Object Oriented Paradigm for Optimization Model Enhancement

A Thesis
Presented to
The Academic Faculty

by

Selçuk Cimtalay

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in Mechanical Engineering

Georgia Institute of Technology
October 2000

Object Oriented Paradigm for Optimization Model Enhancement

Approved:

Dr. Robert E. Fulton, Chairman

Dr. Russell Peak, Co-Chairman

Dr. Daniel P. Shrage

Dr. Suresh K. Sitaraman

Dr. Madhavan Swaminathan

Dr. Charles Ume

Date Approved: 11/27/00

DEDICATION

To my family

ACKNOWLEDGEMENTS

During the Ph.D. marathon, I appreciate the invaluable help and encouragement from many friends, colleagues, and members of my family, to whom I will attempt to express my deepest gratitude in the paragraphs that follow.

First, I must start thanking my dear advisor, Dr. Robert E. Fulton for sharing his technical and managerial knowledge and vast experience with me as a father figure. I also want to thank him for giving me the opportunity to participate in real, meaningful projects that enriched my experience as a Ph.D. student.

I want to thank Dr. Russell S. Peak for his continuous guidance, technical and moral support throughout my research and during the production of this thesis. I also thank him for his friendship, his thoroughness, insightful, diligent comments and for helping me keep the quality of this thesis always on check.

Additionally, I would like to thank the other distinguished members of my thesis committee: Drs., Daniel P. Schrage, Suresh K. Sitaraman, Madhavan Swaminathan and Charles Ume for their time, patience and valuable feedback.

I would like to thank my fellow members of Engineering Information Systems Laboratory at Georgia Tech team: Angela Birkes, Ashok Chandrasekhar, Tal Cohen,

Chien Hsiung, Haruko Peak, M.C. Ramesh, Andy Scholand, Diego Tamburini and Miyako Wilson for their constant support and friendship. I would also like to thank Donna Rogers for her continuous help and for being so supportive with the numerous administrative issues that arose during my studies.

I was very privileged for having the opportunity to participate in several industry-sponsored projects that enriched my Ph.D. experience – and this thesis - enormously. I would like to extend my gratitude to those companies and individuals that provided their support and guidance for these projects. From the TIGER project: Lynwood Hines, Gerry Graves and Mike Stiteler (SCRA); Ken Buchannan and Darla Nettles (Arthur D. Little); Adam Owsley (ITI); Brian Anderson and Jeff Lewis (Holaday Circuits); David Cristenson, Mike Keenan and Greg Smith (Boeing). From the Product Simulation Integration (PSI) Structures Project: Hugo Korwaser and David Leal (CAESAR Systems); Mats Kjellberg, Mats Nilsson, Nigel Shaw and Staffan Westbeck (EuroSTEP); Jack Blaylock, Kal Brauner, Dave Briggs, Richard Brooks, Rod Dreisbach, Dave Jack, Lee Krueger, Dan Kwong, Bob Masters, Dale Novstrup, Marty Prather, Frank Robl, Joe Sharp, Bob Thomas, Scott Tsao, Ching-Yang Wang, Peter Wilson and King Yee (Boeing Commercial Airplane Group). Special thanks to my colleagues at Georgia Perimeter College (Soheyla Mohebbi, Margaret Major, Kari Jones).

Thanks also to my dear friends Lala Cochran, Meredith Golde, Kristin Jones, Sinan Lazoglu, Erol Tutumluer. I thank them for their constant support and encouragement.

Finally, I would like to thank my family for their vote of confidence and patience.

TABLE OF CONTENTS

Dedication iii

Acknowledgements iv

Table of Contents vi

List of Tables ix

List of Figures x

Summary xiii

1. Introduction 1

1.1 Information Integration for Optimization 3

1.2 Enhancing Optimization Models During Design Progression 4

2. Background and Related Research 7

2.1 Background: Optimization and Analysis 7

2.2 Design and Analysis Integration Review 11

2.3 Analysis and Optimization Integration 13

2.4 GIT Research at the Engineering Information Systems Laboratory Design-Analysis Integration 17

2.4.1The Multi-Representation Architecture (MRA) 17

2.4.2 Team Integrated Electronic Response (TIGER) Project 20

2.4.3 Product Simulation Integration (PSI) Structures Project 21

2.5 Gaps in Literature 23

3. Problem Definition 25

3.1 Integration among Optimization, Analysis and Design Representations 26

3.2 Building and Enhancement of Optimization Models 29

3.3 Linkage among Classes of Optimization and Analysis Models 30

3.4 Automated Optimization Method and Tool Selection 31

3.5 Support Enhancement of Optimization Models for Disciplinary Systems 32

3.6 Summary of the Gaps 33

3.7 Objectives of Thesis 34

4. Enhanced Optimization Model (EOM) Representation 39

4.1 Partition Of Engineering Optimization Domain 40

4.2 Enhanced Optimization Model (EOM) Information Structure 45
4.2.1 EOM Representation Information Meta Model 48
4.2.1.1 Overview 48
4.2.1.2 Definitions 52
4.2.1.3 Mathematical Description of EOM 59
4.2.2 EOM Description Languages 62
4.2.2.1 EOM Lexical Forms 63
4.2.2.2 Graphical Representation Languages 71
4.2.3 EOM Protocol 78

5. EOM Pepresentation Implementation 93

5.1 Computer Implementation EOM Representation 93

6. Test Cases 110

6.1 Prototype: Flap-Link 113
6.1.1 Physical Description 113
6.1.2 Engineering Optimization Model 115
6.1.3 EOM Representations of FlapLink 116
6.1.4 Enhancement of EOM 140

6.2 Structural Optimization: 10-Bar Truss 153
6.2.1 Physical Description 153
6.2.2 Engineering Optimization Model: 154
6.2.3 EOM Lexical Forms: 155

6.3 Solder Fatigue Case Study 159
6.3.1 Physical Description 159
6.3.2 Analysis Models (CBAMs) 160
6.3.3 Optimization Models 166
6.3.4 EOM Representation 179

7. Evaluation 185

7.1 Evaluation Approach and Scope 185

7.2 Evaluation of EOM Representation 186
7.2.1 Optimization Integration 188
7.2.2 Enhancement of Optimization Models 190
7.2.3 Representation supports automated Optimization Method and Tool Selection 198
7.2.4 Optimization Models from Various Engineering Disciplines 199
7.2.5 Extend and Utilize Multi-Representation Architecture (MRA) 199
7.3 EOM Representation versus Classical Optimization Model 200

8. Closure 203

8.1 Uniqueness of Solution and Benefits 204

8.2 Recommended Extensions 206

APPENDIX A EOM Schemas for Flap Link 210

APPENDIX B CBAM Schema for Flap Link 214

APPENDIX C ABBs 217

APPENDIX D APM Schema for Flap Link (Preliminary Design) 227

APPENDIX E EOM, CBAM AND APM Instance Forms **228**
References **231**
Vita **236**

LIST OF TABLES

Table 4-1 EOM Languages 63
Table 6-1 Material Table for Flap Link 142
Table 6-2 Ten-Bar Optimization Data 154
Table 6-3 Optimum Results Comparison for Ten Bar Truss 158
Table 6-4 Material Properties of PWA 166
Table 6-5 Design Variable and Nf Final Results 169
Table 6-6 Maximum Nf values for Different Materials 170
Table 6-7 Enhanced Optimization Model Results 174
Table 6-8 Design Variables, Shear Strain and Nf Final Results 177
Table 7-1 EOM Representation Evaluation 187
Table 7-2 EOM Models for Flap Link 192
Table 7-3 CBAMs: Flap Link Case Study Analyses 194
Table 7-4 Product Models for Solder Joint Fatigue Case Study 194
Table 7-5 EOM Models for Solder Joint 195
Table 7-6 CBAMs: Solder Joint Fatigue Case Study Analyses 197
Table 7-7 Product Models for Solder Joint Fatigue Case Study 197

LIST OF FIGURES

Figure 1-1 Enhancement of Analysis- Optimization Integration 5
Figure 2-1 Optimization and Analysis 7
Figure 2-2 Design - Analysis - Optimization Process 8
Figure 2-3 Multi-Representation Architecture for Design-Analysis Integration 18
Figure 3-1 Framework for Problem Definition 25
Figure 3-2 Associtivity among Optimization, Analysis and Design Models 28
Figure 3-3 Construction of Alternative Optimization Models 30
Figure 3-4 Disciplinary Design Example 32
Figure 3-5 Disciplinary Model Enhancement 33
Figure 4-1 Partition of Engineering Entities 41
Figure 4-2 Entities of Optimization, Analysis and Design in EOM 44
Figure 4-3 Main Components of EOM Representation 46
Figure 4-4 Overview of the EOM Description Languages 47
Figure 4-5 A View of the EOM Information Model 49
Figure 4-6 EOM Attribute Classification 52
Figure 4-7 EOM Relation Classification 56
Figure 4-8 Information Model for EOM Language Generation (Express-G Diagram) 64
Figure 4-9 Object_ Category: Categories of EOM, CBAM and APM 67
Figure 4-10 Illustration of EOM Lexical Form (COS) 69
Figure 4-11 COI for eom1 70
Figure 4-12 Fundamental Constraint Schematic Notation 73
Figure 4-13 EOM Constraint Schematic Form 75
Figure 4-14 EOM Process Diagram 78
Figure 4-15 Tasks for Input and Data Execution in EOM Protocol 84
Figure 4-16 Extended Constraint Network Diagram Example 85
Figure 4-17 Tasks for Data Execution in EOM Protocol 86
Figure 4-18 Classes of Optimization Methods and Tools 88
Figure 5-1 EOM Implementation with its Phase and Computer Architecture 95
Figure 5-2 Methods and Tools used for EOM Implementation 99
Figure 5-3 Optimization Agent Implementation Illustration 101
Figure 5-4 Usage of BolinkGeneral Optimization Program 104

Figure 5-5 Execution of Optimizer for Fatigue Life Optimization 105
Figure 5-6 EOM Browser 107
Figure 5-7 EOM Grapher for Flap Link 108
Figure 6-1 Test Cases versus Main Functions of EOM 110
Figure 6-2 EOM Representation Application and Testing 112
Figure 6-3 Flap Link Engineering Drawing 113
Figure 6-4 Flap Link Design Model (Detailed) 114
Figure 6-5 Flap-Link Design Model (Preliminary Stage) 114
Figure 6-6 Engineering Optimization Model 115
Figure 6-7 Express_G Diagram of Flap Link (Preliminary Design) 121
Figure 6-8 Flap Link COB Definition at Detailed Design Stage 123
Figure 6- 9 Relations and Associativity for Flap Link for eom1 124
Figure 6-10 EOM Constraint Object Relation, er1 125
Figure 6-11 Associativity Links (ala, adl and alo, er1) for Flaplink in eom1 127
Figure 6-12 Fidelity in Analysis and Product Models 129
Figure 6-13 Extensional Rod Analysis Model 130
Figure 6- 14 Constraint Schematic for *Extensional Rod* 131
Figure 6-15 COB Structure Definition for Flap Link 132
Figure 6-16 Representing a Flap Link Analysis as CBAM: Linkage Extensional Model. 133
Figure 6-17 Definition Language for Linkage Extensional Model CBAM 134
Figure 6-18 Higher Fidelity Flap Link CBAM: Linkage Plane Stress Model 135
Figure 6-19 COS - link plane stress ABB 136
Figure 6-20 Buckling Rod Model 137
Figure 6-21 ABB for *Buckling Rod* 138
Figure 6-22 EOM Model for Flap Link 139
Figure 6-23 Class of Optimization Models for Flap Link Example 140
Figure 6 -24 Engineering Optimization Models I and II 141
Figure 6-25 Constraint Schematic Diagram of eom1 143
Figure 6-26 Constraint Schematic Diagram of eom2 144
Figure 6-27 Engineering Optimization Models III and IV 145
Figure 6-28 Constraint Schematic Diagram of eom4 146
Figure 6-29 Detailed Design of Flap Link 147
Figure 6-30 Flap Link FE Analysis 148
Figure 6-31 Engineering Optimization Model V 148

Figure 6-32 Iteration History of Weight of Flap-Link 149
Figure 6-33 Flap Link Detailed -Two Design Variables tw and tf 150
Figure 6-34 Engineering Optimization Model VI 150
Figure 6-35 Results -Optimization Model VI 151
Figure 6-36 Objective Function Weight Iteration History 152
Figure 6-37 Flap Link Detailed Design (4 Design Variables) 152
Figure 6-38 Ten-Bar Truss Example 153
Figure 6-39 Engineering Optimization Model 155
Figure 6-40 EOM for Ten Bar 157
Figure 6-41 Iteration History of Volume of Ten-Bar Truss 158
Figure 6-42 Soldered Component on a PWB 159
Figure 6-43 Engelmaier Model for Solder Joint Deformation 161
Figure 6-44 N^2 Diagram - Prediction of Fatigue Life 162
Figure 6-45 Plane Strain Model 164
Figure 6-46 Idealized Solder Joint Geometry 165
Figure 6-47 FE Element Meshing for Rectangular and Detailed Solder Geometry 165
Figure 6-48 Classes of Optimization Models Based on
Two Different Fatigue Analysis Models 167
Figure 6-49 Iteration History of Design Variables for Scenario 1 169
Figure 6-50 Maximized Nf Values for Different Solder Materials 170
Figure 6-51 Integration of Analysis and Optimization 172
Figure 6-52 Initial and Optimized Design of eom4 Model for Rectangular Solder Joint .. 175
Figure 6-53 Left :Height vs Iteration History Right: Fatigue Life vs Iteration History 175
Figure 6-54 Initial and Optimized Design of eom7 Model for Detailed Solder Joint 178
Figure 6-55 Left: Fatigue Life vs Iteration History
Right: h3, l3 and l4 vs. Iteration History 178
Figure 6-56 Optimization-Design-Analysis
Associativity in a Solder Joint Analysis Model 179
Figure 7-1 Classical Engineering Optimization Model 200
Figure 7-2 EOM Representation-Constraint Schematic Graph 201
Figure 8-1 Design-Analysis-Optimization Progression
Correlation with Cartesian View 207

SUMMARY

Designing a complex product requires a multidisciplinary approach from design to manufacturing. Engineers often face alternatives and trade-offs during the design process to achieve better products. Facilitating information flow in an engineering system among various disciplines has a significant effect throughout the design process. With increasing product complexity, tools that automatically search for an optimum design among a myriad of alternatives become increasingly necessary. The challenge is the management of diverse product types, analysis disciplines, and optimization methods and tools. The modeling process which transforms a detailed product design and analysis models into an optimization model is a non-trivial task requiring large amounts of information, engineering theory, and experienced-based heuristics, as well as the support of multi-fidelity optimization and analysis models. This research views engineering optimization as an information-intensive problem that requires engineering information solutions.

This thesis has focused on developing a new information representation of optimization models, termed ***Enhanced Optimization Model (EOM),*** during design progression. ***EOM*** *represents an information framework for an object-oriented design methodology for optimization model construction, enhancement, classification and solution.*

EOM utilizes a combination of constraint graph and object techniques to provide semantically rich mappings and increased flexibility among models, methods and tools for design, analysis and optimization.

The **EOM representation** consists of an information model structure and protocol together with modeling languages for creating EOM objects. Specifically, EOM is an information representation that partitions the optimization area to achieve more trackable, reusable, flexible and modifiable objects. Key distinctions are the explicit representation of associativity between an optimization model and its analysis and product models, as well as the ability to capture multiple fidelities of optimization models.

An **EOM prototype** has been developed and implemented in the Java programming language in conjunction with optimizers (Bolink, CONMIN etc.), analyzers (Ansys FE) and symbolic solvers (Mathematica). Structural analysis and electronic packaging test cases illustrate the different characteristics and help to evaluate the EOM representation with respect to the thesis objectives. Results show that the EOM enhancement capability enables designers to capture optimization model building information and modify the models easily.

CHAPTER 1

INTRODUCTION

Designing a complex product requires a multidisciplinary approach from design to manufacturing. Design engineers often face alternatives and trade-offs during the design process to achieve better products. Facilitating the information flow in an engineering system among various disciplines for alternative models, methods and tools has a significant effect throughout the design process. When complex products become even more complex, tools that automatically search for an optimum design among a myriad of alternatives become increasingly necessary.

The optimization problem is often treated as finding the minimum in a fixed design space. In practical cases, however, the landscape is anything but fixed. New constraints are identified; *analyses are modified*; and new technologies may become available during the design process. This is especially true when optimization is used for design, as the optimizer often tends to exploit errors or omissions in the analyses or problem formulation. Most work on optimization ignores the *evolution of design complexity*, dealing instead with the problem of minimizing a specified function with respect to a specified set of design parameters. Yet, the most compelling examples of complex system design, such as those provided by nature, are the result of a design process that includes *varying complexity* as a fundamental characteristic. Recent work suggests that

methods which permit this change in design complexity can be more efficient and can lead to surprising results. Additional work in this area is needed [Kroo, 1997].

The main aim of this research is to develop an object oriented paradigm for optimization models to facilitate the design process. **This thesis presents research that seeks to manage the design complexity in the context of optimization model construction, enhancement, classification and solution by utilizing engineering information techniques.** The product of this research is a methodology, called *Enhanced Optimization Model (EOM)* to develop a knowledge representation for optimization models. Particular emphasis is given to optimization model building with product and analysis specific information, optimization model enhancement and automated optimization capability. EOM has been developed to handle a diversity of analysis models, optimization models, methods and tools in a variety of products. EOM is represented in such a way that the solution of the optimization problem can be obtained by using existing computational tools which are integrated into an information framework.

Chapter 2 describes the background and related research about design optimization and engineering information management. In Chapter 3, the identification of the gaps is addressed together with the desired characteristics of the Enhanced Optimization Model (EOM). Chapter 4 describes the EOM Representation, which consists of an information model structure and protocol, and modeling languages for creating EOM objects.

Implementation of EOM concepts is presented in Chapter 5. EOM Tool prototype has been developed and implemented in Java in conjunction with representative optimizers (Bolink, CONMIN etc.), analyzers (Ansys FE) and symbolic solvers (Mathematica). Test cases are illustrated in Chapter 6. For that purpose, three applications associated with (1) aircraft flap link, (2) PWA solder joint fatigue life and (3) ten-bar truss optimization have been used to illustrate the different characteristics of EOM. Chapter 7 summarizes an evaluation of the EOM representation with respect to the thesis objectives stated in Chapter 3. Chapter 8 describes the contributions of this research, recommended future research and the conclusion.

1.1 Information Integration for Optimization

Optimization-Analysis-Design integration requires information intensive mapping among design, analysis and optimization models. In this thesis one of the emphases is establishing a formal associativity between optimization entities and analysis and design entities. Design to/from analysis integration has been extensively studied in the last decade. Here, the emphasis is the associativity with respect to optimization.

The data required for optimization spans multiple design and analysis information sources to establish an optimization model that can be performed to obtain desired results. As the product requires multiple interactions, management of the data (searching, retrieving, linking etc.) becomes challenging.

Optimization of most of the complicated products requires CAD-CAE tools. For instance, the solder fatigue life analysis of a PWA can be done using FE analysis tools using information from design and analysis data. Optimization of solder fatigue life involves the transformation and interaction of design and analysis data into the optimization model. Another example is the optimization of a mechanical component; this requires information about FE analysis, the geometry of the component and the properties of the material of which it is made. Integrating the information from multiple sources requires a significant amount of engineering knowledge, which needs to be captured explicitly if the data transformations among optimization, analysis and design are to be automated. Additional control parameters may be required to perform optimization. Meanwhile, additional data may be needed to perform analysis and design.

Association links of EOM Representations basically perform mappings among optimization, analysis and design objects. Multi-directional information flow is needed to transform information from one end to the other whenever necessary.

1.2 Enhancing Optimization Models during Design Progression

Depending on the stage and the requirements, various fidelities of engineering models can be used to evaluate the product being developed. Figure 1-1 shows a flap link example that has been used in the prototype development. Enhanced optimization models are developed in conjunction with various fidelities of analysis and design models during the design progression. Flap link optimization models are enhanced by adding or

subtracting objectives, constraints and design variables as seen in Figure 1-1. Optimization models are constructed by defining the optimization parameters and associating them with the analysis and design data.

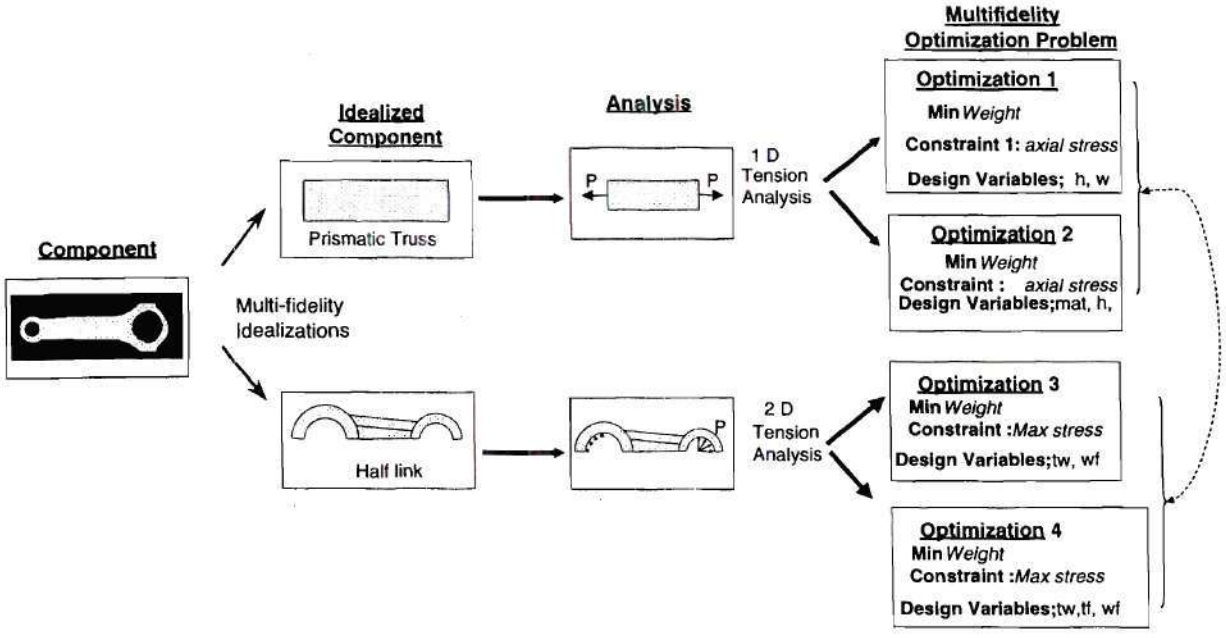


Figure 1-1 Enhancement of Analysis- Optimization Integration

1.2.1 Multifidelity Product Structure:

In some cases, much of the product data representing the final design is not used in optimization models. Often, optimization and analysis models do not need all the detailed geometry and material information. Therefore, the product is simplified. The same design model may be used at a different level of abstraction based on the analysis and optimization model fidelity. For example, Figure 1-1 illustrates two views of the flap

link; one view provides the I-beam link and the other provides a more detailed half of the flap link to be used in the optimization and analysis.

1.2.2 Multi behavior Analysis:

A *variety* of analyses and their responses are utilized for a variety of optimization models. Each mode requires an analysis application to predict the effects and responses of the system. For a given mode, an analysis may be performed using several solution methods (e.g., formula-based, finite-element analysis, etc.) at multiple levels of fidelity. The information requirements vary from one analysis and solution tool to another. The choice of a particular combination of analysis models and solution methods will depend on the level of accuracy desired and the computer resources available.

This chapter has introduced the problem area. Specifically, EOM representation is developed as an information representation by focusing on the optimization aspect to partition the optimization area and to make it more trackable, reusable and remodifiable. The key distinctions of this research are the explicit representation of associativity links of optimization domain to analysis and product models, as well as the enhancement ability to capture optimization model building information, to modify the models easily, and to provide flexibility to designer. The next chapter introduces the background, literature survey and ongoing research in the integration between optimization and analysis-design area.

CHAPTER 2

BACKGROUND AND RELATED RESEARCH

This chapter gives a brief overview of optimization and analysis. Before explaining the thesis work, a literature review was conducted on object oriented approaches to analysis-optimization to determine the status of related research in this area.

2.1 Background: Optimization and Analysis

Analysis can be used to find the response of a given design during the loading conditions. Conversely, design optimization performs in the reverse direction, determining the design that achieves the desired performance as shown in Figure 2-1.

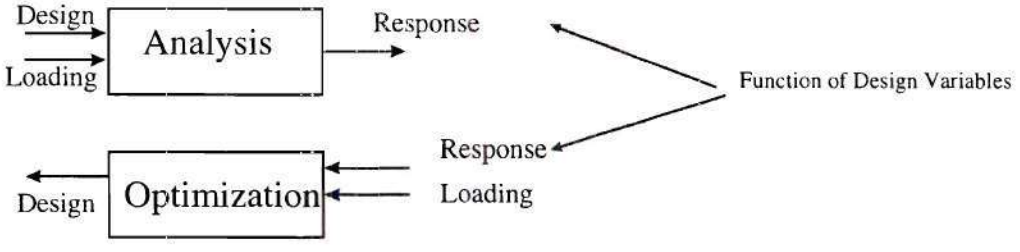


Figure 2-1 Optimization and Analysis

A change in design may impact several types of performance (behaviors). Everything needs to be considered at the same time during the optimization. For instance, modifying

the height of a fin in a heat sink affects heat transfer, cost, manufacturing disciples and so forth.

The engineering design process is also iterative as seen in Figure 2-2 with a heat sink parameter design example [Cimtalay and Fulton, 1994].

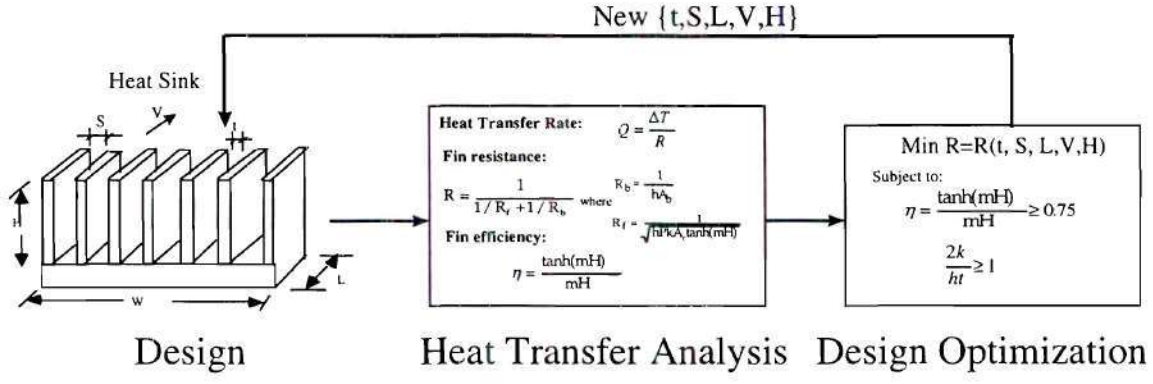


Figure 2-2 Design - Analysis - Optimization Process

It demonstrates how design, analysis and optimization fit together. After the design is created, analysis is performed to check performance and response, and how it conforms to design requirements. If the response and performance of the system is inadequate or unsuccessful, designers modify the design and perform reanalysis. The design process is time consuming and costly. Design optimization performs the same modification, better and faster, in an automated fashion. If analysis models and tools are easily obtainable, design optimization can usually be applied to those disciplines/cases easily. Those tools (analyzers) are used in some cases as black-boxes in design optimization. For instance, FE (Finite Element) analysis is a well-developed method and many tools are available in

engineering practice. FE analysis is considered as a black box and its results are used in optimization model [Esping, 1995].

Within the design concept, there are variables which specify the dimensions, proportions and other details of a product or system to be designed. Optimization consists in the selection of these values in such a way that the design will be the one that satisfies all the limitations and restrictions placed on it and is best in terms of cost, weight or other performance criteria (figures of merit). Some of the other definitions of design optimization are given in [Haftka, 1990], [Bloebaum, 1991] and [Wilde, 1978].

During engineering design optimization, design variables are selected. Then objective function and constraint functions are formulated in terms of the design variables. Finally, minimization /maximization of the objective function subject to the constraints yields successive solutions. The whole process is referred to as design optimization. The design variables are the parameters which describe the system that should be optimized. During the design optimization, improvements to those variables are computed by optimizers in such a way that improvements to the objective function values are achieved while satisfying the restrictions imposed on the system. Optimization problems can be formulated mathematically as an explicit or implicit function of the design variables. Some upper or lower limits are applied on functions which are known as constraint functions. Limits applied to the design variables are known as side constraints. If all the design variables satisfy all the constraints, then the design is in the feasible region, if one

does not then it is in the infeasible region. The design variables can be determined by an optimization procedure. For instance, in structural optimization, it is common to minimize the weight of the structure subject to stress and/or displacement restrictions [Kamat, 1993].

The general single optimization problem can be formulated in the following mathematical form.

Maximize/ Minimize

The objective function, measure of the system performance

$f(x, p)$

x, Vector of independent design variables

p, Vector of constant design parameters

Subject to

System constraints for the solution to be feasible

$h_i(x,p) = 0 \quad i=1,\ldots\ldots, k$

$g_i(x,p) \leq 0 \quad i=n+1,\ldots\ldots, l$

Side constraints

Lower and upper bounds on the system

$x_{low} \leq x \leq x_{up} \quad j= 1,\ldots\ldots, n$

Besides the above fundamental mathematical description, there are many different ways to describe the mathematical model. Classification of those mathematical models is based on abstraction and usage of the engineering problem to be optimized. Some examples of those mathematical models include a multi-objective approach (goal programming) and multidisciplinary optimization [Balling et al., 1997].

As a design gets more complicated depending on the stage and detail and, where specific and limited scope is not adequate, a transition from design optimization to multidisciplinary optimization is necessary.

2.2 Design and Analysis Integration Review

[Bernett et al., 1997] briefly categorizes the integration of analysis and design into a single unified disciplinary design process for vehicles. The authors categorize the focus of the papers they review as:

1) How to employ, coordinate and control existing analysis and design techniques in disciplinary design environments.
2) How to store, maintain and share analysis and design related data for different disciplines.
3) How to coordinate and control design decisions affecting more than one discipline.

This paper is the about general framework from the design stage to the manufacture stage for cars. It mainly contains high level information without going into the details during the design progress. It lacks any optimization aspect of the research.

A group of people, in collaboration with the NASA Langley Multi Disciplinary Optimization group, is developing a general distributed computing system named FIDO (Framework for Interdisciplinary Design Optimization). This system is a programming environment for executing multidisciplinary computations on networked heterogeneous

clusters of workstations and parallel computers. This work is still in progress [http://fmadwww.larc.nasa.gov/mdob/]. The framework enables analysis and optimization tools to perform optimization and was originally developed as a system that could effectively distribute time-consuming analyses among various platforms including super-computers. A High Speed Civil Transport design has been adopted from FIDO and is implemented in IMAGE [Hale, 1996]. This research is focused on the development of an open computing infrastructure that facilitates Decision Based Design, Theory of Languages, Information Science, and Integration Technology. An architecture for exploring design called DREAMS (Developing Robust Engineering Analysis Models and Specifications) has been developed which supports the activities of both meta-design and actual design execution. In addition, a software infrastructure that embodies these concepts was developed and is called IMAGE (Intelligent Multidisciplinary Aircraft Generation Environment).

Product Model-Based Analysis Models (PBAMs) represent engineering analysis models and include linkages to product model design information [Peak and Fulton, 1993]. They have been presented as highly automated analysis modules for designer usage. Via solder joint fatigue models, it has been shown that explicit design-analysis associativity linkages enable seamless interfaces to solution tools (e.g., finite element analysis systems) [Peak et al., 1995]. PBAMs perform routine analysis by linking detailed design models with analysis models of varying complexity and application. The user selects a PBAM from the catalog. Then, required product and analysis

entities are connected to the selected PBAM. The creation, execution and interaction of submodels within the analysis are done automatically.

[Cimtalay et al., 1996] added an initial optimization capability around PBAMs creating two alternative optimization models and solving them based on an Engelmaier fatigue life analysis model obtained from a PBAM catalog. An optimization agent becomes the "user" of the PBAM in the preceding scenario and employs the PBAM to obtain analysis results. The optimization agent automatically determines PBAM inputs and judges PBAM results. EOM is an extension of this early work.

2.3 Analysis and Optimization Integration

This section reviews the research on optimization analysis and design integration.

Analysis-Optimization Integration (Object Oriented Approach) Review

The work of [Eldred et al., 1995] is based on creating an object oriented software using C++ for application of optimization techniques to broad classes of engineering applications. Their work, defined as object oriented software design with C++, has been adopted as a tool to establish the protocol for wrapping parameter optimization around computationally-intensive simulations. They applied optimization in heat transfer and nonlinear solid mechanics. The basic goal of this paper was to utilize a broad class of computational methods by providing them with a general optimization capability. The paper illustrates how to interface different optimization and analysis tools in an OO

environment. It is categorized in the optimization-integration area of the research. Its main aim is to wrap the software tools, interfacing in a heterogeneous environment with different machines and tools. But, it lacks the higher level information content in analysis and optimization model creation from a product specific point of view and lacks the ability to enhance of those models or support alternative models. In other words, it is static for the given model situation and doesn't support the progress in design or detailing of the analysis and optimization models.

Another paper describes Integrated Intelligent Optimization (IIO) Software System, which has been developed [Cha J. et al., 1992] and is based on techniques of intelligence engineering. The authors' goal was to establish an environment for automatic processing on optimization models on the prototype level. The system they created (IIO) consists of a meta system, a toolbox and seven subsystems (preprocess, algorithm selection, parameter settings, process monitor, error analysis, sensitivity analysis and counter graphics). Their toolbox contains software tools shared by subsystems, such as IE (Intelligence Engineering) and KB (Knowledge Base Manager). In this paper, there are many ideas, tools and techniques on an abstract level. It is not obvious how the expert system performs the management capability or how the system components fit together and are implemented. It does not support the model enhancement capability for design progress.

The paper written by [Fujita et al., 1992] uses planar link mechanism optimization with an object oriented capability. In this paper, an optimal shape design system for planar link mechanisms is developed using an object-oriented modeling technique. In this system, link mechanisms are represented using an object-oriented modeling technique based on their graph representation. The formulation for optimal design is performed automatically. The system has effective user-interface functions for setting initial shapes of mechanisms and showing results of optimization calculations. The techniques used in their system object oriented representation; symbolic programming techniques and an algebraic programming language are applied to generate an automated optimization model. The focus of this paper was preliminary the optimal shape design of planar link mechanisms and the automation of the tools to achieve optimum design of the mechanism.

[Kodiyaman et al., 1994] uses optimization with a finite element code and an object-oriented approach for satellite structure applications. They found that OO provides powerful mechanisms for advanced mechanics of satellite structures. They developed a software system for optimization system called SAT-OP in General Electric Co. to address the optimization of satellite structural subsystems. It discusses a software system being developed which is based on a combination of finite-element analysis using NASTRAN, numerical optimization, approximation concepts, and object-oriented technology. Two examples were illustrated: a honeycomb panel with doublers and satellite structural subsystems. The examples are limited to single level optimization of

subassemblies within the satellite structural subsystems. Also, it does not support various disciplinary systems although they refer to it as a recommended work.

The CWP Object-Oriented Optimization Library (COOOL) consists of a collection of C++ classes implementing a variety of optimization algorithms. COOOL was developed at the Center for Wave Phenomena, Colorado School of Mines. A set of optimization tools in the library helps the user select proper optimization algorithms for a given model. The motivation was to collect optimization routines in a library, establish a programming environment for rapid development of new algorithms and establish a new test bed. This work is mainly focused on geophysics. It is primarily a collection of optimization algorithms, and it does not support product specific information or model enhancements during the design process.

[Tiller et al., 1996] used the ideas of OO design and divided the optimization problem into three distinct entities: the simulator (analyzer), the optimizer and the problem specification. They interact through a formal protocol. The main purpose of this paper was to utilize analysis information and sensitivity analysis and numerical optimization flavored with OO concepts.

2.4 GIT Research at the Engineering Information Systems Laboratory: Design-Analysis Integration

This section overviews the design-analysis_optimization integration research activity at the Engineering Information Systems Laboratory (EIS Lab) at the School of Mechanical Engineering of the Georgia Institute of Technology.

The research in EIS Lab has been mainly focused on the design-analysis integration area. EIS Lab has been participating in several industrial projects on design-analysis integration. This subsection overviews some of these research efforts that contribute to leverage the development of this thesis, which is the first major effort in the lab towards optimization integration.

2.4.1 The Multi-Representation Architecture (MRA)

[Peak et al., 1993; 1998] developed the *multi-representation architecture* (MRA, Figure 2-3), a design-analysis integration strategy that views CAD/CAE integration as an information-intensive mapping between design models and analysis models. Peak argues that the gap between design and analysis models is too large for a single general integration bridge; Peak therefore divides the MRA into four information representations that act as stepping stones between the design and analysis tool extremes. These four information representations are: solution method models (SMMs), analysis building blocks (ABBs), product models (APMs), and product model-based analysis models (CBAMs).

