The Reusable Engineering Analysis Preprocessing Methodology to Support Design-Analysis Integration

Ph.D. Thesis Proposal

Presented to
The Academic Faculty

By

Chien H. Hsiung

In partial fulfillment
of the requirements of the degree
Doctor of Philosophy

SCHOOL OF MECHANICAL ENGINEERING
GEORGIA INSTITUTE OF TECHNOLOGY

July 1998

The Reusable Engineering Analysis Preprocessing Methodology to Support Design-Analysis Integration

ABSTRACT

This thesis focuses on researching the modular reuse of engineering analysis preprocessing to aid in design-analysis integration. A component-oriented approach, referred to as Reusable Engineering Analysis Preprocessing (REAP) methodology, is proposed to investigate current analysis preprocessing methodologies and modifies the current monolith analysis process into a process based on assembly of REAP components. These REAP components are intended for third party composition within a component framework to generate a finite element analysis model ready for processing. These components are associated with physical design components, typically "standard parts", and are capable of representing multiple analysis idealizations. This methodology enables "black-box" reuse by accessing the components only through their interfaces.

The methodology provides necessary interfaces for the interoperability of components, framework, and user, and enforces interaction rules to ensure validity of the resulting preprocessed analysis model. This thesis will also identify what information will need to be exchanged across the interfaces to enable compatible assembly of REAP components. Algorithms will be developed to assemble components using only the their interfaces. Component requirements will also be developed to ensure validity of generated analysis model. The methodology will be developed through test cases using a cyclic process of conceptualization, prototyping, testing, and updating.

TABLE OF CONTENTS

1 Introduction

Rapid advances are evident in the field of engineering products, especially in highly competitive industries such as computer and telecommunication. Toward the objective of increasing profit, some of the industry requirements are that the product must be developed and manufactured both *quickly* and *reliably*. By *quickly* it refers to the "time-to-market" issue, where delays in design or manufacturing incur penalties and reduces market share. By *reliably* it refers to how well the product survives the manufacturing process and customer usage, impacting to overall life cycle cost. Reliability analysis is a difficult and time-consuming task, and to save time, many industries either forgo analysis until absolutely necessary or train their designs to perform preliminary analysis. However, most designers are under-trained in analysis and too busy to perform useful analyses [Liker *et al.* 1992]. Reliability study requires the design representation to be idealized and transformed into analysis models that can be solved mathematically. This process is arguably the most important step during the analysis, for it determines whether the results are representatives of the physical phenomenon being studied. This indicates a strong need for design-analysis integration, which aids analysis during design time [Peak *et al.* 1997; Yeh 1992]. Figure 1 illustrates the industry's trade-off problem, their response by having designers perform analyses, the resulting problem of designers lacking experience and time, and finally the research solution of design-analysis integration. This proposed thesis aids design-analysis integration by researching the reuse of engineering analysis preparation.

Despite the fact that past reliability analyses often have parts in common, such parts are not reusable for future analyses. This is because the existing engineering analysis architectures are monolithic, with parts that are not designed for open reuse. This forces new analyses to start from scratch, increasing the difficulty and preparation time. This thesis proposes to research the problem of reusability in engineering analysis preparation by using a component-oriented approach. This component orientation refers to the application of the concepts found in the field of component software, which targets the third-party reuse of pre-fabricated software "modules" [Szyperski 1998]. By focusing on such reusability, this research hopes to support the design-analysis integration efforts by facilitating the reuse of previously validated analysis components, and aid integration of analysis capabilities into existing design tools.

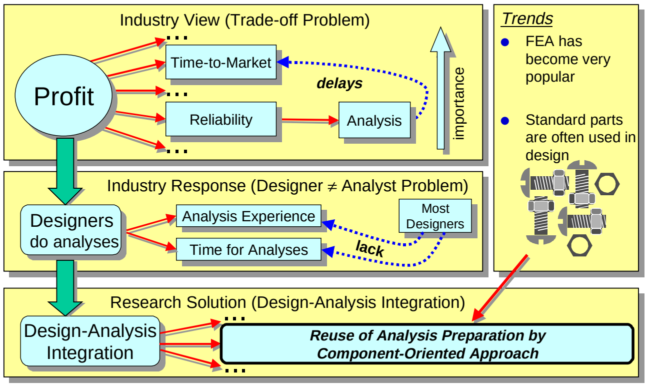

Industry View (Trade-off Problem)
Profit
Time-to-Market
Reliability
Analysis
delays
importance
...
Trends
FEA has become very popular
Standard parts are often used in design
Industry Response (Designer ≠ Analyst Problem)
Designers do analyses
Analysis Experience
Time for Analyses
Most Designers
lack
Research Solution (Design-Analysis Integration)
Design-Analysis Integration
Reuse of Analysis Preparation by Component-Oriented Approach

Figure 1: Overview of the need for design-analysis integration and proposed research

A couple of trends in industry are relevant to this research. The first trend is that "standard parts", chosen from catalogs (either printed or electronic), are often used in design. For example, transistors and memory chips for a circuit board are often simply selected from catalogs based on the required specifications. Another example would be the commonly used nuts and bolts. They are often used because they are readily available, cheaper to buy than to make, and have already been field-tested. This is a prime example of reuse in industry. Another trend is that finite element analysis (FEA) packages has become very popular for analysis due to its flexibility and ability to handle complex geometry [Kurowski 1996]. Finite element analysis is a discretized numerical analysis that can analyze assemblies of parts as a whole, and produce analysis values such as the maximum stress to be used in equation-based analysis such as fatigue calculations. Because of these trends, it is practical to partition analyses into components that correspond to such parts, and assemble them together for finite element analysis of new designs.

The focus of this proposed thesis is to research the issues of reusability in engineering analysis preparation, leveraging the concepts used in component software. A modular approach for third-party composition of analysis components is proposed as a methodology to support reusability. These components are associated with "standard parts" used in engineering design, with prescribed mechanisms of design and analysis parameterizations. The methodology provides interoperation interfaces for the components, the framework, and the user, and enforces interaction rules to ensure validity of the resulting preprocessed analysis model.

The layout of the proposal is as follows: Section 2 (Background) provides additional details on engineering analysis preparations, including finite element preprocessing, and indicates some current shortcomings. It is followed by a brief discussion of part libraries for design-analysis integration, and an introduction to the concept of component software. Section 3 (Related Research) is a literature survey providing some descriptions on prior researches related to design-analysis integration. This is followed by Section 4 (Discussion of Gaps) which discusses the identified research gaps found within the literature survey. This discussion then leads to Section 5 (Research Objective) where the main objective of this proposed thesis is described in detail. The proposed methodology of achieving this research objective is detailed in Section 6 (Proposed Work). To better understand the application of the methodology, an example scenario is described in Section 7 (Example Scenario). To examine the proposed approach, test cases are created and are described in Section 8 (Representative Test Cases). Section 9 (Validation) outlines the criteria used to validate the usefulness of the methodology, followed by a work schedule. This proposed thesis concludes, in Section 10 (Expected Contribution), with discussions of the expected contributions and the impact of this thesis in the efforts toward design-analysis integration.

2 Background

2.1 Engineering Analysis Preparation

The preparation of an engineering analysis is a creative process that requires experience and insight for a reliable analysis result. Despite the availability of many powerful analysis tools, if one does not understand the applicability, assumptions, and limitations of such tools, misinterpretation of the results will occur. A generalization of the process [Brooke *et al.* 1995] is described as: determine physical phenomenon relevant to the analysis, select appropriate physical (theoretical) model, construct mathematical model, and select method to solve equations of the mathematical model. Many assumptions, also called idealizations, are made during the physical model selection and mathematical model construction steps. The purpose of the assumptions is to constrain and simplify the model so that it can be solved efficiently (or even possibly). Most design representation contains too many details that do not significantly affect the accuracy of the analysis, or may contain redundancies that are not necessary for analysis. Material behaviors are among the most commonly idealized, and can drastically effect the results. Environmental idealizations are also often made to predict the boundary and initial conditions for the analysis.

For geometrically dominant problems, the idealizations can include geometry simplification, feature removal (e.g. removing small holes), dimensional reduction (e.g. 3-D to 2-D), and adding new geometry for attribution or stiffness compensation. As one can imagine, after many idealizations, the appearance of the analysis model may appear significantly different from the original design representation. In fact, analysis models are often of non-manifold geometry, even if the design models are of manifold geometry. In most cases, each phenomenon requires a separate analysis application to predict its effects. Figure 2 shows an example of multiple idealizations for different analysis. Even for the same physical phenomenon, multiple analysis models of different detail levels and localization levels may be necessary to predict the effects.

