basis for a Theory of Designing*. in *Artificial Intelligence in Design*. 2000. Worcester, MA.

[27] Gero, J.S., *Design Prototypes: A Knowledge Representation Schema for Design*, in *AI Magazine*. 1990. p. 26-36.

[28] Chang, E., X. Li, and L.C. Schmidt, *The Need for a Form, Function, and Behavior-based Representation System*. 2000, Department of Mechanical Engineering, University of Maryland: College Park. p. 12.

[29] Balazs, M. and D.C. Brown, *The Use of Function, Structure, and Behavior in Design*. 1994, Worcester Polytechnic Institute: Worcester.

[30] Iwasaki, Y. and B. Chandrasekaran, *Design Verification Trhough Function- and Behavior-Oriented Representations*. 1992, Knowledge Systems Laboratory, Stanford University: Stanford. p. 19.

[31] Mocko, G. and S. Fenves, *A Survey of Design-Analysis Integration Issues, NISTIR 6996*. 2003, National Institute of Standards and Technology: Gaithersburg, MD.

[32] Fenves, S.J., R.D. Sriram, S. R., and F. Wang. *A Product Information Modeling Framework for Product Lifecycle Management*. in *International Symposium on Product Lifecycle Management*. 2003. Bangalore, India.

[33] Peak, R.S., R.E. Fulton, I. Nishigaki, and N. Okamoto, *Integrating Engineering Design and Analysis Using a Multi-representation Approach.* Engineering with Computers, 1998. **14**: p. 93-114.

[34] Fulton, R.E., C. Ume, F.C. Y., R.S. Peak, A.J. Scholand, S. S., M. Stiteler, D.R. Tamburini, F. Tsang, and W. Zhou, *Rapid Thermomechanical Design of Electronic Products in a Flexible Integrated Enterprise, Interim Report*. 1994, Manufacturing Research Center, Georgia Institute of Technology: Atlanta.

[35] Peak, R.S. and R.E. Fulton, *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*. 1994, Manufacturing Research Center, Georgia Tech: Atlanta. p. 22-27.

[36] Wilson, M.W., *The Constrained Object Representation for Engineering Analysis Integration*, in *G.W. Woodruff School of Mechanical Engineering*. 2000, Georgia Institute of Technology: Atlanta, GA.

[37] Kemmerer, S.J., *STEP: The Grand Experience*. 1999, NIST.

[38] Lubell, J., R.S. Peak, V. Srinivasan, and S. Waterbury. *STEP, XML, and UML: Complementary Technologies*. in *ASME 2004 International Design Engineering Technical Conferences And The Computers And Information In Engineering Conference (DETC'04), Paper Number DETC 2004-57743*. 2004. Salt Lake City, UT.

[39] Finn, D.P., *A Physical Modeling Assistant for the Preliminary.* Artificial Intelligence in Engineering Design Manufacturing and Analysis, 1993. **7**(4): p. 231-237.

[40] Grosse, I.R. I-MAD: The Intelligent Modeling, Analysis, and Design Laboratory. http://www.ecs.umass.edu/mie/labs/mda/fea/. May 14 2004.

[41] Mocko, G., R. Malak Jr., C. Paredis, and R.S. Peak. *A Knowledge Repository For Behavioral Models in Engineering Design*. in *ASME Computers and Information in Engineering*. 2004. Salt Lake City, UT.

[42] Malak Jr., R.J. and C.J.J. Paredis. *On Characterizing and Assessing the Validity of Behavioral Models and Their Predictions*. in *ASME 2004 International Design Engineering Technical Conferences And The Computers And Information In Engineering Conference (DETC'04)*. 2004. Salt Lake City, UT.

[43] Zwemer, D., M. Bajaj, R.S. Peak, T. Thurman, K. Brady, S. McCarron, A. Spradling, M. Dickerson, L. Klein, G. Liutkus, and J. Messina. *PWB Warpage Analysis and Verification Using an AP210 Standards-based Engineering Framework and Shadow Moiré*. in *EuroSimE 2004*. 2004. Brussels, Belgium.

[44] http://eislab.gatech.edu/tiger.