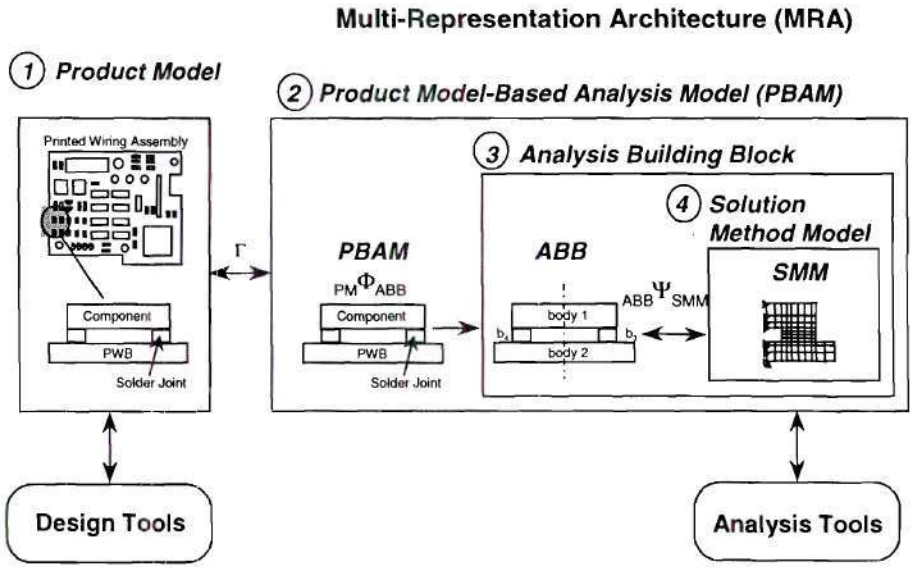


Figure 2-3 Multi-Representation Architecture for Design-Analysis Integration

On the right extreme of the MRA (Figure 2-3) are *solution method models* (SMMs) representing analysis models in relatively low-level, solution-specific form. SMMs combine solution tool inputs, outputs and control into a single information entity (an object) to facilitate automated solution tools access and results retrieval. SMMs are object-oriented wrappers around solution tools (e.g., FEA systems) that utilize an agent-based framework to obtain analysis result in a highly automated manner. *Analysis building blocks* (ABBs) represent engineering analysis concepts in a manner that is largely independent of product application and solution method. ABBs obtain results by generating SMMs through transformations (labeled ${}_{ABB}\Phi_{SMM}$) that are based on solution method considerations. *Product Models* (PMs, on the left extreme) represent detailed, design-oriented product information. A PM is considered the master description of a product which supplies information to other life cycle tasks, including engineering

analysis and manufacturing. To enable usage by potentially many analysis applications, PMs in the MRA go beyond their traditional role and support idealizations that relate detailed, design-oriented attributes with simplified, analysis-oriented attributes. Finally, *product model-based analysis models* (PBAMs) contain linkages (labeled ${}_{PM}\Phi_{ABB}$) that represent design-analysis associativity between PMs and ABBs. These associativity linkages indicate the usage of idealizations for a particular analysis application. PBAMs can be used to create catalogs of ready-to-use analysis modules for applications such as solder joint deformation and fatigue, PWB warpage, and plated-through holes [Peak 1993; Peak and Fulton 1993b; Peak, Scholand et al., 1996].

From the MRA viewpoint, providing solutions to the design-analysis integration problem involves defining these four representations (SMMs, ABBs, PMs and PBAMs) and two inter-representation mappings (${}_{ABB}\Psi_{SMM}$ and ${}_{PM}\Phi_{ABB}$). The MRA achieves flexibility by supporting different solution tools and design tools, and by accommodating analysis models of diverse discipline, complexity and solution method.

[Cimtalay, 1996] introduces an optimization technique closely integrated with the MRA. In this technique, modular software entities called *optimization agents* use the analysis results obtained by PBAMs for design optimization, by plugging them into the objective and/or constraint functions of its internal optimization model and obtaining new design variables needed to reduce the objective function. These new design variables are fed back into the product model and the process is repeated until the objective function value converges. This technique enables a closed-loop process that improves designs by

meeting some selected criteria and constraints. The designer can choose the proper optimization agent based on the complexity of the analysis, types of models and tool availability. The paper provides more details on how optimization agents are integrated with PBAMs and product models.

2.4.2 Team Integrated Electronic Response (TIGER) Project

The goal of the TIGER project was to demonstrate a collaborative design and manufacturing scenario in which a small manufacturing enterprise (SME) exchanged design information with the prime contractor early in the design process, thus reducing the iterations necessary to produce a successful design. For this purpose, a suite of design, manufacturing, and communications tools integrated across the Internet was made available to the product development team. The domain demonstrated was the design, fabrication, and assembly of printed wiring boards (PWBs) and printed wiring assemblies (PWAs).

In the TIGER scenario, a PWA designer generated PWA/B design information and sent it to a PWB manufacturer in STEP AP210 format. When the PWB manufacturer received this file, he uploaded it to an Internet-based engineering service bureau over the Internet [Scholand et al., 1997] that provided a variety of design and analysis services including design-for-manufacturability (DFM) and thermomechanical analysis. These services were integrated in an analysis environment developed for TIGER called *DaiTools-PWA/B* [Peak et al., 1997].

2.4.3 Product Simulation Integration (PSI) Structures Project

The concepts and techniques demonstrated in TIGER are being extended and applied towards the Product Simulation Integration (PSI) Structures Project. The PSI project is a multi-team, multi-year project conducted by The Boeing Commercial Airplane Group in Seattle, Washington. The objective of the PSI project is to define and enhance the processes, methods and tools to integrate structural product simulation and analysis with structural product definition (Prather and Amador 1997). This includes automated engineering analysis as an integral component of the product definition. The EIS Lab team has been contributing to this effort since September of 1997 with the application of its MRA/APM techniques in the airframe structures, extending beyond the electronics domain explored in earlier work. Specifically, it is providing design-analysis associativity techniques that are crucial to the true simulation integration the project wants to achieve [Peak, Fulton et al. 1999].

The Product Simulation Integration (PSI) Structures project is under way in Boeing Commercial Aircraft Group (BCAG) to reduce costs and cycle time in the design, analysis, and support of commercial airplanes. The objective of the PSI project is to define and enhance the processes, methods, and tools to integrate structural product simulation with structural product definition. This includes automated engineering analysis as an integral component of the product definition. Subprojects have been defined and working selected topics toward accomplishing the objectives of the PSI for BCAG Structures. Formalized

integration activities have also been identified to support the PSI subprojects through their technology life cycle. [Prather & Amador, 1997]

As part of PSI, Georgia Tech has contributed an information modeling language, termed constrained objects (COBs), that is aimed at next-generation stress analysis tools. COBs combine object and constraint graph techniques to represent engineering concepts in a flexible, modular manner. COBs form the basis of the extended multi-representation architecture (MRA) for analysis integration, which is targeted at environments with high diversity in parts, analyses, and tools [Peak *et al.* 1998]. A key MRA distinctive is the support for explicit design-analysis associativity (for automation and knowledge capture) and multidirectional relations (for both design sizing and design checking). Another MRA characteristic is to use COBs to represent and manage complex constraint networks that naturally underlie engineering design analysis.

Using a case study approach, lug and fitting design guides have been recast as example reusable COB libraries. The use of these and other COBs on structural parts relevant to the aerospace industry has been demonstrated. These case studies utilize *XaiTools*, a toolkit implementation of MRA concepts, which interfaces representative design tools (CATIA CAD, materials and fasteners libraries) and general purpose analysis tools (Mathematica solver, ANSYS FEA).

It is anticipated that COBs and the MRA will contribute key technologies to the overall PSI next-generation analysis tool architecture. The potential impact of explicit design-analysis

associativity is significant. Capturing such knowledge, which is largely lost today, enables libraries of highly automated analysis modules and provides a precise reusable record of idealization decisions. User adaptation/creation of existing/new analysis templates is also possible. Today creating views of analysis results such as internal analysis documentation (strength check notes) and regulatory agency summaries typically requires extensive manual effort. While COBs focus on core associativity and analysis computation relations, their combination with technology like XML should enable interactive "pullable views" to help streamline this analysis task. Other COB applications are anticipated, including upstream analysis-associativity.

2.5 Gaps in Literature

In light of the literature survey of the preceding sections, the following main items stand out as requiring additional attention. Those items are more extensivley explained and broken down in the next chapter.

- *Lack of an optimization modeling representation tailored to design-analysis integration*

  There is a need for a general information representation that addresses the special needs of optimization to design-analysis integration. That representation should be easily defined. Also, it should capture the information in an optimization model with associations to the product and analysis specific information.

- *Lack of an optimization modeling representation that capture multifidelity optimization capability*

The representation should be easily defined, and enhanced. It should provide a means of creating multiple fidelity optimization, analysis, and design models. The engineering optimization should be semantically rich and, where merits, constraints and design variables defined in terms of product and analysis specific terms, as well as being flexible to enhance the model during the design progression.

CHAPTER 3

PROBLEM DEFINITION

This chapter explains some of the needs in the current optimization technology to be addressed in this thesis. Identifying these needs also helps to construct the main structure of problem definition. The main focuses for the problem definition are graphically shown in Figure 3-1 with question marks adjacent to the needs.

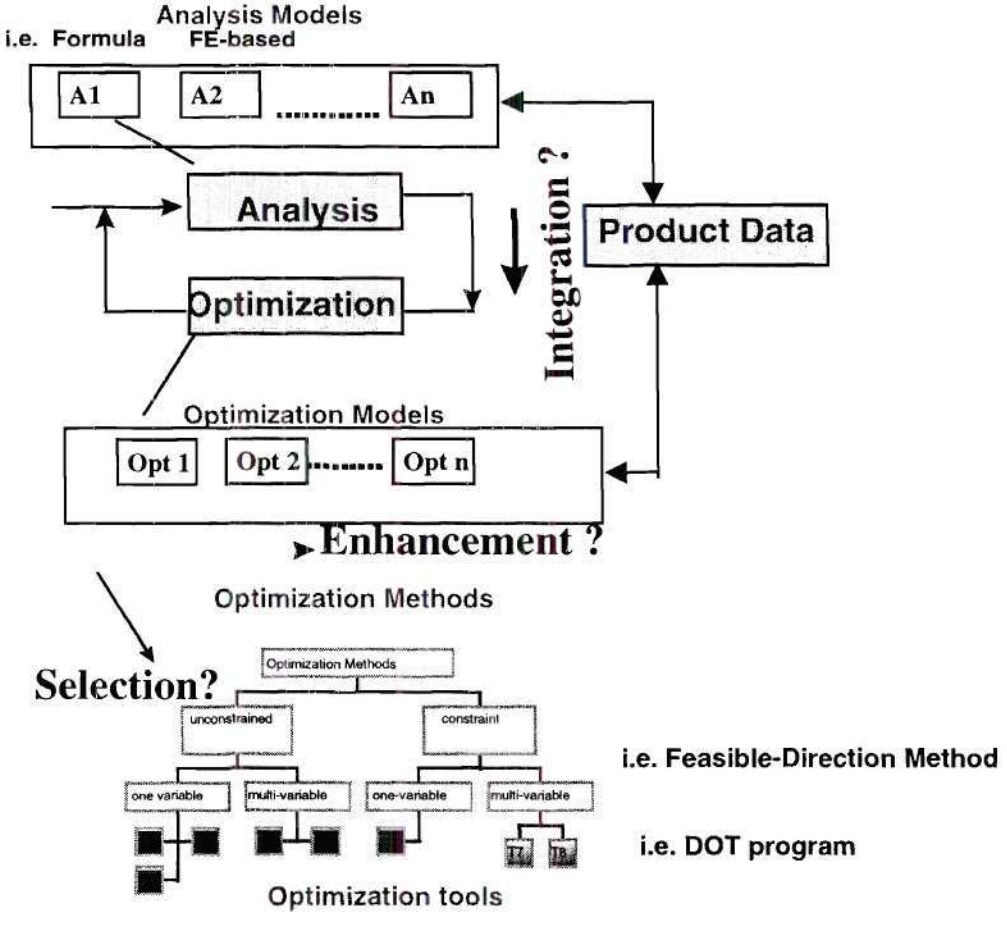


Figure 3-1 Framework for Problem Definition

The main entities used in the Figure 3.1 are analysis, optimization and product data models. Different fidelity models of all those main entities are implemented during the different stages of a design process.

The following subsections emphasize the specific needs and articulate them through illustrations.

3.1 Integration among Optimization, Analysis and Design Representations

As a design progresses, it requires more information for its analysis and optimization models to cope with the evolution of design complexity. Engineers, depending on the stage of the design, use models of various degrees of detail and alternatives for their models and solutions. For instance, a designer can choose an analysis module (A1) in Figure 3-1, say a formula-based Engelmaier Model, from Product Based Analysis Models (PBAMs) [Peak and Fulton, 93] for solder joint fatigue life and construct an optimization model (Opt1). Optimization Model (Opt1) includes the objective and constraints, based on information from the analysis model (A1), and other design models such as CAD and product data. ***There is a need for an effective integration strategy to share and pass information between optimization and analysis representations that can support using different tools, computer environments and locations, emphasizing and linking product specific information.*** The optimization model requires multiple sources of information and human interaction (decision making). Selection of design variables, objectives and constraint functions requires broad knowledge and different sources such

as product data (CAD), analysis results and design specifications. There is a need for building a bridge from the information sources to the optimization model.

The example in Figure 3-2 illustrates the information requirement for an optimization model. The optimization model entities (objective, constraints and design variables) require information from product and analysis entities. For instance, the objective is to maximize the fatigue life. This requires information about shear strain that is determined in the analysis system, which in turn receives information from product data. Another optimization entity is design variable, pwb material type, which requires link to product data.

A knowledge representation is required to capture optimization models in a computable form at a higher level of context than currently exists. For instance, a typical optimization model has little, if any, explicit information about how its attributes (Objectives, constraints and design variables) are derived from the product and analysis information.

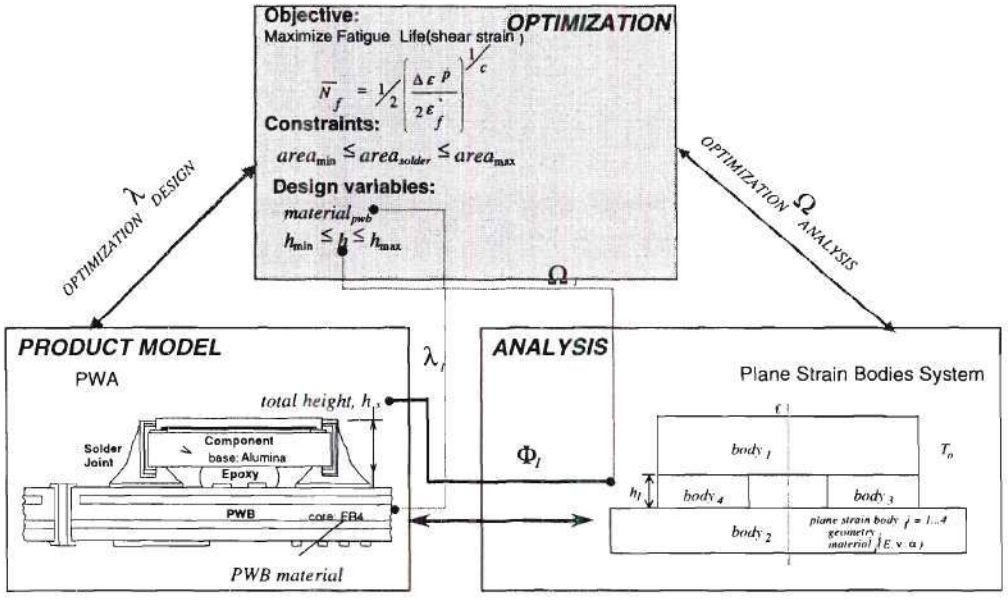
The representation should support information associations:

(1) between optimization and product models

(2) between optimization and analysis models

(3) between analysis models and product models

The focus of this objective with respect to this thesis is to have the optimization model representation capture the associtivity among the optimization, analysis and product entities. The optimization data must be linked with the analysis and product models to receive input from them and to provide outputs to them when it is required.



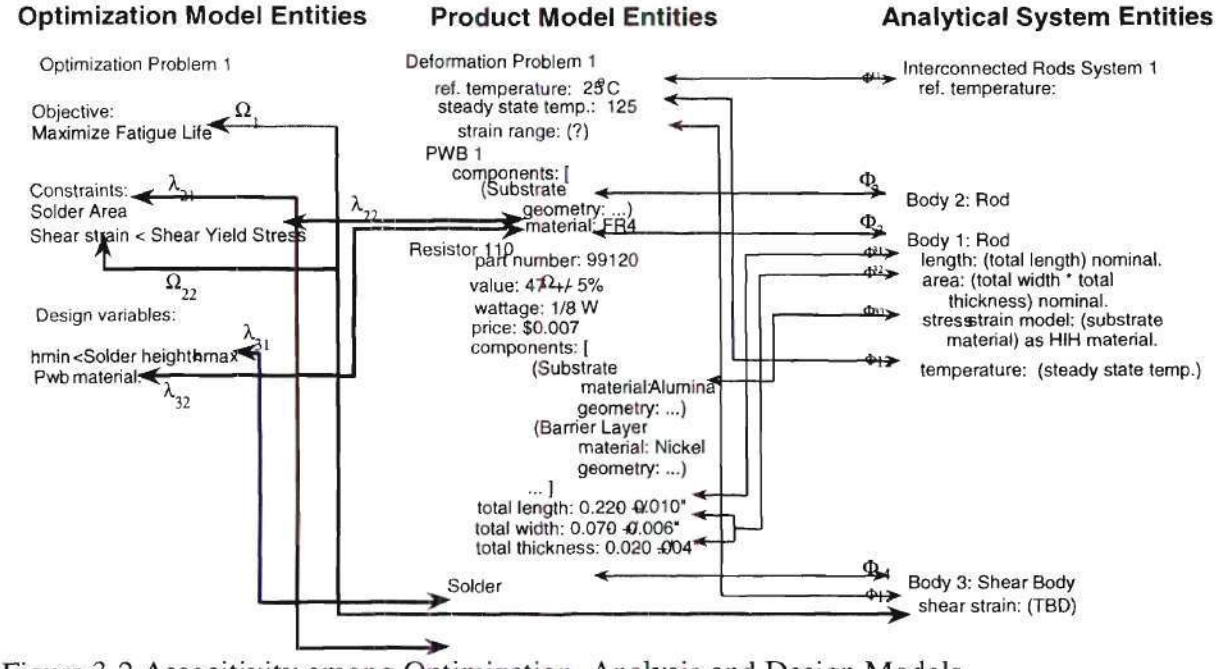


Figure 3-2 Associtivity among Optimization, Analysis and Design Models

3.2 Building and Enhancement of Optimization Models

As another objective, optimization and analysis models of varying complexity levels should be supported by the optimization model representation. Depending on the design progression, a simple analysis model with a formula-based solution (Analysis Model A1) or a more detailed analysis model of varying fidelity (Analysis Model A1) might be used, which requires a finite element solution (Figure 3-1 and Figure 3-3).

For the chosen analysis model, say A1 in Figure 3-3, the designer may construct alternative optimization models that will help in the decision process. The reasons behind creating different optimization models may include modification of requirements and merit functions, answering what-if questions, capturing design progress (e.g. availability of higher fidelity design details) and adapting new optimization technology. Thus, ***there is a need to enhance, link and extend from one optimization model (Opt_i) to another (Opt_j) during the construction of alternative optimization models based on the same analysis, A1.*** For instance, the Engelmaier model is represented as a CBAM in a catalog of solder joint analysis modules. Based on this same analysis model, two alternative optimization models can be created and solved [Cimtalay et al., 1996]. The objective is to maximize the solder joint fatigue life for the electronic product in both models. In the optimization model Opt1, fatigue life is a function of solder joint height. And, in model Opt2, it is both a function of solder height and PWB material (See left of Figure 3-3).

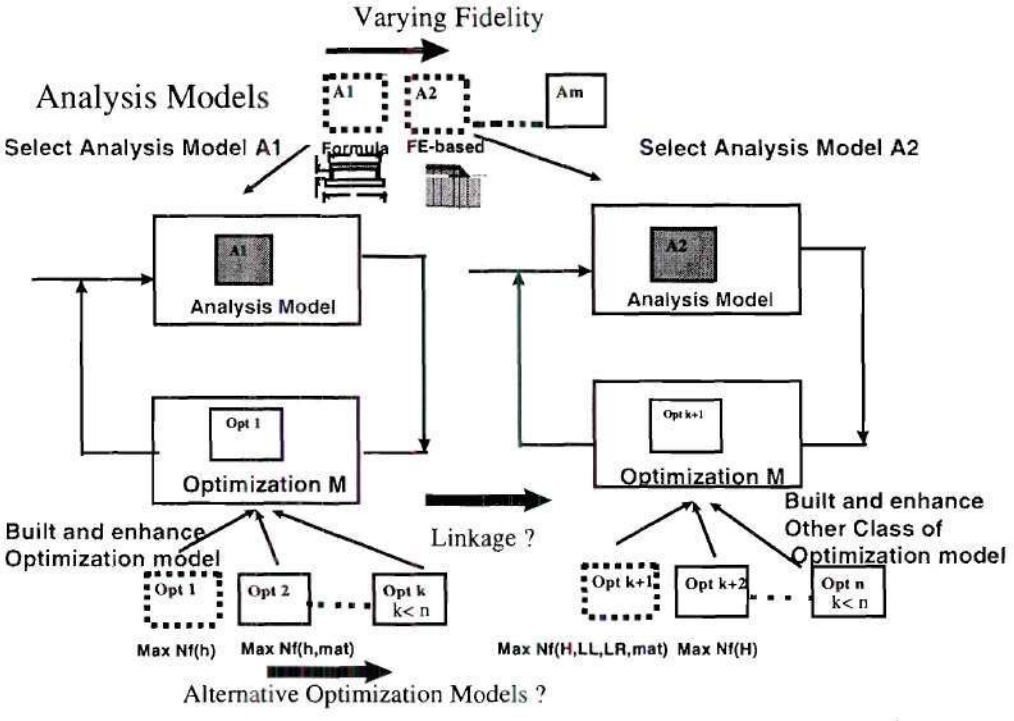


Figure 3-3 Construction of Alternative Optimization Models and Support Classes of Analysis Models

3.3 Linkage among Classes of Optimization and Analysis Models

Third, if the designer chooses a more complex analysis module (A2) from the class of analysis modules instead of (A1), how will the existing optimization model and its alternatives be affected? (Figure 3-1). ***Hence, there is a need to link alternative optimization models from one analysis model (A_i) to another (A_j) to create and adapt new optimization models.*** The reason may be to enhance the models during the design process. For instance, the designer may use the finite element based plain strain model instead of the closed form Engelmaier model (See right side in Figure 3-3). Then the

question arises, how will the original model be affected? How will analysis and optimization integration be affected? Can we establish a linkage from the earlier case?

3.4 Automated Optimization Method and Tool Selection.

In addition to the above needs, each optimization model constructed requires proper solution methods and related tools to obtain optimization results for further evaluation in decision making. Some of those optimization methods have been categorized in [Vanderplaats, 1984] and [More et al., 1994]. The optimization problem may be solved using the techniques ranging from theoretical closed form solutions to numerical computer based methods. Selection of the proper optimization tool depends on the mathematical content of optimization model. ***There is a need to find an automated strategy to select a proper optimization method and tool based on the character of optimization models to be solved.*** Ideally an algorithm can automatically choose an optimization solution tool (Ti) from the optimization tool library that is, compatible with optimization model (Opt_i). For instance, in solder joint fatigue life optimization (Opt1) (Figure 3-3), solder joint was a function of height (a continuous variable), which makes it a candidate for a gradient -based method. On the other hand, in Opt 2, height and PWB material are the variables and this model has a mixture of continuous and discrete design space. This model can be a candidate for a mix-integer solution class. This objective is to alleviate the human engineer from performing tedious tasks without risking errors.

3.5 Support Enhancement of Optimization Models for Disciplinary Systems

Furthermore, some products require various disciplinary analyses. If the designer wants to substitute different analysis models or enhance the models, then how will the optimization models be affected? Changing disciplines and the analysis model in the system requires enhancement of the optimization model downstream. Figure 3-4 shows a multidisciplinary case for a PWA system which requires heat sink analysis, solder joint fatigue analysis, and warpage analysis. During the design progress, the designer may want to add new disciplines, such as PTH analysis. New analyses bring new information to the system, including new variables and constraints. Alternately, the designer may want to exchange the analysis models with higher fidelity ones for the specific disciplines (i.e. instead of using a 1D Engelmaier model, they may exchange it with a 2D plane strain model while keeping the other discipline analyses the same). Construction and enhancement of the optimization models are necessary. ***There is a need to support the enhancement of optimization models for various disciplinary systems.***

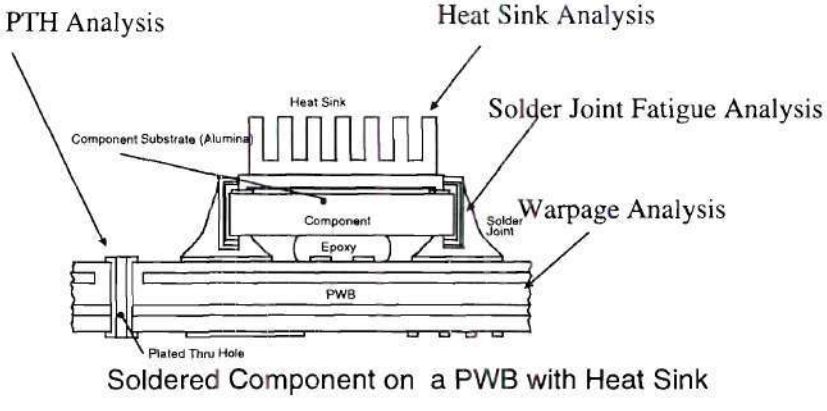


Figure 3-4 A Disciplinary Design Example

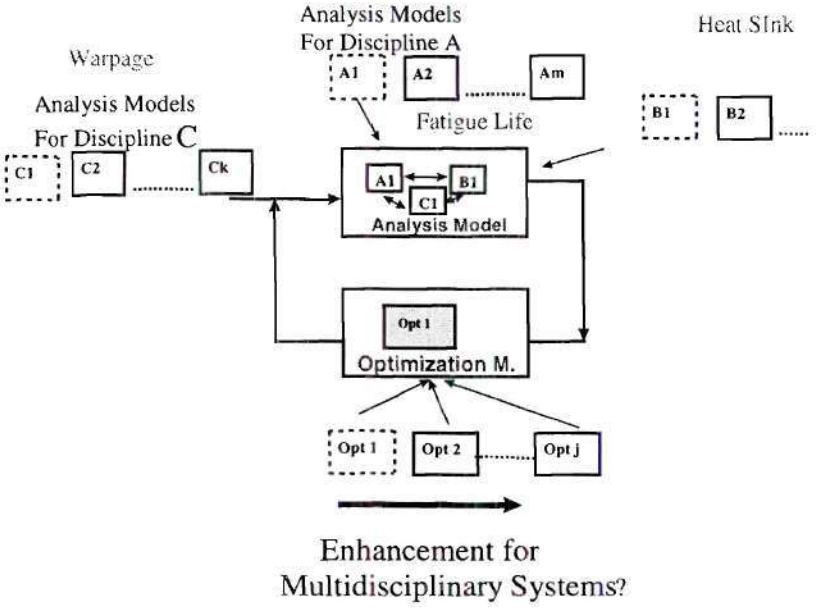


Figure 3-5 Disciplinary Model Enhancement

3.6 Summary of the Gaps

We can summarize the gaps identified above briefly as:

- Integration between analysis and optimization representations, emphasizing product specific linkages
- Building and enhancement of optimization models
- Linkage among classes of optimization and analysis models
- Automated optimization method and tool selection
- Support enhancement of optimization models for various product and disciplines.

In the previous sections, the primary gaps have been explained. Next, the objectives of this thesis will be described based on these needs.

3.7 Objectives of Thesis

This section defines the capabilities that an information representation should have to address the needs described in previous sections. The following items are the desired characteristics of the representation described in the next chapter. Facilitating these capabilities into one representation has been major objective of this research, as summerized here:

The objective of this thesis is to develop an information representation named ***Enhanced Optimization Model (EOM)*** *that facilitates creating multiple fidelity optimization, analysis, and design models emphasizing optimization association with product and analysis specific information. The main goal is to facilitate optimization-design-analysis integration considering modularity. EOM is used for capturing the information in an optimization model with association linkage to the product and analysis specific information. The term "enhanced" in EOM means an engineering optimization model which is semantically rich and; where merits, constraints and design variables are defined in terms of product and analysis specific terms, as well as flexible to enhance the model during the design progression.*

- **Thesis Objective 1:** ***Capture associativity between an optimization model and its design and analysis models***

A general optimization representation should address the needs of optimization-analysis-design-integration. This representation needs mechanisms for defining modeling objects and the relationships between their attributes, while at the same time capturing the

semantics specific to optimization-analysis-design entities. This representation must have a formal structure composed of an information model structure and protocol, along with languages (lexical form and graphs) for modeling specific problems. Constructs of the representation should be introduced with its rules. Optimization models based on this representation should be easily defined, constructed and modified.

The representation should enable optimization models to retrieve the data they need from design and analysis representations instead of considering all the entities combined as a whole in the optimization model. Data with its semantics and values retrieved from analysis and design representations can be utilized in the optimization model. Thus, engineers should understand the representation of an optimization model with their ingredients. Common analysis and design data could be potentially shared among several optimization models instead of replicated in the codes of each individual application.

- **Thesis Objective 2:** ***Provide modeling language for defining optimization models***

The representation should support modeling languages that allow engineers to easily create, modify or extend optimization models without requiring extensive effort. Related with the representation, the languages have to capture the main entities of optimization models. They must be independent of the product domain and computer applications used.

- **Thesis Objective 3:** ***Provide the explicit representation of data integration knowledge***

Since the optimization model requires the retrieval of design and analysis information scattered across several repositories and combined for optimization purposes, a formal and domain independent mechanism for defining the transformations will provide an explicit description of the data integration process. Generally, these transformations are not explicitly defined anywhere today. Instead, they are implicitly placed inside the code of the optimization and analysis applications, making it difficult to reuse or modify them, as well as to understand them.

- **Thesis Objective 4**: ***Provide a way to partition among optimization, product and analysis models***

Today optimization models are mixed with analysis and product models. As a result, there is not a clear distinction among the core attributes and relations that belong to the optimization model, the product, and those that correspond to the analysis model. The importance of this partition is to allocate the information to its main domains to increase clarity, reusability and portability (where the information is generated and where to get it and where to transform it). The intrinsic product attributes and relations are independent of the environmental conditions to which the product is subjected, whereas analysis attributes characterize the behavior of the product under specific environmental conditions and provide the response of the system under these conditions. Optimization attributes include what is the merit function to be maximized or minimized, what kind of technical or managerial constraints are applied to the system, and what are the design variables that can be manipulated to achieve the merit function.

- **Thesis Objective 5:** ***Support for building and enhancing multi-fidelity optimization models***

This will provide modularity to multi-fidelity optimization models tailored with of varying complexity of analysis and design models. Modularity of knowledge representations has been defined as "the ability to add, modify, or delete individual data structures more of less independently of the remainder of the database, that is, with clearly circumscribed effects of what the system knows" [Barr and Feigenbaum, 1981] and [Peak, 1993]. An engineer can easily build and modify optimization model instances with varying complexity. This is one of the major concerns in current approaches to integrate optimization-design-analysis. The users generally have little means for adding or changing optimization models and retrieving data without a major adaptation of their analysis and product data.

- **Thesis Objective 6:** ***Support automated selection of optimization methods and tools***

Based on optimization information representation, it should be possible to select the proper optimization methods and tools for a particular problem. It is also possible to define the mathematical structure of the optimization model by utilizing an optimization information model to capture the necessary data. Thus, an automated selection algorithm based on the mathematical structure should be provided, which will then enable finding the optimum solution.

- **Thesis Objective 7:** ***Support optimization models from various engineering disciplines***

Most of the products require numerous analysis disciplines to simulate and check different physical behaviors. Those responses are reflected in the optimization models. The representation should support the disciplinary approach to optimization where models from different disciplines are involved.

- **Thesis Objective 8:** ***Provide capability to extend and utilize the Multi-Representation Architecture (MRA)***

EOM Representation should be compatible with the MRA approach [Peak, 1995]. The EOM Representation should utilize the MRA by interacting with the product information and analysis information required by EOM. This will add the capability of extending MRA to the optimization domain, meanwhile facilitating design representation and analysis representation already available. The EOM Representation should provide a mechanism for interaction CBAM and APM models. This mechanism also supports the recursive nature of optimization tools during the data interaction. Thus, transformation of the information will be managed in efficient manner.

The next chapter describes EOM representation in a formal and generic manner with its definitions and its components.

CHAPTER 4

ENHANCED OPTIMIZATION MODEL (EOM) REPRESENTATION

This chapter introduces a formal presentation of the **Enhanced Optimization Model (EOM)** Information Representation and its main components. The chapter begins by describing the identification and partition of optimization-design-analysis entities into subgroups in the engineering domain (Section 4.1). This section provides a layout for an information framework through an exploration of optimization, analysis and design area by identifying the main entities and the interaction of those pieces among each other in the engineering context. Next, Section 4.2 overviews EOM Representation with its main components: EOM Representation Information Model, EOM Description Languages; lexical and graphical views and EOM Protocol; these are formally presented in the remaining sections of the chapter. Section 4.2.1 introduces the EOM Information Representation Model, which contains the definitions and constructs of the EOM Representation. Section 4.2.2 introduces the description languages that are lexical and graphical forms. Lexical languages are the information modeling languages used to define EOMs and EOM instances respectively (Section 4.2.2). Graphical languages are information graphical structures that represent EOM concepts (EOM EXPRESS-G Diagrams, EOM Constraint Schematics Diagrams, and EOM Specialized Objects) (Section 4.2.3). Finally, Section 4.3 describes a group of EOM information access operations known as the EOM Protocol.

A prototype implementation of the concepts introduced in this chapter is presented in the next Chapter 5. Several test cases validating these concepts are presented in Chapter 6.

4.1 Partition Of Engineering Optimization Domain

Figure 4-1 shows the whole system partitioned into the main entities that are associated with each other. The main rectangles that have subgroups in the Figure 4-1 represent ***optimization, analysis*** and ***design*** related entities. The main entities are subdivided further into subgroups to capture information content, interactions and transformations among design, analysis and optimization entities. The main focus of this thesis is on the optimization block and the interaction of this block with design and analysis entities as shown by the shaded area with red arrows in the Figure 4-1. The arrows represent the associativity links to the product model that represents the design entities and to the context based analysis model that represents the analysis entities. From the EOM point of view, providing and managing the integration problem described above involves information representation of optimization models. This is achieved by partitioning the optimization area into entities and keeping interaction among those entities in an information framework. This also includes establishing the interaction among those entities inside the optimization domain as well as the related entities in design and analysis domain. Thus, partitioning provides a structured way during optimization model building to easily track, reuse, and modify them; and also provides the identification of

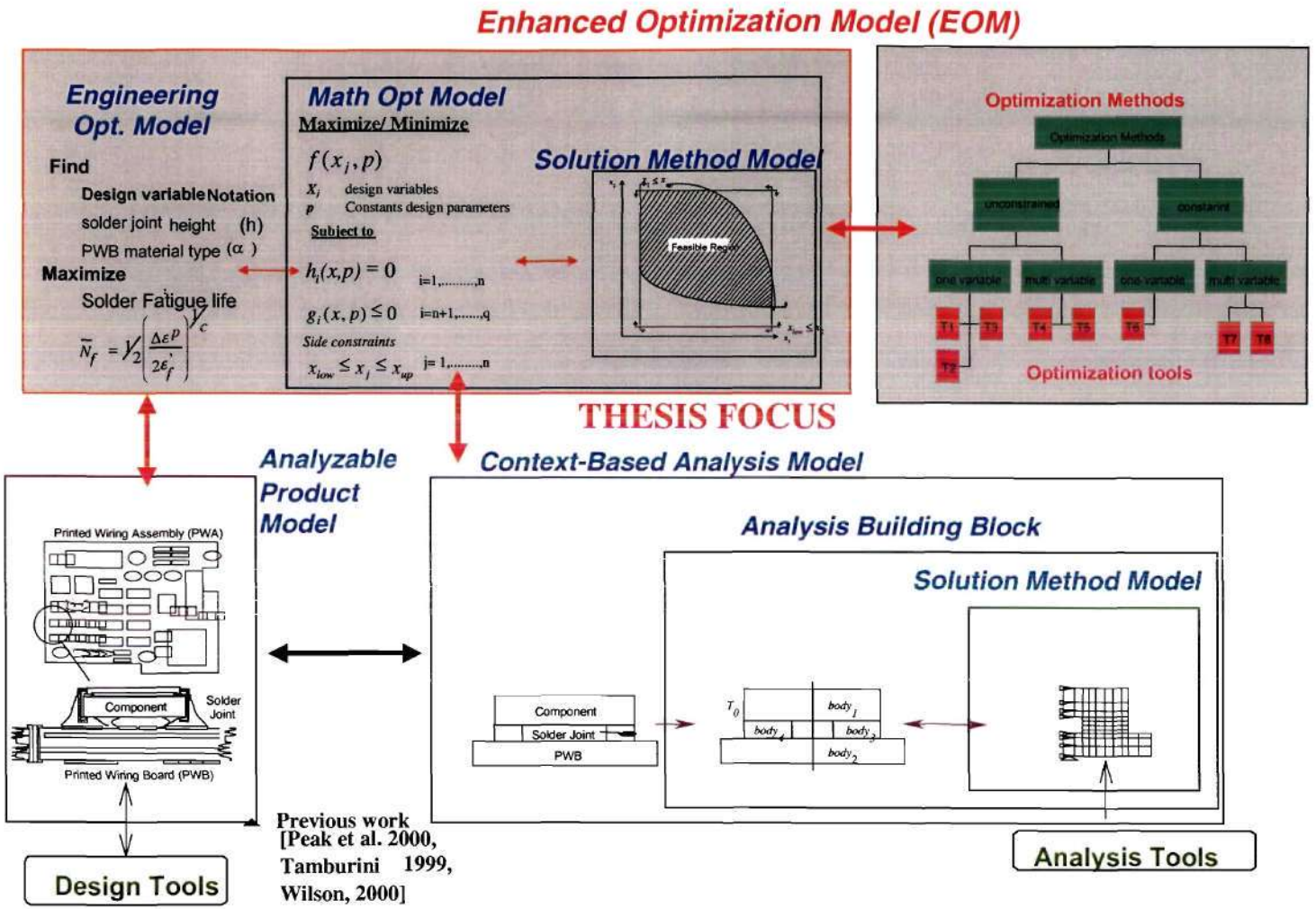


Figure 4-1 Partition of Engineering Entities

Page missing from thesis

Optimization Solution Method Model (OSMM):

OSMM is relatively low level, solution method specific form. OSMM's tools input, output and control information form a single information entity to facilitate results.