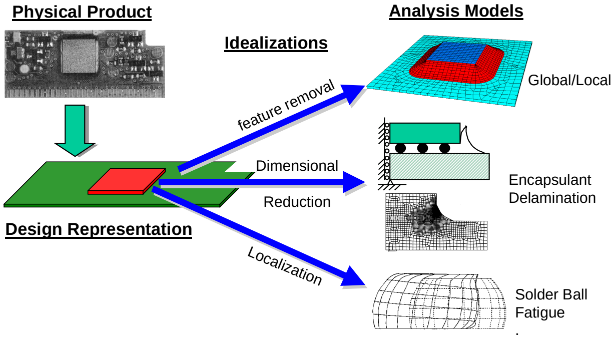


Figure 2: Different idealizations for different analysis models

Much time and effort are spent in creating multiple analysis models; however, the resulting analysis models are often not reused, since they are for a particular product. Even for an adaptive design, analysts often re-develop the analysis models from scratch for several reasons [Zhou 1997; Arabshahi *et al.* 1993]. One reason is that the analysts do not completely trust the existing analysis models, because either someone else created the existing models, and/or the assumptions of the existing models were unclear or unknown. Another reason is that the existing models were created on different software system, resulting in software incompatibility. One more reason is that the "pieces" of existing models can not be easily extracted or modified for creating a new model. A conclusion can be drawn from these various reasons: The existing architecture for analysis does not support open reusability.

2.2 Finite Element Analysis Preprocessing

Finite element analysis (FEA) is a popular, general purpose, discrete numerical analysis tool that is widely used in industry. The preprocessing stage inherits the same characteristics from the above section on analysis preparation. A typical FEA process is shown as a flowchart on Figure 3. Although the pretty pictures shown at the postprocessing stage give the impression that the solution has been solved accurately, it may not predict the correct behaviors [Russell 1996]. FEA will always give some sort of solution, even for a poorly developed model; however, the resulting solution may be highly incorrect for the analysis of the physical phenomenon. Usefulness of the FEA result depends highly on the preprocessing stage, which currently has some important shortcomings. These additional shortcomings are described below.

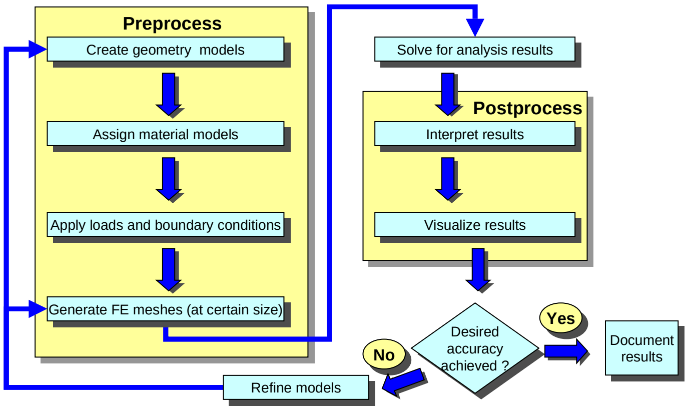


Figure 3: Typical FEA process

- ***User resource intensive***: The process of creating the geometry model (or translation from design geometry), idealizing it, specifying mesh controls, etc., consumes too much of the user's time. It may account for up to 80% of the total analysis time [Zhou 1997]. The advances in computing power may dramatically reduce *CPU* time, but not necessarily *USER* time. In addition, CPU intensive jobs may be queued for batch processing overnight or weekends. Furthermore, given increased computational power, the user may select more complicated model that may actually increase total time without gaining significant accuracy increase.
- ***Awareness of assumptions made***: Many idealizations are made during this stage. The user may inadvertently simplify a section that contains the high stress region, or the user may erroneously use a 3D solid element with no bending stiffness to model a shell structure [Kurowski 1996].
- ***Design models not suitable***: Often a geometric model translated from the design tools is not directly usable for analysis. As discussed in the previous section, analysis geometry is often an abstract representation of the original design geometry for the particular analysis. The original design model often contains presumed idealizations, too many details, and lacks much of the information needed for analysis (e.g. material properties, environmental properties).
- ***Brute force modeling from scratch***: FE models are often started from scratch, since FEA packages stores the model with the lowest level of semantic details. It is very difficult to extract a portion of a mesh from a previous product to be used in a new model, and still maintain compatibility.
- ***Valid meshing is not simple***: Given an idealized solid model, it is still not a simple task to mesh the model. Auto-meshing algorithms have been improving, but are still not perfect and not very accurate. Much of the user's time is spent specifying mesh control over the solid model to generate a good mesh.

2.3 *Design-Analysis Integration Library*

For products assembled mostly from standard parts, a library of standard part models can be very useful. However, most of the libraries available are for design use, not analysis use. A library of standard parts for both design and analysis can aid significantly toward design-analysis integration. Such libraries would contain high semantic level translations of the product data into analysis models. In the case of an assembly-based product, there would be translations of the assembly of design parts into an assembly of analysis components, rather than just a pure geometry translation. The user would then be able to clearly relate the appropriate analysis components to its corresponding design part. This would be one way to reuse previous modeling efforts and reduce the time spent remodeling.

There are some repositories, for analysis, of commonly uses parts used in some researches [Shephard *et al.* 1995; Zhou 1997], but they are either imbedded in knowledge bases or in forms not easily used by others. This means that these repositories are not designed with reusability in mind. The ability for a third party to assemble analysis components together to form a compatible analysis model remains an open issue, to be addressed in this proposed thesis. Once such capability is available, and with widely accepted specifications, different analysis software vendors can then independently develop analysis components. This would create a component market for flexible design-analysis integration libraries.

2.4 *Software Component Technology*

In this proposed thesis, the term “component” has a dual meaning. The simple meaning of “component” is a physical substructure or its associated computer model, ready to be combined in an assembly. The alternate meaning of “component” refers to the software technology call *component software*. This is an emerging technology in software industry that builds on existing object-oriented approaches to deliver reusable, “off-the-shelf” software components for incorporating into large applications. Software components provide a vehicle for planned and systematic reuse. They have an interface (pre-defined methods for accessing both its data and functionality), encapsulated internal details, and are documented separately. It is required that software components be easily combined with each other, and especially without knowing each other’s existence. The primary intention in reusing software components is so that one can take a component and integrate it into a software system [Sametinger 1997].

Some of the fundamental properties of component technology are concepts such as late integration, combinatorial explosion of possible system configuration, “black-box” abstractions, and component safety through conservative software engineering. To adopt component technology requires adaptation of principles of independence and controlled explicit dependencies, as well as enforcing modularity of requirements, architecture, designs, and implementations. The benefits of using component software are enhanced adaptability, scalability, and maintainability. Currently there are three major forces in component software arena that provide the low-level “wiring” standards: Object Management Group’s CORBA (Common Object Request Broker Architecture), Microsoft’s COM (Component Object Model), and Sun’s JavaBeans. Beyond the “wiring” level standards, the modeling of component-based system is still largely an unresolved problem [Szyperski 1998].

In the scope of this proposed thesis, the software system is the system in which the design-analysis integration is implemented, which can even be the design-tool software system. Since computer-based engineering analyses are intimately tied to software systems, it make sense to research analysis reusability by applying concepts from software reusability. The modeling of a component-based engineering analysis system will be targeted in this thesis.

3 Related Research

Researches in design-analysis integration efforts have been very active in the past few years, since it is of great interest to the engineering industry. As mentioned before, industries are now training their design engineers to perform some preliminary analysis to roughly validate their design [Deitz 1997]. The goal of the design-analysis integration researches is to make it feasible for the designers to perform analysis, using tools that they are familiar with, namely computer-aided design (CAD) tools. Numerous publications were surveyed in the course of developing this proposed thesis. These publications dealt with various topics such as integration frameworks, data exchange technologies, analysis preprocessing issues, analysis techniques in certain domains, software technologies, etc. Among these literatures the most relevant ones to this proposed thesis are selected and briefly described in the following section (Individual Descriptions). Although some of the described researches may not directly relevant, they are mentioned because they provide valuable insights to the problem of design-analysis integration. Literature in general software components, although reviewed, are not described here because they do not generally consider the engineering analysis problem. After the descriptions of selected researches, a summary classification (Summary Categorization) of the covered issues is discussed.

3.1 Individual Descriptions

Zhou [1995-1997] developed a methodology, called Modular & Parametric Finite Element Methodology (MP/FEM), aimed at a seamless integration between electronic packaging design and thermo-mechanical analysis. Modular FEM is achieved using a group technology called Modularized Generic Primitive (MGP), which is used to categorize families of similar shapes. Each MGP supports a number of 3D shapes, is modeled independently, has common boundaries to its neighbors (to maintain compatibility), and defines shapes using forming rules. A Constructive Module Assembly Tree (CMAT) is a graph tree representing overall relationships among MGPs, components, assembly, and the product. Parametric FEM is achieved through use of templates that accepts feature-based parameters for geometric data, material properties, analysis control, assembly control, and meshing control. These concepts are implemented in a component library structure that allows rapid generation of FE models for both local and global/local analysis. A product level thermo-mechanical analysis is carried out as a test case.

Shephard and Finnagan [1988, 1989] describe the structure to a geometry-based FE preprocessing system for the integration of geometric modeling and finite element preprocessing. Three general data structures (topology and geometry data, mesh data, and attribute data) to support both geometric modeling and FE preprocessing are described. Five categories of geometric communication operators needed in FEM from geometric modeling systems are identified: basic query, derived data query, geometric modeling operation, attribute specification, and general utility. The authors indicate that such integration is a major step toward fully automated mesh generation. A framework of a design system, named IDEALZ, to support analysis idealization control and automated discretization (e.g. meshing) is described in Shephard *et al.* [1990a] and in Shephard and Wentorf [1994]. This framework consists of functional modelers, geometry modelers, and attribute modelers that interact with analysis applications and applies AI techniques to guide the user through idealization and analysis processes. Techniques for controlling the idealization errors are described in Shephard *et al.* [1990b]. Specific application of this framework to global/local thermal and thermo-mechanical analysis of multichip modules is described in Shephard *et al.* [1992]. Graphical user interface for the goal manager of IDEALZ, described in Wentorf and Shephard [1993], is later implemented in X-Windows for design experts. Visualization of analysis models and graphical attribution system is later described in O'Bara, Beall, and Shephard [1995].