[45] Wang, Y. and P. Hassell, *Measurement of Thermal Deformation of BGA Using Phase-shifting Shardow Moire.* Experimental/Numerical Mechanics in Electronic Packaging, Society for Experimental Mechanics, 1998. **2**: p. 32-39.

[46] Zewi, I.G., I.M. Daniel, and J.T. Gotro, *Residual Stresses and Warpage on Woven-Glass/Epoxy Laminates.* Experimental Mechanics, 1987: p. 44-50.

[47] Peak, R.S. X-Analysis Integration / Multi-Representation Architecture (XAI/MRA) Overviews. http://eislab.gatech.edu.

[48] Peak, R.S., M.W. Wilson, I. Kim, N. Udoyen, M. Bajaj, G.M. Mocko, G. Liutkus, L. Klein, and M. Dickerson. *Creating Gap-Filling Applications using STEP Express, XML, and SVG-based Smart Figures - An Avionics Example*. in *NASA-ESA Workshop on Aerospace Product Data Exchange (APDE)*. 2002. Noordwijk (ZH), The Netherlands.

[49] Bajaj, M., R.S. Peak, M.W. Wilson, I. Kim, T. Thurman, M. Benda, M.C. Jothishanka, P. Ferreira, and J. Stori. *Towards Next-Generation Design-for-Manufacturability (DFM) Frameworks for Electronics Product Realization*. in *IEMT, Semicon West*. 2003. San Jose, California.

[50] Peak, R.S., M. Bajaj, M.W. Wilson, I. Kim, T. Thurman, M. Benda, M.C. Jothishanka, P. Ferreira, J. Stori, D. Mukhopadhyay, D. Tang, G. Liutkus, and L. Klein. *Enhancing Design-for-Manufacturability Using ISO 10303 Standard for Electronics Design: AP210*. in *Aerospace Product Data Exchange (APDE) Workshop*. 2003. NIST, Gaithersburg, MD.

[51] Peak, R.S. *An AP210-based Repository for Collaborative Electronics Engineering*. in *NASA STEP for Aerospace Workshop*. 2001. Pasadena, California.

[52] Pahl, G. and W. Beitz, *Engineering Design*. Second Edition ed. 1999: Springer.

[53] Brooks, R.J. and A.M. Tobias, *Choosing the Best Model, Level of Complexity, and Model Performance.* Mathematial and Computer Modelling, 1996. **24**(4): p. 1-14.

[54] Symonds, J., J.P. Vidosic, H.V. Hawkins, and D.D. Dodge, *Strength of Material*. 10 ed. Marks' Standard Handbook for Mechanical Engineers, ed. T. Baumeiser III. 1996, New York: McGraw-Hill.

[55] Welcome to the Protégé Project. 2004. http://protege.stanford.edu/. April 15 2004.

[56] Mocko, G., J. Panchal, M. Fernandez, C.J.J. Paredis, and R.S. Peak. *Towards Reusable Knowledge-Based Idealizations for Rapid Design and Analysis*. in *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. 2004. Palm Springs, CA.

[57] Conner Seepersad, C., *A Robust Topology Exploration Method for Designing Multi-Functional Cellular Materials*, in *Mechanical Engineering*. 2002, Georgia Insitute of Technology: Atlanta.

[58] Hayes, E.E. and W.C. Regli. *Integrating Design Process Knowledge with CAD Models*. in *2001 ASME Design Engineering Technical Conference & Computers and Information in Engineering Conference*. 2001. Pittsburgh, Pennsylvania.

[59] Peak, R.S., R.E. Fulton, I. Nishigaki, and N. Okamoto, *Integrating Engineering Design and Analysis Using a Multi-Representation Approach.* Engineering with Computers, 1998. **14**(2): p. 93-114.

[60] Matsuki, R., R.S. Peak, S. Zeng, M.W. Wilson, I. Kim, and M. Bajaj. *Design-Analysis (Thermal and Mechanical) Integration Research for Electronic Packaging*. in *SEMICON: RATS of Japan Institute of Electronics Packaging Symposium*. 2002. Japan.

[61] Szykman, S., R.D. Sriram, C. Bochenek, and J. Senfaute. *Design Repositories: Next-Generation Engineering Design Databases*. in *IEEE Intelligent Systems and Their Applications*. 2000.