Analysis Domain Partition

Context Based Analysis Models (CBAMs):

CBAMs are developed to capture the associativity with the context of the analysis. It answers what, why and which circumstances the analysis is performing. The analysis context object includes a **link** to product data, **condition** objects, such as loads, boundary conditions, **behavior mode,** such as static structural analysis and **design responses,** such as stress calculation.

Analysis Building Blocks (ABBs):

ABBs represent analysis concepts as reusable, modular, and adaptable objects that are product and tool independent.

Solution Method Models (SMMs):

SMMs combine analysis solution tool input, output and control into a single information entity to facilitate automated solution tools access and results retrieval.

Design Domain Partition

Analyzable Product Models (APMs):

APMs representation defines formal, generic computer-interpretable constructs to create and manipulate analysis oriented views of engineering parts or products [Tamburini, 1999].

Those engineering entities described above are related to each other in a higher level as shown below.

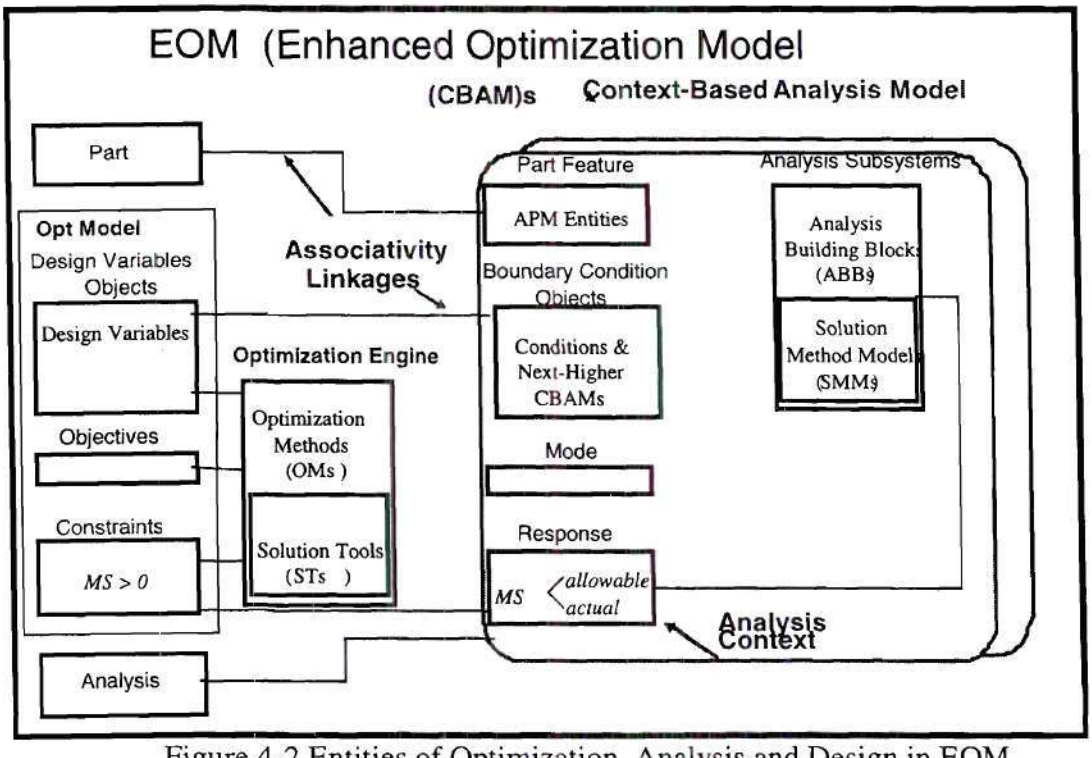


Figure 4-2 Entities of Optimization, Analysis and Design in EOM

4.2 Enhanced Optimization Model (EOM) Information Structure

This section formally introduces the main component of the **Enhanced Optimization Model (EOM)** representation based on constrained object technology [Wilson, 2000]. **EOM** represents an information framework for an object oriented design methodology. This methodology provides optimization model construction, enhancement, classification and solution.

EOM utilizes a combination of constraint graph and object techniques to provide highly semantic mappings by increasing flexibility among design, analysis and optimization tools. The EOM information representation is specifically tailored to optimization.

The main goal is to facilitate optimization-design-analysis integration considering modularity. EOM is used for capturing the information in the optimization model with association linkage to the product and analysis specific information. Enhanced optimization model (EOM) means an engineering optimization model which is semantically rich and; where merits, constraints and design variables are defined in terms of product and analysis specific terms, as well as flexible to enhance the model during the design progression. EOM contains the fundamental building blocks used to define optimization models.

EOM representation described with the following components; EOM representation information model, EOM information description languages and EOM protocol. The main components that constitute the EOM Representation are shown in Figure 4-3 below.

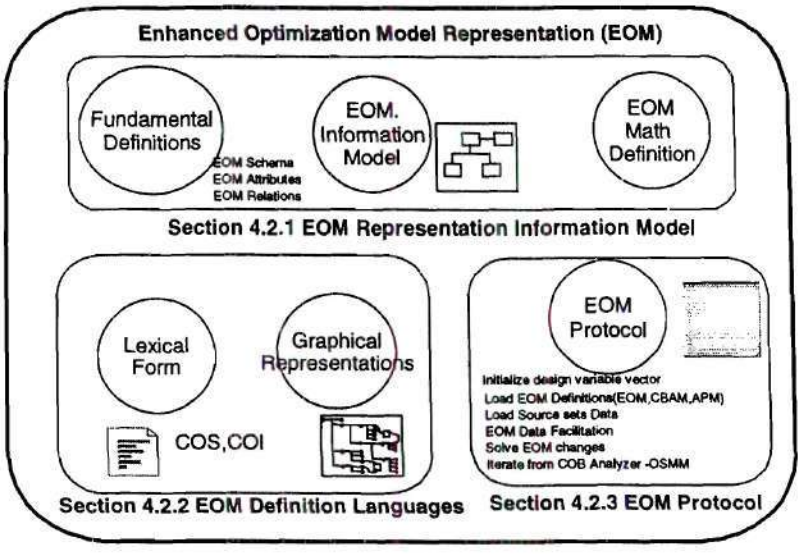


Figure 4-3 Main Components of EOM Representation

First, the representation information model is a meta-model describing the main architecture (constructs) of EOM representation. The information model provides a foundation for the rest of the EOM Representation components. The EOM information model is described by Express-G diagram followed by formal definitions of those pieces given. Finally, EOM was expressed in mathematical terms in set theory.

Second, the information languages are described in lexical and graphical forms. Lexical form is considered as the master view of the EOM, where EOM graphs can be generated from it. This form consists of enough language constructs to describe enhanced

optimization models (*eoms*) of varying fidelity with its association to product and analysis entities of a particular case. Lexical forms have a defined structure consisting of objects interacting with other objects and data instances of those objects. Graphical forms are the views that help easier comprehension of the domain, the concepts of object oriented and the constraint graphs. Graphical forms consist of Constraint Schematic, Express-G and EOM specialized objects. Below, Figure 4-4 gives the overview of the EOM Description languages with lexical and graphical forms. Section 4-2 describes them in more detail.

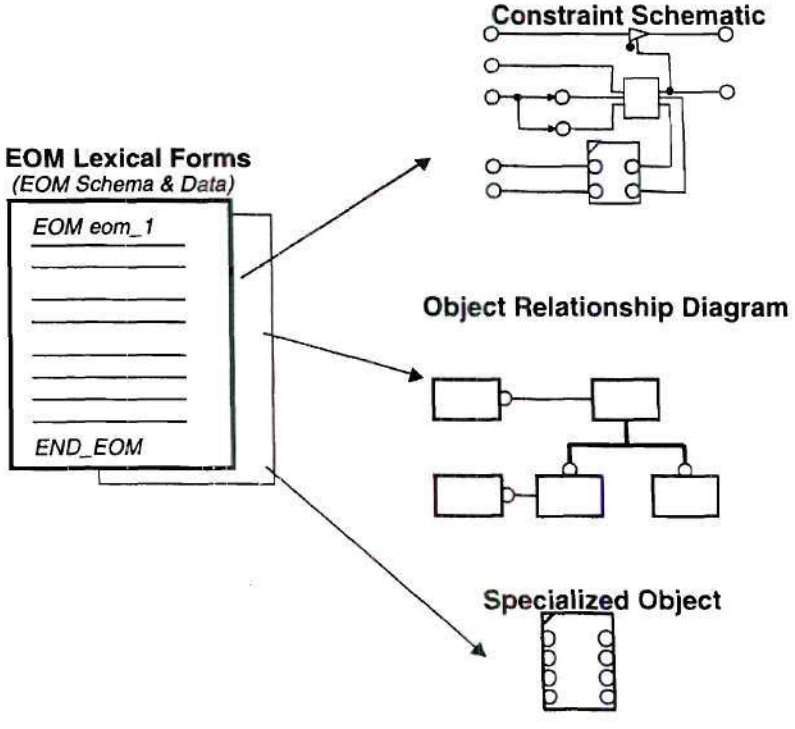


Figure 4-4 Overview of the EOM Description Languages

Finally, protocol as a component of EOM representation describes the implementation of those concepts.

The more detailed description of the EOM components are presented at the following Sections 4.2.1, 4.2.2 and 4.3.3.

4.2.1 EOM Representation Information Meta Model

4.2.1.1 Overview

Information techniques have been utilized to describe the EOM representation that compasses entities describing optimization domain and how to glue them together. These techniques form the information medium in which EOM is represented.

The main building blocks are the constraint objects (cobs) used for this information medium. It has a formal defined structure with its attributes and operations. The EOM representation combines constraint graph techniques and object-oriented principles to represent optimization concepts associated with analysis and design concepts. EOM attributes and relations are formally defined and in the description of the EOM representation. They intend to capture the essence and the functionality (association, enhancement, automation) of engineering optimization domain.

The fundamental defined constructs contained in the EOM Information Model provide development of the EOM Representation. Their definitions are expressed later in terms of set theory notation. Thus, they are independent from any particular data modeling or programming language.

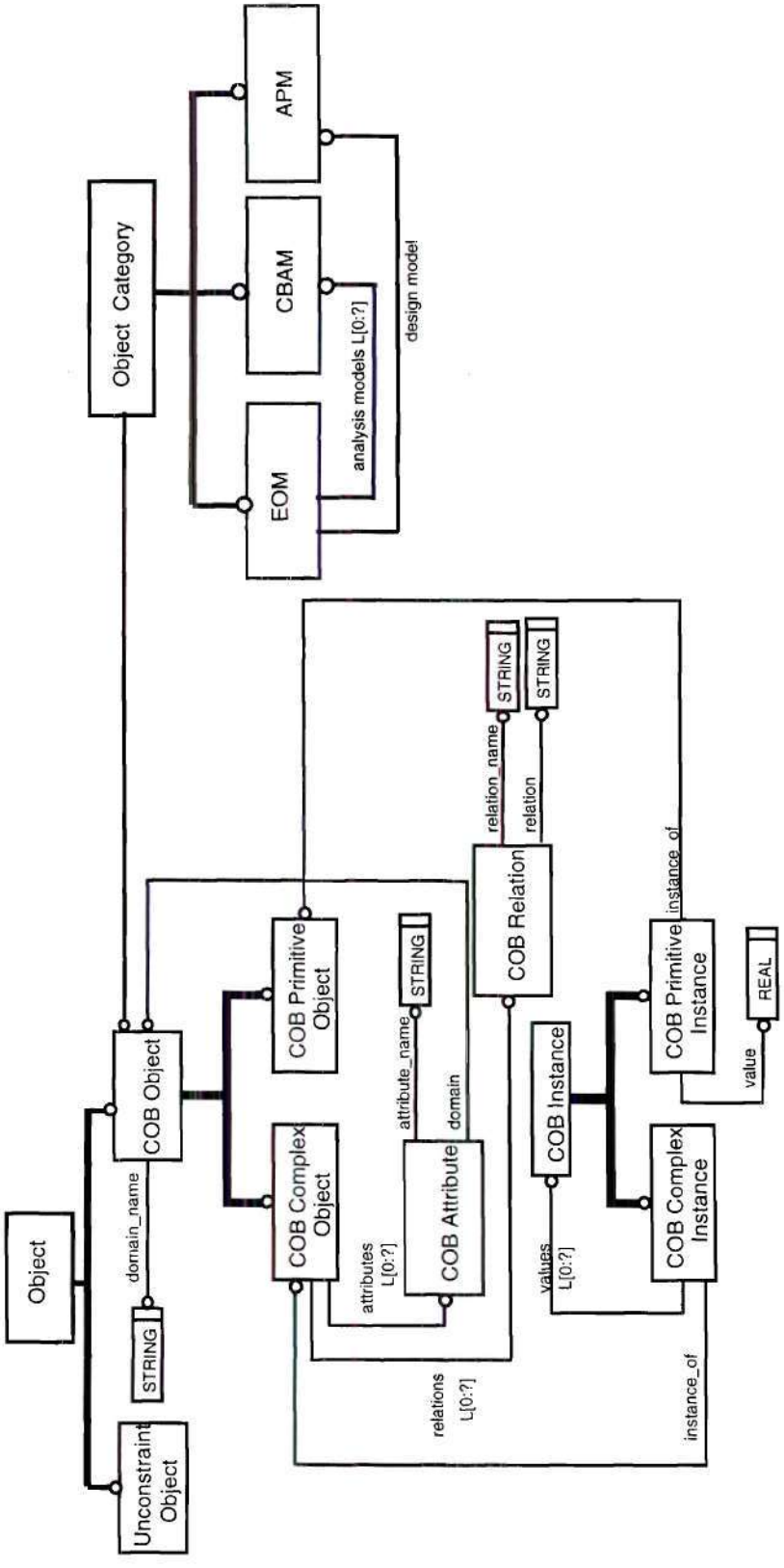


Figure 4-5 A View of the EOM Information Model

Figure 4-5 is a simplified view of the EOM Information Model showing its fundamental groups of constructs ***EOM Schema, EOM Name, EOM Attributes and EOM Relations***. The following description is based on that Figure 4-5 Information Model of EOM in EXPRESS-G (for more about Express-G see Section 4.2.2.2)

An Enhanced Optimization Model contains particularly the all EOM Object Sets that define an optimization-oriented view of a given product. It is possible to join a set of objects that belong to any category through association links. The EOM category contains optimization domain related objects and links. EOM associates optimization objects with CBAM analysis and APM design object sets in its information model and utilizes those objects in the any category. In other words, the EOM provides a starting point to all schema, attributes and relations that constitute an optimization model.

An object is the source origin of all objects with its identity, attributes and relations. Constraint object (cob) is an object such that its relation is defined in mathematical relations based on constraint graph theory. A constraint object is developed for the purposes of information modeling as applied to describing the information required to design, analysis and optimization integration at EISLAB.

The EOM uses the same structure of constraint object defined in the higher level information model (left side of the figure). The first level is the related constructs that create the structure of language. The second level is the instance level constructs associated with the structure of the upper level. Together, (The structure and Instance) they will define the complete set of constructs that make up a lexical language that can be

used for domain specific definitions of optimization, analysis and product data (See Figure 4-8 information model level 1 and 2-Section 4.2.2.1).

Specific Constructs define attribute and relation definitions specific to the discipline in this context here. For instance, an analysis context can be described with the following attributes: the part, the conditions of that part, the behavior type (mode), which it is tested against and the finally response attributes. It is very desirable to have context-based attributes and relations to a developer and user. In section 4.2.1.2, there will be formal definitions that makes up optimization related entities.

A Categorized-object is a set of all complex and primitive objects used for the generation of models in a specific domain area such as optimization; EOM, analysis; CBAM, design; APM. EOM, CBAM and APM are all subtypes of the object category. The EOM objects represent a set of optimization domain related entities, whereas the CBAM represents a set of analysis domain related entities and finally, the APM represents set of product related entities. EOM utilizes CBAMs and APMs. CBAM utilizes APMs in their model generation. For instance, the optimization category object (EOM) represents all complex and primitive constraint objects used in the description of a specific optimization model. It has its own specialized attributes and relations. EOM has optimization_model attributes, further classified as under attribute category of objective, constraints and design variables, as well as CBAMs and APMs. Similarly, the EOM has a number of relations under the relation category. They are optimization relations (er), associativity relations of optimization to design (ald), associativity relations of

optimization to analysis (ala) and associativity relations of optimization to optimization (alo).

4.2.1.2 Definitions

Above, the information structure of EOM has been explained briefly. The above objects are defined more precisely in this section.

EOM Schema:

> **Definition 4.1:** *EOM schema is a set of items that are grouped for the purpose of optimization-analysis-design integration.*

EOM Name:

> **Definition 4.2:** *EOM name is an ordinary name of specific eom schema.*

For instance, the optimization of the flap link of an airplane could be "flap_link_optimization1"

EOM Attributes:

EOM attributes may be grouped into the following categories that describe the main attributes of engineering optimization.

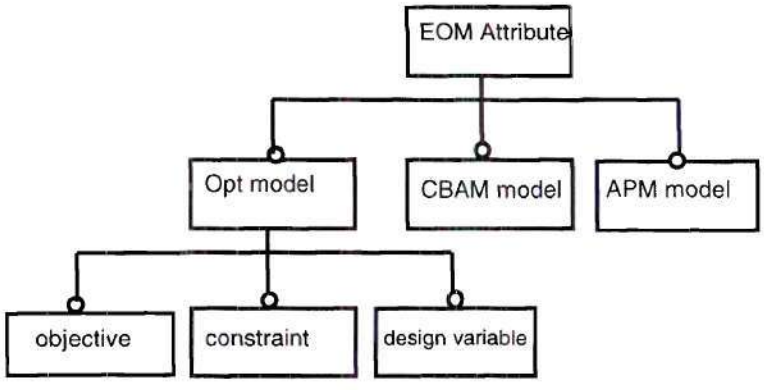


Figure 4-6 EOM Attribute Classification

Optimization Model Category (EOM) Attributes include, Objective Attributes, Constraint Attributes and Design variable attributes *as a part-of Optimization Model attributes,* as well as Analysis Categorization (CBAM) and Design Categorization (APM) attributes.

EOM Optimization Attributes are defined in the content of this thesis as follows:

> **Definition 4.3:** *An EOM attribute is an EOM attribute if it is part of the set of minimum and necessary attributes that optimize a product.*
>
> **Definition 4.3.1:** *An EOM optimization model attribute is an abstract EOM Attribute that has EOM objective, constraint and design variable attributes.*

More specifically, subtype of EOM *Optimization Model* Attributes are defined as follows:

> **Definition 4.3.1.1:** *An EOM Objective Attribute that belongs to the optimization model is an EOM Attribute which has to be maximized or minimized.*

An EOM Optimization objective can usually be defined as a merit to achieve. Examples of *Optimization Objective* attributes are a mathematical function, such as the weight of a truss, cost of a product and efficiency of an engine.

> **Definition 4.3.1.2:** *An EOM Constraint Attribute that is a part of optimization model is an EOM Attribute, which specifies the limits.*

An EOM *Optimization Constraint* can usually be expressed as a managerial or physical limit that is imposed on the system. Some examples of *Optimization* constraints are the stress margin of safety of a product under loading conditions, the limits on the dimensions of a product, and the specific number of materials that can be

used in the development of a product. They can all be transformed as mathematical inequalities or equalities and expressed in relations.

> **Definition 4.3.1.3:** *An EOM Design Variable Attribute that is a part of the optimization model is an EOM Attribute which defines the variable parameters where their value is allowed to change in their bounded limits in such a way that they achieve the merit and satisfy the all the constraints.*

An EOM Optimization design variable attribute can be described as the variable parameters that are chosen. Objective and Constraint attributes are explicit or implicit functions of those variables that formalize the mathematical optimization problem. Changing the value of design variables in their allowable limits through an optimization strategy facilitates finding the optimum results. Some examples of *Optimization design variables* are the height of a solder joint whose fatigue life is to be maximized, areas of truss structure where weight can be minimized and the velocity of the air flow through the fins of the heat sink on a chip where heat loss is to be maximized.

Attributes of the product and analysis are often needed for optimization because optimization models are expressed mathematically in terms of their attributes.

Briefly, we can introduce the analysis (cbam) or design (apm) related attributes that have been utilized in the EOM.

> **Definition 4.3.2:** *A CBAM attribute is part of the set of minimum and necessary attributes to analyze a product.*

The Analysis Constraint Object Category is represented as CBAM. The CBAM attributes include analysis context attributes with subclass attributes response, conditions,

mode and the product model and (APMs) attributes. For instance, stress analysis of a flap link includes context information. Features and parameters of the part are retrieved from the product model, static stress analysis is treated as behavior mode, loading and boundary conditions as the condition of the part exposed and the calculated stress treated as a response from the analysis [Prather, 1997] and [Wilson, 2000].

> **Definition 4.3.3:** *An APM attribute is a part of the set of minimum and necessary attributes to define an analyzable product.*

Design Constraint Object Category (APM)'s attributes include product and idealized attributes. Product attributes are the subset of EOM attributes that are associated directly with geometry and material information where as idealized attributes are the approximated attributes that are used in analysis and optimization. They are derived from product attributes. An example of idealized attributes is the "effective" length of a link. Idealized attributes are the result of simplifying or transforming product attributes using heuristic knowledge [Tamburini, 1999].

EOM Relations:

EOM Relations and *transformation/associativity* have been classified as shown in Figure 4-7.

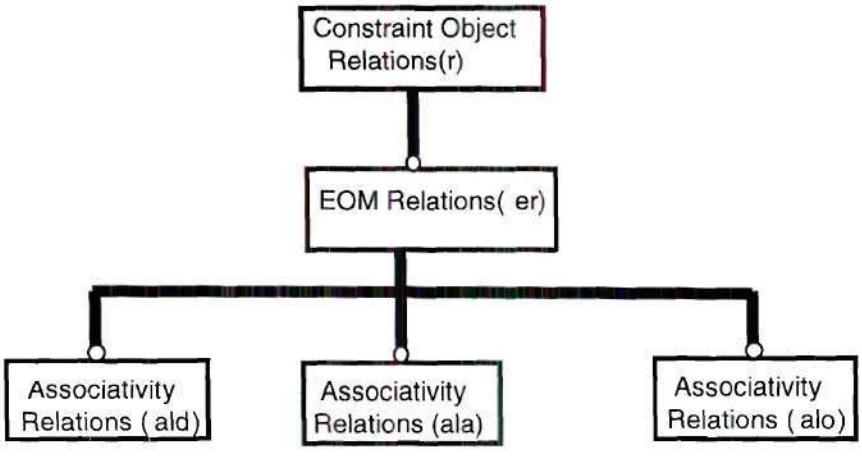


Figure 4-7 EOM Relation Classification

EOM Relations **(er)** are inherited from constraint object relations. This provides a mathematical operation in a bi-directional manner, and also allows black-box transformations. Associativity transformation relations are subtypes of constraint object relations linking objects with its semantics and its content. EOM association relations are sub-classified as *optimization-analysis transformation* **(ala),** *optimization-design transformation* **(ald)** and *optimization-optimization transformation* **(alo).**

EOM Optimization Relations are defined as follows:

> **Definition 4.4.1:** *An EOM relation (er) is a constraint object relation that represents mathematical or black-box operations relevant to the optimization domain.*

Relations are also a part of the definition of constraint objects. EOM relations are defined as mathematical equations or functions about optimization entities such that relations are based on constraint graph theory. EOM Relations are a part of the definition of constraint object. The following definitions are given to describe and formalize the linkages of the main partition entities.

One of the main features of the EOM representation is that it provides a structure for linking the design oriented product data and the analysis models into the optimization model via transformations called associativity. Association relations are also utilized between the entities of the optimization models.

> **Definition 4.4.1.1:** *An optimization-design transformation relation (ald) is an EOM relation that links optimization attributes and design attributes.*

> **Definition 4.4.1.2:** *An optimization-analysis transformation relation (ala) is an EOM relation that links optimization attributes and analysis attributes.*

> **Definition 4.4.1.3:** *An optimization-optimization transformation relation (alo) is an EOM relation that links optimization attributes to optimization attributes.*

Transformation relations can achieve the association defined in EOM through linking objects. Optimization-analysis transformation (ala) relates analysis information to optimization information along with the requirement of the optimization modeling. This also true for optimization-design transformation (ald) where it relates analysis information to optimization information. They are described as associativity relations that, respectively, transform analysis information and design information into

optimization information or vice-versa. Finally, it is desirable to have a semantic match between the two-optimization objects. An optimization-optimization transformation relation (alo) provides this kind of associativity.

All of EOM relations, relation (er), optimization-design transformation (ald), optimization-analysis transformation (ala) and optimization-optimization transformation (alo) are relations that have bi-directional forms. A bi-directional form is a determination of the terminal values of an input-output system that is possible from the opposite direction. For instance, a calculation of axial stress of a simple bar is a function of the axial force applied to the ends of the bar and the area of the bar. If someone knows the values of the force and the area of the bar, stress can be determined. The inverse is also possible. If a person knows the values of the stress and the force, the area can be determined. Bi-directional operation is also applicable for association relations. For instance, one design variable attribute, say ***diameter_out,*** in the optimization model can be associated with outer **diameter** of the sleeve attribute in product data, or vice versa. This concept of bi-directionality can be achieved through constraint-graph theory. A Constraint schematic is based on constraint graph theory (see brief explanation in Section 4.2.2.2). EOM relations, ***ers*** and transformation relations, ***alds, alas*** and ***alos*** can be represented in the constraint schematic of an EOM.

Based on the previous definitions of the EOM, the Mathematical Description of Enhanced Optimization Model is given below.

4.2.1.3 Mathematical Description of EOM

*An **EOM for an optimization model*** is defined as follows:

$\mathbf{eom}_i = (\ name\ ,\ \{\{\ obj_1 \ldots , obj_k\},\ \{g_1 \ldots , g_l\},\ \{\ x_1 , x_2 , \ldots , x_m\ \},\ \{\ cbam_1^t\ , cbam^t{}_2 ,\ldots ,\ cbam_n^t\ \},\ apm_k^p\ \}\},\ \{\{\ ala_1,\ldots .\ ,ala_p\},\{ald_1,\ \ldots,\ ald_r\},\},\{alo_1,\ \ldots,\ alo_s\},\{\{er_1,\ldots\ldots\ ,\ er_w\}\})$

Where:

$\mathbf{eom}_i$: is a specific EOM for optimization model i ;

name is the name of the eom;

attributes

$\{\ obj_1 \ldots , obj_k\}$ is the list of objective attributes of the EOM;

$\{\ g_1 \ldots \ldots g_l\}$ is the list of constraint attributes of the EOM;

$\{\ x_1 , x_2 ,\ \ , x_m\}$ is the list of design variable attributes of the EOM;

$\{\ cbam_1^t\ , cbam^t{}_2 ,\ldots ,\ cbam_n^t\ \}$ is the list of analysis categorization attributes of the EOM at fidelity level t;

apm_k^p is the product categorization attribute of product, p at design stage k;

relations

ala_p is the p^{th} relation for associativity linkages between optimization and analysis attributes;

ald_r is the r^{th} relation for associativity linkages between optimization and design attributes;

alo_s is the s^{th} relation for associativity linkages between optimization attributes;

er_w is the w^{th} enhanced optimization relation ;

Properties:

$name \in S$;

$obj_k \in$ ***EOM*** or ***COB***;; $g_l \in$ ***EOM*** or ***COB***;; $x_m \in$ ***EOM*** or ***COB***;;

$cbam_j \in$ ***EOM*** or ***COB***; $apm \in$ ***EOM*** or ***COB***;

Individual EOMs are grouped to form the ***Set of EOMs of a product p*** ($\boldsymbol{EOM^{(p)}}$) as follows:

$\boldsymbol{EOM^{(p)}} = \{ \mathbf{eom_1^{(p)}}, \mathbf{eom_2^{(p)}}, \ldots, \mathbf{eom_n^{(p)}} \}$

There may be more than one $\mathbf{eom_n^{(p)}}$ for a given product **p**. The reason for this is that, for modularity and efficiency purposes, it may be preferable to have several specialized **eom**s for the product to test different scenarios rather than a single, monolithic eom that is difficult to adapt to the model change required by progression of design.

Each $\mathbf{eom_i^{(p)}}$ could support a combination of different *sets* of analyses (cbams) with different fidelity and mode and different fidelity of product data (apm_k).

For example, ***eom***$_i$ could be defined for the optimization of an *airplane flap link* with its product data representation with $\mathbf{apm_1^{(p)}}$ at design **stage 1** or more information intensive product data representation $\mathbf{apm_2^{(p)}}$ at design **stage 2** and with its analysis objects, **cbam1** and **cbam2.** The first group of cbam1's are to support extensional analyses, and the second group cbam2's are to support buckling analyses. Even those analyses have different degree of complexity (fidelity). The determination of the number of EOMs that should be defined for a given product should be based on the how depth and breath are required by both the developer and user.

The objective is to keep EOMs small and simple, while at the same time sharing as much information as possible.

Since EOM's building blocks are constraint objects, the generic definition of a constraint object is given below in mathematical terms

A *Constrained* ***Object*** is defined as follows:

$cob_i = (object\_name , \{ a_{s+1} , a_2 , \dots , a_n \} , \{ r_{p+1} , r_2 , \dots , r_m \}, supertype\_object)$

Where:

$\boldsymbol{cob_i}$: is a specific Constraint Object;

$object\_name$ is the name of the object;

$\{ a_{s+1} , a_2 , \dots , a_n \}$ is the ordered list of local attributes of $\mathbf{cob_i}$;

s is the number of all attributes (local and inherited) of $supertype\_object$;

$\{ r_{p+1} , r_2 , \dots , r_m \}$ is the ordered list of local Relations of cob_i;

r_k is the k^{th} relation of cob_i;

p is the number of all relations (local and inherited) of $supertype\_object$;

$supertype\_object$ is the parent object of cob_i;.

Properties:

$object\_name \in \mathbb{S}$, where $\mathbb{S}$ represents the set of strings;

$a_j \in \mathcal{A}$, where $\mathcal{A}$ is the set of Attributes;

$r_k \in \mathcal{R}$, where $\mathcal{R}$ is the set of EOM Relations;

$supertype\_object \in COB$, where COB is defined below.

supertype_object provides the means for defining inheritance hierarchies between Objects. Following the object-oriented paradigm, a given Object cob_i inherits the attributes and relations of its parent *supertype_object*. Thus, in the definition above{ a_{s+1} , a_2 , ... , a_n } are called *local attributes*, whereas { a_1 , a_2 , ... , a_s } are called *inherited attributes*. Similarly, relations { r_{P+1} , r_2 , ... , r_m } are called *local relations*, whereas { r_1 , r_2 , ... , r_p } are called *inherited relations*. In this work, a given object can only have *one* parent object. In other words, only *simple inheritance* is allowed [Tamburini, 1999] and [Wilson, 2000].

After the information model description with formal definitions of the pieces and mathematical description of EOM were given, the description languages developed and utilized have been explained in the following section.

4.2.2 EOM Description Languages

The **EOM Lexical form** is the language for defining data structure and relations for a specific to optimization domain. A specific type of EOM, a template, is generated by a populating EOM meta-level structure. The application of this template to a specific design instance is achieved by instantiating this template. Table 4-1 shows tabular information views of the structure. Also, Table 4-1 summarizes the purpose of each view. These views emphasize different aspects of the information structure.

EOM Languages	
EOM Lexical Form	- Information template in lexical form for optimization domain - Defined by constraint objects(EXPRESS like information modeling language)
EOM Instance	- Defined by constraint object instances - Instances of structure defined in the EOM

EOM Graph Forms	
Engineering Optimization View	- It's a mathematical optimization template with its symbols representing engineering domain
Constraint Schematic	- Graphical view emphasizing relations among Attributes
Object Relationship Diagram	- Graphical view emphasizing *part-of* and *is-a* relations. Based on EXPRESS-G
Specialized Object	- Encapsulated view used in other constraint schematics of optimization domain
Subsystem Instance View	- Similarly annotated subsystem view
Graph Instance View	- Annotated constraint graph showing usage of COB instance

Table 4-1 EOM Languages

4.2.2.1 EOM Lexical Forms

The EOM is based on constraint object structure definition language and instance definition language such that both structure and instance definition languages form a complete, generic unit for the description of engineering related data and behavior. The purpose of EOM Structure is to define classes of EOM with its data structure and type of relations defined in Section 4-2. The EOM Instance Definition Language is associated with EOM structure. It contains the data instances of the defined objects of the optimization domain. Attributes and methods become meaningful to the optimization area by classifying them according to their content.

Below, a description of an EOM lexical form is expressed by STEP EXPRESS language (ISO 10303-11). EXPRESS is an object-oriented information model specification

language. It was developed to generate formal information models describing engineering products. It has been accepted as part of the STEP standard for product data exchange in the industry. This EXPRESS view also corresponds to some sections of EXPRESS_G diagram in Figure 4-8

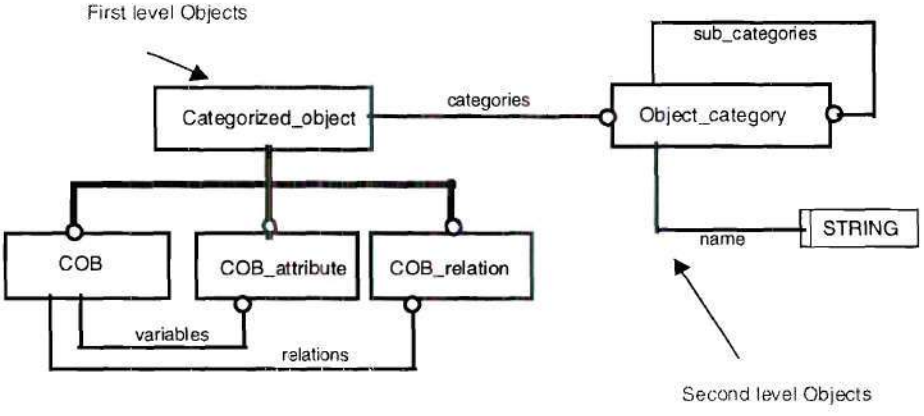


Figure 4-8 Simplified Information Model for EOM Language Generation (Express-G Diagram)

The first part of the EXPRESS below defines the second level objects where particular domain objects are created such as EOMs, CBAMs, APMs.

```
ENTITY categorized_object;
    categories : LIST OF [0, ?] object_category;
END_ENTITY;

ENTITY object_category;
    name : STRING;
    subcategories : LIST OF [0, ?] object_category;
END_ENTITY;
```

The second part of the EXPRESS below deals with the first level objects describing the constraint object. A detailed description of those objects, *cob_attribute, cob_relation, and constrained_object are all of type categorized_object*, has been given in [Wilson, 2000].

```
ENTITY constrained_object SUBTYPE OF categorized_object;
    attributes : LIST OF [0, ?] cob_attribute;
    relations : LIST OF [0, ?]  cob_relation;
END_ENTITY;

ENTITY constrained_attribute SUBTYPE OF categorized_object;
    uid : STRING;
    attribute : STRING;
    comment :  OPTIONAL STRING;
END_ENTITY;

ENTITY constrained_relation SUBTYPE OF categorized_object;
    uid : STRING;
    relation : STRING;
    comment :  OPTIONAL STRING;
END_ENTITY;
```

The Second level object called Object_category can be generated from the first level object by implementing ISO-10303-21. The Object_category, EOM, can be seen with its basic attribute and relation types below.

```
First class objects (in Express part 21 form)
ISO-10303-21;
HEADER;
FILE_SCHEMA (('FIRST_LEVEL_COBS_SCHEMA'));
ENDSEC;

DATA;

/* cob = categories, attributes, relations */
#10 =CONSTRAINED_OBJECT ( (#1000), (#200, #210, ...), (#300, #310, ...) );
#200 = COB_ATTRIBUTE.....
#300 = COB_RELATION ( (#1010), ' er1', '<objective>  ==(<o1>+<o2>)/2', ' This is an
optional comment for  er1');

/* These are standard instances */
# 1000 = OBJECT_CATEGORY  ('EOM', () );
# 1010 = OBJECT_CATEGORY  (' OPTIMIZATION_RELATIONS', () );
.....
```

The formal, standard EOM lexical language generation based on Express-G with its optimization model is a tedious job as seen in the illustration above. It may be difficult to comprehend the ISO Standard Part 21 form for generating EOM. Although a formal definition of EOM has been established from the Express Standard Information Modeling form above, the EOM Representation utilizes (facilitates) two lexical information definition languages. Those languages are called COB Structure Definition language (COS) and COB Instance Definition language (COI) to express EOM lexical forms. The EOM lexical form enables the generation of enhanced optimization models related to a specific domain with its links to product and analysis data. The COS is used to define structure of constraint objects with the schema, cobs, attributes, relations and cob set links.

Below, cob object set called *object category* has been shown with its name, description and attribute and relation types (see Figure 4-9). Then, EOM, CBAM and APMs inherit the constraint object characteristics adding domain specific knowledge. For instance, as described in the previous definition section, EOM has the optimization model attributes, objective, constraint, design attributes, in addition to cbams and apms. Also, EOM has relations; er, ald, ala and alo. Then each category has a list of attributes of relations specific to the domain to be modeled (i.e. how many design variables, what are their names, which type).