Tamburini [1996, 1997] describes an analysis representation called Analyzable Product Model (APM) to facilitate design-analysis integration. The APM is an integrated source of data for analysis, representative of a design product. It is an objected-oriented data model that contains data, the mappings required to retrieve design information from several sources, and the transformations required for idealizing this information. The APM enables reusability inherently by supporting data and idealizations shared among multiple analysis and by using a generic specification.

Peak [1995] approaches design-analysis integration with a system called multi-representation architecture (MRA). The MRA approach bridges the gap in design-analysis integration with four "stepping stones" intermediate representations. These representations are solution method model (SMM), the analysis building block (ABB), the product model (PM), and the product-based analysis model (PBAM). The SMM is the detailed solution-method specific model that combines solution tool input, output, and controls. The ABB represents analysis concepts independent of product applications and solution methods. The PM contains the detailed design-oriented product information. The PBAM contains the link between PM and ABB, showing idealization usage. The MRA supports routine analysis automation, described in Peak [1993] and Peak *et al.* [1996].

Yeh [1992] takes an information management approach to the integration of printed wiring board (PWB) design and analysis. An integrated design process modeling methodology (IDPMM) is developed to model the design process of PWB, and an integrated framework called thermal structural electromagnetic testability (TSET) is developed through use of a common database. Five levels of integration are identified: data, standard, control, platform, and user-interface. Knowledge base and expert system shell is used to guide the framework.

Arabshahi *et al.* [1991] models the two different FEA processes using IDEF0 models. The first one is the traditional FEA process where the FE model is built-up using the direct method. The second one is the alternate FEA process where is FE model is broken down from a solid model. At first it seems that the second process is preferable. However, Arabshahi *et al.* [1993] indicates that the complete geometry in the form of solid model is seldom taken advantage of because of the amount of time required to simplify and idealize the geometry for subsequent meshing stage. The paper also overviews a future system that would allow more automated CAD-FEA transformation. Such system would have functional components that consist of: attribute editor (for applying attributes), detail editor (feature removal), dimensional reduction aid, subdivision feature recognizer (divide complex model into mappable regions), macro-feature builder (assembling of subparts), and cutting surface facility (for splitting and built-up).

Armstrong [1994] discusses the modeling requirements for FEA. It points out that the major factor limiting the wider application of FEA is the effort and time required to produce analysis model, and that abstraction (idealization) process rely on the judgement of skilled analyst. For determining abstractions, it views "deep" knowledge (based on physical laws and computational algorithms) as better than "shallow" knowledge (based on expert opinion and experience; e.g. knowledge base). The medial axis and surface transformation is then introduced as a way to simplify geometry and reduce dimensionality without using "shallow" knowledge. The medial axis and surfaces computation accomplish these by providing information on the proximity of edges or faces that can be removed.

Finn [1993] describes an interactive modeling system that aids engineers in selection and evaluation of various model options that finally leads to an idealized model. The system employs AI technique (propose-critique-modify) to provide expert advice to the user, but does not take control away from the user (as in complete automation). User interactivity is indicated as preferred over the automated "black-box" approach of idealization.

Hardwick *et al.* [1996] describes an ongoing research on an information infrastructure that uses standards to reduce tool incompatibility occurring in manufacturing. The prototype combines Internet with STEP for data exchange and CORBA for interoperation of application systems. Specifically, the data definition language of CORBA (IDL), is combined with the data definition language of STEP (EXPRESS). Note that this research involves manufacturing interoperability, but not engineering analysis.

3.2 Summary Categorization

The selected researches described in the previous section (Individual Descriptions) are categorized according to the issues that they address. Although these issues may not be a complete set of all the issues related to design-analysis integration, they span a broad-enough spectrum to cover those related to the focus of this proposed thesis. This categorization is shown in Table 1, along with this thesis's expected coverage (under the column "Hsiung* 98"). The categories are divided into three groups: those that are associated with modeling (Modeling Issues), those that are associated with functional capabilities (Functional Issues), and those that are associated with how it is used (Usability Issues). These categories are described in Table 1 under the column "Descriptions". It is through this categorization that gaps are identified in current related researches (circled in Table 1), to be discussed in the next section (Discussion of Gaps).

Authors

Expected coverage of this proposed thesis

	97	95	97	97	92	93	94	93	96	98	
	Zhou	Shephard	Tamburini	Peak	Yeh	Arabshahi	Armstrong	Finn	Hardwick	Hsiung *	Descriptions
Modeling Issues											
Idealization		✔	✔			✔	✔	✔		✔	Discusses analysis idealizations, the control of its errors, or idealization selection
Data Exchange			✔	✔		✔			✔		Involves translation of design-analysis information (e.g. STEP)
Data Structure	✔	✔				✔	✔			✔	Involves development of new data structures to support design-analysis integration
Relational Mapping			✔	✔	✔					✔	Contains mapping relations among attributes from various models
Process					✔	✔					Performs modeling of any processes (e.g. IDEF0)
Generic Product-based			✔	✔							Includes methodology includes mapping to non-specific products
Physical Component-based	✔									✔	Involves analysis model built-up from smaller modular physical components
Constructive Assembly										✔	Involves analysis model built-up constructively and algorithmically
Functional Issues											
Repository Reusability	✔	✔	✔	✔		✔			✔	✔	Contains repository of models to be re-applied to other models
Integrated Framework		✔		✔	✔	✔			✔		Introduces an integrated framework for design-analysis
Abstract Operators		✔				✔			✔	✔	Utilizes generic, cross-platform, high-level operations (e.g. CORBA)
Global/Local Analysis	✔	✔					✔			✔	Supports global/local analyses
Iterative/Adaptive Analysis		✔				✔		✔			Supports iterative and/or adaptive analyses
Component Software									✔	✔	Uses component software for independent reusability (e.g. CORBA, JavaBeans)
Usability Issues											
Visualization		✔				✔				✔	Involves visualization of design models or analysis models
Knowledge-base/AI		✔			✔						Uses knowledge-base or AI to increase usability
Automation	✔	✔		✔						✔	Discusses automation (complete analysis without much user input)
Interactivity								✔		✔	Allows flexible interactive user control (versus complete automation)

Table 1: Related researches (and this proposed thesis) versus issues they cover - gaps circled

4 Discussion of Gaps

All the above mentioned researches lend valuable insights to the problems associating with the design-analysis integration. As seen in Table 1, different people have different approaches to the design-analysis integration problem, covering various issues. Also shown in Table 1, three large gaps are identified in the related researches. These gaps are not intended to be a comprehensive list of gaps in design-analysis integration; rather they are gaps relevant only to the focus of this proposed thesis.

4.1 The Three Gaps: Constructive Assembly, Component Software, and Interactivity

The first gap involves the ability to *constructively assemble* analysis components to form new analysis models. Such methodology would be both algorithmic and information-intensive, but could potentially save a lot of time on remodeling or building new models. The closest research to covering this gap is by Zhou [1995-1997], who offered component-based modeling; however, Zhou did not offer generic methodology for constructive assembly, which greatly limits its reusability and adaptability.

The second gap deals with the use of *component software* (CORBA) for independent reusability and interoperation. Although Hardwick *et al.* [1996] discusses the importance of this issue extensively, the research focuses only on its application to manufacturing, not to engineering analysis. Application to engineering analysis would greatly assist the integration of analysis components into existing engineering design systems.

The last gap involves the ability for the user to interactively control the analysis preparation. User *interactivity* has typically only been an issue in computer science and psychology under usability studies. Software developers for engineering analysis typically consider usability not as important as features and computational performances. This may have once been true when computation cost is very high, but with the increasingly less-costly and higher-performance computers readily available, the cost of user resources exceed that of computers. This is especially important within design-analysis integration since the primary benefactors are not analysis experts.

4.2 The Three Gaps Summarized as Under the Reusability Gap

These three gaps can be summarized as part of the *reusability gap* in current researches, illustrated in Figure 4. The reusability gap is identified as an important obstacle to the goal of design-analysis integration. There are many other forms of reuse, such as source code libraries, designs, or model repositories. In fact, reuse is a common concept used everywhere, especially in engineering (e.g. standard parts). In the context of this proposed thesis, reusability refers to the ability to take "pieces" of previous analysis and allow easy creation of new analysis models. However, a distinction is made here regarding the granularity of reusability. The reusability addressed in this proposed thesis is applied to fine level of the *individual components* that composes the analysis model, and not applied to coarse level of the *completed* analysis model. This means that parameterized analysis models (e.g. routine analysis in Peak *et al.* [1996]) are not considered as reusable components within the scope of this research.

Since design-analysis integration is typically a heterogeneous integration between different disciplines of engineering, and perhaps even different domains (e.g. electrical versus mechanical engineering in electronic packaging), the ability to *reuse* analysis implementations in design systems is critical. By targeting reusability, efforts on the analysis side can then be easily integrated into the design side without too much restructuring of current design system. Why has reusability not been prevalent in design-analysis integration researches? A possible explanation is that software reusability has only recently been on the verge of success, and then only in business applications. Engineering analysis is a complex problem that is not easily solved by software engineers. It is a goal of this proposed thesis to apply this emerging software success to the integration of engineering design and analysis.

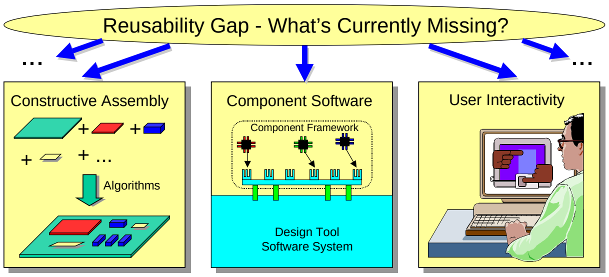


Figure 4: Summary of Research Gaps – Reusability Gap

5 Research Objective

5.1 Component-Oriented Approach to Reduce the Reusability Gap

The objective of this thesis is to address the reusability gap in design-analysis integration by an approach hereby called the Reusable Engineering Analysis Preprocessing (REAP) methodology. This is an information-intensive, component-oriented approach that applies concepts and principles found in component software technology. In general, this methodology investigates the current analysis preprocessing methodologies and attempts to modularize current monolith systems into assembly of REAP components. These REAP components are intended for third party (ones who did not implement the components) composition within a component framework that generates new preprocessed analysis models (PAMs). Such PAMs are ready for computation by commercial analysis applications (e.g. ANSYS).

The REAP methodology directly addresses the three gaps discussed in the previous section. The primary functionality of a REAP system (a realization of REAP methodology) is to support the *constructive assembly* of prefabricated REAP components into a complete PAM. The REAP system directly applies *component software* concepts and principles, enabling independent reusability and interoperation. The REAP components are designed to be *interactive*, by having prescribed mechanisms (interfaces) for the user and the system to select design parameters, analysis parameters, and various analysis idealizations. The *interactivity* concept has been expanded into the interfacing between the REAP component and: another REAP component, the assembly user, and the REAP framework. These analysis components are associated with physical design components, typically “standard parts”, and are capable of representing multiple analysis idealizations. A generalized overview of what REAP system does is illustrated in Figure 5. The ultimate question to be asked here in this proposed thesis is whether engineering analysis preparation can be modularized into a component system. The hypothesis is “yes”, and it is a thesis objective to prove it using REAP methodology.

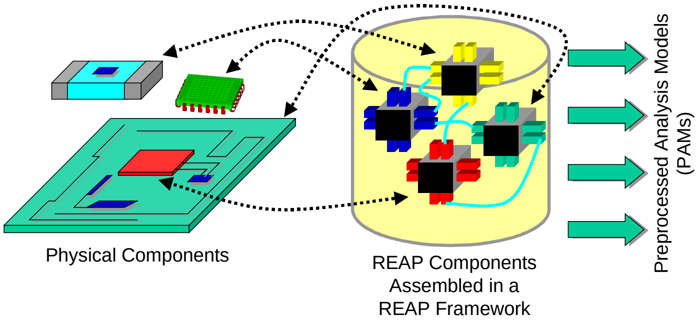

Physical Components
REAP Components Assembled in a REAP Framework
Preprocessed Analysis Models (PAMs)

Figure 5: General overview of how a REAP-compliant system works

5.2 Approach Conceptualization

A REAP system consists of the analysis components, a component framework that governs the interactions between the components, algorithms for assembly, and a set of rules to ensure validity of analysis model. One may ask why associating REAP components with physical design components is appropriate. Here is the reason. Many new product designs often have physical components (e.g. standard parts) that are in common with older designs. Since these physical components are reused in new designs, it makes sense to based “reusable-ness” of analysis components on such structures. In addition, for design-analysis integration, there exists a natural semantic mapping between the physical design component and the virtual analysis component for the product designer. This makes the integration of analysis capabilities to design tools less intrusive and easier to understand. Physical components are mechanically joined together during manufacturing; similarly, REAP components are constructively assembled together using algorithms and forming rules to generate the analysis model. A similar approach is also found in Zhou [1996-1997] in a mechanical component library for electronic packaging reliability analysis.

What makes the REAP approach different is that it is component-oriented, focusing on the independent reuse of individual analysis components. The REAP methodology is *not* about *how to implement* an analysis component, nor is it the related data structures. Rather, it is about *how to use* analysis components by using only its interfaces (a.k.a. “black-box” reuse), and the corresponding rules of interactions. This includes interfaces that exchange attributes and data structure on several different perspectives (e.g. analysis-control related, geometry related, etc.).

A REAP framework enforces various interaction rules that ensure valid and compatibility PAMs. How a REAP component is implemented internally does not matter to the workability of the methodology. In fact, a component's implementation may not even be object-oriented, or may even be scripts executing on a commercial application. By concentrating on interfaces and rules, this methodology allows flexible, independent implementations of REAP components that are guaranteed to be "plug-and-play". For example, one REAP component could be implemented as a parametric model in ANSYS PREP7 files [Zhou 1996], and another component could be implemented, by someone else, as a geometric formation (based on boundary conditions and constraints [Wu 1997]) from a simulation program. By communicating only though the interfaces, these two different and independently developed components can be assembled together to form a compatible FE model.

Another difference in the REAP approach to other design-analysis integration researches is in the granularity of reuse. The REAP methodology allow reuse of "finer" grained components that can be generically assembled to form a PAM. Other approaches that reuses a completed parameterized analysis model is considered a "coarser" grained reuse (e.g. routine analysis in Peak *et al.* [1996]). These "coarser" grained modules cannot be generically combined together, nor do they interface with each other, to formulate a new analysis model.

One may also ask how the REAP methodology is different from pure software component technology today. Typically, such technology is business-oriented, exchanging business data and enforcing business rules. The datasets involved in typical engineering applications are far more complex in both types and sizes [Birkes *et al* 1995]. Specifically, for REAP system, the data and data structures exchanged are very complex, involving interactions, geometry, and analysis. Furthermore, the assembly algorithms and rules enforcing compatibility require an in-depth understand of engineering analysis, computer-aided design geometry, and computer-aided engineering. These very engineering-oriented requirements of the REAP methodology (identifying the datasets, forming the algorithms and analysis-validation rules) makes it beyond the expertise of most software engineers.

5.3 Scope of REAP Methodology

Complete coverage of an all-purpose REAP methodology is considered to be too lofty of a goal for this proposed thesis. Therefore, several restrictions are hereby placed to scope the problem down to a reasonable size. The following is a list of preliminary restrictions placed artificially on the proposed methodology:

- ***Geometry Restriction***: Product to analyze is assembled from several physical components (e.g. standard parts) with geometrically simple interconnect-boundaries that do not overlap (e.g. electronic components on printed circuit board).
- ***Analysis Model Restriction***: Assembled PAM will be a FE model, since FEA is popular, able to represent complex geometry, and also employs a built-up technique (element assembly).
- ***Idealization Restriction***: Focus will be placed mainly on geometry issues, including mesh geometry, rather than on material properties.

5.4 Questions to Address

In the course of researching the REAP methodology, the following preliminary set of high-level questions will be addressed in respect to engineering analysis preparation. This set of questions is not comprehensive, but illustrates the technical challenges in the methodology.

- What information must be exchanged across the component interfaces for constructive assembly to work?
- What functionalities must a component support for constructive assembly?
- What minimum set of features must a component support? How to access extended features generically?
- How does a component map to a physical design component? How to access design parameters generically?
- How to define extensible interfaces?
- How does a component interact with another component, the framework, and the assembling user?
- How to constructively assemble components to generate PAMs?
- What is the format of PAMs? Discrete meshes, parametric FE model, FEA package specific?
- What rules must the REAP framework enforce to achieve Finite Element compatibility in PAMs?
- How to resolve conflicts among requirements of individual components (e.g. mesh division)?

- How to address the problem of combinatorial explosion of possible system configurations?
- Can any analysis models be broken-up into components? If so, how?

6 Proposed Work

To research the validity and feasibility of the REAP methodology, a preliminary structure of the proposed work is hereby described. This structure is designed to address the questions raised in the previous section. The preliminary deliverables for this proposed thesis are listed below and shown in Figure 6, illustrating the layout of the interfaces and rule enforcement. Each one of the deliverables will then be described in detail, and a discussion of the research strategies will conclude this section.

1. ***Interfaces***: Set of interfaces for component and framework to interoperate.
2. ***Required Information Exchange***: Information needed to exchange across interfaces.
3. ***Rules***: Rules of interaction and interoperation to achieve consistent and valid PAMs.
4. ***Assembly Algorithms***: Algorithms to test the generic assembly and the interfaces
5. ***Requirements***: A set of general and feature requirements for component implementations.
6. ***Guidelines***: For development of component and framework, system integration, and user interaction.