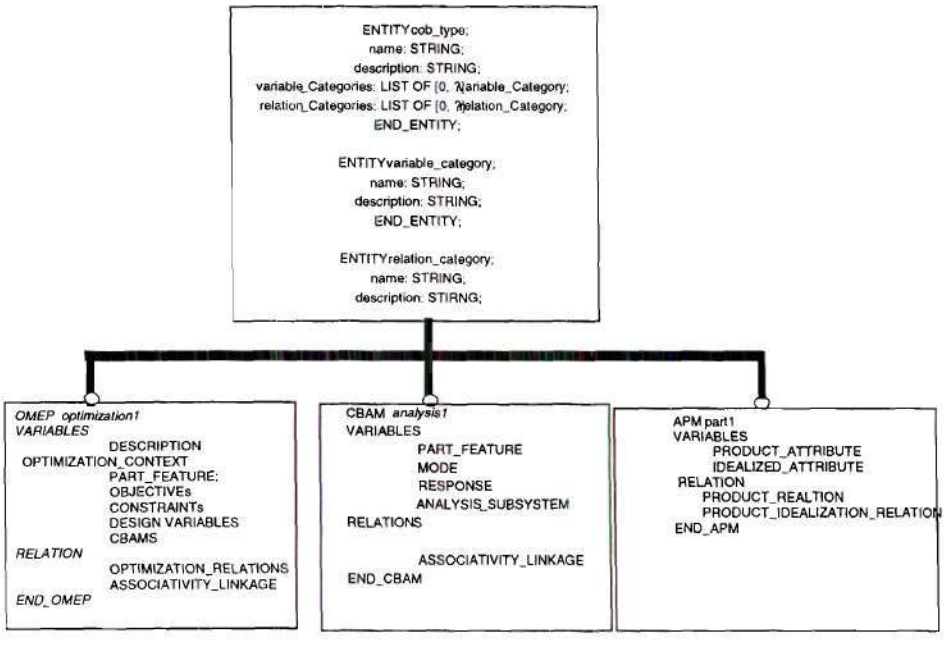


Figure 4-9 Object_ Category: Categories of EOM, CBAM and APM

The COB structure definition language (COS) is developed based on the general purpose STEP EXPRESS language. Constraint Object language has been developed in EISLAB

to define information during optimization-design-analysis integration by extending EXPRESS.

The COI is used to define instances of objects defined in COS. The COB Instance Definition language (COI) has similar characteristics with the general-purpose STEP Part21 File format (ISO 10303-21). Both COI and STEP Part21 may be used to define constraint object data. STEP Part21 is developed to exchange data between different CAD/CAE tools. COI is developed for the purpose of easy creation and better readability. The COI is used to define instances of objects defined in COS. Both COB and COI are stored in files called: the COB and, COI Definition Files, respectively [Wilson, 2000].

The EOM lexical form is illustrated in Figure 4-10. This template shows the structure of lexical form. Under each category attribute, there may be a list of attributes. For instance, attribute **optimization_model.objectives.minimize** has three attributes named respectively, o1, o2 and objective. The naming of those attributes is arbitrary. Generally, they are chosen specific to the domain being modeled. An illustration of a semi- generic eom lexical form is shown below.

```
EOM eom1 SUBTYPE_OF optimization_model;
DESCRIPTION
    "Represents minimization and/or maximization of merits with respect to design or analysis entities"
OPTIMIZATION-MODEL
 OBJECTIVES
  minimize
    o1: REAL;
    o2:REAL;
    objective: REAL;
  maximize
       Null
 CONSTRAINTS
  equality_constraints( = 0)
       Null
  inequality_constraints( < 0)
    g1: REAL;
    g2: REAL;
DESIGN VARIABLES
    x1:REAL;
    x2:REAL;
    x3;REAL;
ANALYSIS MODELS
    cbam1b : cbams.link_plane_stress_model.name;;
    cbam2a : cbams.link_buckling_model.name;
DESIGN MODELS
    apm1a; link.name;
RELATIONS
OPTIMIZATION_RELATIONS
 er1 : <objective>  ==(<o1>+<o2>)/2";
ASSOCIATIVITY_LINKAGE_OPTIMIZATION_ANALYSIS
ala1 : "<m1> == (1.0/(<cbam1b.sx_mos_model.margin_of_safety> + 1.0))-1";
ala2 : "<m2> ==(1.0/(<cbam2a.bx_mos_model.margin_of_safety> +1.0))-1";
ASSOCIATIVITY_LINKAGE_OPTIMIZATION_DESIGN
ald : "<obj1> == (<area>*<cbam2a.link.effective_length>)*<cbam2a.link.material.density>";
ASSOCIATIVITY_LINKAGE_OPTIMIZATION_OPTIMIZATION
alo1: "<g1> ==  <m1>;
alo2: "<g1> ==  <m2>;
END_EOM;
```

Figure 4-10 Illustration of EOM Lexical Form (COS)

EOM Instance Form

The Figure 4-11 shows an example of EOM instance form based on COI. This is one instance of eom1 data file based on the structure defined in Figure 4-10. It shows all

attribute values. The lines in the example with question marks will be derived from the relations. The other lines are input values of those attributes.

```
DATA ;
INSTANCE OF eom1;
DESCRIPTION */
     description: "Represents minimization of merits with respect to design or analysis entities";
/*OPTIMIZATION MODEL*/
  <m1> == ?;
  <m2> ==?;
/* OBJECTIVES*/
    minimize
      o1: ?;
      o2:?;
      objective: ?;
    maximize
          Null
 /* CONSTRAINTS*/
     euquality_constraints( = 0) */
          Null
     ineuquality_constraints( < 0) */
      g1: ?;
      g2: ?;
/* DESIGN VARIABLES*/
      x1: 1.0 ;
      x2:1.5;
      x3 :5.0;
/*ANALYSIS MODELS */
     cbam1b :"analysis1_developed";
     cbam2a : "analysis2_preliminary";
/*DESIGN MODELS*/
     apm1a;" pm_product_x";
END_INSTANCE
END_DATA;
```

Figure 4-11 COI for eom1

The next section describes the graphical forms used in the description of EOM.

4.2.2.2 *EOM Graphical Representation Languages*

Object relationship diagrams

An **object relationship diagram** is utilized as an another view of an EOM. It is a partial graphical view of EOM representation (information that emphasizes the **is-a** and **part-of** relations between objects used to represent optimization, analysis and product models. Numerous notations exist that could be used for this purpose, IDEF1X [Bravoco and Yadav, 1985c], entity-relationship (ER) diagrams [Chen, 1979], and the Object Modeling Technique (OMT) Object Model notation [Rumbaugh, et al., 1991] and more general UML Unified Modeling Language. [Booch et al., 1998]

Object relationship diagrams of EOM emphasize EOM attributes and EOM hierarchies. As described in the earlier section, the EOM lexical structure is the main description for optimization related entities, having all the content of the other views but in lexical form. For example, the notation *a.d* indicates *d* is an attribute of *a*, denoting the *parent-child* relationship. Generilization-specilization and parent-child concepts are presented in a graphical manner in object-relationship diagrams. They aid in comprehending the object relationships.

EXPRESS-G [ISO 10303-11] has been used in this research as the notation for object relationship diagrams. Because, it is an ISO standard and it enables the leverage of research being conducted in EISLAB. EXPRESS-G is the graphical representation of EXPRESS lexical models. EXPRESS-G is described in detail in the STEP standard.

EXPRESS-G diagrams are useful in representing entities, their attributes, subtypes and supertypes.

The object relationship diagram shows which objects are specializations of other objects (the is-a relationship - bold lines), and which objects are parts of other objects (the part-of relationship - thin lines). Thus, an object relationship diagram shows which *attributes* are inherited from superclass objects; however, it does not show which *relations* are inherited.

EOM Constraint Schematic Forms:

Constraint Schematic Forms aid the development, implementation, documentation and usage of EOMs. Below, a brief description of constraint schematics is given first, then the EOM constraint schematic form is explained.

Constraint Schematic Generic:

The other diagram **constraint schematics** graphically emphasizes relationships among objects. In this notation (Fig. 4-12), an attribute can be a simple object like a number, or it can be a complex object having attributes that are themselves complex objects. The strength of this notation is depicting relations among complex objects (analogous to electrical schematics showing connections between complex integrated circuits). This capability is achieved by graphically showing *attribute-subattribute* relations, by abstracting complex objects as subsystems (analogous to integrated circuits), and by supporting hierarchical nesting of other constraint schematics within these subsystems. An option category indicates alternate subgraphs within a constraint schematic (e.g. the

value of f depends on option category 1 in Figure 4-12, where $f = s.d$ for option 1.1, while $f = g$ for option 1.2)[Peak, 1993].

The constraint object views are analogous to flow charts in procedural programming, as they are largely independent of a specific computer implementation.

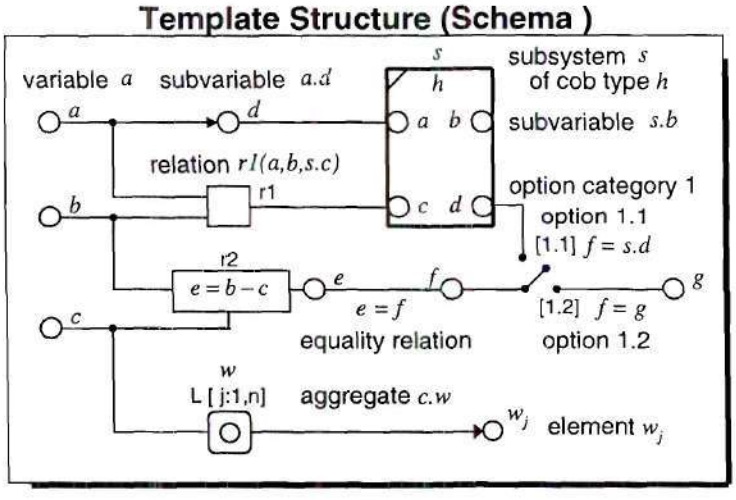


Figure 4-12 Fundamental Constraint Schematic Notation

EOM Constraint Schematic Form:

This section shows how the defined concepts fit into the EOM representation in the Constraint Graph. The constraint schematic has been utilized to show another view of the general structure of EOM. The basic structure of an EOM is illustrated in the form of a Constraint Graph view in Figure 4-13. Examples of constraint schematics for different domains will be explained in Prototype development for eom flap link optimization and implementation in Chapter 5 and Test Cases in Chapter 6. The EOM constraint schematic is one of the EOM deliverables.

Referring to Figure 4-13 again, typical categories of each type of attributes and their relations contained in EOM are shown.

The EOM has an **optimization attribute**, optimization model attribute which has in turn sub-attribute categories of objectives, constraints and design variables. Each has a list of its own attributes named arbitrarily based on their domain being optimized. In Figure 4-13 the objective attribute has list of attributes named as o_1, o_2, ..., o_n. Similarly, constraint attributes are named as g_1, g_2, ..., g_n, where as design_variable attributes are named as x_1, x_2,....., x_n. These optimization attributes are the minimum set of constructs needed to establish a mathematical model. A mathematical optimization model can be generated from these attributes after extracting or linking the engineering entities to those attributes.

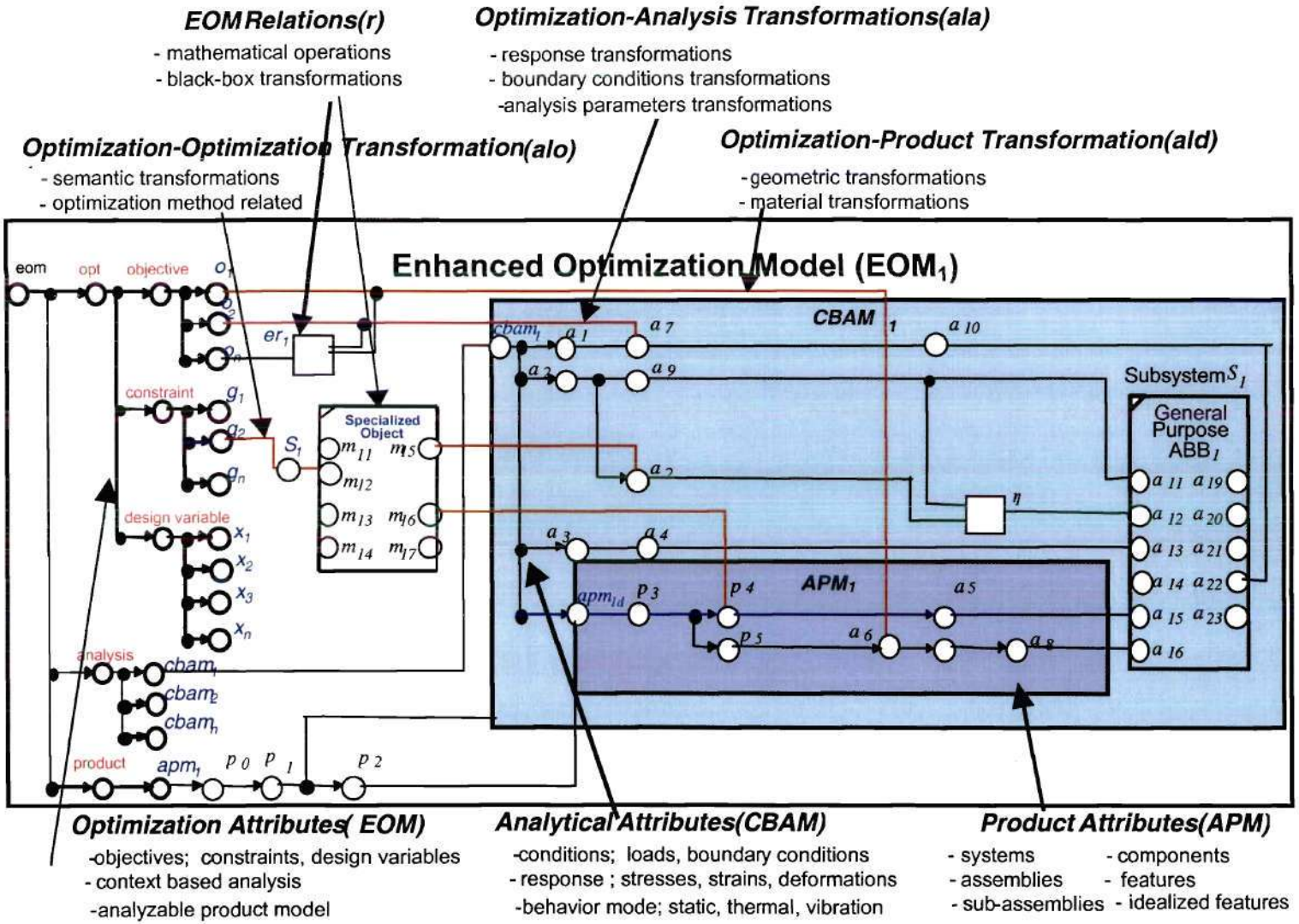


Figure 4-13 EOM Constraint Schematic Form

The EOM has one analyzable product model (apm1) of a specific product p_o at any particular design stage k. The product has a number of **product attributes** (p_1 and subattributes of p_2 in EOM$_1$ in Figure 4-13). APM1 decomposes these attributes into their subattributes (p_3, p_4, p_5 in APM$_1$) per the part-of relationships. Product attributes can be systems, assemblies, sub-assemblies, components, features, and idealized product attributes. Optimization and analysis models utilize those product attributes by means of relations and associativity linkages.

The EOM has one or many analysis models (cbams) representing different behavior and level fidelity. CBAMs can also be included in the scope of the EOM. CBAM1 decomposes **analytical attributes** (a_1 through a_{10} in CBAM$_1$). Analysis attributes can be decomposed into subattributes in the same manner as product attributes. Analysis attributes can be conditions, responses and behavior modes. This can be broken down further into subattributes such as thermal loads, boundary conditions, stress, strains, etc. as part of a relationship until their basic primitive attributes and their values of real and integer types.

Optimization-analysis transformations (alas) are used to link optimization attributes with some of these analytical attributes. For example, an EOM-specific ala exists between a_7 and o_2. An **alo** is contained in the product model as a_6 is linked o1, where as a6 is related to p5 via the part-of relationship.

Similarly, optimization-design transformations (alds) are used to link optimization attributes with some of these product attributes.

Also, there may be **optimization-optimization transformations (alo)** within the EOM scope that relate analytical attributes (e.g., er_1 in EOM$_1$). The emphasis of this type of linkage is relating optimization attributes within an EOM scope and transforming optimization attributes between specialized objects.

An EOM is considered as a **specialized object** if it can contain a general purpose specialized object as exemplified by the presence of $spob_1$ with attributes m11,....,m17. A specialized object is facilitated when special mathematical modules or specific operations are performed many times on attributes of optimization. For instance, linear approximation of objective functions based on Taylor Series expansion can be treated as specialized object. Other examples can be the normalization of constraints and response surface of approximation. Both product and analytical attributes in the scope of the EOM can be typically related to the optimization attributes to such a specialized object. For example, EOM-specific *ala* exist directly between a_2 and special object attribute $spob_1.m_{15}$. Similiarly, EOM-specific *ald* exist directly between p_4 and special object attribute $spob_1.m_{16}$. Finally, EOM-specific *alo* exist directly between s_1 and special object attribute $spob_1.m_{12}$.

Optimization relations (er) are utilized for calculations by mathematical operations.

4.2.3 EOM Protocol

This section describes the major operations of EOM called EOM protocol. These operations have been supported by the implementation phase. Thereby, EOM data can be accessed and utilized in the optimization process. First, the main tasks in the high level has been described by a process diagram (Section 4.2.3.1), then the sequence of tasks performed by EOM-driven applications using the EOM Protocols has been listed, finally that list has been explained further.

4.2.3.1 EOM Protocol Process Diagram

The Figure below shows a basic process diagram for EOM protocol in IDEF0 [U.S. Air Force, 81]. IDEF0 provides a structured representation of functions, information and objects.

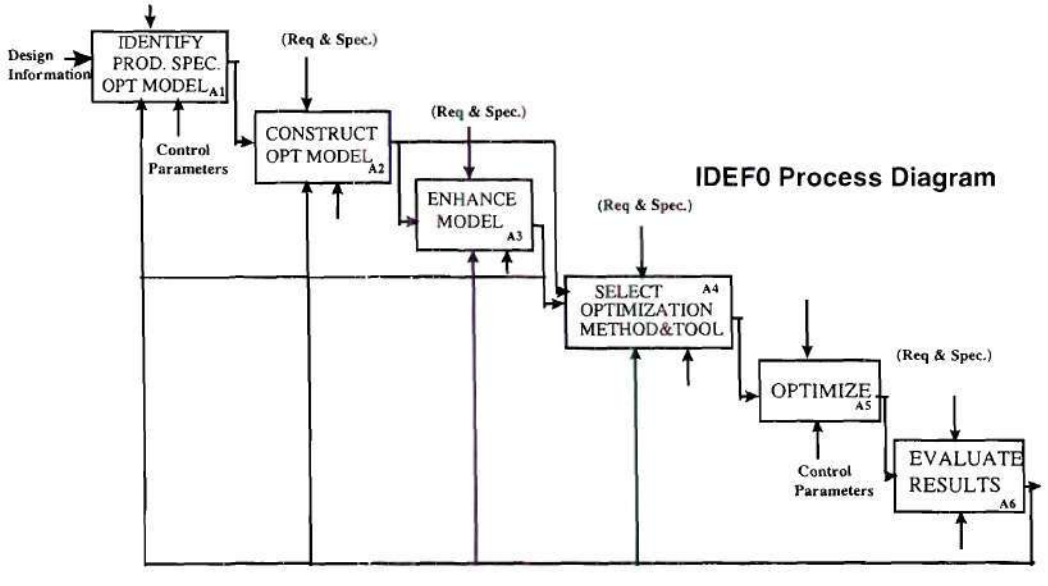


Figure 4-14 EOM Process Diagram

In Figure 4-14, the process starts by identifying the product specific optimization model by engineers. The engineers determine the objective, constraints and attributes in terms of product and analysis data. Then, they prepare enhanced optimization models (**eom**s), with their context based analysis models (**cbam**s) and analyzable product models (**apm**s). Linkages among the product model, analysis model and optimization model are hereby established. Then, the optimization agent selects automatically the proper optimization method with user options and related tools from the optimization tool library. The agent is represented as a class and its methods for the automatic selection of optimization tools. The Optimization Solution Tool Method (OSMM) manages the model and tool interactions. The agent again tells the system to generate OSMM for the specified method and the tool. OSMM handles the preprocessing, execution and post-processing with the optimization model information and control parameters necessary for the specific optimization tool. The optimizer runs, updates and transfers data in conjunction with the analysis and design tools. The process stops when the optimum is achieved according to specified criteria.

Finally, the results are posted. Due to the evaluation of the results, the designer can go back to change the optimization method and tools for the same model.

If the models need to be enhanced, then the Optimization Enhancement will support the changes in the optimization model. The EOM allows enhancement of the optimization model by adding or subtracting objectives, constraints and design attributes. It also permits changes in the analysis programs and product data. The EOM will utilize the

enhancement in the optimization model and archives of the optimization models during the design process.

4.2.3.2 Sequence of tasks performed by EOM-driven applications using the EOM Protocols

The following is an overview of a typical sequence of tasks and their sub-groups performed by any EOM-driven applications using the EOM Protocols. A more detailed explanation follows the itemized overview below.

Overview

1) Definition of Optimization Model Enhancement Models

-identifying and constructing an engineering optimization model

-partitioning of optimization, analysis and design models into their attributes and methods.

-building models (Preparing eoms, cbams and eoms with their schema and data).

2) Creation of cobs (EOMs, CBAMs, APMs) in EOM Protocol

-loading schema (definition information)

-loading data

3) Execution of EOMs Data.

-accessing data in the memory defined with schema and data

- solving relations that set forth in EOM by constraint graph theory (COB Based Analyzer).

-generating Optimization Solution Model Method (OSMM).

-deciding automatically the optimization method and tool based on eom data.

-getting information related to model and tool parameters from OSMM

-initializing of optimizer (pre-processing).

-executing of the optimizer

-executing of the analyzer

-transferring generated data between optimizer and analyzer

-updating new data from optimizer and analyzer

-terminating the process

4) Data Saving

The following is a more detailed explanation of the itemized tasks above.

Description of the Itemized tasks of EOM Definition and Protocol

1) Definition of Optimization Model Enhancement Models

This stage deals with the context definition of the optimization model. In order to define the optimization model, the series of tasks are performed. Those tasks are identifying and constructing optimization problem, partitioning the problem into its optimization, analysis and design entities, and building EOM models.

The *identification and construction* of an engineering optimization model for a product is the first step during the definition of **eom**s with their desired merits, constraints to be satisfied and design variables to be decided in their allowable limits. In this engineering context, the optimization model derives which kind of product attributes are required

from product data and which kind of analyses are required from analysis models and, how theses data are associated in **eoms**. The engineering optimization model is another deliverable of the models; it helps engineers unfamiliar with domain or information management issues to grasp the classical engineering view of the optimization model, in addition, it prepares them toward new information representations of optimization, analysis and design.

The requirements for the optimization model demand the ***partitioning*** of entities into their domains and the ***utilization*** of optimization-analysis-design information. The main purpose of partitioning is to develop modularity that allows model enhancement. Thus, optimization models can be easily adapted to different analysis and optimization tools. Data required for building an optimization model (EOM) can be retrieved from cbam and apm attributes through associativity linkages, as well as optimization attributes in EOM. Partitioning allows reuse of the analysis and design models, their attributes and relations if they are available and associate the necessary data with the attributes in the optimization models. If the analysis and design models, their attributes and relations are not available, modify the cbams and apms by adding attributes and relations whenever necessary or create them in the EOM.

The *building models* for EOM include the preparation of the definition files (EOM.cos, CBAM.cos and APM.cos) and data files files (EOM.coi, CBAM.coi and APM.coi)) that characterize the enhanced optimization model in lexical forms with its analysis and design data information. An example can be found in Section 6-1 in the prototype

development. This information will be input into computer memory as described in the next stage and manipulated further in the execution stage.

2) Generation of EOMs

The EOM Protocol provides a way to input **eom**s into computer memory. The operations are utilized as EOM Representation. In this thesis, lexical language has been taken as a base to input the model definition into computer memory.

First, ***loading schema*** has to be issued in order to make the meta data that correspond to schema (the domain–specific model template) available. This is done by parsing the schema information. Schema described with EOM lexical language and stored in COS files. The schema contains the information related to optimization models (eoms) tailored to analysis models (cbams) and product models (pms).

Second, ***loading data*** has to be issued to create the domain-specific model (an instance of the schema) by parsing the COI data file that describes the model with EOM instance definition language. Eom, cbam and apm instance files corresponding to each eom, cbam and apm schema files have been loaded. After loading schema and data, the information content in the eom model is ready, in the memory, to be accessed and operated on. The tasks listed above are constructed from one or more EOM methods as seen Figures 4-15. There are other possible approaches for input. One is an interactive problem definition through a Graphical User Interface (GUI's).

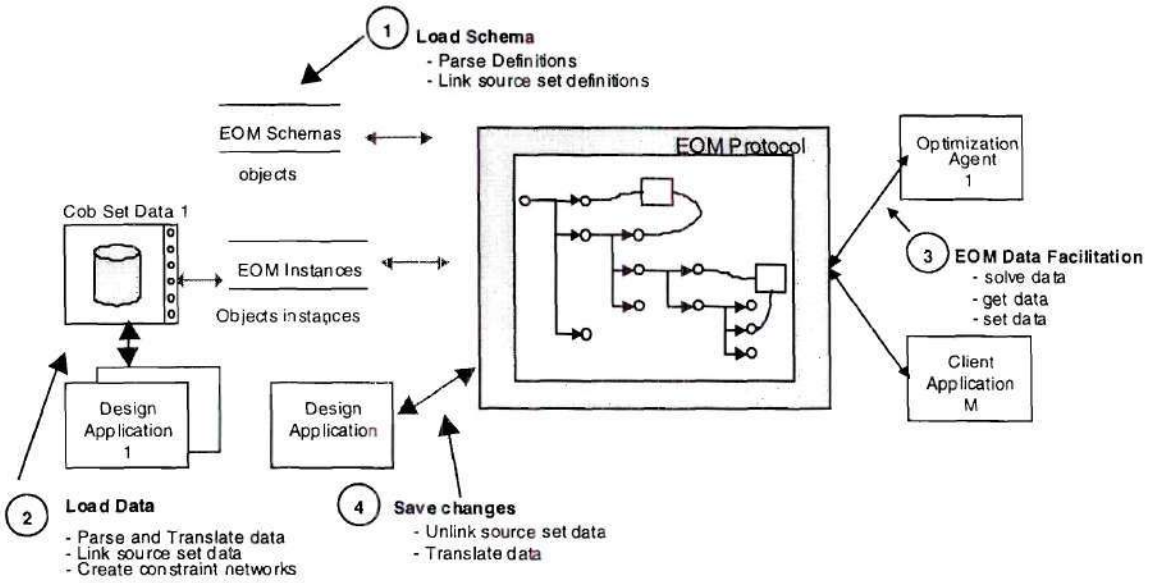


Figure 4-15 Tasks for Input and Data Execution in EOM Protocol

After that *creation of constraint network* must be issued for each domain-specific model.

3) Execution of EOM Data

In this main stage, since necessary definitions and data have been in the memory, data manipulation will be performed.

-accessing data in the memory defined with schema and data:

After constructing the *eoms*, the EOM Protocol must provide a way to ***access*** to instances of eom. Any attributes of an instance are accessed via *get attribute* with *full attribute name* as its argument. **The *full attribute name*** is obtained by concatenating the attribute names all the way up to the *eom* receiving the method in the attribute hierarchy tree from the primitive. To get and set primitive attribute values, *get value* and *set value* may be issued in *eom that is a specialized cob* with *full attribute name* as its argument.

For example, in Figure 4-16 a value of the real attribute **b1** in *cob* **B** can be obtained by sending *get value (b1)* to cob **B**, *get value (a3.b1)* to *root cob* **A**, or *get value( )* to **b1**, the primitive itself.

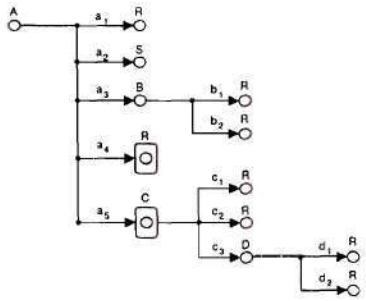


Figure 4-16 Extended Constraint Network Diagram Example

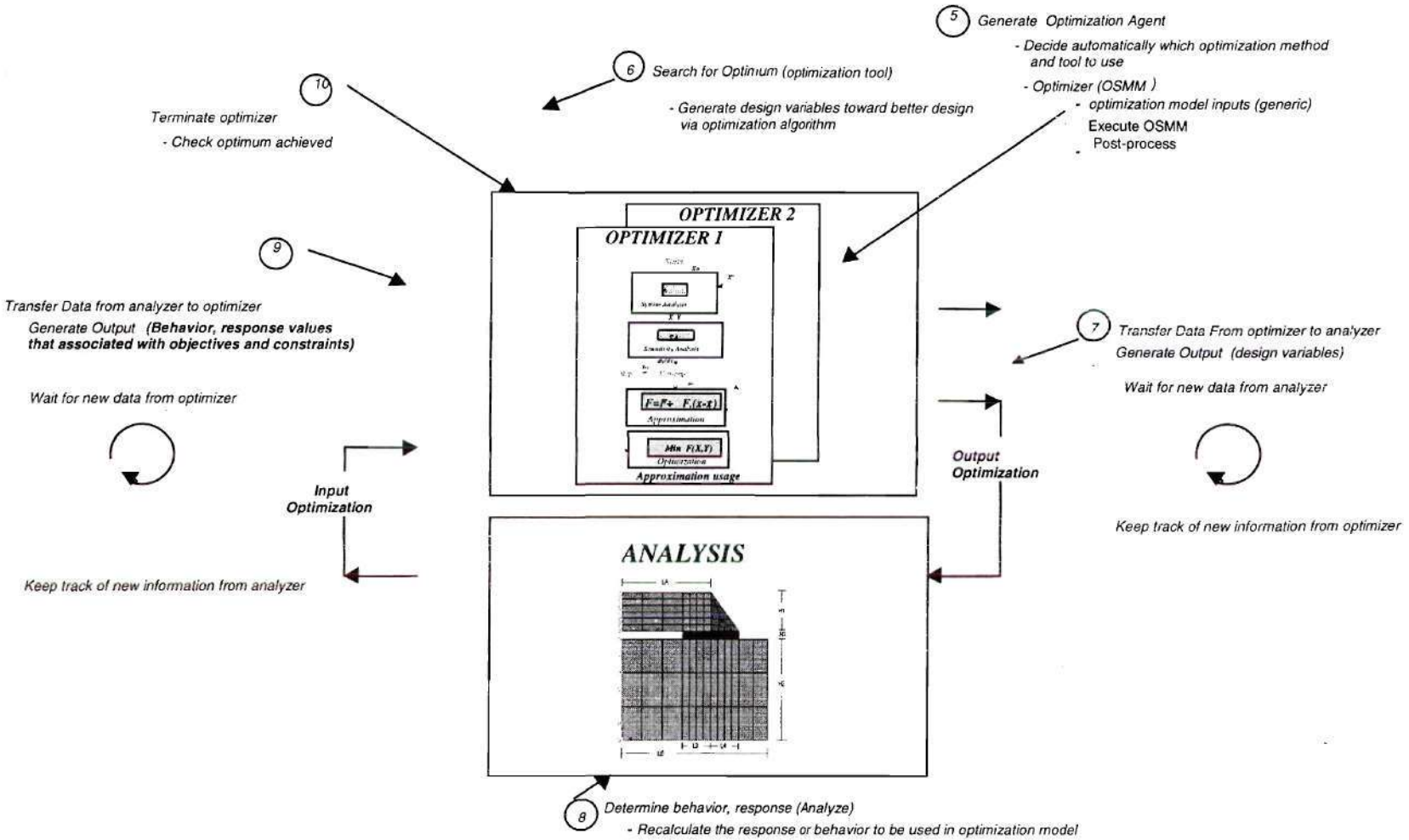


Figure 4-17 Tasks for Data Execution in EOM Protocol

-***solution*** relations that set forth in EOM by constraint graph theory (COB Based Analyzer). ***Solution of*** unknown variables of domain-specific instance can be achieved via *solve* operation available in *root cob*. Unknown variables trigger to create a series of simultaneous equations from constraint network to be solved. The number of the simultaneous equations is the same as the number of sub-graphs in the constraint network. Simultaneous equations are created from two sub-graphed constraint network to solve unknown variables.

To perform design checking or sizing, the directionality (input/output) of COB real-instance may be changed. This can be archived by issuing *set as output* to change from input to output. In a similar way, change from output to input, *set as input* must be issued. Also, relations may be relaxed (dis-activated) or activated. In order to relax a relation currently active, *set active(false)* must be issued. To activate a relation currently relaxed, *set active(true)* must be issued [Wilson, 2000].

Generate Optimization Agent:

This includes the following main tasks to be performed in this process. One of them is the ***decision*** *process* to identify automatically the suitable optimization methods and tools based on eom schemas. The other one is the ***generation*** of Optimization Solution Model Method (OSMM) with subtasks of preprocessing, running and post-processing.

Decision to pick the right optimization methods and tools based on eom schemas requires determining the structure of the optimization problem, and selecting appropriate method

and algorithm from Optimization Library. Then, it requires running the specific optimization tool based on the selected method. This is done automatically by selecting the proper optimization method and the software. Since the knowledge about the optimization problem with its attributes and methods are loaded into the memory of the computer, the information related to the structure of optimization can be extracted and tested with a class with its methods in object oriented approach.

Figure 4-18 shows a partial view of the classes of optimization methods and tools.

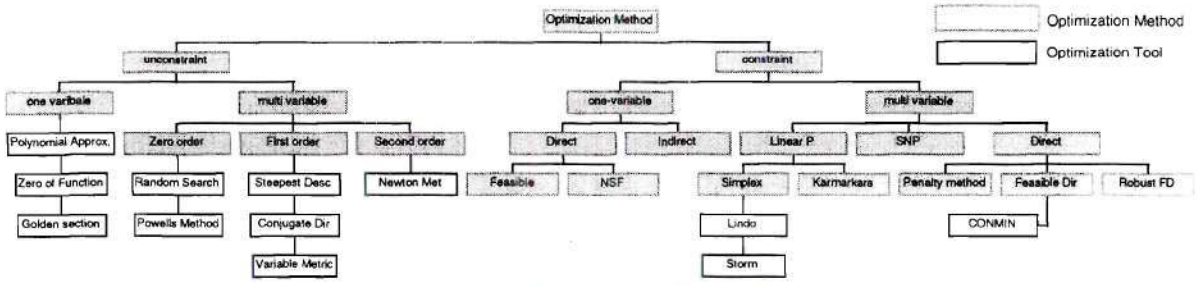


Figure 4-18 Classes of Optimization Methods and Tools

For instance, in the implementation, choosing the methods and tools is done by deciding the design variables (parse the design variables, count them, see if they are continuous or integer, evaluates them all together if continuous, discrete or mix_based problem). The algorithm decides which method to use and which proper optimization tool to launch, based on the optimization software from the library.

After the tool has been selected, Optimization Solution Model Method (**OSMM**) is **generated** to manage the inputting the necessary model and tool parameter to optimizer (preprocessing), running it and post-processing it. Information related to model and tool

parameters are used for the purpose of initialization of optimizer (pre-processing). The first set of information is related to the ***modeling information*** that is independent from optimization method and tool specification (The main focus). This requires the specification of how many design variables with their upper and lower limits and starting values (starting vector) are available. The modeling information also includes the answers to the following questions; what are they, what are the objectives and constraints functions, how many of them and are they defined as mathematical functions or black-box form (math module, software, statistical tool, in-house code etc.)? The second set of information is specific to method and tools selected. It includes the *preprocess control parameters* such as how many iterations are intended and what is the convergence criteria needed to stop the optimization process. Others are the options to control in the tool. They include how the modeling information described above are formatted at preprocessing stage (GUI, Data files).

-***Execution*** of optimizer:

Optimizer - Analyzer - Design tools are at interaction among each other in the optimization process. Generally, optimization tools work in a recursive manner. Starting with initial design variable vector defined in the definition stage, optimizer evaluates the objectives and constraints calculated inside the optimization algorithm if they are simple enough. Otherwise, they are generally transferred (linked to) from other sources where more complex analysis might be required by specialized analysis modules. For instance, these modules can be wrapped in COB Analyzer as an external tool in the implementation. Calculation in the optimizer is based on some sort of search algorithm

to find optimum value of the objective function by evaluating the objectives and constraints (see optimization methods [More, 1994]). The optimizer is stopped when desired convergence or iteration number achieved. Otherwise, they give a new set of variables toward better objectives during the search at each iteration cycle of the optimization process.

The new design variables are used to find the new responses and behaviors to calculate the objective and constraint functions. Generally, **analyzers** are used for that purpose.

-***Transfer*** generated data between optimizer and analyzer (Cob Analyzer):

The new design variables are ***transferred*** to analyzers and design tools. Again, responses from the analyzers are calculated and the objective and constraint data is transferred into the optimizer. In this thesis, we utilized a cob analyzer to calculate necessary information by benefiting of object oriented approach. When information from the optimizer is ready, COB Analyzer utilizes this data to find the desired behavior or responses, otherwise analyzer waits for data until the data from optimizer is available. Utilization of this data from the optimizer includes retrieving the new design variable value, setting the design variables with new values and re-solving the problem again.