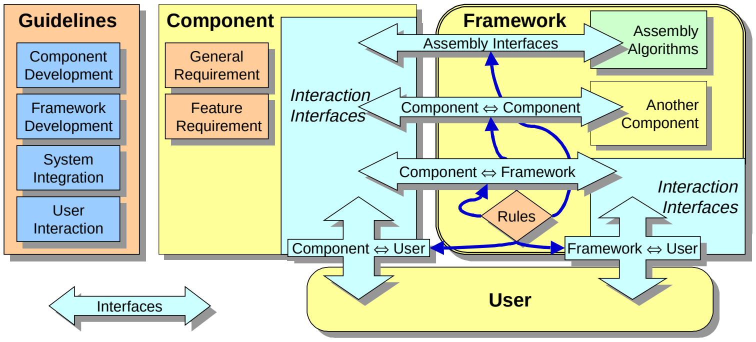


Figure 6: Preliminary deliverables for the proposed thesis

6.1 *The Deliverables*

6.1.1 Interfaces, Information Exchange, and Rules

For the "pieces" of a REAP system to interoperate, a set of interaction interfaces is proposed. To be able to constructively assemble "black-box" components together, details about the geometric interconnect-boundary must be known by the assembly algorithms. This information is accessed through the assembly interfaces. The assembly interfaces are functional interfaces used during assembly to access characteristic information needed about the interconnect-boundaries between the components. This include the operations needed to query, compare, and modify interconnect properties. The set of assembly interfaces is expected to be well defined. Examples of interconnect information exchanged through these interfaces are contact types (e.g. point, edge, or surface contact), topology (e.g. number of loops), shape (e.g. rectangular, single faced), and FEA specifics (e.g. element size, DOF, divisions). Assembly-specific rules may govern how the algorithms access and modify the interconnect information.

When one component is connected to another component, the two can exchange information using the component-component interfaces. The are two categories of component-component interfaces. The first category deals with the interconnect information exchange between the two components. This first set of interfaces is expected to be well defined, and the type of information exchanged is similar to that of the assembly interfaces. It is through these interfaces that two components negotiate to reach a compatible boundary interconnect. The second category of component-component interfaces is high customizable, allowing components of the similar family to exchange highly technical and detail information to interoperate. It is through this category that product-specific extensions can be added. This second set of interfaces is expected to be highly mutable. Rules of interaction will strictly govern the first category of component-component interfaces to ensure validity, but may not govern the second category.

The communication between the framework and the components is through the component-framework interfaces. This set of interfaces is expected to be well-defined and not overly technical. The information exchanged between the framework and the components are mostly high-level directives, component statuses, and interconnect conditions. By designing these interfaces to be high-level, the framework can then easily integrated into other systems.

Since REAP methodology allows user to interactively control the system, two sets of interfaces to the user are defined. The component-user interfaces allow the user to interact directly with the component. The information passed through the component-user interfaces may be highly customized to the implementation of the component, and can include parameters such as design, analysis idealization selection, mesh density, etc. It is expected that the component-user interfaces are highly mutable, with perhaps a few defined ones for interconnect-boundary manipulations. The framework-user interfaces are expected to be high-level and well defined. It is through these interfaces that user issue commands to the framework, to be performed upon a number of components. The type of information passed is similar to that of the component-framework interfaces.

The proposed set of interfaces may be summarized as follows. The interfaces relating to interconnects are well defined and highly technical. The interfaces to the components are highly customizable and not as well defined. The exception is when the interfaces are for the framework, where they are of high-level, low technicality, and are well defined. It is expected that the rules will be encapsulated as software components, but not as REAP components, since they are not for assembly.

6.1.2 Assembly Algorithms, Requirements, and Guidelines

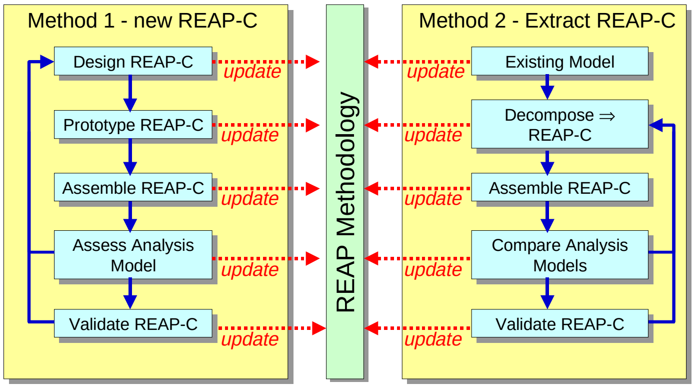
Computer algorithms will be prototyped to test the interfaces and the feasibility of the REAP methodology. These algorithms will assemble the REAP components into PAMs by using only the assembly interfaces. One strict requirement is that these algorithms must always generate valid, converging FE models (e.g. maintain FE compatibility). The prototyping of these algorithms will aid in the research of REAP interfaces, since new insights and requirements will appear during the course of the process. Some attempts will be made to handle different combinations of physical interfaces. However, since the prototyping of algorithms is not a primary objective of this proposed thesis, there will not be an attempt to cover all possible combinations of physical interconnects. Example combinations are edge-to-edge, surface-to-surface, edge-to-surface, etc. It is also expected that the algorithms will be encapsulated as software components (not REAP components), used inside the component framework.

Just specifying the interfaces for REAP components is not enough to ensure workability of the methodology. Therefore, two sets of requirements are placed on the implementation of the REAP components. The first set consists of the general requirements, and these are the absolute requirements that all REAP components must address. For example, a REAP component must be able to enumerate its idealizations, show the number of interconnect-boundaries it has, what physical component it represents, etc. The second set consists of the feature requirements, and these are the required and optional features, related to constructive assembly, that a REAP component supports. Some examples of optional features are the ability to dynamically alter its interconnect-boundary to meet mating condition, and the ability to calculate the stiffness matrix for an equivalent resolved super-element.

Notes and suggestions will be recorded during the prototyping of the REAP methodology. At the conclusion of this proposed thesis, these notes and suggestions will be compiled into guidelines to assist future development using REAP methodology. Four types of guidelines are currently identified: REAP component development guidelines, framework development guidelines, system integration guidelines, and user interaction guidelines. The REAP component development guidelines will provide the “how to” suggestions of developing components. The framework development guidelines will describe and make recommendations on how implement the various pieces of the framework. The system integration guidelines will describe how to integrate the REAP framework into existing design and/or analysis systems (e.g. design tools). The user interaction guidelines will describe how to increase usability of REAP methodology. Possible additional algorithms developed in the course of the research may also be included in the guidelines.

6.2 REAP Research Strategy

The proposed research strategy is the cyclic process of conceptualization, prototyping, testing, and modification. This is an often-used concurrent engineering process for building something that is not fully specified. Two methods are illustrated in Figure 7 for researching and validating the REAP methodology. The first method involves the creation of new REAP components from scratch, prototyping them, assembling them, and then assessing the validity of the PAM they generate. During this process, new insights and requirements will be identified and used to update the REAP methodology. The second method is similar to the first with the exception that the REAP components are decomposed from an existing analysis model, and the validation scheme is to compare the result from the assembled PAM to that of the original analysis model. The second method will also show whether it is feasible to convert an existing analysis model into a set of REAP components that, when assembled, give similar analysis results. Several iterations of these two methods will be done to fully develop the methodology.



* REAP-C = REAP Components

Figure 7: Two methods of building the REAP methodology

7 Example Scenario

An example scenario is described here to aid in the understanding of how REAP components can work together as “black-boxes” to generate finite element models. This scenario will show how the REAP components can exchange the needed information across its interfaces to achieve compatibility. Two types of processes will be shown. The first process is the refinement of a meshed edge that propagates to its connected neighbors. The second process is the altering of idealization that triggers changes throughout the assembly. The product being modeled here is a simple surface-mounted resistor on a printed circuit board. For simplicity, this scenario will in illustrated in 2D.

There are three distinct REAP components in this scenario: resistor, board, and solder body. These components are associated to the physical product, illustrated on the far-left side of Figure 8. There are four instances of REAP components, since there are two solder bodies (left and right). In the initialization stage the components are assembled together, shown on the middle-left side of Figure 8. Note that each component has two interfaces boundaries, illustrated by the two curved lines connected to each component. On the right side of Figure 8, some geometry parameters and options are shown for each component. The names and syntax of the parameters are fictitiously created solely for the purpose of this example. The description of the parameters is shown in Figure 9. Note that the resistor component (“R”) has only one shape (rectangular), but has two optional boundaries. The solder body component (“SL” and “SR”) has two possible shapes (rectangular and custom), and can be used to represent an alternate idealization.

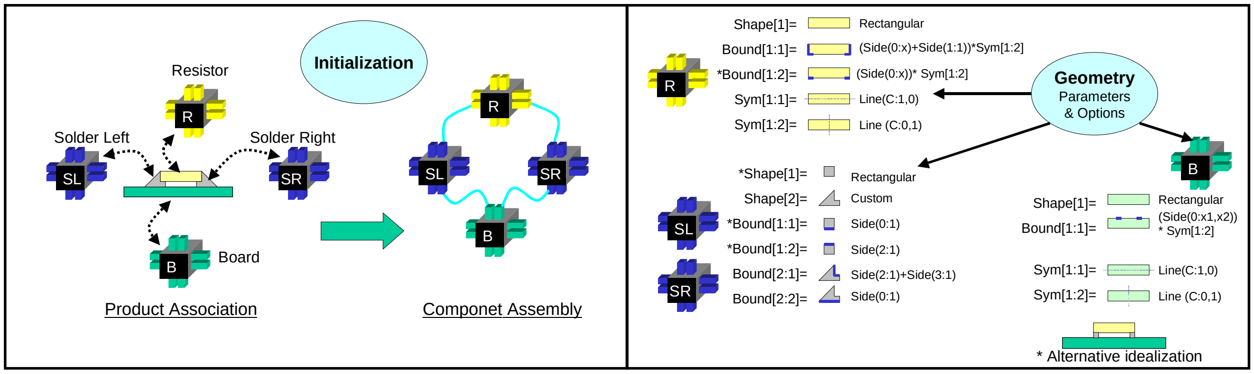
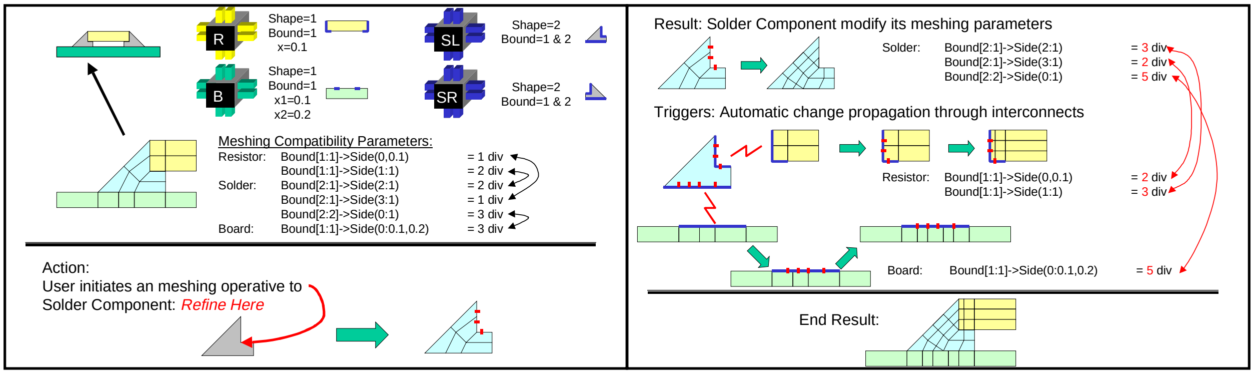


Figure 8: Scenario – Initialization and Parameters

- **Shape**[*shape_id*] indicates the shape(s) that the component can assume.
- **Bound**[*shape_id: version_id*] describes the boundary of component .
    - **Side**(*side_id: fraction*) describes an edge on component , which may be partial if *fraction* < 1. *Side_id* is counted clockwise from the bottom.
    - **Side**(*side_id: fraction_starting, fraction_ending*) is partial edge not starting from 0.
    - **Side**() may be added up, and/or multiplied mirror reflection along line of symmetry.
- **Sym**[*shape_id: version_id*] describe the line of symmetry
    - **Line**(*point_intersected: Vector X comp., Vector Y comp.*) describes a line passing through *point_intersected* (*C* is the geometric centroid) in direction indicated by the vector components.

Figure 9 : Scenario – Parameter list descriptions

For further simplicity, let the assembly be represented by only the left symmetric half-model. Figure 10 illustrates what happens when the user requests a refinement of FE edge. Initially, the assembly is meshed with the default number of FE edge divisions, illustrated on the left side of Figure 10. Note that the “Meshing Compatibility Parameters” chart show the respective number of divisions along boundaries of individual components, and curved arrows show the matching numbers between components that share a common boundary. On the lower-left corner of Figure 10, an action is taken to refine the edge divisions on a boundary of the solder body. Note that red coloring is used on Figure 10 and Figure 11 to indicate the propagated change due to an action. For example, small red ticks along a boundary indicate a new edge division. As a result, shown on the right side of Figure 10, the component self-modifies its meshing, which also changes the number of edge divisions on its other boundary. Because other components are connected to these changed boundaries, messages are sent from the solder body to the resistor and the board. A connected component receives the new edge division and applies it to its boundary, then self-modifies its meshing. The end result is shown on the lower-right corner of Figure 10.



Meshing Compatibility Parameters:

Resistor:	Bound[1:1]->Side(0,0.1)	= 1 div
	Bound[1:1]->Side(1:1)	= 2 div
Solder:	Bound[2:1]->Side(2:1)	= 2 div
	Bound[2:1]->Side(3:1)	= 1 div
	Bound[2:2]->Side(0:1)	= 3 div
Board:	Bound[1:1]->Side(0:0.1,0.2)	= 3 div

Action:
User initiates an meshing operative to
Solder Component: *Refine Here*

Result: Solder Component modify its meshing parameters

Solder:	Bound[2:1]->Side(2:1)	= 3 div
	Bound[2:1]->Side(3:1)	= 2 div
	Bound[2:2]->Side(0:1)	= 5 div

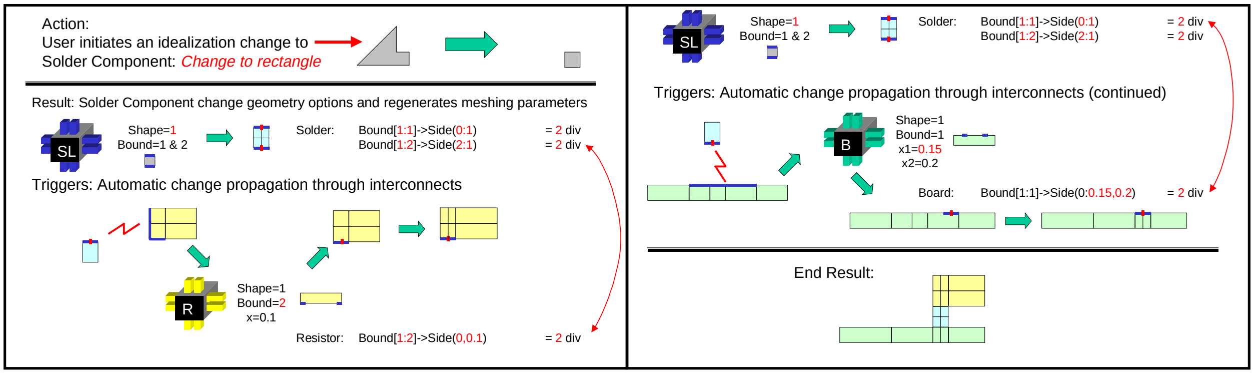
Triggers: Automatic change propagation through interconnects

Resistor:	Bound[1:1]->Side(0,0.1)	= 2 div
	Bound[1:1]->Side(1:1)	= 3 div
Board:	Bound[1:1]->Side(0:0.1,0.2)	= 5 div

End Result:

Figure 10: Scenario - Mesh refinement

Figure 11 illustrates what happens when the user requests a change in idealization of the solder body from the custom shape to a rectangular shape. The solder body component changes its boundaries to a different version and regenerates its meshes. Due to the change in boundaries, messages are sent to connected components and a similar change propagation process occurs. The end result is shown on the lower-right corner of Figure 11.



Action:
User initiates an idealization change to
Solder Component: *Change to rectangle*

Result: Solder Component change geometry options and regenerates meshing parameters

Solder:	Bound[1:1]->Side(0:1)	= 2 div
	Bound[1:2]->Side(2:1)	= 2 div

Triggers: Automatic change propagation through interconnects

Resistor:	Bound[1:2]->Side(0,0.1)	= 2 div

Solder:	Bound[1:1]->Side(0:1)	= 2 div
	Bound[1:2]->Side(2:1)	= 2 div

Triggers: Automatic change propagation through interconnects (continued)

Board:	Bound[1:1]->Side(0:0.15,0.2)	= 2 div

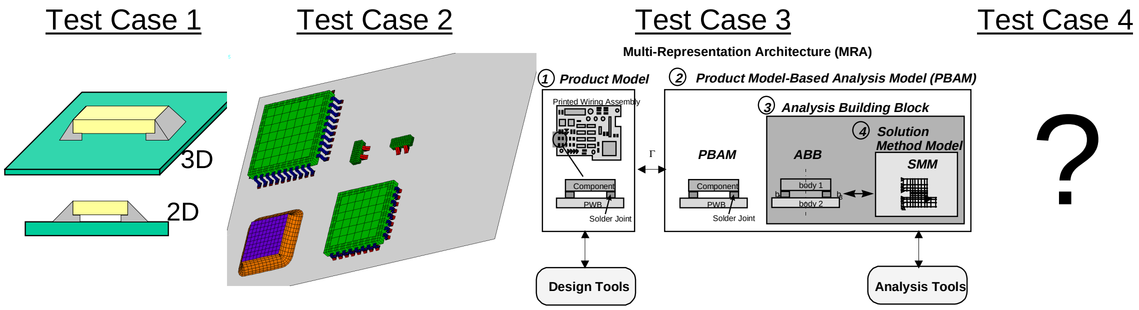
End Result:

Figure 11: Scenario – Idealization Change

The changes may propagate to other connected components, and if a conflict occurs (loop-back), some sort of negotiation among the components would occur. This is not shown in this simple example, and will be addressed.

8 Representative Test Cases

The preliminary test cases, illustrated in Figure 12, are used to examine the validity and usefulness of the REAP methodology. The first two test cases are applications of REAP research strategy, for their development will update the methodology. The third test case will be examining the feasibility of integrating REAP system into an existing architecture. The fourth test case is to be determined, depending on the outcome of the other test cases. Details of each test case are described below.