The same is also true for the optimizer, after the first initialization, the optimizer runs and outputs a new data used by the analyzer, otherwise, it waits for the data from analyzer. The crucial data from the optimizer is the new design variables transferred to the analyzer. The crucial data from the analyzer is behavior or response values used in the objective and constraint functions.

update new data from optimizer and analyzer :

This requires updating the new design variables, behavior variables, constraints and objectives in every recursion cycle. This can be thought of as broadcasting. This information could be used in interactive (dynamic) iteration history, post-processing, CAD models and log information.

terminate the optimization process by optimizer due to a criteria:

Recursion process is terminated by the optimizer, when the convergence or iteration criteria is met. Criteria are being checked in every iteration cycle. Convergence criteria is generally based on checking the successive absolute normalization values of the objective values and/or design variables not to exceed the specified error. Iteration criteria must not to exceed allowable iteration number depending on method and heuristic knowledge. Both cases are shown below.

$$\left|\frac{objective_{i+1} - objective_i}{objective_i}\right| \leq \varepsilon_1 \text{ or } \left|\frac{x_{i+1} - x_i}{x_i}\right| \leq \varepsilon_2$$

$$IterationNo \leq MaximumIterationNo$$

4.2.3.3 Data Saving

The EOM Protocol provides a way to save data. The instances created or modified during the utilization of the EOM may be saved as linked or unlinked forms. The linked instances conform to the linked version of EOM schema that has one cob-set merged

from two or more cob-sets. The linked instances can be separated into the unlinked instances that conform to the original individual cob-sets of the COB [Wilson, 2000].

Chapter 4 presented the main building blocks of EOM representation with its information model, description languages and its protocol in implementation independent form. The next chapter explains how those concepts are implemented in a prototype.

CHAPTER 5

EOM REPRESENTATION IMPLEMENTATION

The previous Chapter 4 formally introduced the EOM Representation. This section briefly explains a prototype implementation of the EOM Representation with the flap link illustration. The concepts presented earlier will be implemented in a developed computer program called EOM Tool. Then, a prototype example presents the main functions of EOM Representation implementation through the EOM Tool. Subsection 5.1 presents the implementation phases in the computer environment. Later, these phases are paralleled to the EOM protocol. Then, a flap link optimization example illustrates the main functionality of the EOM representation. The next chapter will present the test cases from different domains to further demonstrate the functionality, diversity and capability of the EOM representation.

5.1 Computer Implementation of EOM Representation

The EOM Representation is implemented with an object-oriented programming language, Sun Microsystem's Java interacting with Optimization Programs (CONMIN 1978), Constraint Solver (Mathematica), and Analysis Programs (ANSYS Finite Element

Program). Text files are also used for defining schema and instance files. The simple architecture of the EOM Representation is shown in Figure 5-1.

Java is used as a Browser for loading the schema and instance files into memory and then to execute the objects defined in the eom files for the purpose of optimization. Entities in the EOM Information model are converted to Java classes and EOM Protocol operations. Protocols are converted to methods using these classes.

The browser also is used to execute the data for the solution by constraint solver, allowing bi-directional calculation, interaction with optimizers and analyzers. Optimizers are used in conjunction with the analyzer. The optimization model's information is supplied by the analyzer. Optimizer algorithms can be transformed into computer languages. They can be commercial or research algorithms developed by organizations and universities. CONMIN, Modified CONMIN and BFGS are used during the implementation of prototype development and test cases in Chapter 7.

Some functions of the Xaitool have been utilized and extended to the level of optimization model building, running the optimizer in addition to analysis capabilities. The EOM Tool utilizes and extends the XaiTools™. *The* EOM Tool is a Java-based prototype-tool for optimization-analysis integration. Basically, EOM Representation sweeps concepts of optimization, analysis and design taking optimization as a base. The EOM Representation extends the MRA [Peak, 1995]. EOM concepts are implemented in the EOM Tool.

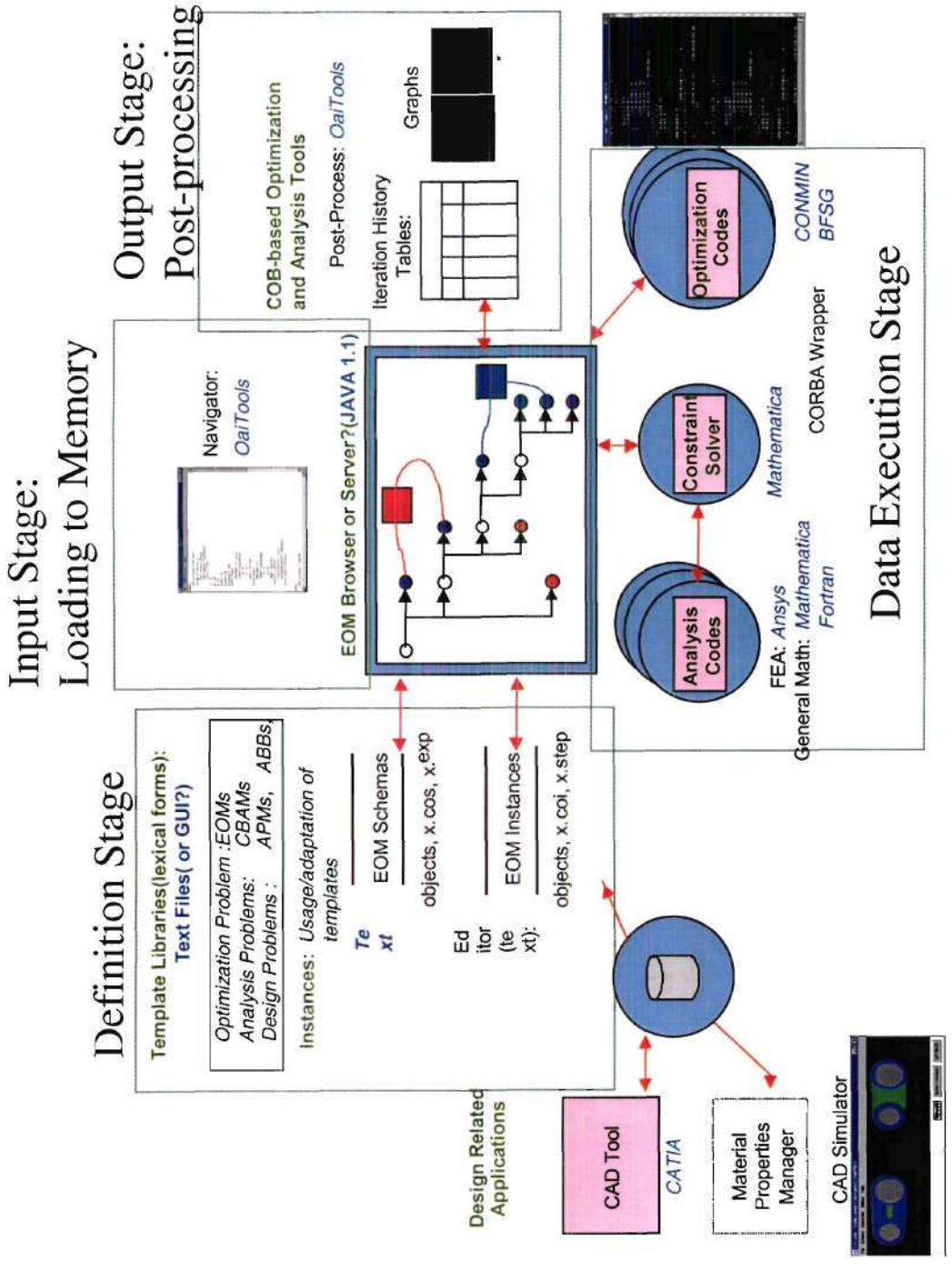


Figure 5-1 EOM Implementation with its Phase and Computer Architecture

The current tool for architecture, XaiTools, is explained in [Wilson, 2000]. Some functional characteristics of XaiTools include the integration with representative analysis tools FEA (Ansys) and general mathematics symbolic solver (Mathematica). Mathematica is used as a main engine for the calculation of relations represented with the constraint graph theory. When special analysis is required, the analysis is handled through that engine (Mathematica). For instance, an external tool (Ansys FE) is required to calculate the stress for the given geometry and loading parameters defined in the cbam, executed through Mathematica.

Basically, EOM *Representation* implementation has been also divided into different phases as described in the EOM Protocol Section. The phases with tools in the architecture are described below.

1) Definition Stage

This is the Enhanced Optimization Development stage. Lexical forms are used to define the necessary objects describing the optimization domain with its links to the analysis and product domain. For that purpose, the text editor has been used to specify eom, cbam and apm schema files with cos extension. The text editor also has been used to define the data file with coi extension. Data files correspond to schema files. Express files with exp extension and Part 21 files are also supported. Eom schema files represent the optimization related objects with links to cbam and apms. Cbams represent the analysis objects with links to design objects. Apm objects define the design information with its geometric and

material data schema. Schema and object definitions are given in the text files with cos extension.

Tool: *A text editor.*

Representative definition files (eom.cos and related cbam.cos and apm.cos) in the flap link example have been presented in section 6.1 in the next chapter. The data files (eom.coi and related cbam.coi and apm.coi) are prepared in the same way. It is also possible to use a graphical user interface (GUI) instead of lexical forms.

2) **Input Stage(loading)**

Solver: Java Browser is used to load schema and to load files.

Tool: Java - GUI

As described earlier in Chapter 4, Schema Definition and Instance files have been defined in text files with a specified format in lexical forms. Those files are loaded to the EOM Tool browser to send the information content from the files to the computer memory. Schema files contain the meta data about the domain related enhanced optimization model information with its attributes and relations. The **eom.cos** and **eom.coi** files for the flap link prototype example have been prepared in the definition stage by the engineers. During this stage, the text files prepared at the definition stage have been loaded. The Java Browser has been developed to load the definition and data files. The Java browser is capable of parsing and reading the EOM lexical forms.

In Figure 5-1, the input stage can be seen. The user picks the proper definition and data files prepared for the optimization of the particular part from the browser. Specifically, the user chooses the buttons:

a) **Loading** for an eom_cos file with the links to CBAM and APM using EOM browser. This action leads to creating constraint object structure

b) **Loading** for instances of eom.coi and cbam.coi and apm_geometry.coi and apm_material.coi.

Data Execution Stage

The information model about the optimization problem with its attributes and methods has been loaded into the memory of the computer in the previous stage. Then this data will be manipulated to find an optimum solution. For that reason, the actions described in the protocol section have been implemented with the following method and tools. Those tools are specified in the Figure 5-1 with the blue text next to the figures.

EOM-Agent: Java Class

Solver: Used Constraint Graph Theory -

Tool: Mathematica

Optimizer :

Methods: Modified Feasible Direction Method

Variable Metric Method

Tools: CONMIN, NASA FORTRAN based optimization software

BolinkGeneral, FORTRAN based optimization software

Modified BFGS FORTRAN based optimization software

Analyzer: FE Element Analysis, Formula, In-house

Tools: ANSYS, Mathematica, FORTRAN, Java

Figure 5-2 Methods and Tools used for EOM Implementation

The optimization agent is a Java class with its attributes and methods; it works underneath the EOM Browser. The Optimization Agent **determines** the character of the optimization problem based on eom information contents, and it also chooses the proper optimization method and tool. Then, it **generates** Optimization Model Methods (OSMM) based on the selected tool for handling the preprocessing, execution and post processing tasks.

The information about the structure of optimization can be explicitly retrieved and tested with a class or methods. The algorithm decides which method and which proper optimization tool to use based on the optimization software from the library. Figure 5-3 shows some part of the optimization agent by the methods in pseudo code. That is an illustration of the optimization agent implementation. Basically, the agent parses the design variables, counts them to see if each design variable can be treated as continuous or integer, then evaluates them all together to determine whether the optimization model requires the continuous, discrete or mix_based optimization tool. The second part is to transfer the necessary optimization model information and tool dependent information to the tool for automation purposes.

```
/*_AFTER, THE FIRST READING EOM, DECIDE THE CHARACTER OF THE PROBLEM,
CHOOSE THE METHOD AND PREPARE THE OSMM (PREPROCESSING, SOLUTION AND
POSTPROCESSING)
*/
//How many design variables are available and, which kind are they?
/* This section is about getting x*** attributes (design variables) from the memory */
String[] design_variables = new String[ parseDesignVar("x").length];


System.out.println ("CHOOSING THE APPRPRIATE METHOD");
 /* The java method in bold, does the following task
First sort it out continuous variables from discrete, then count each
If all continuous then
"CONTINUOUS" method
If all discrete method
" DISCRETE"
if  between them
"MIX_INTEGER method."
*/
public String decideDesVar(String DomainName,String[] DvName)


/*PREPROCESS OF OSMM : */
/*OSMM INFO to a file (transformation) */
```

```
/*A- (OPTIMIZATION MODEL INFORMATION TRANSFORMATION)*/

public void Transfer(String[] objective,String[] constraint,String[] desvar)
   {System.out.println("INITIAL TRANSFORM TO OPT");

      String outputFileName = "INITIALOptOSMM.txt";

/*B- (OPTIMIZATION TOOLS CONTROL PARAMETERS)*/

public void Transfer(String[] epsilon, String[] max_iter_no , String[] )

/*
-RUNNING OPTIMIZER FROM LIBRARY CLASS OPTIMIZER (They have a library of tools and
which environment to run them).  In the case of flap-link implementation tools are bolinkgeneral.exe for
continuous bolinkgeneral1.exe for mixed and discrete.exe for discrete*/
om= new Optimizer1();

if (gg.equals("c") )

 {
 om.optim("D:\\XaiTools\\v0.3\\optimization3-cob-improve\\bolinkgeneral.exe");

 System.out.println("Continuous"); }

if (gg.equals("mix") )
 {
 om.optim("D:\\XaiTools\\v0.3\\optimization3-cob-improve\\bolinkgeneral1.exe");
   System.out.println("Mixed");

   }
if (gg.equals("d") )
 {

    System.out.println("Discrete");
    proexec("D:\\XaiTools\\v0.3\\optimization3-cob-improve\\discrete.exe");

   }
        }
```

Figure 5-3 Optimization Agent Implementation Illustration

Optimizers

There are numerous algorithms and codes of optimization available in the optimization field for research and commercial usage. Each handles a specific class of optimization problems with its strengths and weaknesses in functionality, performance and accuracy. During the preparation of the optimization library, the aim was to illustrate the functionality

of EOM more than performance. For that reason, programs in the optimizer library are chosen or developed on the basis of availability and familiarity. The library can be extended to cover a variety of optimization problems with optimization tools that are more accurate, powerful and have better performance.

In the prototype implementation, the **BolinkGeneral** program has been used with its CONMIN subroutine [Vanderplaats, 1978]. CONMIN is a NASA developed optimization program based on the method of Feasible Direction. It is written to solve linear and non-linear constraint and unconstrained optimization problems. The advantage of that program is that it is written as a subroutine and easily interfaced with another program such as FE programs. It is also a FORTRAN program that is publicly available and widely used in the structural optimization area. CONMIN is a general purpose numerical optimization subroutine which can be used to solve a variety of optimization problems. CONMIN will change the input values of the analysis to optimize the user defined objective, subject to constraints that are generally supplied from analysis. The response values from analysis define the objective and constraint function values. The responses calculated in the analysis programs can be linear or nonlinear functions of design variables. They may be simple formula based analytical functions or they may be highly complicated implicit functions of design variables. The latter may require special computational analysis packages to obtain the desired responses used in the optimization model.

BolinkGeneral calls the analysis programs repeatedly while searching the optimum of the optimization problem.

"CONMIN is a FORTRAN program, in subroutine form, for the solution of linear or nonlinear constrained optimization problems. The basic optimization algorithm is the Method of Feasible Directions. The user must provide a main calling program and an external routine to evaluate the objective and constraint functions and to provide gradient information. If analytic gradients of the objective or constraint functions are not available, this information is calculated by finite difference. While the program is intended primarily for efficient solution of constrained problems, unconstrained function minimization problems may also be solved, and the conjugate direction method of Fletcher and Reeves is used for this purpose."[CONMIN Manual, 1978].

In many mathematical problems, it is necessary to determine the minimum or maximum of a function of several variables, limited by various linear and nonlinear inequality constraints. It is rarely possible to solve these problems directly, and iterative methods are used to obtain the numerical solution. CONMIN is a FORTRAN program, in subroutine form, for the minimization of a multi-variable function subject to a set of equality and inequality constraints.

OBJ need not be a simple analytic function, and may be any function that can be numerically evaluated. Figure 5-4 shows the main usage of the Bolinkgeneral program with its analysis and optimizer subroutines.

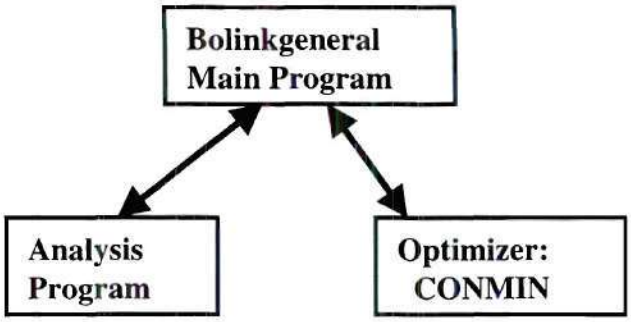


Figure 5-4 Usage of BolinkGeneral Optimization Program

EOM Agent launches the executable version of the structure shown in the Figure 5-5. Generally, the analysis function or subroutine is provided to evaluate the objective and constraint values. In the prototype implementation, the COB analyzer has been used to direct the necessary analysis evaluation. The main program of the optimizer calls the CONMIN to proceed with the optimization. The optimizer modifies the design variables during the search for the optimum. When the optimizer requires the values of the objective function and constraints corresponding to the optimization model, the main program obtains the evaluated objective and constraint values transferred from the analyzer.

BolinkGeneral1 is a modified version of BolinkGeneral which is used to simulate a mix-integer algorithm developed by the author. It finds the continuous optimum solution for holding constant each discrete set by using CONMIN. Then, it sorts the best objective function value among the discrete set where the continuous optimum has already been found. This program can be easily replaced by a more efficient and faster program doing the same function. In the truss test cases in Chapter 6, BolinkGeneral and BolinkGeneral1 are utilized again.

Another program based on BFGS method is also utilized for the Solder Joint Fatigue case (Chapter 7). That program is also extended to handle the mix-integer optimization problems.

BFGS is a variable metric method. The prototype implementation is based on FORTRAN. The aim of variable metric methods is to accumulate information successive line minimization so that such line minimizations lead to the exact minimum of a quadratic form in N dimensions. The basic idea of the variable metric method is to build up, iteratively, a good approximation to inverse Hessian Matrix [Press et al., 1992]. It could be the feasible direction method if it is appropriate for the mathematical structure of the problem. The Figure 5-5 below shows a screen shot of the fatigue life optimization program.

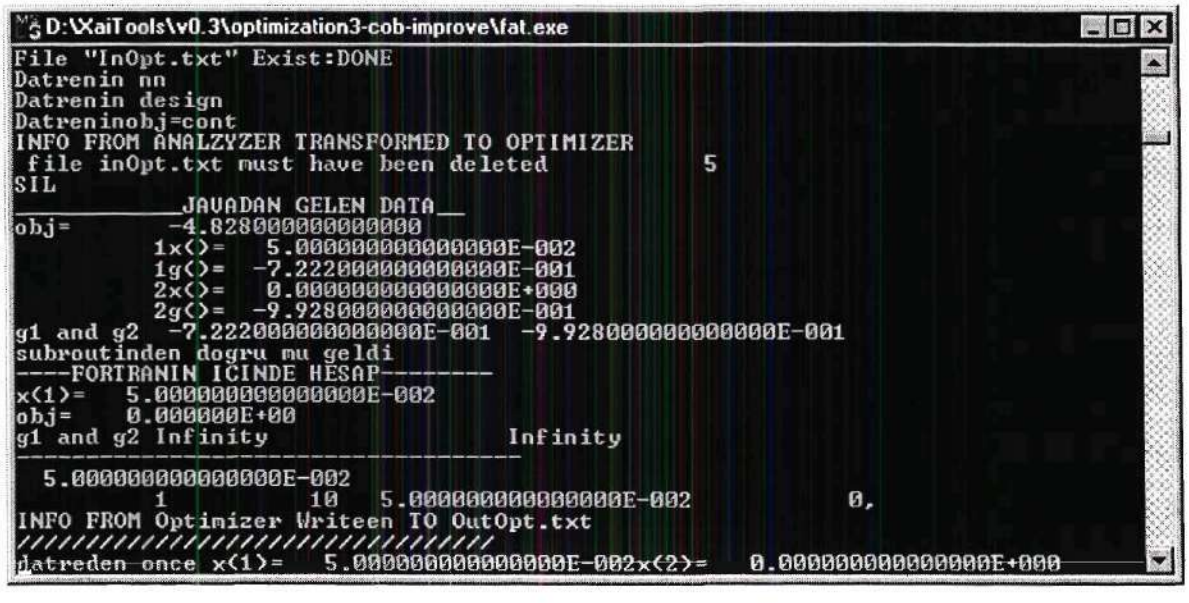


Figure 5-5 Execution of Optimizer for Fatigue Life Optimization

Furthermore, ANSYS optimizer routines are sometimes utilized when ANSYS is used as the analyzer. These optimizer routines are highly coupled with the ANSYS analyzer. ANSYS optimizer implementation will be discussed in the related test cases.

Analyzer:

For formula based analysis, FORTRAN program and Mathematica have been used to find the desired outputs from analysis.

FE element analysis has been used for higher fidelity analysis, where formula based analysis is not adequate to model and solve the analysis problem. ANSYS FE program has been used in this thesis for that purpose.

3) **Output Stage** (Post-processing)

Eom Browser:

The EOM Browser is generated to load the schemas and data, then solve for the loaded system with their initial design configuration. Loading schema and data to the memory and generating constraint network helps the next step to manipulate all those data. As described in earlier sections, the domains are partitioned as eom, cbams and apms.

Figure 5-6 shows the EOM browser with its initial configuration as defined in the schema and data files. Flap_link optimization has objective named as obj and two constraints as g1 and g2. It also has analysis object cbam1 and design object apm (truss_link), which has its own attributes shown in the figure.

An engineer first loads eom objects with links to cbam and apm objects [Wilson, 2000], and then loads instances of eom, cbam and apm and material objects consecutively by Eom Browser. This will create a constraint network. Mathematica is utilized to solve the necessary desired responses and behaviors used in optimization model of eom.

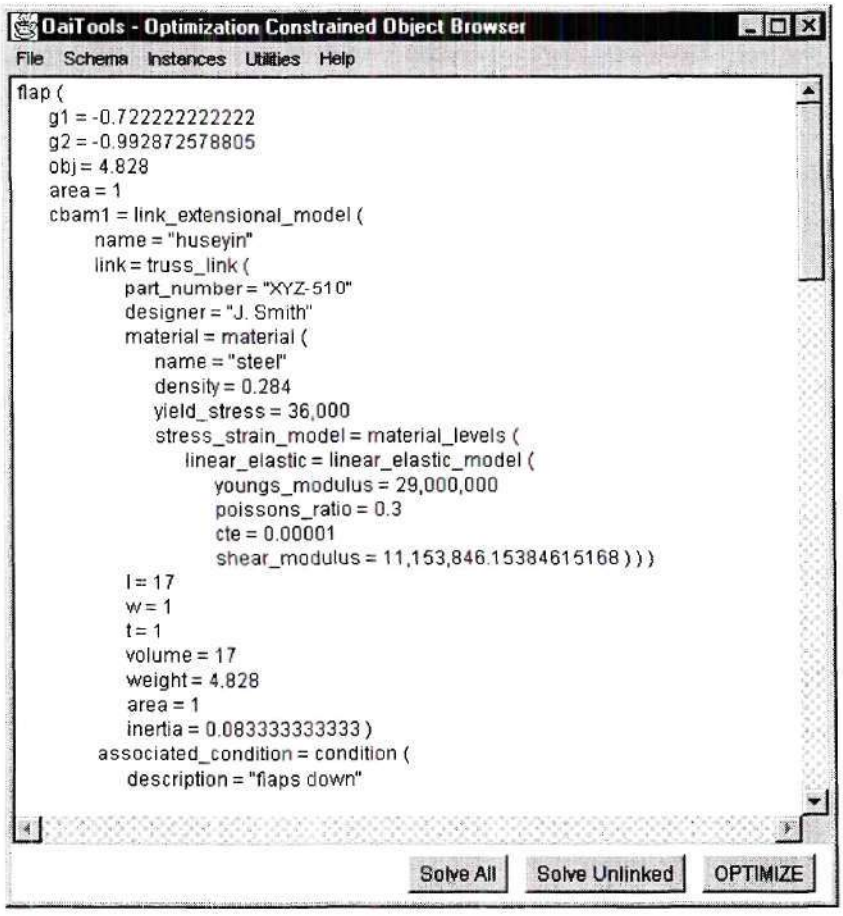


Figure 5-6 EOM Browser

Later, the optimization and analysis interaction occur and corresponding data are updated in each iteration of the optimization process.

EOM Grapher:

Although XaiTools supports integration with representative design tools: geometric CAD (CATIA), libraries (materials, fasteners) [Tamburini 1999, Chadenskar 1999, Wilson 2000]. The Geometric Modeler simulation for flap link has been implemented in Java using Graph classes. Figure 5-7 shows Optimization Grapher for flap_link test case. Figure 5-7 shows the flap link geometry before optimization and after optimization such that the design changes of the geometry of the flap link for every iteration loop.

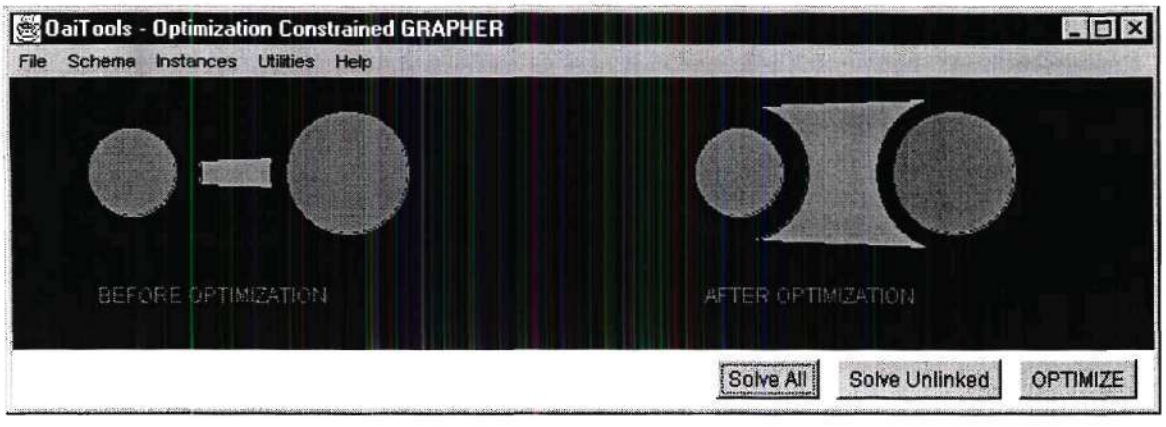


Figure 5-7 EOM Grapher for Flap Link

The main purpose of the grapher is the simulation of CAD model and data interaction between the optimizer and CAD. In other words, Figure 5-7 demonstrates the graphical form of the object. It is a simulation to geometric model to demonstrate how certain parameters changing dynamically in each iteration.

Optimization Iteration History

EOM Tool generates another view to keep track of the iteration history of the objectives, constraints and design variables in each recursive step. Display is a frame in Java as a

tabulated form. It is also possible to generate graphical iteration history of the objectives and constraints using the tabulated data.

The next chapter explains test cases in which EOM concepts are developed, and implemented with the prototype described in this chapter.

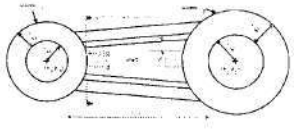
CHAPTER 6

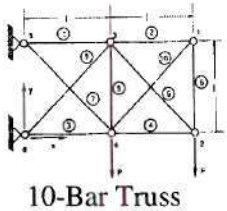
TEST CASES

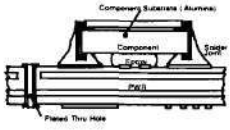
This chapter presents a series of cases that test and validate the EOM Representation. The results of these test cases will be used in Chapter 7 to evaluate the EOM Representation. This evaluation will be compared with the objectives set in Chapter 3 in order to demonstrate the EOM Representation capabilities. For that purpose, three optimization problems with their different models are presented in this chapter. These test cases, each corresponding to a different engineering part, are respectively,

1. Flap Link Optimization (Subsection 6.1);
2. Ten Bar Truss Optimization (Subsection 6.2);
3. Solder Joint Fatigue Life Optimization (Subsection 6.3);

As illustrated in Figure 6-1, these test cases include Test EOM Definitions and Applications.







gaps addressed \ test case	Flap_Link	10-Bar Truss	Solder Joint
Optimization Integration	✓	✓	✓
Enhancement	✓		✓
Method & Tool Identification	✓		✓
Extend MRA	✓	✓	✓

Figure 6-1 Test Cases versus Main Functions of EOM

Section 6.1 presents a prototype example of a flap link, for explanatory purposes. The flap_link_preliminary design with its low fidelity analysis is simple enough to show basic ideas about the product data, the detail analysis, the model enhancement and the optimization method selection aspects of the system. Starting with a simple example demonstrates the development of eom models in conjunction with cbam and apm models. First, eom has been developed with preliminary apm and low fidelity cbam. This eom shows how the integration has been performed through eom representation. Lexical and graphical forms of EOM have been demonstrated in this example. EOM Definitions that utilize the EOM Definition Languages are introduced in Chapter 4. Section 6.1 presents EOM Applications that demonstrate the use of the Java classes and methods presented in Chapter 5: EOM Implemention. Enhancement of optimization models based on analysis model A1 (1D case), and the later higher fidelity analysis model A2 (2D FEA) has been illustrated. Results of the flap link optimization case have been presented. Selection of the optimization method and tool based on the information content of the optimization model has been demonstrated with this prototype.

Also shown in Figure 6-2 is the test case for the ten bar truss presented in Section 6.2. The main emphasis for this case is to compare the optimization results with the published results. This example also shows that flap_link_preliminary product data can be used for this case. The links of the ten-bar truss have the similar geometry and material data.

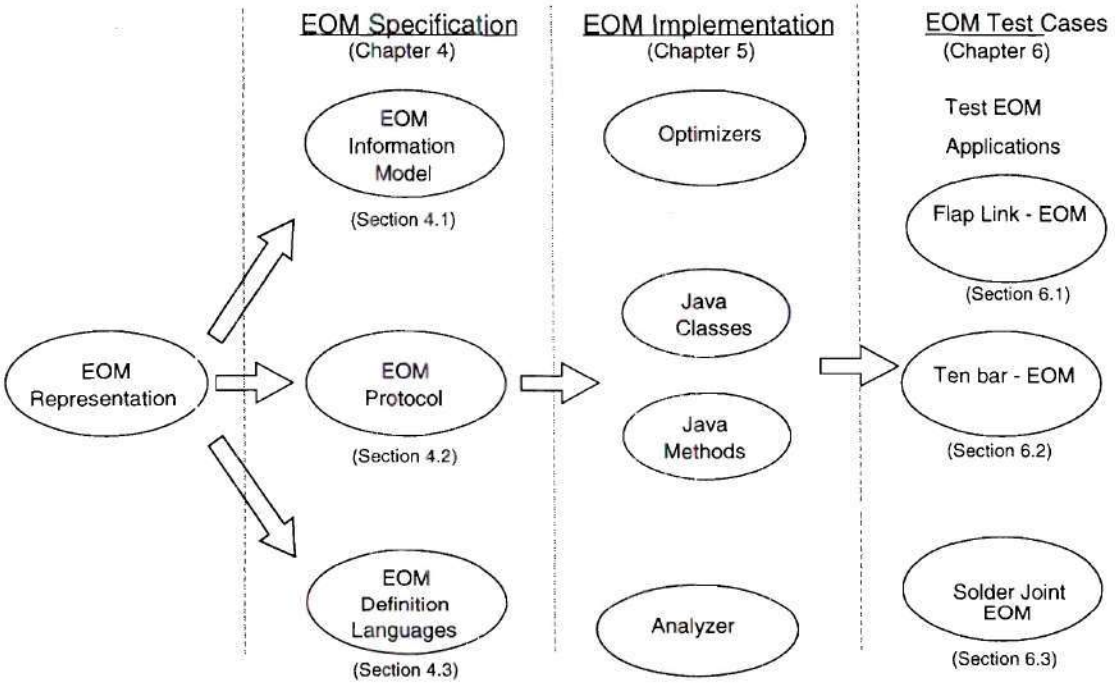


Figure 6-2: EOM Representation Application and Testing

Finally, Section 6.3 presents test cases demonstrating an EOM approach with a solder joint fatigue life optimization case. This example utilizes the Engelmair fatigue model for the lower fidelity analysis. Based on that model, optimization models have been developed. Later, the fidelity of the analysis was increased as the 2D Finite Element of the rectangle and a detailed cross-section of the solder joint were used to calculate fatigue life. Again, optimization models have been generated based on those models. The eom representation of this case with associativity links has been demonstrated.

6.1 Prototype: Flap Link

6.1.1 Physical Description

The flap link has been used as a prototype first test case. The engineering drawing of a flap link is shown in the Figure 6-3 below. This flap link prototype example was developed in EISLAB during the PSI Boeing project [Peak et al., 1999]. It is a simple rod composed of two sleeves (sleeve1 and sleeve2), a shaft, and two ribs (rib1 and rib 2). This product is designed to work at compression and tension loading conditions. The main emphasis in this prototype is to illustrate the main concepts of EOM representation starting with a simple design.

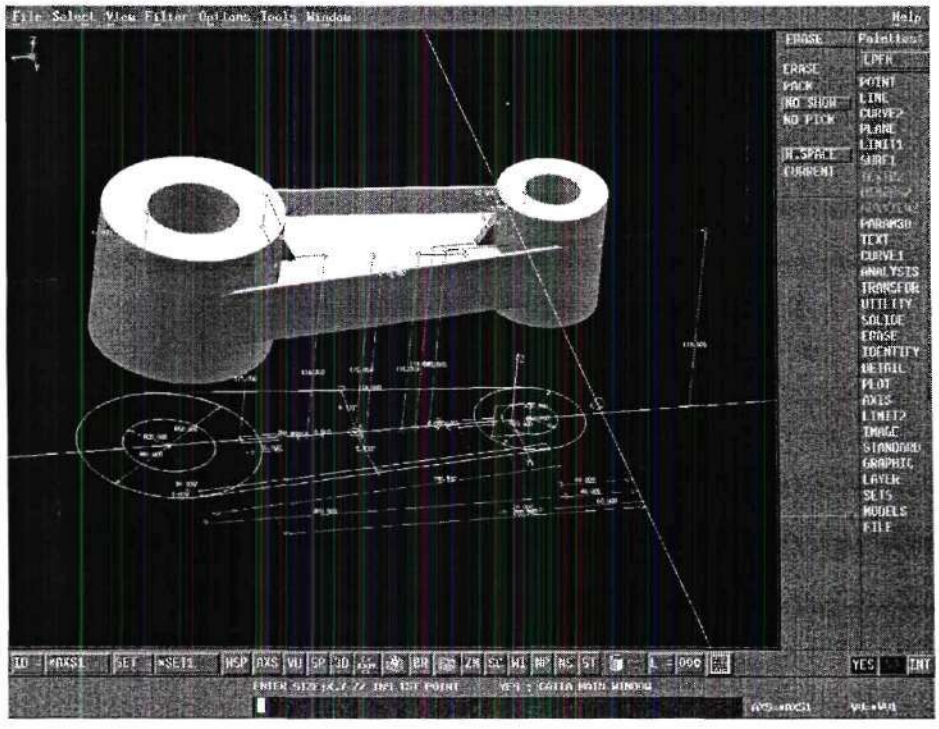


Figure 6-3 Flap Link Engineering Drawing (CATIA)[Chandeskar, 1999]

This design evolves from the preliminary to the detailed stage. Figure 6-4 represents the engineering in detailed design stage ready for manufacturing. During the product development, iteration is required at different stages of the design. As the design evolves, it requires more extensive optimization and analysis models to satisfy the design requirements.

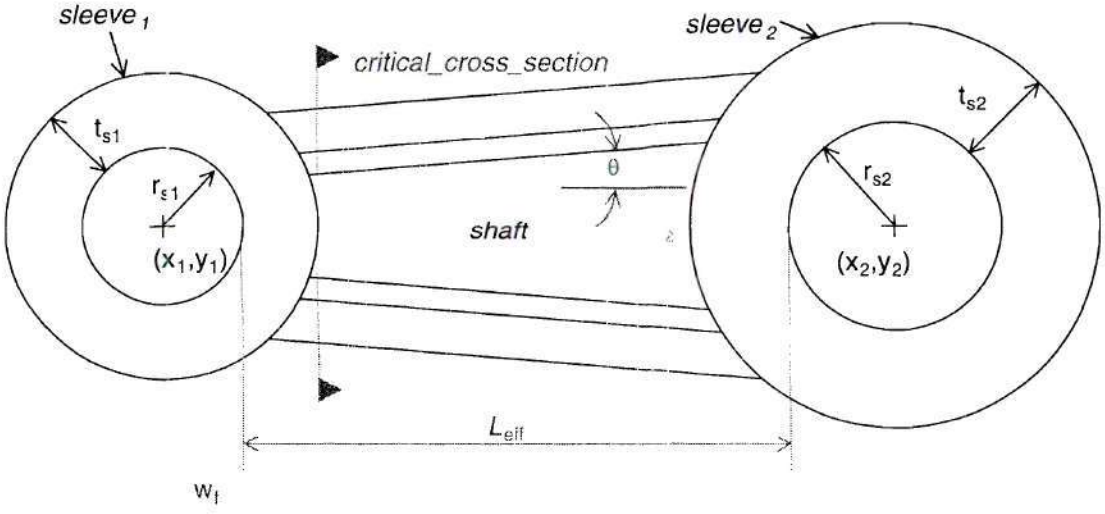


Figure 6-4 Flap Link Design Model (Detailed)

The early design stage of a flap link has been shown in Figure 6-5. This simple flap link design can be treated as either the design at an early stage or as an idealized view of the detailed flap link shown in the Figure 6-3.