Test #	Case Name	Description	Analysis Types	Engineering Software
1	SMT Resistor	Surface-Mounted Resistor on a PWB	Thermal, 2D, 3D, multiple idealizations	ANSYS, IDEAS(?) ProE(?)
2	PWA	General Printed Wiring Assembly	Global/Local, Thermomechanical, 3D, multiple idealizations, etc.	ANSYS, IDEAS(?) ProE(?)
3	MRA Intergration	Integration of REAP with Multi-Representation Architecture [Peak 1995]	Varies	ANSYS
4	To be determined	To be determined	To be determined	To be determined

Figure 12: Illustration of test cases 1 - 4

- ***Test Case 1 - Surface Mounted Resistor***: This test case will involve thermal analysis of a surface-mounted resistor on a Printed Wiring Board (PWB). This test will involve four REAP components (the resister, two solder bodies, and the board), and will have multiple idealizations such as 2D, simplified solder bodies, etc. This will be done using method 1 (described in REAP Research Strategy), as well method 2, using the pre-developed parametric model developed in the Tiger PWA Toolset [Peak 1997].
- ***Test Case 2 – Printed Wiring Assembly***: This test case will involve general thermomechanical analysis of Printed Wiring Assembly (PWA). Several REAP components will be developed using both methods 1 and 2, with existing analysis models from Zhou [1997]. It is expected that this will be the most time-consuming test case, and it will most heavily test and update the REAP methodology.
- ***Test Case 3 – MRA Integration***: This test case will examine the system integration issues by attempting to integrate REAP system into MRA [Peak 1995].
- ***Test Case 4 – To be determined***: This test case is to be determined later, based on the outcome of others.

9 Validation

The purpose of the validation is to show that the REAP methodology works and assess its usefulness in reducing the reusability gap in the design-analysis integration. A number of preliminary validation criteria, along with metrics, are created for comparing the results from the test cases. These criteria are based on the expected benefits of the methodology, and are shown in Table 2. The criteria are related to the benefits of the analysis preparation and of the component-oriented approach. Some predicted metrics values are entered for test case 1, and others are to be developed. The list of criteria is by no means complete and will be further developed in the course of this research. This is indicated in the estimated work schedule, illustrated in Table 3.

Criteria	Metrics	TC1	TC2	TC3	TC4
Produce compatible models	% compatible	100%			
Reduce user time	% time saved	> 50%			
Multiple idealizations	# idealizations	4+			
Multiple analysis models	# analysis models	2+			
Accurate analysis results	% error	< 10%			
Multiple analysis applications	# applications	2			
Can convert existing models	% converted	100%			
Generate new components	# components	3			
Reusable components	# reused	1			
Allow diverse implementations	# different types	2			
Interfaces are sufficient	# not workable	0			
Customizable interfaces	# customized				
Integrate to existing system	% workable	-			
etc…					

Table 2: Validation criteria versus the test cases

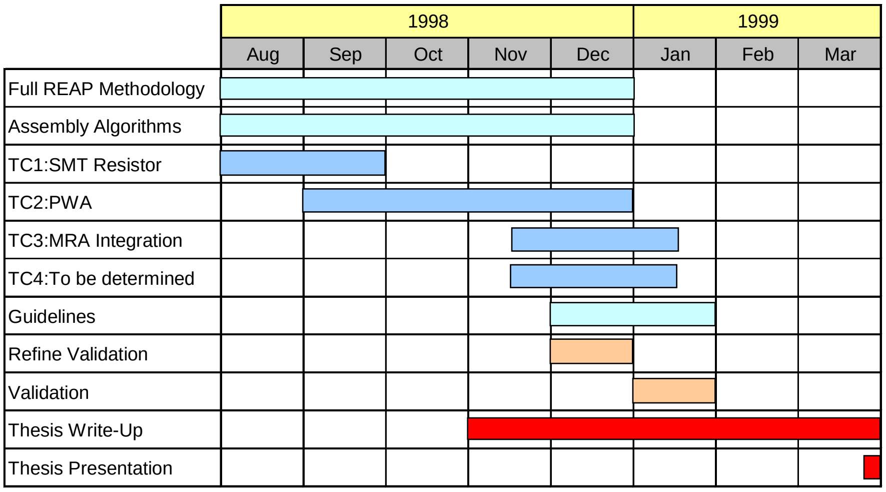


Table 3: Estimated Work Schedule

10 Expected Contribution

Successful research of the Reusable Engineering Analysis Preparation (REAP) methodology will contribute significantly to the efforts in design-analysis integration, as well as in general engineering analysis. This methodology separates the involved “analysts” into two categories: experienced analyst that implement their expertise into REAP components, and “lesser experience-required” analyst (e.g. designers) that reuse expert implemented REAP components to generate new analysis models. Thus, for design-analysis integration, a designer can perform flexible, effective reliability analysis on new designs both quickly and easily. This is the primary benefit of targeting third party reuse. By mapping REAP components to physical design components, REAP methodology also facilitates natural semantic mapping and less intrusive integration of analysis capabilities into the design environment.

This methodology also reaps the benefits from component software technology. This methodology inherits the enhanced adaptability, scalability, and maintainability. It allows multiple independent development of REAP components that can interoperate, so that the component customer can safely choose among different component implementations to best represent the problem. It allows unique and innovative preprocessing techniques to be combined with more traditional preprocessing, since only the interfaces have to be compatible. If something like REAP methodology becomes an accepted standard, a market for analysis components can then be created. This would revolutionize the way engineering analysis is prepared and introduce commercially something like the Rapid Application Development (RAD) environment for engineering analysis.

References

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

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

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

Baldwin, D. F., Beerensson, J. T., and Sitaraman, S. K. (1996). "Thermal management in direct chip attach assemblies." *American Society of Mechanical Engineers, EEP*, 17, 29-37.

Birkes, A. Y., Hsiung, C. H., Cohen, T., and Fulton, R. E. (1995). "A Prototype Multimedia Auto Broker." , *The 1995 ASME 15th Annual Computers in Engineering Conference*, Boston, MA.

Brooke, D. M., Pennington, A., and Bloor, S. M. (1995). "An Ontology for Engineering Analyses." *Engineering with Computers*(11), 36-45.

Buratynski, E. K. (1993). "Thermomechanical modeling of Direct Chip Interconnection assembly." *Journal of Electronic Packaging, Transactions of the ASME*, 115, 382-391.

Burkett, W. C., and Yuhwei, Y. (1995). "The STEP Integration Information Architecture." *Engineering with Computers*(11), 136-144.

Cimtalay, S., and Fulton, R. E. (1994). "Parametric Design of Heat Sink: Multiple Trade Offs." *ASME Winter Annual Meeting*, EEP-9, 53-58.

Cimtalay, S., Peak, R. S., and Fulton, R. E. (1996). "Optimization of Solder Joint Fatigue Life Using Product Model-Based Analysis Models." *ASME Winter Annual Meeting*, 18.

Cohen, T., Birkes, A. Y., Hsiung, C. H., and Fulton, R. E. (1995). "Modeling Patient Care for Multimedia Databases." , *The 1995 ASME 15th Annual Computers in Engineering Conference*, Boston, MA.

Cox, F. L., and Jazbutis, G. B. (1997). "Models of Electronic Package Engineering." To be published, , *InterPACK '97, ASME*, Kohala Coast, Hawaii.

Deitz, D. (1996). "Java: A New Tool for Engineering." Mechanical Engineering, 68-72.

Deitz, D. (1997). "The Convergence of Design and Analysis." *Mechanical Engineering*, 119(3), 93-100.

Dragan, R. V. (1997). "Java Tools Get Real." PC Magazine, 181-214.

Finn, D. P. (1993). "A Physical Modeling Assistant for the Preliminary Stages of Finite Element Analysis." *(AI EDAM) Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 7(4), 275-286.

Hardell, C. (1996). "An Integrated System for Computer Aided Design and Analysis of Multibody Systems." *Engineering with Computers*(12), 23-33.

Hardwick, M., Spooner, D. L., Rando, T., and Morris, K. C. (1996). "Sharing manufacturing information in virtual enterprises." *Communications of the ACM*, 39(2), 46-54.

Korngold, E. V., Shephard, M. S., Wentorf, R., and Spooner, D. (1989). "Architecture of a Design System for Engineering Idealizations." 1, , *ASME Design Automation Conference*, 259-265.

Kurowski, P., and Dvorak, P. J. E. (1996). "Good solid modeling, bad FEA. (finite element analysis)." *Machine Design*, 68(November 21), 67-73.

Lau, J. H. (1993). "Thermomechanics for Electronics Packaging." Thermal Stress and Strain in Microelectronics Packaging, J. H. Lau, ed., Van Nostrand Reinhold, New York, 1-77.

Laurance, N. (1994). "A High-level View of STEP." *Manufacturing Review*, 7(1), 39-46.

Le Gall, C. A., Qu, J., and McDowell, D. L. (1996). "Delamination cracking in encapsulated flip chips." *Proceedings - Electronic Components and Technology Conference*, 430-434.

Liker, J., Fleischer, M., and Arnsdorf, D. (1992). "Fulfilling the Promises of CAD." Sloan Management Review, 74-86.

Mortenson, M. E. (1985). *Geometric Modeling*, Wiley & Sons, New York.

O'Bara, R. M., Beall, M. W., and Shephard, M. S. (1995). “Analysis model visualization and graphical analysis attribute specification system.” *Finite Elements in Analysis and Design*, May 4 1995, p 325-345.

Otte, R., Patrick, P., and Roy, M. (1996). *Understanding CORBA (Common Object Request Broker Architecture)*, Prentice Hall PTR, New Jersey.

Peak, R., S. (1993). “Product-Based Analytical Models (PBAMs): A New Representation of Engineering Analysis Models,” Doctoral Thesis, Georgia Institute of Technology, Atlanta.

Peak, R. S. (1997). “DaiTools - A Design-Analysis Integration Toolkit.” , http://eislab.gatech.edu/.

Peak, R. S., and Fulton, R. E. (1993). “Automating Routine Analysis in Electronic Packaging Using Product Model-Based Analytical Models (PBAMs), Part I: PBAM Overview.” *ASME Winter Annual Meeting*.

Peak, R. S., and Fulton, R. E. (1993). “Automating Routine Analysis in Electronic Packaging Using Product Model-Based Analytical Models (PBAMs), Part II: Solder Joint Fatigue Case Studies.” *ASME Winter Annual Meeting*, 93-WA/EEP-24.

Peak, R. S., and Fulton, R. E. (1993). “A Multi-Representation Approach to CAD/CAE Integration: Research Overview”, Report No. MS-93-03, Manufacturing Research Center, Georgia Institute of Technology, Atlanta, GA.

Peak, R. S., Fulton, R. E., Nishigaki, I., and Okamoto, N. (1995). “Integrating Engineering Design and Analysis Using a Multi-Representation Approach.” *Engineering with Computers*, to appear.

Peak, R. S., Fulton, R. E., and Sitaraman, S. K. (1997). “Thermomechanical CAD/CAE Integration in the TIGER PWA Toolset.” *InterPACK '97*, To be published.

Peak, R. S., Scholand, A., and Fulton, R. E. (1996). “On the Routinization of Analysis for Physical Design.” *1996 ASME International Mechanical Engineering Congress and Exposition*, 18, 73-82.

Porter, S. (1990). “FEA faces a changing role: 'easy-to-use' tools prompt debate.” Computer Graphics World, 39-47.

Rocheleau, D. N., and Lee, K. (1987). “System for interactive assembly modelling.” *Computer-Aided Design*, 19(2), 65-72.

Rosen, D. W., Dixon, J. R., and Dong, X. (1991). “A Methodology for Conversions of Feature-based Representations.” *Proceedings of the Design Theory and Methodology - DTM '91*, 31, 45-51.

Russell, R. (1996). “Don't trust the pretty pictures.” *Machine Design*, 68(May 23), 68-84.

Sahu, K., and Grosse, I. R. (1994). “Concurrent Iterative Design and the Integration of Finite Element Analysis Results.” *Engineering with Computers*, 10(4), 245-257.

Sametinger, J. (1997). *Software Engineering with Reusable Components*, Springer, Berlin ; New York.

Satoh, R. (1993). “A Prediction of the Thermal Fatigue Life of Solder Joints Using Crack Propagation Rate and Equivalent Strain Range.” Thermal Stress and Strain in Microelectronics Packaging, J. H. Lau, ed., Van Nostrand Reinhold, New York, 500-531.

Scholand, A. J., Peak, R. S., and Fulton, R. E. (1997). “The Engineering Service Bureau - Empowering SMEs to Improve Collaboratively Developed Products.” *CALS Expo USA*, To be published.

Scholand, A. J., Peak, R. S., and Fulton, R. E. (1997). “Improving Electronic Packaging Manufacturing Through Product and Process-Driven Analysis: A PWB Case Study.” *ASME International Mech. Engr. Congress & Expo.*

Shephard, M. S., Baehmann, P. L., Georges, M., K., and Korngold, E. V. (1990b). “Framework for the Reliable Generation and Control of Analysis Idealizations.” *Computer Methods in Applied Mechanics and Engineering*, 82, 257-280.

Shephard, M. S., Baehmann, P. L., Le Coz, Y. L., and Sham, T. L. (1992). “Methodology for the Integration of Global/Local Thermal and Thermo-Mechanical Analyses of Multichip Modules.” *Computer Aided Design in Electronic Packaging - ASME*, EEP- Vol 3, 65-72.

Shephard, M. S., and Finnagan, P. M. (1988). “Integration of Geometric Modeling and Advanced Finite Element Preprocessing.” *Finite Elements in Analysis and Design*, 4(2), 147-161.

Shephard, M. S., Korngold, E. V., and Wentorf, R. (1990a). “Design Systems Supporting Engineering Idealizations.” Geometric Modeling for Product Engineering, M. J. Wozny, J. U. Turner , and K. Preiss, eds., Elsevier Science Publishers, North-Holland, 279-299.

Shephard, M. S., and Wentorf, R. (1994). “Toward the Implementation of Automated Analysis Idealization Control.” *Applied Numerical Mathematics*, 14(1-3), 105-124.

Shigley, J. E., and Mischke, C. R. (1989). *Mechanical Engineering Design*, McGraw-Hill.

Sizemore, J., and Sitaraman, S. K. (1995). "Elastic-Plastic and Elastic-Viscoplastic Modeling of Plated-Through Holes in Solder Shock Test,." *ASME Intl. Mech. Engr. Congress & Expo*, HTD-Vol. 319/EEP-Vol. 15.

Spooner, D., and Hardwick, M. (1993). "Using Persistent Object Technology to Support Concurrent Engineering Systems." Concurrent Engineering: Methodology and Applications, P. Gu and A. Kusiak, eds., Elsevier Science, New York, 205-234.

Subramanyam, S., and Rolph, W. D. (1989). "Thermal analysis of direct chip mount electronic systems using ADINA-T." *Computers and Structures*, 32(3-4), 853-859.

Szyperski, C. (1998). *Component Software: Beyond Object-Oriented Programming*, Addison-Wesley, New York.

Tamburini, D. R. (1997). "The Analyzable Product Model Representation to Support Design-Analysis Integration," Doctorial Thesis Proposal, Georgia Institute of Technology, Atlanta, GA.

Tamburini, D. R., Peak, R. S., and Fulton, R. E. (1996). "Populating Product Data for Engineering Analysis With Applications to Printed Wiring Assemblies." *1996 ASME International Mechanical Engineering Congress and Exposition*, EEP-18, 33-46.

Tamburini, D. R., Peak, R. S., and Fulton, R. E. (1997). "Driving PWA Thermomechanical Analysis from STEP AP210 Product Models." *1997 ASME International Mechanical Engineering Congress and Exposition*, To be published.

Tummala, R. R., Rymaszewski, E. J., and Klopfenstein, A. G. (1997). *Microelectronics Packaging Handbook (Part I, II and III)*, Chapman & Hall, New York.

Vanhelsuwe, L. (1997). *Mastering JavaBeans*, Sybex Inc.

Wentorf, R., and Shephard, M. S. (1993). "User Interface Functions for an Analysis Idealization System." *Microcomputers in Civil Engineering*, 8(2), 85-95.

Wu, X., Hu, K., Dou, X., Mui, G., Yeh, C.-p., and Wyatt, K. (1997). "Solder joint formation simulation and reliability prediction." , *TMS Annual Meeting*, Orlando, FL, USA, 179-185.

Yang, H., and Hoeltzel, D. A. (1992). "Automatic finite element mesh generation over domains comprised of multiple connected, intersecting, rigid body-movable subdomains." *Journal of Mechanical Design*, 114(December), 603-615.

Yeh, C.-p. (1992). "An Integrated Information Framework for Multidisciplinary PWB Design," Doctoral Thesis, Georgia Institute of Technology, Atlanta, GA.

Yeh, C.-P., Ume, C., Fulton, R. E., Wyatt, K. W., and Stafford, J. W. (1993). "Correlation of analytical and experimental approaches to determine thermally induced PWB warpage." Thermal Stress and Strain in Microelectronics Packaging, J. H. Lau, ed., Van Nostrand Reinhold, New York, 305-327.

Yu, Q., Shiratori, M., and Wang, S.-B. (1994). "Thermal conduction and thermal stress analyses of surface-mount assembly with a solder joint element." *American Society of Mechanical Engineers, Applied Mechanics Division, AMD*, 2, 205-211.

Zheng, W. C. (1996). "Curing analysis of a flip-chip-on-board electronic package." *American Society of Mechanical Engineers, EEP*, 17, 13-22.

Zhou, W. X. (1996). "Modularized & Parametric Modeling Methodology for Concurrent Mechanical Design of Electronic Packaging," Doctoral Thesis, Georgia Institute of Technology, Atlanta.

Zhou, W. X., and Fulton, R. E. (1995). "Modularized & parametric finite element methodology for concurrent design, analysis, and simulation." *American Society of Mechanical Engineers, EEP*, 10(1), 173-185.

Zhou, W. X., Hsiung, C. H., Fulton, R. E., Yin, X. F., Yeh, C. P., and Wyatt, K. (1997). "CAD-Based Analysis Tools for Electronic Packaging Design (A New Modeling Methodology for a Virtual Development Environment)." To be published, , *InterPACK '97, ASME*, Kohala Coast, Hawaii.