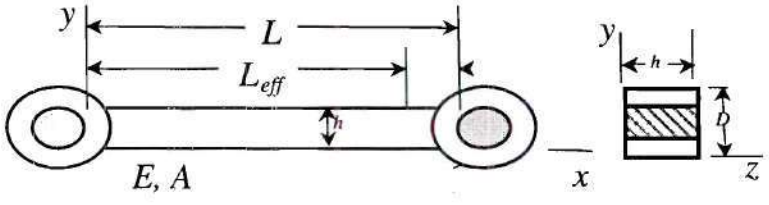


Figure 6-5 Flap Link Design Model (Preliminary Stage)

6.1.2 Engineering Optimization Model

For instance in the development of the flap link, the requirements on the product are such that product must be extremely light in weight. Meanwhile, the normal stress and buckling due to its loading condition must not exceed the allowable limits. Those limitations are the constraints imposed on the optimization model. In this case, the only parameter that is a variable is chosen to be the area of the link. In other words, a design variable is the area of the link. The engineering optimization model for preliminary flap link is shown below. The model is developed based on those requirements. Note that the engineering optimization model can be easily enhanced by adding more design variables in addition to area, or by adding more constraints and objectives even for this simple design, as requirements. The enhancement will be described in the next Chapter.

ENGINEERING OPTIMIZATION MODEL

Minimize

$f_1 = \rho LA$ Weight

Subject to

$g_1 = NC_{stress}(A) < 0$ *Normalized Stress Constraint*

$g_2 = NC_{buckling}(A) < 0$ *Normalized Buckling Constraint*

Design variables

$\mathbf{X} = \{A\}$

Figure 6-6 Engineering Optimization Model

Generally, engineers develop design optimization models with engineering entities. Those entities in the first cut are combined in the model. Note that the optimization model is built in terms of domain-specific notation in the engineering domain. It is convenient to partition the model, its optimization, analysis and design objects. In the model, the objective function f_1, the minimization of the weight of a prismatic link (See Figure 6-5), is subject to structural constraints g_1 and g_2 that require specific analysis

such as stress and buckling analysis. It is also required that the relevant constraints are to be satisfied. The stress and buckling analysis are treated as analysis objects that supply information constraint objects in the optimization model.

The parameters that describe the weight, density of material, length and area are the attributes of design belonging to the Product Model. Similarly, they supply information to the optimization and analysis models. The engineering optimization model is one of the deliverables of the EOM representation. The next step in the EOM model is the preparation of eom.cos with its link to a cbam.cos and apm.cos.

6.1.3 EOM Representations of Flap Link

The lexical forms are computed by the EOM Tool. It uses the EOM representation with its languages and protocols. The template corresponds to **engineering optimization model 1** shown below. It is an application of the lexical form defined in Chapter 4. Those lexical forms are loaded to the EOM Tool. The EOM Tool uses the objects defined in those forms and performs analysis and optimization.

6.1.3.1 EOM Lexical Forms:

Example: The EOM lexical forms of flap link optimization are given below.

```
EOM flap_optimization SUBTYPE_OF link_optimization_model;
DESCRIPTION
    "Represents minimization of weight of flap_link with respect to area"
PART_STRUCTURE
        link:flap_link_preliminary;
OBJECTIVES
        min
    obj: REAL;   /*WEIGHT OF FLAP LINK */
```

```
        max
        Null
CONSTRAINTS
        euquality_constraints = 0
        Null
        ineuquality_constraints < 0
      g1: REAL;  /*STRESS CONSTRAINT*/
      g2: REAL;  /*BUCKLING CONSTRAINT*/
DESIGN VARAIABLES
       area:bound_continuous;
 ANALYSIS MODELS
        cbam1a : cbams.link_extensional_model;  /*STRESS ANALYSIS 1D FORMULA*/
        cbam2 :  cbams.link_buckling_model;     /*BUCKLING ANALYSIS FORMULA*/
RELATIONS
 ASSOCIATIVITY_LINKAGE_CBAM
ala1 : "<area> ==<cbam2.deformation_model.area>";
ala2 : "<g1> == (1.0/(<cbam1b.sx_mos_model.margin_of_safety> + 1.0))-1";
ala3 : "<g2> ==(1.0/(<cbam2.buckling_mos_model.margin_of_safety> +1.0))-1";
ASSOCIATIVITY_LINKAGE_PM
ald1:"<obj1>==
(<area>*<cbam2.link.effective_length>)*<cbam2.link.material.density>";
END_EOM;
```

The **EOM flap_link_optimization model** has two analysis attributes named *cbam1b* and *cbam2*. Cbam1b has a link to *cbams.link_extensional_model* object. Cbam2 has a link to *cbams.link_buckling_model*. Below, only *cbams.link_extensional_model* has been shown with its template lexical form as an illustration. The analysis object has its own objects to define the analysis context. The **EOM flap_link_optimization model** also uses the cbams.link_buckling_model; it is not shown here (see Appendix).

CBAM Analysis Models:

```
CBAM link_extensional_model SUBTYPE_OF link_analysis_model;
 DESCRIPTION   "Represents 1D formula-based extensional model.";
 ANALYSIS_CONTEXT
  PART_FEATURE
   link : flap_link_preliminary;
  BOUNDARY_CONDITION_OBJECTS
```

```
  associated_condition : condition;
 MODE
  "tension";
 OBJECTIVES
  stress_mos_model : margin_of_safety_model;
ANALYSIS_SUBSYSTEMS
 deformation_model : extensional_rod_isothermal;
RELATIONS
 ASSOCIATIVITY_LINKAGE
 al1 : "<deformation_model.undeformed_length> == <link.effective_length>";
 al2 : "<deformation_model.area> == <link.shaft.critical_cross_section.basic.area>";
 al3 : "<deformation_model.material_model.youngs_modulus> ==
                  <link.material.stress_strain_model.linear_elastic.youngs_modulus>";
 al4 : "<deformation_model.material_model.name> == <link.material.name>";
 al5 : "<deformation_model.force> == <associated_condition.reaction>";
 al6 : "<stress_mos_model.allowable> == <link.material.yield_stress>";
al7:"<stress_mos_model.determined>==<deformation_model.material_model.stress>";
END_CBAM;
```

The **EOM flap_link_optimization model** has also a design (part_structure) attribute named *link*. Link has a linkage to the flap_link_preliminary object. Below, flap_link_preliminary has been shown with its template lexical form as an illustration. The design object has its own objects to define the design context in the analyzable product model. Cbam objects also utilize the same design object below.

Analyzable Product Model (APM):

```
APM flap_link_preliminary SUBTYPE_OF part;
  sleeve1 : sleeve;
  sleeve2 : sleeve;
  shaft : prismatic_shaft
END_APM

APM prismatic_shaft;
ESSENTIAL ATTRIBUTES
  l : REAL;
  w : REAL;
  t : REAL;
  area :REAL;
```

```
  volume : REAL;
  weight : REAL;
  IDEALIZED
  effective_length: REAL;
RELATIONS
 PRODUCT_RELATIONS
  pr1 : "<area> == <w>*<t>";
  pr2 : "<volume> == <l>*<w>*<t>";
  pr3 : "<weight> == <volume>*<material.density>";
IDEALIZED_RELATIONS
  pr1 : "<effective_lenth> == 1.3*<l>";
END_APM;
```

Above the flap_link_preliminary design object is represented as an analyzable product model in a lexical form similar to EXPRESS language. Below, the design representation has been explained. An EXPRESS-G Diagram of the flap_link_preliminary has been shown. It is possible to generate an EXPRESS_G Diagram from the EOM lexical forms.

In the context of this thesis, we define a design representation as a product data model tailored to analysis and optimization. The definition of product data is given below from ISO standard as "A representation of information about a product in a formal manner suitable for communication, interpretation, or processing by human beings or by computers" [IS0-10303-1:1994(E), 1994]. APM's represent detailed, design oriented product information. APM is considered to be the master description of a product that supplies information to other product life cycle tasks, including engineering optimization, analysis and manufacturing.

The Product Model represents geometry and material information that is part of a product and support idealization that can be used in different analysis and optimization models. An idealization in the APM context means a derived attribute of a design obtained by functional transformation of product attributes. Then, these attributes can be used in analysis and optimization models. Generally, these attributes are simple approximations to abstract the essence of the product as required by analysis and optimization, such as effective-length, which calculates the critical load in a buckling column. Basically, a product model is an information source to optimization and analysis models; that model contains information related to geometrical and material description of the product. Again, referring to Figure 6-5, the geometric description view of the information modeling language of product at early stages of design with its parameters was described as APM flap_link_preliminary. Those parameters will be determined from analysis and design considerations. Below, Figure 6.7 shows the product data of flap-link corresponding to the above in the Express-G diagram.

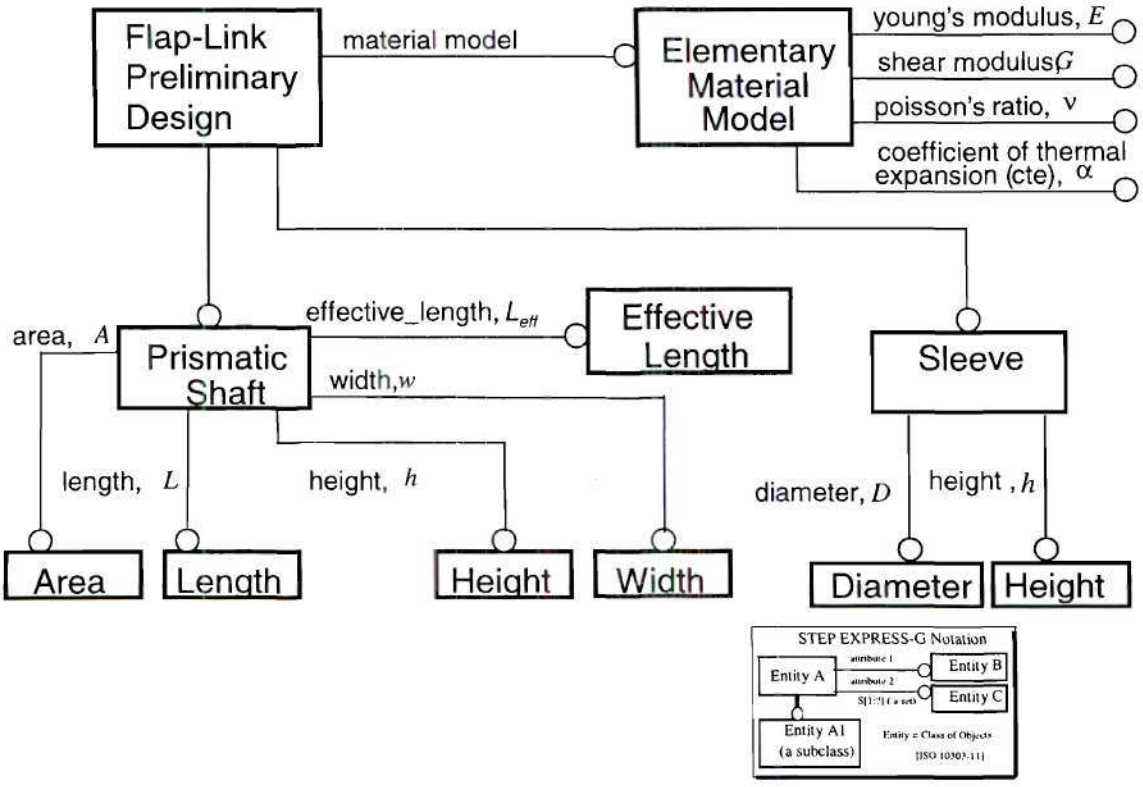


Figure 6-7 Express_G Diagram of Flap Link (Preliminary Design)

The Express-G diagram is an object relationship diagram used as a graphical model representation that emphasizes the is-a and part-of relations between objects used to represent product, analysis and optimization models as explained in Chapter 4.

As an illustration of how to interpret the diagram, the flap link object has a prismatic link and two sleeves connected to the end of the link. This prismatic link has the part's length, width, height, derived attribute area and approximated attribute effective length. Similarly, the sleeve has diameter and width. Link also has a material model with attributes including young's modulus, shear modulus, poisson's ratio and coefficient of thermal expansion. More detailed product data can be generated as design progress with more information content. Figure 6-4 shows such a more developed flap link at the

detailed stage of the design. Its apm lexical constrained object form and its Express_G graphical representation have been given in [Wilson, 2000]. Optimization and analysis models utilize product models. Multiple optimization and analysis models of varying fidelity utilize the product data models by retrieving and passing data. This enables modularity and reuse for enhancement of optimization models.

```
SCHEMA flap_link_apm;

SOURCE_SET flap_link_geometric_model ROOT_COB flap_link;

USE_FROM lib/apm.cos;

COB flap_link SUBTYPE_OF part;
  part_number : STRING;
  inter_axis_length, L<sub>a</sub> : REAL;
  IDEALIZED  effective_length, L<sub>eff</sub> : REAL;
  sleeve1 : sleeve;
  sleeve2 : sleeve;
  shaft : tapered_beam;
  rib1 : rib;
  rib2 : rib;
  allowable_twist : REAL;
  allowable_twist_factor : REAL;
  allowable_inter_axis_length_change_factor, C<sub>&phi;</sub> : REAL;
  allowable_inter_axis_length_change , C<sub>&Delta;La</sub> : REAL;
RELATIONS
 PRODUCT_RELATIONS
  pr1 : "<sleeve1.origin.y> == <origin.y>";
  pr2 : "<sleeve2.origin.y> == <sleeve1.origin.y> + <inter_axis_length>";
  pr3:"<rib1.height>= (<sleeve1.width> -
<shaft.critical_cross_section.design.web_thickness>)/2";
  pr4 : "<rib2.height> == (<sleeve2.width> -
<shaft.critical_cross_section.design.web_thickness>)/2";
  pr5 : "<rib1.thickness> == <shaft.critical_cross_section.design.web_thickness>";
  pr6 : "<rib2.thickness> == <shaft.critical_cross_section.design.web_thickness>";
 PRODUCT_IDEALIZATION_RELATIONS
  pir1 : "<effective_length> == <inter_axis_length> -
(<sleeve1.hole.cross_section.radius> + <sleeve2.hole.cross_section.radius>)";
  pir2 : "<shaft.critical_cross_section.design.total_height> ==
<sleeve1.outer_diameter>";
```

```
  pir3 : "<allowable_twist> == <allowable_twist_factor> * <effective_length>";
  pir4 : "<allowable_inter_axis_length_change> ==
<allowable_inter_axis_length_change_factor> * <effective_length>";
END_COB;

END_SOURCE_SET;

END_SCHEMA;
```

Figure 6-8 Flap Link COB Definition at Detailed Design Stage

6.1.3.2 Constraint Schematic Diagram of EOM of Flap Link Prototype

The Constraint Schematic is another graphical view of lexical form with easy to visualize object constraint concepts as explained in Chapter 4. The EOM relations are essentially constraint objects. Thus, mathematical operations are utilized among the primitive real attributes of EOM.

Figure 6-9 shows the defined relations of EOM in constraint schematic form for the prototype example, flap link optimization with its analysis and product objects.

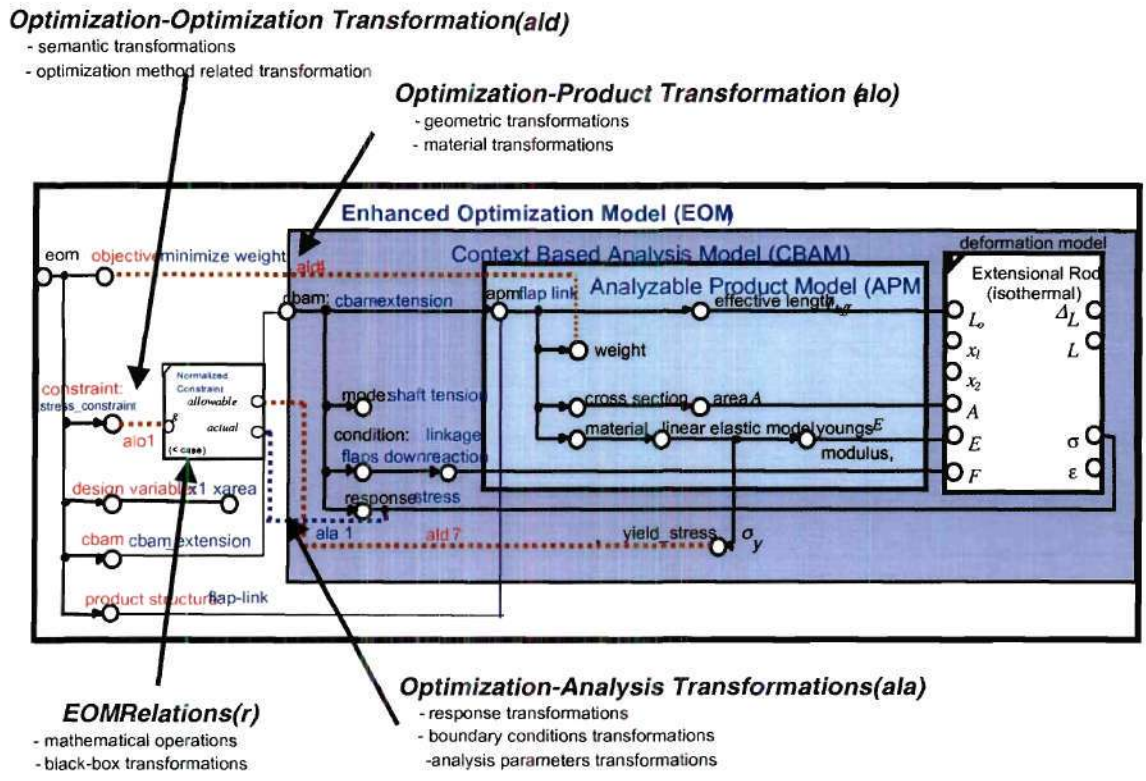


Figure 6- 9 Relations and Associativity for Flap Link for eom1

The EOM Model flap link optimization in Figure 6-9 has an optimization model that in turn has an objective (minimization of weight), a constraint (normal stress_constraint) and a design variable (area of the link). The other constraint (buckling constraint) described earlier has been omitted in this illustration. The EOM Model flap link optimization has also a cbam1 called *cbam_extension* and a product structure called *flap_link_preliminary.*

cbam_extension is an analysis object with its own attributes shown as the cbam block in blue color in Figure 6-9. Similarly, the ***flap_link_preliminary*** design object has been

shown in another block in turquoise color in Figure 6-9. Those objects are the information model ingredients extracted from the engineering optimization, analysis and design models.

The figure below shows a part of the constraint schematic where the product's area, length and density has been multiplied in a relation called r1 and linked to the objective attribute of eom.

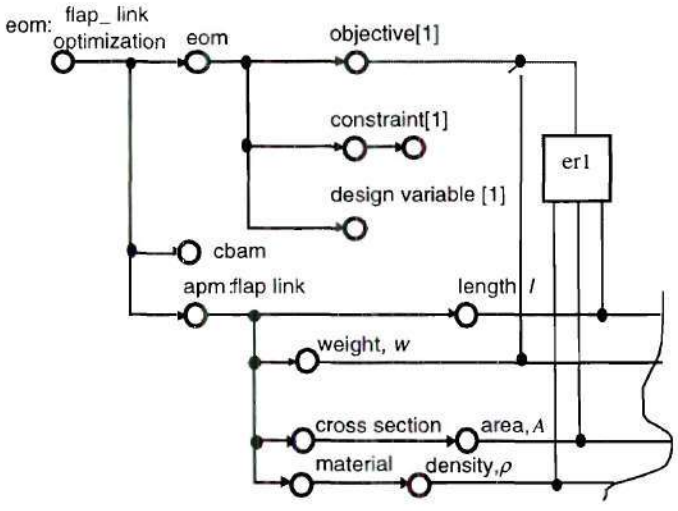


Figure 6-10 EOM Constraint Object Relation, er1

The optimization-design transformation (ald) links those design attributes to optimization model. One specific illustration of the associativity linkage to the design relation is shown in Figure 6-10 (Constraint Schematic Diagram). The **ald4** links the optimization attribute, objective to design attribute and product weight in the following manner.

ald4: oem.objective[1] == apm.link.weight

The above statement is written in the EOM of lexical information language form that is computable. This statement is read as ***fourth associativity linkage to design relation (ald4)*** of the enhanced optimization model linking EOM model to APM model. EOM has a list of objective attributes. The first attribute of that list, called *flaplinkweight,* is linked to analyzable product model (apm), which has an associativity to the product, in this case flap-link, which has a derived attribute weight. Weight is calculated from the geometrical attributes of link (area and length) and material attribute (density) in apm. Some cases are even calculated in eom as an eom relation (**er1**) as seen right of the relation. They are then linked to the objective attribute of eom shown as an option in the above figure.

ald4:oem.objective[1]==
apm.link.crosssection.area*apm.link.length*apm.link.material.density

Similarly, an ***optimization-analysis transformation (ala1)*** is shown in Figure 6-9. A more detailed view is provided in Figure 6-11 below. The ala1 associates the optimization attribute, constraint to analysis attribute (stress) in the following manner.

ala1: normalized_constraint.actual==cbam.deformation_model.stress

The statement above is read as ***first associativity linkage to analysis relation*** (ala1) of enhanced optimization model associates the EOM model to the CBAM model. The normalized_constraint's actual attribute is linked to the context-based analysis model (cbam) which is linked to the deformation-model analysis building block (abb), with attribute stress. This stress is calculated from the cbam attributes in abb. Then the stress

attribute (what is desired from analysis) is linked to both the response attribute of cbam and the actual attribute of normalized_constraint.

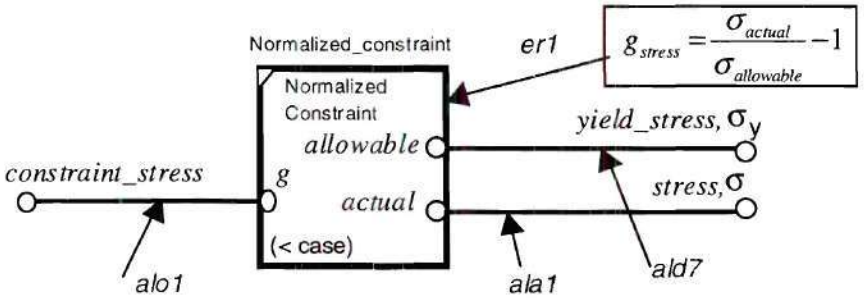


Figure 6-11 Associativity Links (ala, adl and alo, er1) for Flap Link in eom1

The normalized constraint block is an EOM object whose mathematical description is shown in the figure, with its links. Because this relation is commonly utilized, it has been packaged as a reusable **specialized object**. The ***EOM specialized object*** **(spob1)** of the normalized constraint is defined in lexical form below.

spob1: normalized_constraint.g ==
(normalized_constraint.actual/ normalized_constraint.allowable) -1

In Figure 6-11, another optimization-design transformation (ald7) is shown. It associates the allowable constraint object with the flap link materials' yield stress. Its lexical form is given as:

ald7:normalized_constraint.allowable=apm.material.linear_elastic_model.yield_stress

Finally, an optimization-optimizations transformation (**alo1**) is shown in Figure 6-9. The alo1 links the optimization attribute, constraint to another optimization attribute **g** of the normalized constraint. Basically, this link matches semantics related to the domain.

alo1: eom.constraint_ stress == g[1]

Where EOM has a list of constraint attributes, the first attribute of that list, called **constraint_stress**, is associated with generic **normalized_constraint of g.**

6.1.3.3 Context Based Analysis Models for EOM

When the flap link goes through compression, normal stress analysis and buckling analysis are required to assure that the system is safe. The analysis is performed for two modes, which are the behaviors to which the product is exposed. In the optimization model flap link analyzed against the extension and buckling modes. They may have different fidelity (1D, 2D, or 3D).

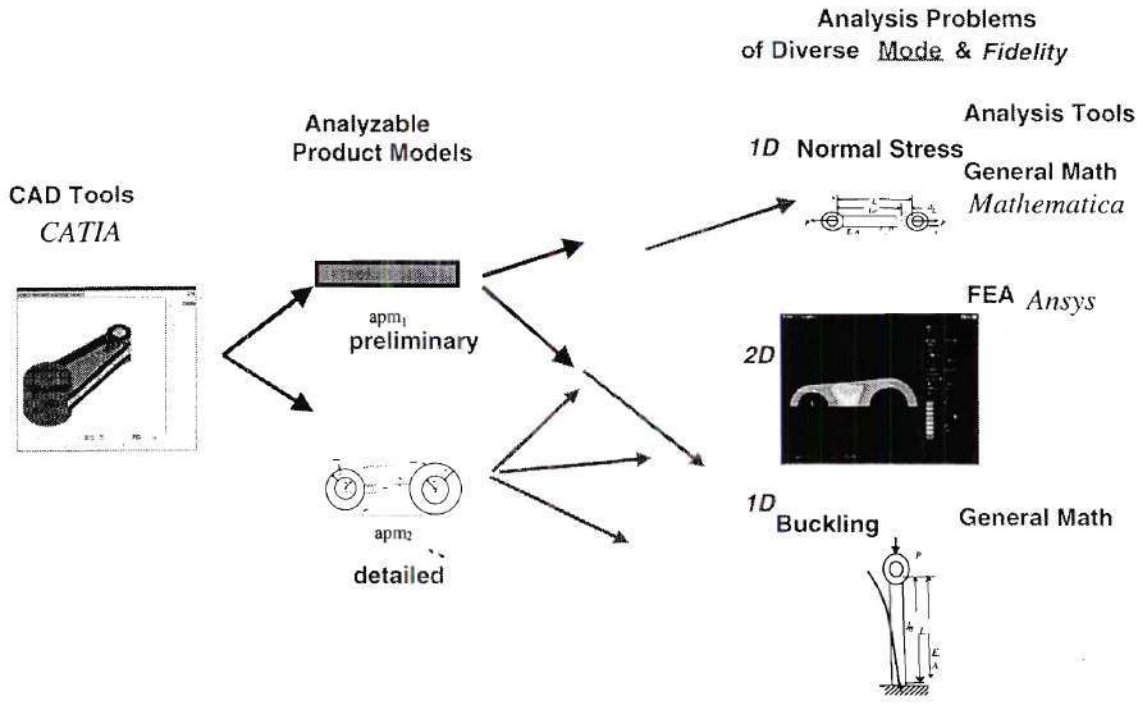


Figure 6-12 Fidelity in Analysis and Product Models

Below, the context based analyses used in EOM flap link optimization model are explained.

Normal_Stress Analysis

Here, two different fidelities of normal_stress of flap_link will be explained. The first one is a context based analysis for 1-Dimensional Normal Stress analysis, and the other one is a 2D Plain Stress Finite Element Analysis.

Their constraint schematics will be shown.

Cbam1a: 1D Normal Stress Analysis

Figure 6-13 shows *Extensional Rod Analysis* for the behavior of slender members due to axial loading. Axial loading can be both tension and compression.

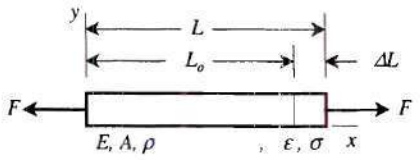


Figure 6-13 Extensional Rod Analysis Model

The relations between strain, ε, total elongation, ΔL, and these loads is given by the following equations.

$$\sigma = \frac{F}{A} \qquad \text{Equation 6-1}$$

$$\varepsilon = \frac{\Delta L}{L} \qquad \text{Equation 6-2}$$

$$\sigma = E\,\varepsilon \qquad \text{Equation 6-3}$$

$$\varepsilon = \frac{\sigma}{E} \qquad \text{Equation 6-4}$$

$$\Delta L = L - L_o \qquad \text{Equation 6-5}$$

Where σ is the normal stress, A is the cross sectional area, ε is the elastic strain due to force (F), L_0 is the length of the rod in the undeformed state, L is the length of the rod after elongation, ΔL is the change in the length of the rod. E and ρ are the material properties, Young modulus and density of the material respectively. Those basic relations are distributed into various Analysis Building Blocks based on their engineering meaning.

The Constraint Schematic in Figure 6-14 shows how the ABB *One D Linear Elastic Model* can be used to represent the ABB *Extensional Rod Model.*

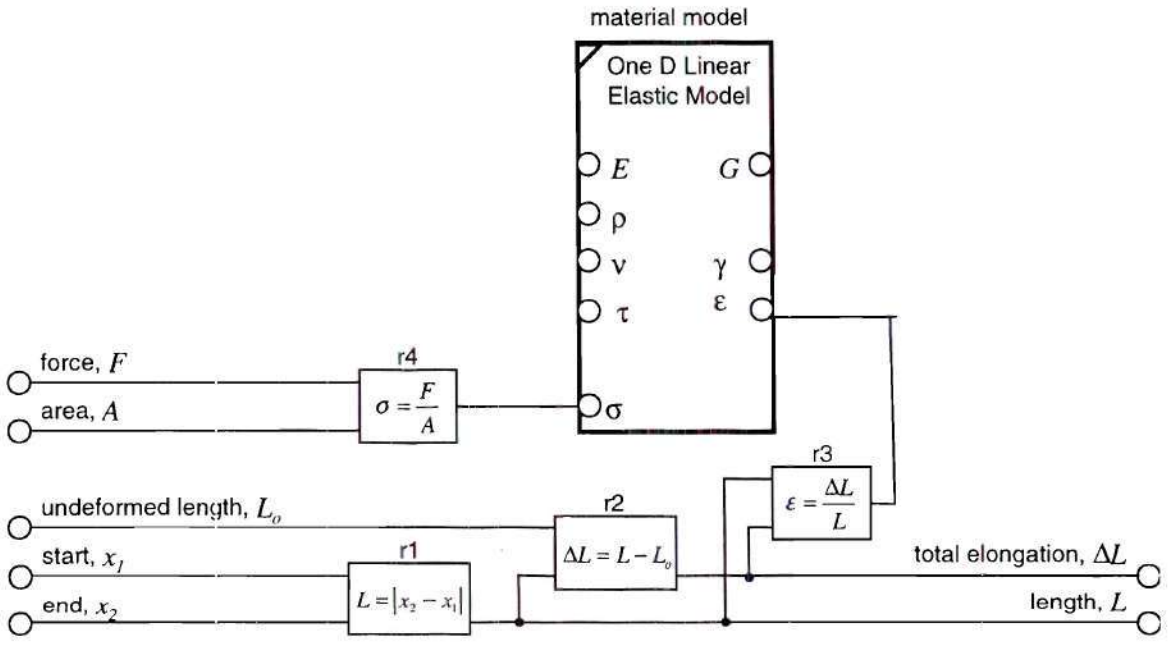


Figure 6- 14 Constraint Schematic for *Extensional Rod*

The COB Structure Definition language of the *Extensional Rod Model* is shown in Figure 6-15. It is actually a subtype of ABB *Slender Body* and inherits four attributes (undeformed_length, reference_temperature, temperature, and material_model) and one relation (material_model.temperature_change = temperature – reference_temperature). Equations 6-5, 6-2, and 6-1 defined above are described as relation r1, r2, and r3 respectively. The assumption is that *One D Linear Elastic Mode l* where there is no shear and no temperature change [Wilson, 2000].

```
CBAM slender_body SUBTYPE_OF deformable_body
   undeformed_length, L<sub>0</sub> : REAL
   reference_temperature, T<sub>0</sub> : REAL
   temperature, T : REAL;
   material_model : one_D_linear_elastic_model
 RELATIONS
   sbr1 : "<material_model.temperature_change> =
             <temperature> - <reference_temperature>"
END_CBAM;

CBAM extensional_rod SUBTYPE_OF slender_body
   start, x<sub>1</sub> : REAL,
   end, x<sub>2</sub> : REAL;
   length, L : REAL;
   total_elongation, &Delta;L : REAL
   force, F : REAL;
   area, A : REAL;
 RELATIONS
   r1 : "<length> == <end> - <start>"
   r2 : "<total_elongation> == <length> - <undeformed_length>"
   r3 : "<material_model.strain> == <total_elongation> / <undeformed_length>'
   r4 : "<material_model.stress> == <force> / <area>"
   mat_sc_r1: "<material_model.shear_stess> == 0"
 END_CBAM;
```

Figure 6-15 COB Structure Definition for Flap Link

Representations of linkage normal-stress analysis for flap link are shown in Figure 6-16. Generally, the analysis notes contain the boundary conditions (for example, force that applies to the flap link), idealized variables (such as effective length), and material property information along with the equations used for the analysis and the results showing the analysis response (normal-stress) in procedural format in engineering manuals. This information is transformed into constrained objects to construct the CBAM representation of the extensional analysis shown in 6-16. The CBAM representation illustrates design-related details of APM entities (i.e., linkage) on the left and a generic analysis system, ABB, (i.e., *Extensional Rod Analysis*) on the right. Between the APM and the ABBs, associativity relations indicate how the APM design entities are idealized as product-independent ABB. For example, Figure 6-16 explicitly

specifies that the area of Extensional Rod ABB equals the cross sectional area of the linkage shaft. The CBAM representation only illustrates APM variables that are associated with ABBs.

The CBAM Representation explicitly relates those product-independent analysis variables defined in ABB back to detailed design parameters defined in APM. Once the analysis result is obtained, it may be compared with an allowable limit (for example, material yield strength with computed maximum stress) to determine the normalized constraint used in the EOM.

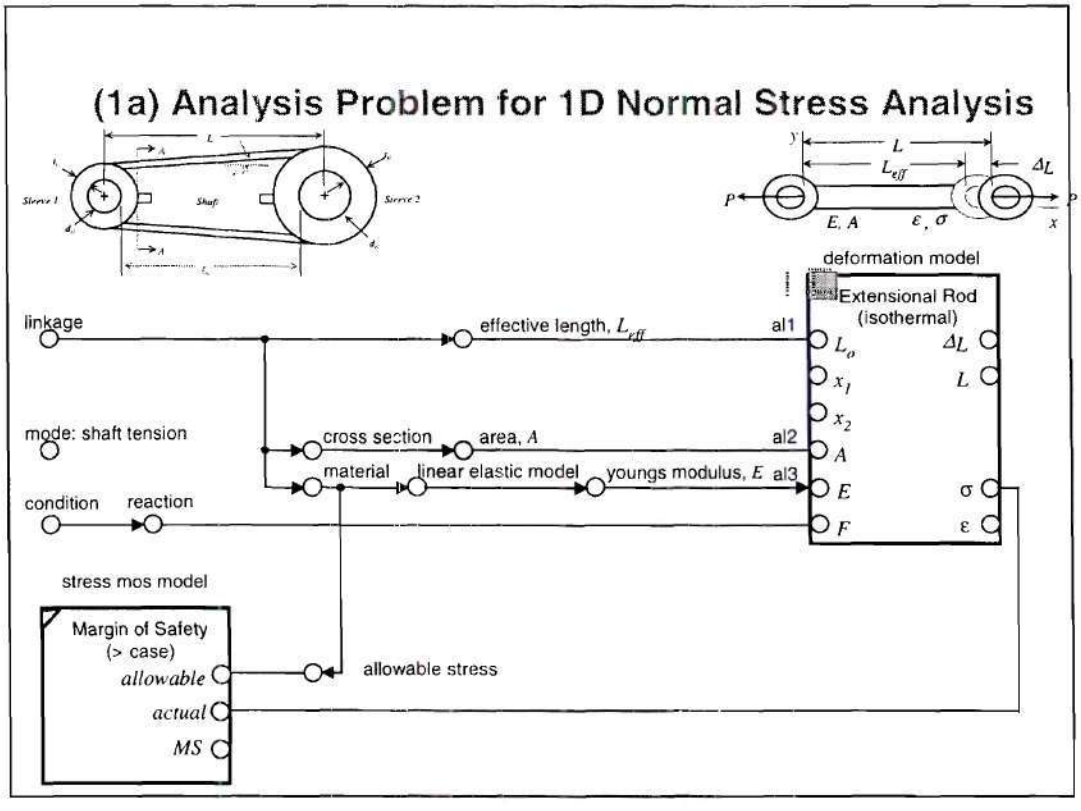


Figure 6-16 Representing a Flap Link Analysis as CBAM: Linkage Extensional Model

Figure 6-17 is the CBAM Definition for the Flap Link Extensional Analysis Model. For example, the link attribute has flap_link and the deformation_model attribute has extensional_rod_isothermal. That attribute captures the basic governing equations 6-1 to 6-5 as seen in Figure 6-14. Next, the deformation_model attribute has been replaced with 2D FE Plain Strain Analysis.

```
CBAM link_extensional_model SUBTYPE_OF link_analysis_model;
  DESCRIPTION
    "Represents 1D formula-based extensional model.";
  ANALYSIS_CONTEXT
    PART_FEATURE
     link : flap_link
    BOUNDARY_CONDITION_OBJECTS
     associated_condition : condition;
    MODE
     "tension";
    OBJECTIVES
     stress_mos_model : margin_of_safety_model;
  ANALYSIS_SUBSYSTEMS
    deformation_model : extensional_rod_isothermal;
 RELATIONS
   a11 : "<deformation_model. undeformed_length> == <link.effective_length>";
   a12 : "<deformation_model.area> == <link.shaft.critical_cross_section.basic.area>";
   a13 : "<deformation_model.material_model. youngs_modulus> ==
            <link.material.stress_strain_model.linear_elastic. youngs_modulus>";

   a14 : "<deformation_model.material_model.name> == <link.material.name>";
   a15 : "<deformation_model.force> == <associated_condition.reaction>";

   a16 : "<stress_mos_model.allowable> == <link.material.yield_stress>";
   a17 : "<stress_mos_model.determined> == <deformation_model.material_model.stress>";
END_CBAM;
```

Figure 6-17 Definition Language for Linkage Extensional Model CBAM

Cbam1b: 2D FE Plain Stress Analysis

Instead of using a 1-Dimensional formula based analysis model, a 2-Dimensional finite element analysis model called *linkage plane stress ABB* is used to determine the

extensional behavior of the flap link. The higher fidelity of the Link Extensional Analysis is shown in Figure 6-18.

The CBAM wraps the FEA analysis tool within a constraint framework with the CobExternalToolFunction relation that creates a parameterized FEA model, performs FEA analysis, and returns results. Figure 6-19, shows the CBAM Definition for the ABB. In this case study, ANSYS is used as the FEA tool. The results returned are maximum stress and maximum deformation in x-direction. With the results, the margins of safety for stress and deformation are determined. The linkage APM and margin of safety ABB are the same objects used in the 1-Dimensional extensional analysis to demonstrate the reusability of the same constrained objects for a variety of analyses.

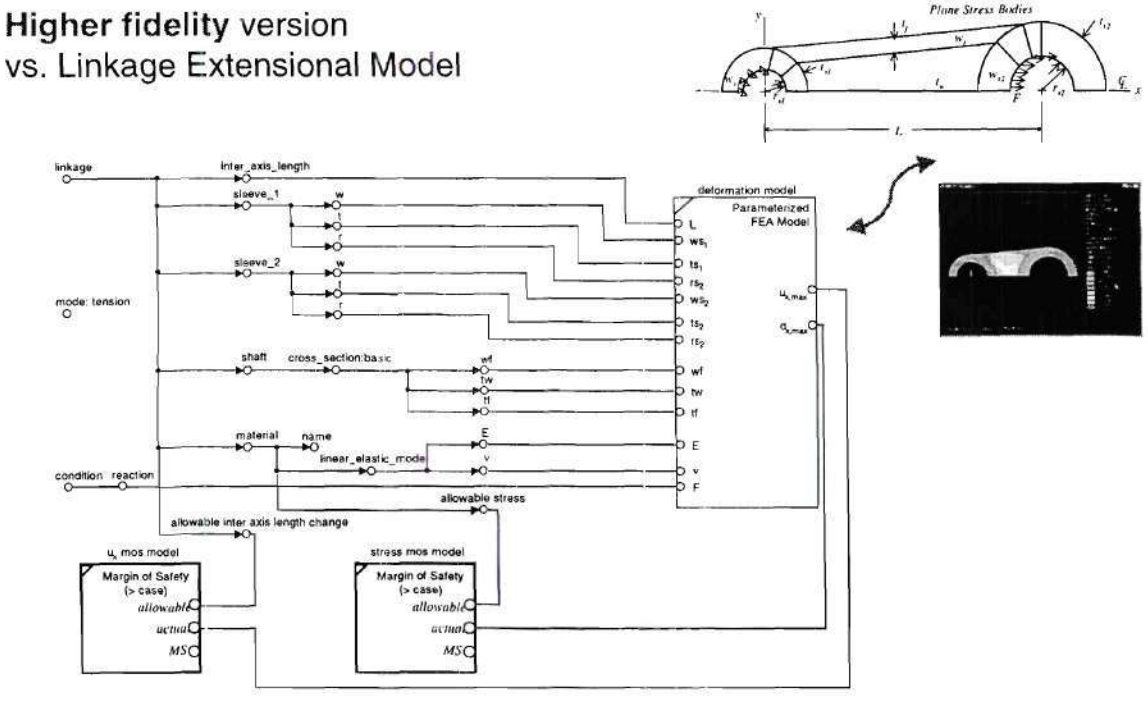


Figure 6-18 Higher Fidelity Flap Link CBAM: Linkage Plane Stress Model

```
COB link_plane_stress_abb;
      l: REAL;     /* length */
      ws1: REAL;   /* width at sleeve1 */
      ts1: REAL;   /* thickness at sleeve1 */
      rs1: REAL;   /* radius Of sleeve1 (rs1<rs2) */
      ws2: REAL;   /* widthth at sleeve2 */
      ts2: REAL;   /* thickness of sleeve2 */
      rs2: REAL;   /* radius of sleeve2 */
      tw: REAL;    /* thickness at shaft_web */
      tf: REAL;    /* thickness of shaft */
      wf: REAL;    /* thickness at shaft_flange */
      force: REAL;       /* total force */
      ex: REAL;    /* youngs modulus */
      nuxy: REAL;  /* poisson's ration */
      ux: REAL;    /* deformation in x dir */
      sx: REAL;    /* stress in x dir */
RELATIONS
   r1: "<ux> == CobExternalToolFunction[ansys,flaplink,ux,
            {<ex>,<nuxy>,<l>,<ws1>,<ws2>,<rs1>,<rs2>,<ts1>,
            <ts2>,<tw>,<tf>,<wf>,<force>}]" ONEWAY;
   r2: "<sx> == CobExternalToolFunction[ansys,flaplink,sx,
            {<ex>,<nuxy>,<l>,<ws1>,<ws2>,<rs1>,<rs2>,<ts1>,
            <ts2>,<tw>,<tf>,<wf>,<force>}]" ONEWAY;
END_COB;
```

Figure 6-19 COS - link plane stress ABB

Buckling Analysis

The previous two analyses are to observe the extensional behavior. Here the flap link is analyzed for the behavior due to buckling. The deformation model used is the ABB buckling rod. The same APM linkage and ABB margin of safety models are used in this model as in the previous analysis models.

cbam 2: Buckling Analysis

Buckling Rod Analysis, describing the behavior of a slender member subject to buckling loading, is shown in Figure 6-20.

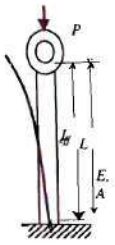


Figure 6-20 Buckling Rod Model

Assuming linearly elastic material behavior, the following equations describe the *Buckling Rod* behavior.

$$I = \frac{wt^3}{12} \qquad \text{Equation 6-6}$$

$$L_e = K.L \qquad \text{Equation 6-7}$$

$$P_{cr} = \frac{\pi^2 EI}{L_e} \qquad \text{Equation 6-8}$$

A is the area of prismatic shaft with dimension width, w and thickness, t. I is the moment of inertia of the prismatic link. L_e is the effective length based on boundary condition. K is the boundary coefficient factor. P_{cr} is the critical load for buckling.

Similar to the *Extensional Rod*, the Constraint Schematic of *Buckling Rod* can be represented with the ABB *One D Linear Elastic Model* as shown in Figure 6-21.

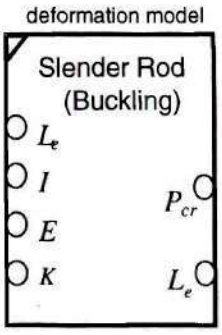


Figure 6-21 ABB for *Buckling Rod*

The *Buckling Rod* is the subtype of ABB *Slender Body*. Since the ABB *Slender Body* is a subtype of ABB *Extensional Rod*, the *Buckling Rod* inherits all attributes and relations from both of the ABBs. The Equations 6-6 to 6-8 defined above are described as relation r1, r2, and r3 respectively.

6.1.3.4 EOM Usage of CBAMs and APMs

The EOM utilizes both apm and cbam objects. In Figure 6-22, below, the eom flap_link optimization model utilizes the flap_link design object. An analyzable product model of flap_link_preliminary design object has been shown in earlier sections. The previous section describes the analysis objects, cbams used in eom. Those cbams are cbam1a (1D Normal_Stress_Analysis) and cbam1b (2D Plain Strain FE Analysis). They represent two different fidelities of normal stress determination which achieve the optimization constraint calculation in EOM. The other analysis is buckling, explained as cbam2. Figure 6-22 shows that eom utilizes cbam1a and cbam2 for different behavior.

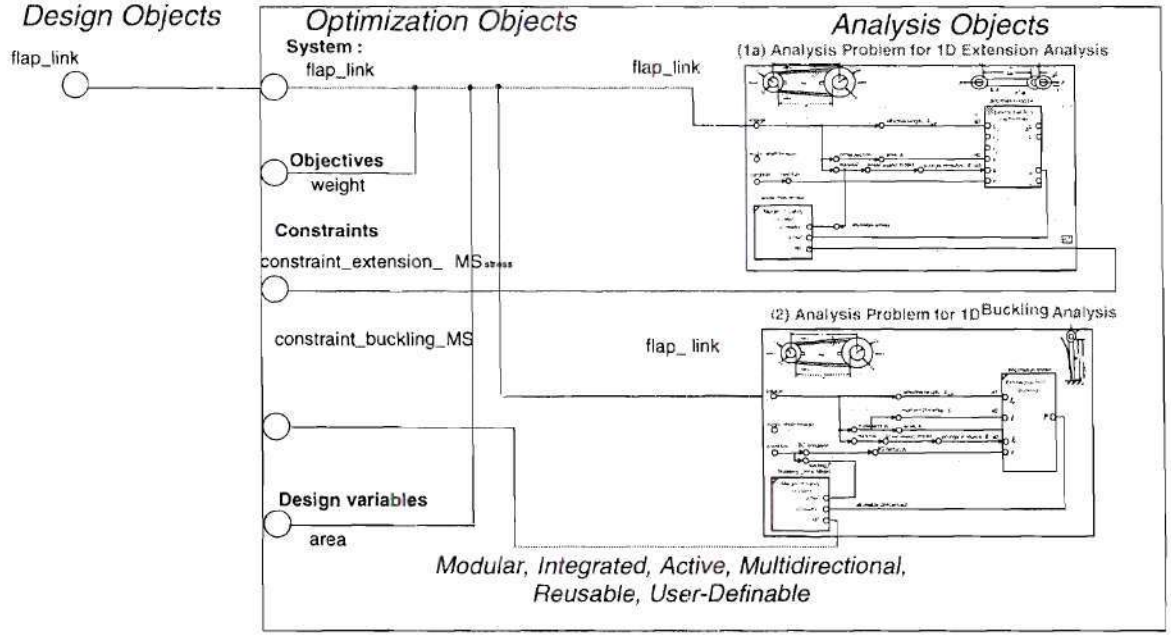


Figure 6-22 EOM Model for Flap Link

6.1.4 Enhancement of EOM

Figure 6-23 shows that the enhancement of different fidelity of analysis and optimization models has been developed, executed and results obtained. It is possible to develop a variety of optimization models, depending on the requirements.

In Figure 6-23, several optimization models have been developed based on cbam1a: 1D Normal Stress Analysis and a given product model apm1a: flap_link_preliminary. Another class of optimization models has been developed based on cbam1b: 2D Plain Stress FE Analysis and a given product model apm1a: flap_link_detailed as shown at the bottom of the figure.

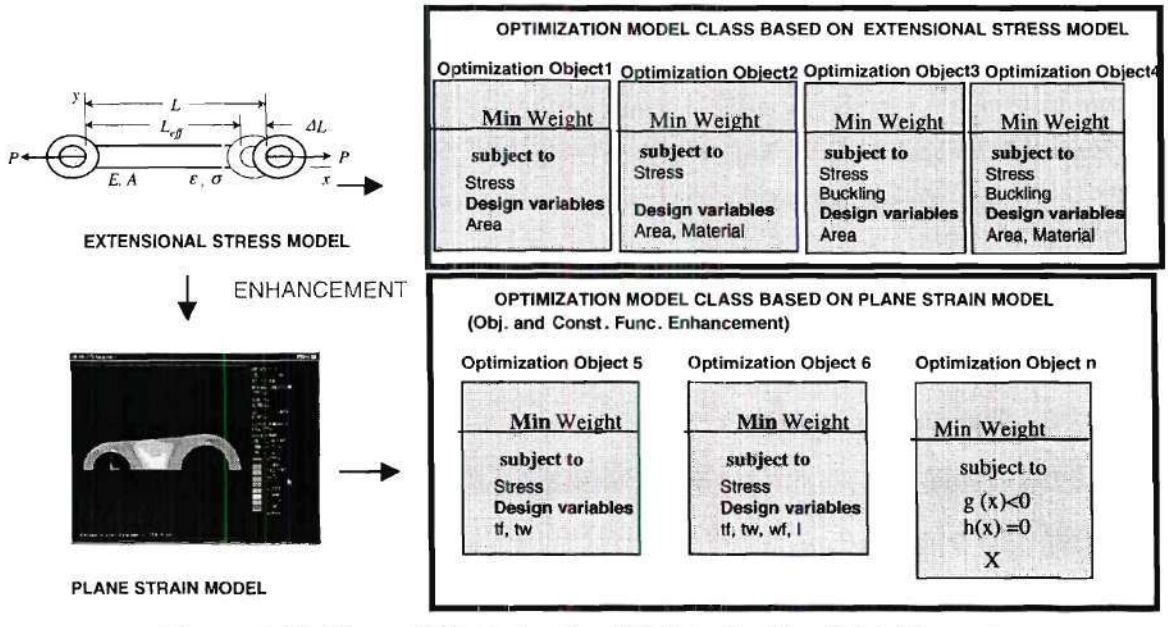


Figure 6-23 Class of Optimization Models for Flap Link Example

Enhancement can be performed several ways, such as adding or subtracting more objectives, constraints and design variables. In this prototype example, the following engineering optimization models are developed and run for the first class of optimization models based on cbam1 a and apm1a.

6.1.4.1 Enhanced Optimization Models based on flap_link_detailed design and 2D FE Analysis

The first model has one objective (minimization of weight of the flap link), one constraint (normal_stress-constraint) and one continuous design variable (the area of the flap link). This model already has been discussed to show different aspects of the EOM representation and how to integrate the different pieces of design and analysis objects into eom representation.

OPTIMIZATION MODEL I

Minimize

$f_1 = \rho LA$ Weight

Subject to

$g_1 = MS_{stress}(A) \geq 0$ *Normal Stress Margin of Safety*

Design variables

X={A}

OPTIMIZATION MODEL II

Minimize

$f_1 = \rho LA$ Weight

Subject to

$g_1 = MS_{stress}(A) \geq 0$ *Normal Stress Margin of Safety*

Design variables

X={A, material}

Figure 6 -24 Engineering Optimization Models I and II

The second engineering optimization model has one objective (minimization of weight of the flap link), one constraint (normal_stress-constraint) and two design variables. One design variable is a continuous design variable (the area of the flap link) and the other one is discrete (the material of flap link among a list of material).

The list of material that flap link can be made of with its properties is shown in Table 6-1 below. The optimization algorithm will search the best combination of material and area due to its objective function.

MATERIAL	SPECIFIC WEIGHT SPE	YOUNG MODULUS OF ELASTICITY E	MAX ALLOWABLE YIELD STRESS SIGMA
STRUCTURE STEEL ASTM -A-36	0.284	29000	36000
STAINLESS STEEL ANNEALED AISI 302	0.284	28000	38000
ALUMINUM ALLOY 2014- T6	0.103	10600	60000
TITANIUM	0.162	16500	120000

Table 6-1 Material Table for Flap Link

Below, the Constraint Schematic Diagrams correspond to Engineering Optimization Models I and II. They have been shown respectively. The only design variable is the area in Figure 6-25. The area is chosen between its allowable upper and lower limits in such a way that the weight of the flap link will be minimized.

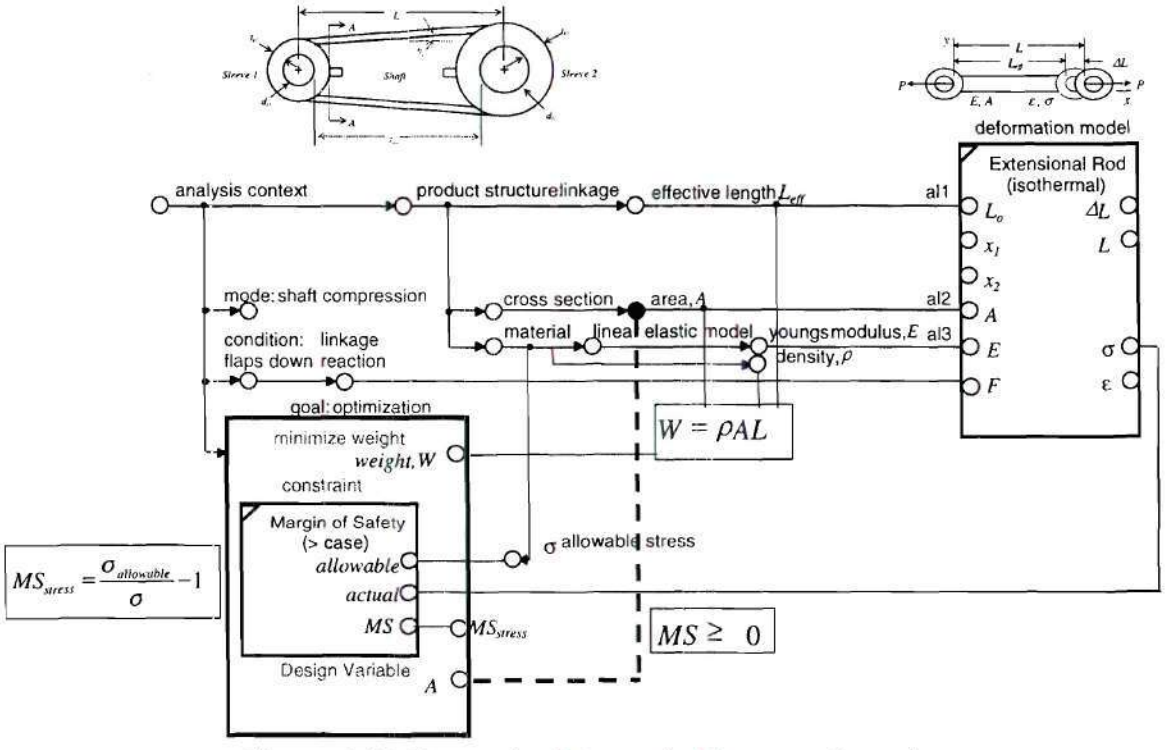


Figure 6-25 Constraint Schematic Diagram of eom1

The parameters that are the design variables are the area and the material of flap link in the Figure 6-25. Simultaneously, the area is determined between its allowable upper and lower limits, and the material is selected from the given list such a way that the weight of the flap link will be the minimum. When a particular material is selected by the optimization, its material properties will be attained from the corresponding values. In this case, the Young's modulus of elasticity and density are affected by the selection.

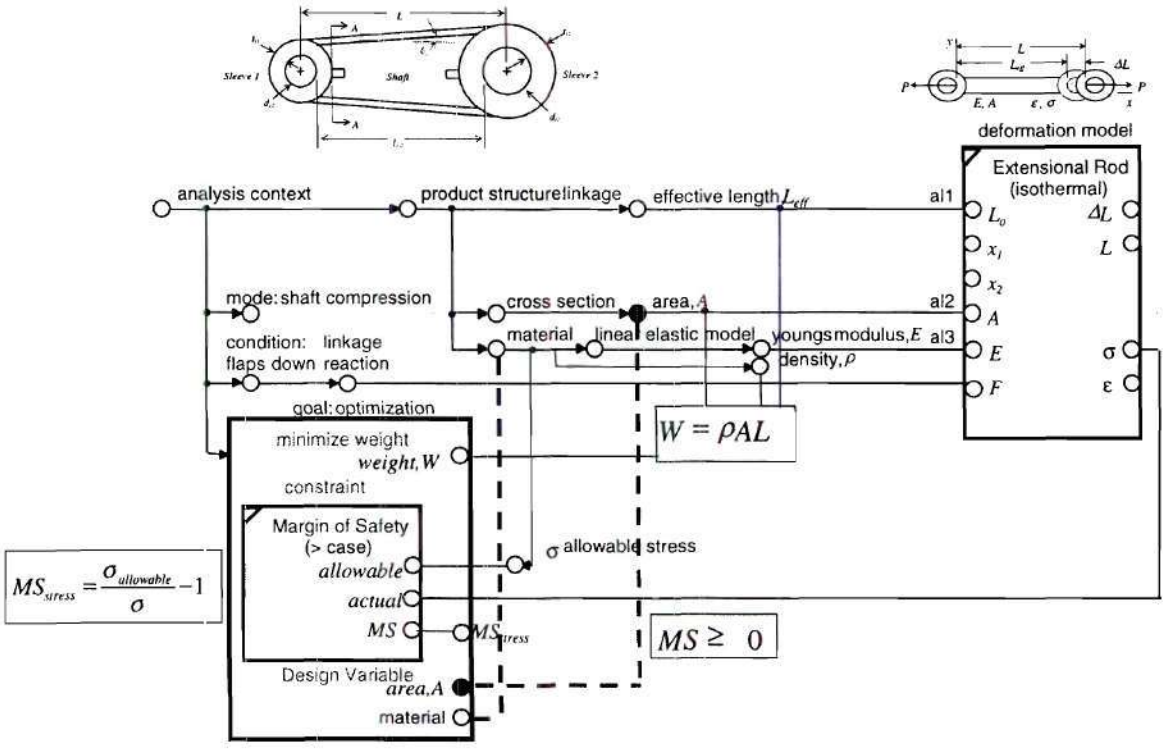


Figure 6-26 Constraint Schematic Diagram of eom2

The third engineering optimization model (Model III) has again one objective (minimization of weight of the flap link), one design variable and two constraints (normal_stress_constraint) and (buckling_stress). The design variable is a continuous design variable (the area of the flap link). The optimization model is very similar to the Model I. It shows the enhancement of constraints. Engineering Optimization Model IV extends Model III by adding additional constraint. Another point is that having only continuous design variables and constraint in the optimization model leads to choosing the continuous, constraint optimization methods and tools as Model I and Model III. On the other hand, having both continuous and discrete design variables as well as constraints in the optimization model leads to choosing the mixed optimization methods.

This information is captured in the eom model. The EOM Tool with eom protocol chooses the proper method and tool as described in the implementation section.

OPTIMIZATION MODEL III

Minimize

$f_1 = \rho LA$ Weight

Subject to

$g_1 = MS_{stress}(A) \geq 0$ *Normal Stress Margin of Safety*

$g_2 = MS_{buckling}(A) \geq 0$ *Buckling Margin of Safety*

Design variables

X={A}

OPTIMIZATION MODEL IV

Minimize

$f_1 = \rho LA$ Weight

Subject to

$g_1 = MS_{stress}(A) \geq 0$ *Normal Stress Margin of Safety*

$g_2 = MS_{buckling}(A) \geq 0$ *Buckling Margin of Safety*

Design variables

X={A, material}

Figure 6-27 Engineering Optimization Models III and IV

Again, the Constraint Schematic Diagrams correspond to Engineering Optimization Models III and IV. They have been shown respectively. Note that the models are obtained by adding another constraint (buckling_constraint) to Engineering Optimization Models III and IV.

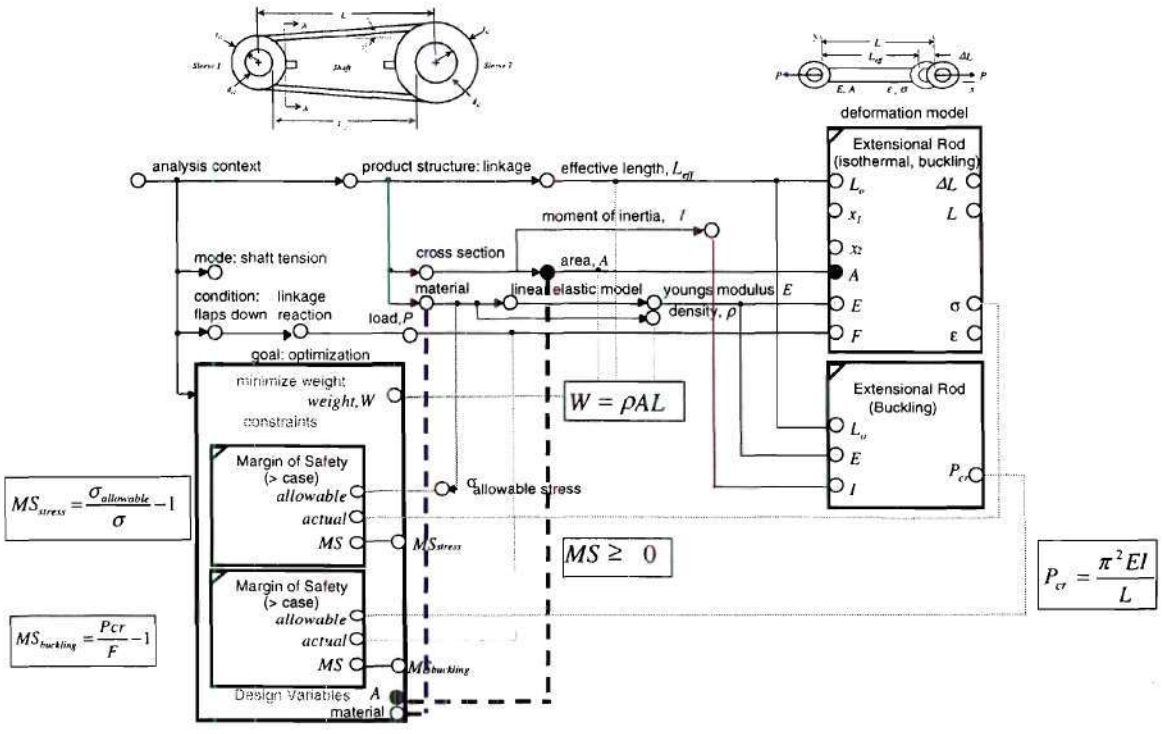


Figure 6-28 Constraint Schematic Diagram of Eom4

The results for eom1 are shown below. The initial value for area was 1.00 in2. The lower limit is 0.1 and the upper limit is 3.00 in^2.

```
**************************************************
* THE LAST INFORMATION =OPTIMAL DESIGN   *
*         POINT                          *
**************************************************
DV(1)=    2.778000000000000E-001

The OBJECTIVE value at this point is:
OBJ=         1.341200000000000
 The corresponding CONSTRAINT values are:
G(1)= -3.480000000000000E-002
weight=      1.341200000000000
```

6.1.4.2 Enhanced Optimization Models based on flap_link_detailed design and 2D FE Analysis

The second group of optimization problems is developed for flap_link_detailed design. That design has a shaft with a simple I beam cross-section as seen in Figure 6-29. The design variables are selected from the dimensions of the I beam and the length of the shaft.

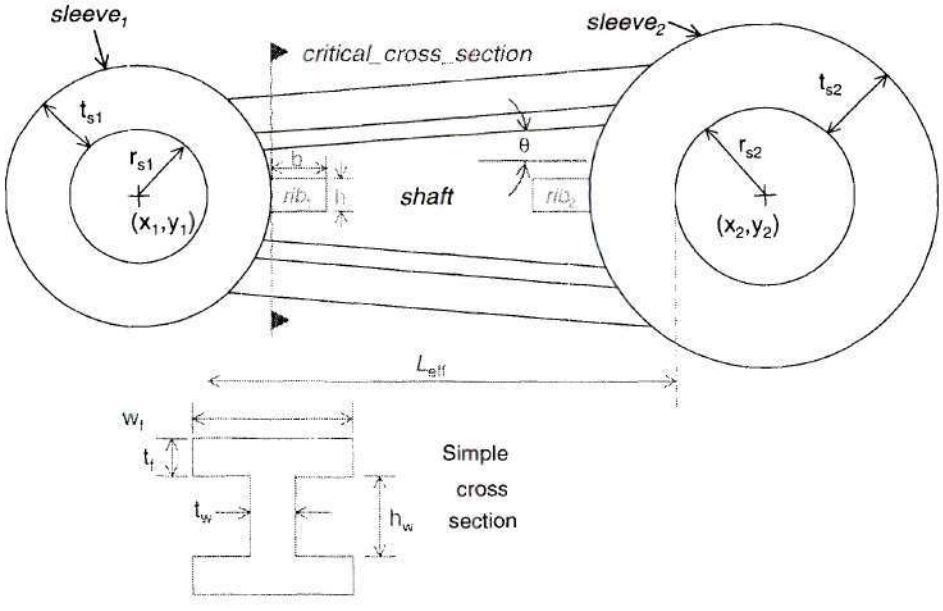


Figure 6-29 Detailed Design of Flap Link

In this case, Ansys FE analysis is performed on the system due to compression pressure inside the sleeves instead of 1D Normal Compression analysis. Still, the desired response from the analysis is the stress value on the flap link. The main difference is that the fidelities of the product model and the analysis have been increased to accommodate the requirements.

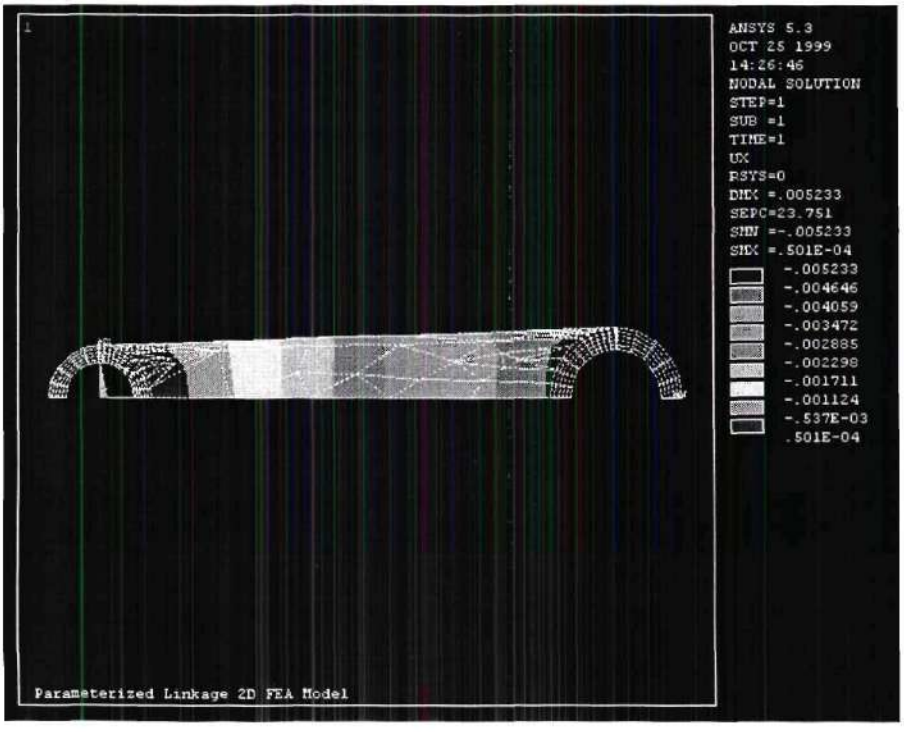


Figure 6-30 Flap Link FE Analysis

Several optimization models have been developed based on this flap_link_detailed and FE Element analysis for stress calculation. Among those engineering optimization models, two (Figures 6-31 and 6-34) have been illustrated to show the enhancements obtained by adding design variables.

Engineering Optimization Model V

Minimize Weight (Volume)

wt = (2*tf*wf+tw*((rs1+ts1+rs2+ts2)/2-tf))*l * rho

Subject to

-30000 < Stress < 30000 psi

Design variables

0.1 < tf <0.25 in

0.1 < tw <1.0 in

Figure 6-31 Engineering Optimization Model V

The results for Engineering Optimization Model V are shown below with the iteration history of objective (Figure 6-32) and design variables (Figure 6-33).

FLAP LINK DETAILED PARAMETERIZED TW AND TF

Optimization

parameter	min	max	toler
SXMAX	.800000000E-30	.360000000E+05	.360000000E+03
SXMIN	-.360000000E+05	.360000000E+05	.720000000E+03
TF	.100000000E+00	.250000000E+00	.150000000E-02
TW	.100000000E+00	.100000000E+01	.900000000E-02
WT	.000000000E+00	.000000000E+00	.200000000E+01

current parameter names and values

BASINC -2359.04695 BB 2359.04695 DEFXMAX .129525671E-03DEL .100000000E-05ES1_1 10.0000000

ES1_2 7.00000000 ES1_3 8.00000000 ES1_XC 3.00000000 ES2_3 2.00000000 ES2_4 6.00000000

ES2_XC 2.00000000 E_LINK 15.0000000 FORCE -10000.0000 L 15.7500000

LOCX1TF -.428571429E-01

LOCX2TF 15.6942857 LOCY1TF 1.49938763 LOCY2TF 1.94920392 LOCYKP15 .000000000

PHI .285753173E-01

RS1 1.00000000 RS2 1.35000000 SXMAX 26661.5531 SXMIN -29724.6993 **TF 103244973**

TS1 .500000000 TS2 .600000000 **TW .103604164** WF 1.50000000 WS1 2.00000000

WS2 2.50000000 **WT 7.52464901**

objective design variables

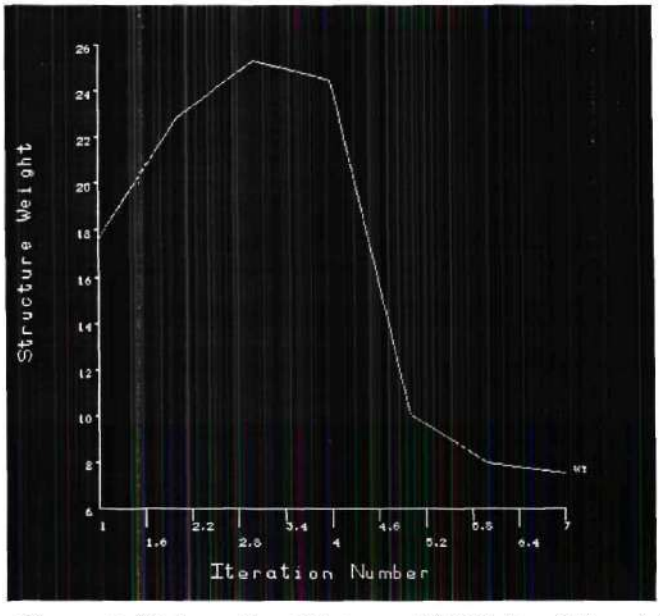


Figure 6-32 Iteration History of Weight of Flap Link

Below, two design variables (tf and tw) are shown during the optimization cycle.

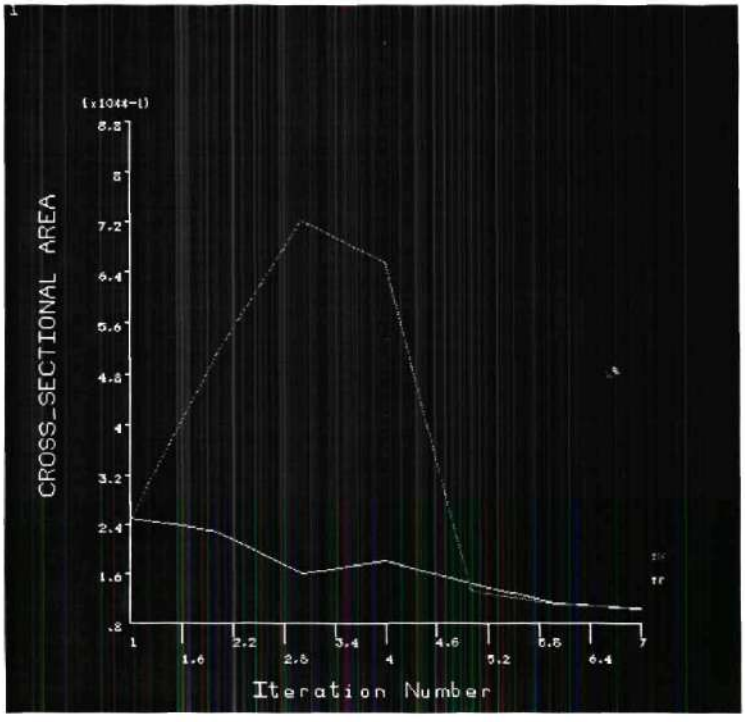


Figure 6-33 Flap Link Detailed-Two Design Variables (tw and tf)

Below, engineering optimization model VI has been derived from V by adding two more design variables (effective length, l and flange width, wf).

Engineering Optimization Model VI

Minimize Weight

wt = (2*tf*wf+tw*((rs1+ts1+rs2+ts2)/2-tf))*l*rho

Subject to

-30000 < Stress < 30000 psi

Design variables

0.1 < tf <0.25 in

0.1 < tw <1.0 in

0.1 < wf <2.0 in

12.0 < l <17.0 in

Figure 6-34 Engineering Optimization Model VI

The final results for a four-design-variable case have been shown below. Two more parameters from the I cross-section beam are added to the optimization model in the case below. The objective function is the weight of the shaft again.

Parameterized Linkage 2D FEA Model

optimization variables

parameter	min	max	toler
SXMAX	.800000000E-30	.300000000E+05	.300000000E+03
SXMIN	-.300000000E+05	.250000000E+05	.550000000E+03
TF	.100000000E+00	.250000000E+00	.150000000E-02
TW	.100000000E+00	.100000000E+01	.900000000E-02
WF	.100000000E+00	.200000000E+01	.190000000E-01
L	.120000000E+02	.170000000E+02	.500000000E-01
WT	.000000000E+00	.000000000E+00	.200000000E+01

current parameter names and values

BASINC	-2359.04695	BB	2359.04695	DEFXMAX	.717417607E-04	DEL	
.100000000E-05	ES1_1	10.0000000					
ES1_2	7.00000000	ES1_3	8.00000000	ES1_XC	3.00000000	ES2_3	2.0000000
ES2_4	6.00000000						
ES2_XC	2.00000000	E_LINK	15.0000000	FORCE	-10000.0000	**L**	**12.358263**
LOCX1TF	-.546193244E-01						
LOCX2TF	12.2872583	LOCY1TF	1.49900525	LOCY2TF	1.94870682	LOCYKP15	.00000000
NUM_N	18.0000000						
PHI	.364209344E-01	RS1	1.00000000	RS2	1.35000000	SXMAX	13823.206
SXMIN	-26470.8767						
TF	**.189284305**	TS1	.500000000	TS2	.600000000	**TW**	**.17865704**
WF	**.574167429**						
WS1	2.00000000	WS2	2.50000000	**WT**	**6.07690647**		

objective

design variables

Figure 6-35 Results -Optimization Model IV

As with the earlier case, the iteration histories of objective (weight of the link) for the 4-Design Variables case are shown in Figure 6-36 and 6-37.

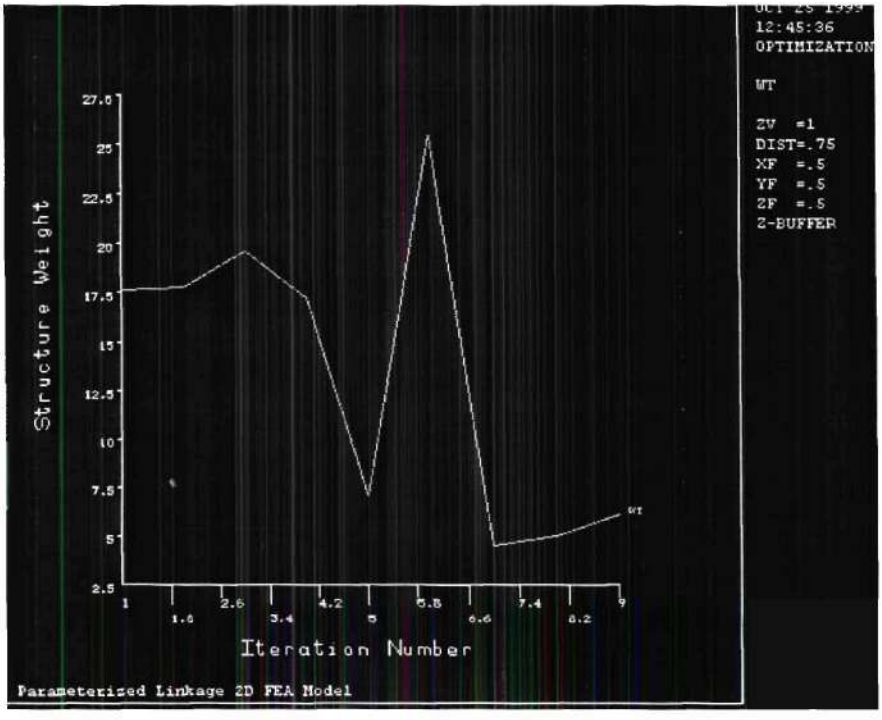


Figure 6-36 Objective Function Weight Iteration History

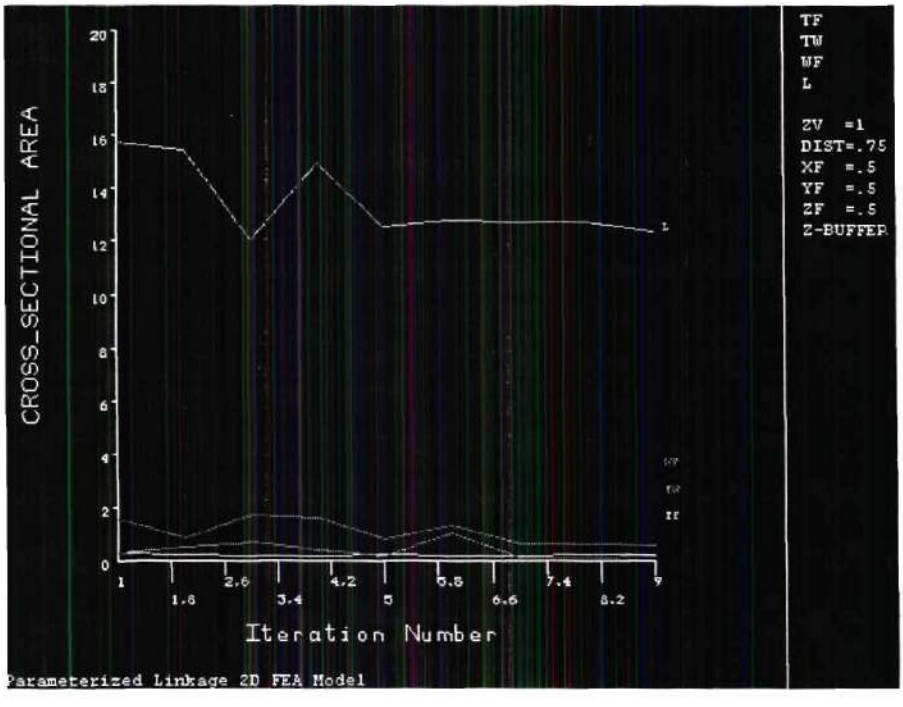


Figure 6-37 Flap Link Detailed Design (4 Design Variables)

6.2 Structural Optimization: 10-Bar Truss

6.2.1 Physical Description

The ten-bar truss is used as a test case. The ten-bar truss is a standard example used by many authors. The minimum weight design obtained by changing the cross-sectional areas of the truss members is sought, subject to stress constraints [Haftka, 1990].

The main emphasis in this example is to compare the results with the published data. The ten-bar configuration is shown in Figure 6-38 below.

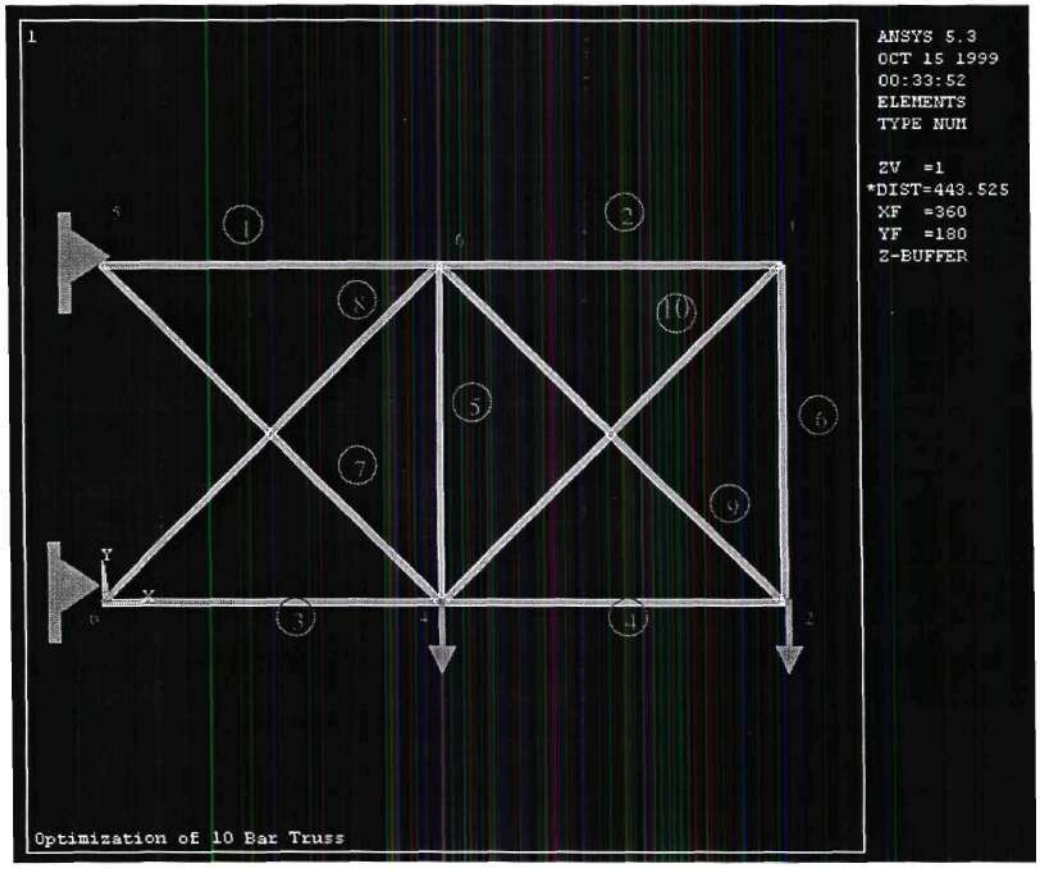


Figure 6-38 Ten-Bar Truss Example

The bars are treated as the same design links used in the flap_link_preliminary product data. The system is a simple rod designed to work in compression and tension loading conditions.

Table 6-2 below shows the input values for modeling the ten bar truss.

Modulus Of Elasticity (aluminum)	10,000 ksi
Material Density	0.10 lb/in^3
Lower Limit on cross sectional areas	0.10 in^2
Initial Value of design variables	10.0 in^2
Stress limit	25 ksi
Number of design variables	10
Number of constraints	10
Number of loading conditions	1

Load Data

Loading condition	Node	Load Components (kips) x	y	z
1	2	0.0	-100.0	0.0
	4	0.0	-100.0	0.0

Table 6-2 Ten-Bar Optimization Data

6.2.2 Engineering Optimization Model:

Weight must be minimized in the ten_bar truss. Meanwhile, the normal stresses on the bars due to the loading condition must not exceed allowable limits. In this case, the parameters that are variables are chosen to be the cross sectional areas of the links. The engineering optimization model of the ten bar truss is shown below. The engineering optimization model can be easily enhanced by adding more design variables in addition to area. Adding more constraints such as deflection limitations and even new objectives also can enhance the model.

ENGINEERING OPTIMIZATION MODEL

Minimize

f_1 *Weight*

Subject to

$g_1 = NC_{stress}(A) < 0$ *Normalized Stress Constraints*

Design variables

$\mathbf{X} = \{A_i\}$

Figure 6-39 Engineering Optimization Model

The template above corresponds to engineering optimization model, shown in Figure 6-39.

6.2.3 EOM Lexical Forms:

Figure 6-40 below shows the EOM lexical form for the ten-bar truss case. Lexical forms are loaded to EOM Tool, which uses the objects defined in those forms to perform analysis and optimization. The external function has been used to facilitate the Ansys FE program to calculate the necessary stress values for each link in the truss.

```
SCHEMA ten_bar_truss_optimization;

SOURCE_SET one ROOT_DOMAIN Truss;
EOM ten_bar;
    xaa: REAL;
    xbb: REAL;
    xcc: REAL;
    xdd: REAL;
    xee: REAL;
    xff: REAL;
    xgg: REAL;
    xhh: REAL;
    xpp: REAL;
    xrr: REAL;
```

```
    obj: REAL;
    g1: REAL;
    g2: REAL;
    g3: REAL;
    g4: REAL;
    g5: REAL;
    g6: REAL;
    g7: REAL;
    g8: REAL;
    g9: REAL;
    g10: REAL;
    sig1: REAL;
    sig2: REAL;
    sig3: REAL;
    sig4: REAL;
    sig5: REAL;
    sig6: REAL;
    sig7: REAL;
    sig8: REAL;
    sig9: REAL;
    sig10: REAL;
    sx: REAL;
    ex: REAL;
    l: REAL;
    nuxy: REAL;
    allow :REAL;
RELATIONS
   r23:"<obj>==
CobExternalToolFunction[ansys,tenbar,obj,{<xaa>,<xbb>,<xcc>,<xdd>,<xee>,<xff>,<xgg>,<xhh>,<x
pp>,<xrr>}]" ONEWAY;
r1:"<sig1>==
CobExternalToolFunction[ansys,tenbar,sig1,{<xaa>,<xbb>,<xcc>,<xdd>,<xee>,<xff>,<xgg>,<xhh>,<
xpp>,<xrr>}]" ONEWAY;

r2:"<sig2>==CobExternalToolFunction[ansys,tenbar,sig2,{<xaa>,<xbb>,<xcc>,<xdd>,<xee>,<xff>,<
xgg>,<xhh>,<xpp>,<xrr>}]" ONEWAY;

r3:"<sig3>==CobExternalToolFunction[ansys,tenbar,sig3,{<xaa>,<xbb>,<xcc>,<xdd>,<xee>,<xff>,<
xgg>,<xhh>,<xpp>,<xrr>}]" ONEWAY;
 r4:"<sig4>==
CobExternalToolFunction[ansys,tenbar,sig4,{<xaa>,<xbb>,<xcc>,<xdd>,<xee>,<xff>,<xgg>,<xhh>,<
xpp>,<xrr>}]" ONEWAY;
    r1:"<sig5>==
CobExternalToolFunction[ansys,tenbar,sig5,{<xaa>,<xbb>,<xcc>,<xdd>,<xee>,<xff>,<xgg>,<xhh>,<
xpp>,<xrr>}]" ONEWAY;
    r2:"<sig6>==
CobExternalToolFunction[ansys,tenbar,sig6,{<xaa>,<xbb>,<xcc>,<xdd>,<xee>,<xff>,<xgg>,<xhh>,<
xpp>,<xrr>}]" ONEWAY;
    r1:"<sig7>==
CobExternalToolFunction[ansys,tenbar,sig7,{<xaa>,<xbb>,<xcc>,<xdd>,<xee>,<xff>,<xgg>,<xhh>,<
xpp>,<xrr>}]" ONEWAY;
```

```
   r2:"<sig8>==
CobExternalToolFunction[ansys,tenbar,sig8,{<xaa>,<xbb>,<xcc>,<xdd>,<xee>,<xff>,<xgg>,<xhh>,<
xpp>,<xrr>}]" ONEWAY;
   r1:"<sig9>==
CobExternalToolFunction[ansys,tenbar,sig9,{<xaa>,<xbb>,<xcc>,<xdd>,<xee>,<xff>,<xgg>,<xhh>,<
xpp>,<xrr>}]" ONEWAY;
   r2:"<sig10>==
CobExternalToolFunction[ansys,tenbar,sig10,{<xaa>,<xbb>,<xcc>,<xdd>,<xee>,<xff>,<xgg>,<xhh>,
<xpp>,<xrr>}]" ONEWAY;
   r3 : "<g1> == (<sig1> / <allow> - 1.0)";
   r4 : "<g2> == (<sig2> / <allow> - 1.0)";
   r3 : "<g3> == (<sig3> / <allow> - 1.0)";
   r4 : "<g4> == (<sig4> / <allow> - 1.0)";
   r3 : "<g5> == (<sig5> / <allow> - 1.0)";
   r4 : "<g6> == (<sig6> / <allow> - 1.0)";
   r3 : "<g7> == (<sig7> / <allow> - 1.0)";
   r4 : "<g8> == (<sig8> / <allow> - 1.0)";
   r3 : "<g9> == (<sig9> / <allow> - 1.0)";
   r4 : "<g10> == (<sig10> / <allow> - 1.0)";

END_EOM;

END_SOURCE_SET;

END_SCHEMA;
```

Figure 6-40 EOM for Ten-Bar

Note that eom model for ten-bar lexical form in Figure 6-40 includes analysis and design attributes. It is more desirable to partition the analysis and design attributes into cbams and apms. But, the concentration in this example is on the validity of results rather than the enhancement of the optimization model.

Results:

Table 6-3 below shows the results of EOM Real, in which EOM is strongly coupled with ANSYS FE optimizer and reference [Haug and Arora, 1979]. EOM Real describes a lexical form prepared such a way that Bolinkgenereral optimizer and Ansys. FEA work iteratively by transforming data between them. Strongly coupled means the optimizer inside the analysis package (Ansys Optimizer) performs the optimization without an external optimizer.

Member No	Optimum Cross Sectional area		
	ANSYS FE Based	Real	Reference
1	8.0057	7.9396	7.9379
2	0.1617	0.1	0.1
3	8.2114	8.065	8.0621
4	4.0209	3.9379	3.9379
5	0.1	0.1	0.1
6	0.1617	0.1	0.1
7	5.8788	5.7462	5.7447
8	5.5887	5.5696	5.5690
9	5.5836	5.5712	5.5690
10	0.1702	0.1	0.1
Optimum Volume(in^3)	16205.88	15935.75	15931.8

Table 6-3 Optimum Results Comparison for Ten Bar Truss

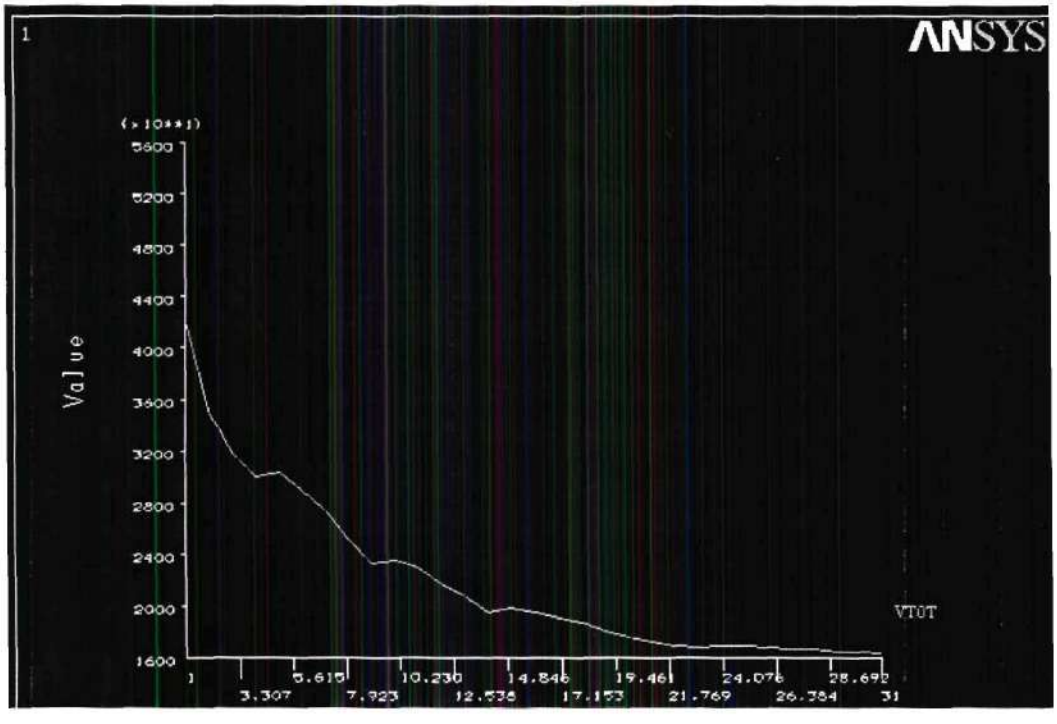


Figure 6-41 Iteration History of Volume of Ten-Bar Truss

6.3 Solder Fatigue Case Study

This section describes how the EOM technique has been applied to solder joint fatigue optimization cases based on Engelmaier's model [Engelmaier, 1983, 1989]. CBAMs representing this model, including finite element-based variations, are explained briefly later in this chapter.

6.3.1 Physical Description

Figure 6-42 shows a surface mount resistor soldered on a PWB.

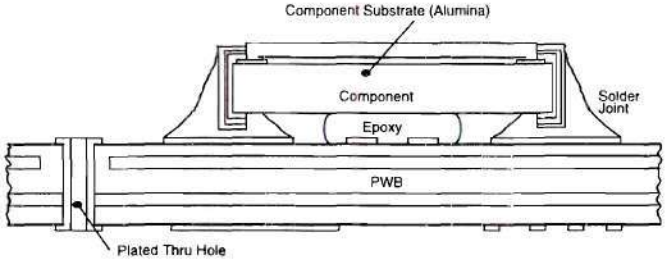


Figure 6-42 Soldered Component on a PWB

The above physical realization can be modeled using various approaches and degrees of complexity. The solder joint can be seen as a 2D geometric entity. In the first group of optimization and analysis models, the height of the solder and the material of the solder are the main concerns for analysis and for design. For the higher fidelity FE cases, 2D rectangles and detailed shapes for the geometry have been used.

6.3.2 Analysis Models (CBAMs)

$\overline{N_f}$	average cycles to failure
$\Delta\varepsilon^p$	plastic cyclic strain range
c	fatigue ductility exponent
ε_f'	fatigue ductility coefficient
$\overline{T}$	mean cyclic temperature, (°C)
f	load frequency, (cycles/day, $1 \leq f \leq 1000$)
$\overline{T}_{sj}$	mean cyclic solder joint temperature (°C)
$\Delta\gamma_{sj}$	solder joint shear strain range
F	adjustment factor
$\Delta(\alpha\Delta T)$	steady state thermal expansion mismatch
T_o	reference temperature
T_{ss}	steady state temperature
L	length
h	height
E	Young's modulus
ν	Poisson's ratio
α	coefficient of thermal expansion (CTE)
σ_Y	yield stress
d	mesh density
n	number of load steps
a	load yield factor
e	convergence criteria
Subscripts	
pwa	printed wiring assembly (PWA)
pwb	(bare) printed wiring board (PWB)
c, s, sj	component, substrate/PWB, solder joint

Subsections 6.3.2.1 and 6.3.2.2 below overview how to analyze solder joint reliability using analysis models 1:Engelmaier Model for Solder Joint Deformation and Analysis Models: 2D FE Model for Rectangle and Detailed Solder Joint Deformation 2a and 2b. Those analysis models are utilized as case studies in this research. Then, optimization models are formed based on these models, and results are explained respectively in sections 6.3.3.1, 6.3.3.2 and 6.3.3.3.

6.3.3.1 Analysis Model 1: Engelmaier Model for Solder Joint Deformation

[Engelmaier, 1983, 1989] developed the following relations by assuming that the solder joint undergoes uniform shear strain as shown in the Figure 6-43. The materials (alumina for ceramic components and FR4 for PWBs) are assumed to be linear elastic. The CTEs needed in these relations are given in Table 6-4

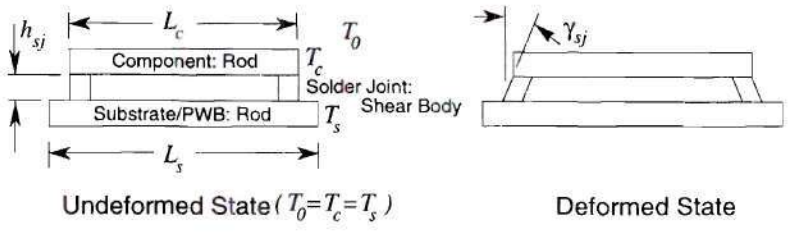


Figure 6-43 Engelmaier Model for Solder Joint Deformation

The major steps to calculate the fatigue life of solder joints are presented in an N^2 diagram [Rogers, 1990] in Figure 6-44. Every box in this diagram shows a module which is a major calculation step towards prediction of fatigue life. Feed forward information is represented by connections above the diagonal of the N^2 diagram. For example, information from module 1 is used in modules 2, 3 and 4. There is no feedback for this case, as indicated by the lack of connections below the diagonal. These calculation steps are represented by a modular nesting of analysis calculations.

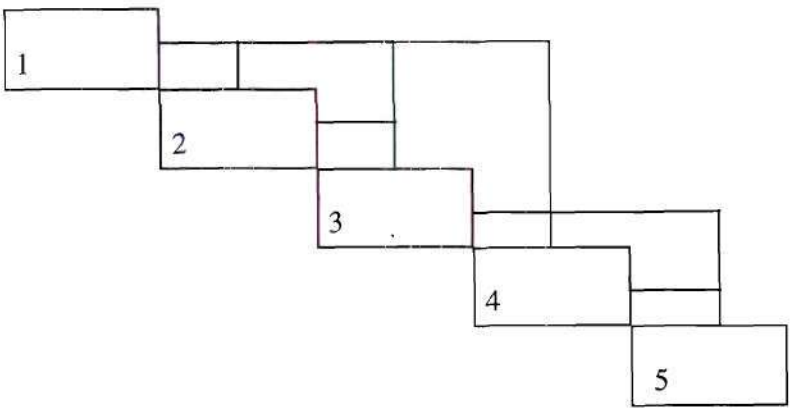


Figure 6-44 N^2 Diagram - Prediction of Fatigue Life

1. Global PWA Thermal State

Power Cycling Case (Operating temperatures)

Turning on/off an electronic product is the most common example of such a case. Temperature differences between components and the PWB will occur during operation, causing strain in the solder joints due to CTE mismatches. In this study, we assume a thermal analysis has already been done, where T_0 = 20°C, yielding T_c = 89°C and T_s = 88°C.

Global PWA Thermomechanical state

Global warpage effects are not considered in this case study. Boundary conditions for the model are assumed to be zero.

Local Solder Joint Thermomechanical State

Engelmaier [1983, 1989] developed the following relations by assuming uniform shear strain in the solder joint.

$$\Delta(\alpha\Delta T) = \alpha_s(T_s - T_o) - \alpha_c(T_c - T_o)$$

$$\gamma_{sj} = \frac{L_c \Delta(\alpha \Delta T)}{2h_{sj}}$$

4. Solder Properties

The following relations are developed by Engelmaier and are used to calculate fatigue life in Step 5.

$$c = -0.442 - 0.0006\overline{T} + 0.0174 \ln(1 + f)$$

$$\overline{T} = \overline{T}_{sj} = \frac{1}{4}(2T_O + T_C + T_S)$$

Solder Joint Fatigue Life

The strain range is calculated using the below two relations, assuming that plastic deformation dominates. F is a correction factor based on the type of solder joint and the experimental results.

$$\Delta\gamma_{sj} = F\left|\gamma_{sj}\right|$$

$$\Delta\varepsilon^p = \Delta\gamma_{sj}$$

The fatigue life model is based on a low-yield stress and on creep under small loads. A modified Coffin-Manson relation for low-cycle fatigue is used by Engelmaier. The exponent c is frequency and temperature dependent and comes from Step 4.

$$\overline{N}_f = \frac{1}{2}\left(\frac{\Delta\varepsilon^p}{2\varepsilon_f'}\right)^{1/c}$$

Given the above, one can see that the case study analysis model, though formula-based, is non-trivial, as it involves several nonlinear relations and numerous variables.

6.3.3.2 Analysis Model 2: Plain Strain Model for Solder Joint Deformation

In this case, a plane strain model is developed to determine the solder joint fatigue of a surface mount chip resistor on an FR4 PWB. The Plane Strain model has been used to calculate shear stress. Leadless modules are attached to the printed circuits with Pb/Sn solder alloy. Due to the thermal expansion mismatch between the modules and the printed circuit boards, thermal and mechanical stresses are developed in solder joint. The plane strain model [Lau, et al, 1986] is used in the optimization models.

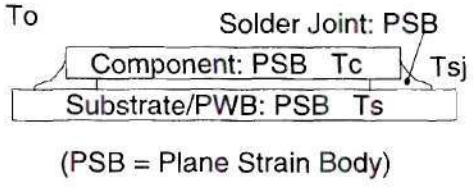


Figure 6-45 Plane Strain Model

Finite element-based solutions are used to capture the plane strain model stated above with two solder geometry, rectangle and detailed.

Analysis Models 2a & 2b : 2D FEA for Rectangle and Detailed Solder Joint

CBAMs can support different types of analysis modeling options with different fidelity in the same model.

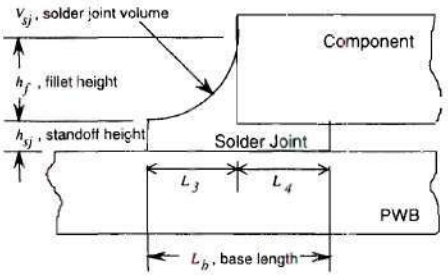


Figure 6-46 Idealized Solder Joint Geometry

Two FE models have been developed using linear elastic solder behavior with rectangular and detailed solder joint geometry. Those models are developed relatively simply in order to demonstrate the model's fidelity rather than concentrating on the model itself. Those models can be replaced with better and more detailed FE models.

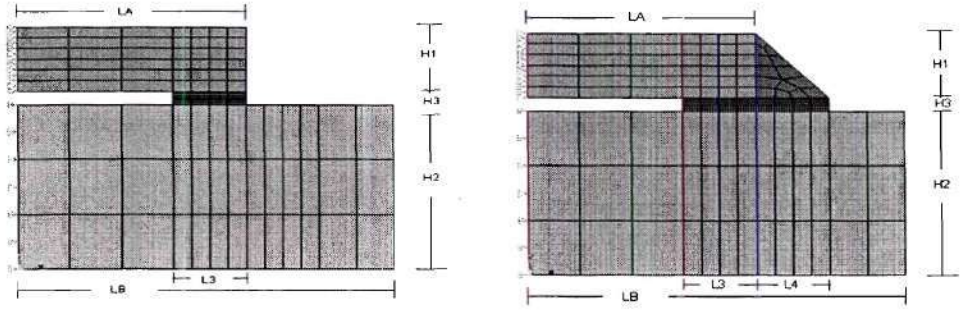


Figure 6-47 FE Element Meshing for rectangular and detailed solder geometry

On the symmetry plane, the component and board are constrained against the x-displacement. To avoid rigid body motion in the y direction, the point on the left bottom corner of the board is constrained against the y-displacement. Temperatures are applied

as body force to study effects of thermal expansion or contraction. The applied temperatures are: To= 20 °C, and Ts = 88 °C. T =89 °C [Wilson, 2000].

The frequency is assumed to be 1 cycle/day. Static structural analysis is performed and the model is assumed to have linear material. The material properties of the components of PWA are shown below:

Material Peoperties at room temperature (24°C)

Material	Young's modulus Mpsi	Poisson's ratio	CTE expansion 10^{-6}in/in °C
resistor (Almumina)	37	0.3	6.7
Board (FR-4)	1.6	0.28	15
Solder (Pb-Sn)	1.5	0.4	21

Table 6-4 Material Properties of PWA

6.3.3 Optimization Models

Figure 6-48 shows the optimization models generated from the analysis models described in the previous section. Two different classes of optimization models are being built to determine the maximum fatigue life of a solder joint are being built. These two models (Engelmeier and Plain Strain Model) have different complexity.

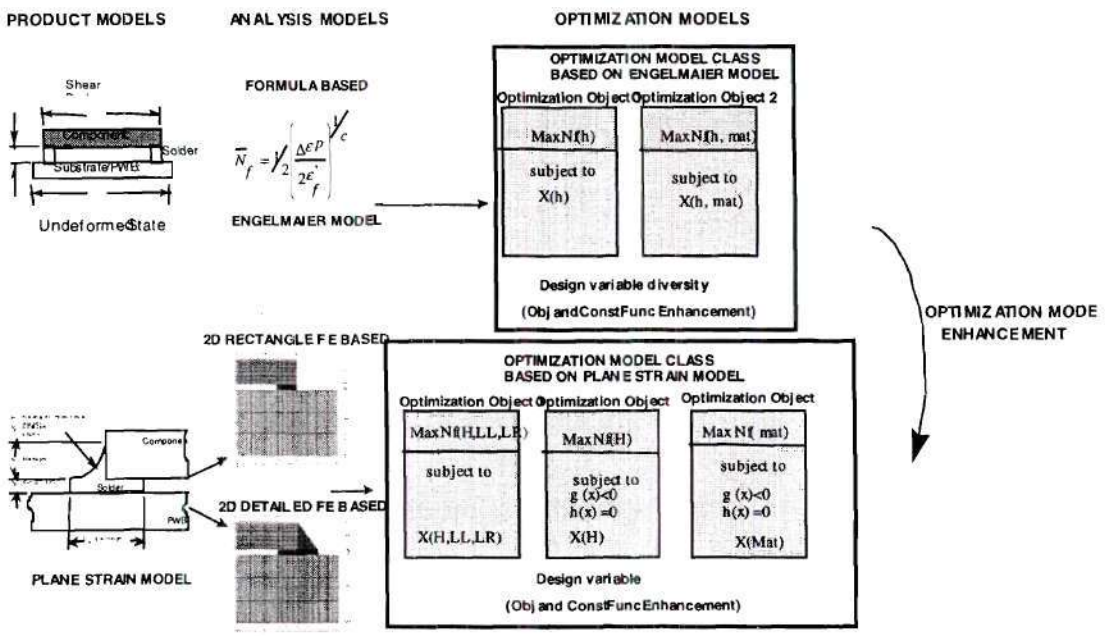


Figure 6-48 Classes of Optimization Models Based on Two Different Fatigue Analysis Models

6.3.3.1. Optimization Models for Analysis Model 1

The case study optimization models are summarized by the following format.

Find

Design variable	Notation
solder joint height	**(h)**
PWB material type	(α_S) ***Model 2 &3***

Maximize

Solder Fatigue life: (Objective Function)

$$\overline{N}_f = 1/2\left(\frac{\Delta\varepsilon^p}{2\varepsilon_f'}\right)^{1/c}$$

Bounds on variables

$0.0001 \le h(in) \le 0.1$

$\alpha_s \text{x} 10^{-6} / \text{C}^\circ \in \{10,15.7,21\}$ ***Model 2***

$\alpha_S \text{x} 10^{-6} / \text{C}^\circ \in \{1,2,3,4,5,6,8,910,15.7,21\}$ ***Model 3***

Three models have been investigated. In the first model, solder joint fatigue life is maximized with respect to solder joint height, with all other parameters held constant. Solder joint height can range from 0.001 to 0.1 inch. This design variable is considered a continuous variable. In this model the PWB material has a coefficient of thermal expansion of $\alpha_S = 15.7 \times 10^{-6}$ /°C .

In the second model, the primary pwb material is added as another design variable which determines α_S in the analysis inputs. This problem is mixed in nature. PWB material is a discrete variable, while solder joint height is a continuous variable. Different optimization tools must be used to handle this case. Three pwb materials are investigated with the following properties:

Material 1	$\alpha_S = 10. \times 10^{-6}$
Material 2	$\alpha_S = 15.7 \times 10^{-6}$
Material 3	$\alpha_S = 21 \times 10^{-6}$

The third model is the same as the second, except that maximized Nf values are calculated from a larger set of materials.

Results

The results obtained for design variables and objective functions for each model are shown in Table 6-5.

	eom1	eom2	eom2'
h(in)	0.1	0.1	0.1
α_S **(/ C)**	-	$\alpha_S = 10.x\ 10^{-6}$	$\alpha_S = 6x\ 10^{-6}$
Nf (cycles)	$3.05x10^6$	$27.8x10^6$	$560.4x10^9$

Table 6-5 Design Variable and N_f Final Results

Figure 6-49 gives the design variable value in eom1 at every iteration of the optimizer, and shows how this value converges rapidly.

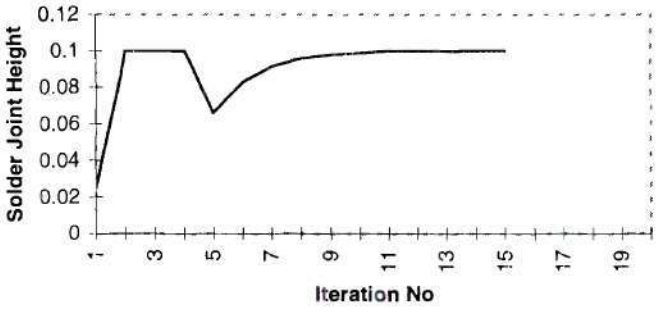


Figure 6-49 Iteration History of Design Variables for Scenario 1

The results for the third model (eom2') are tabulated in Table 6-6. Figure 6-50 also shows Maximized Nf values for various material with different CTE based on Table 6-4.

α_S x10^{-6}	Nf x10^5
1	77
2	116
3	192
4	372
5	968
6	5604
8	2316
9	625
10	278
15.7	30.5
21	11

Table 6-6 Maximum Nf values for Different Materials

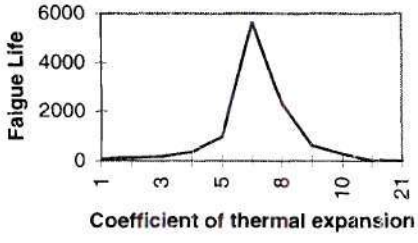


Figure 6-50 Maximized Nf values for Different Solder Materials

The optimizer is the Broyden-Fletcher-Goldfarb-Shanno algorithm based on variable metric method and written in Fortran [Press et al., 1992]. Results are obtained as seen in the tables. The initial design point is the same for all scenarios. Design variable "h" converges smoothly after an initial peak value (Figure 6-50).

The strategy to ensure that the global optimum has been achieved is to search from different initial points in the design space and to verify that the results are converging to the same optimum value. For that reason, different starting design variables have been

tried, and they all converged to the same values shown in the result tables. The optimum solder joint height is found to be the upper limit value of the design variable (as would be expected from examining the relations in Section 6.3.3.1). Solder joint fatigue life increases as solder joint height increases.

Two design variables, one discrete and the other continuous, are used to achieve maximum fatigue life in the second scenario. For this and the third model, one would expect the optimum α_S to be the value that yields zero thermal mismatch, $\Delta(\alpha\Delta T)$: α_S,optimum = $6.7x10^{-6}$ in this case. This is confirmed in Table 6-6 where the PWB material with CTE closest to α_S,optimum gives the maximum fatigue life. Meanwhile, solder again height reached the maximum allowable limit.

Materials with different CTEs have been optimized in the third model in order to demonstrate that the technique correctly predicts that fatigue life is maximum when α_S, and α_S,optimum are closest to each other, even with α_S candidates on either side. One can conclude that fatigue life increases as CTE of PWB material approaches to $6.7x10^{-6}$ value and decreases when the CTE diverges from that value as seen in Figure 6-50 and Table 6-6.

Analysis is coupled with an optimizer using the following approach:

-Initialize optimizer parameters.

-Call the optimizer.

-Supply design variables from optimizer to the analysis model, and then get the objective function value from the analyzer and send it to the optimizer.

-Continue until the objective function value converges to the optimum.

Figure 6-51 shows the data flow between analysis and optimization entities during the iteration process.

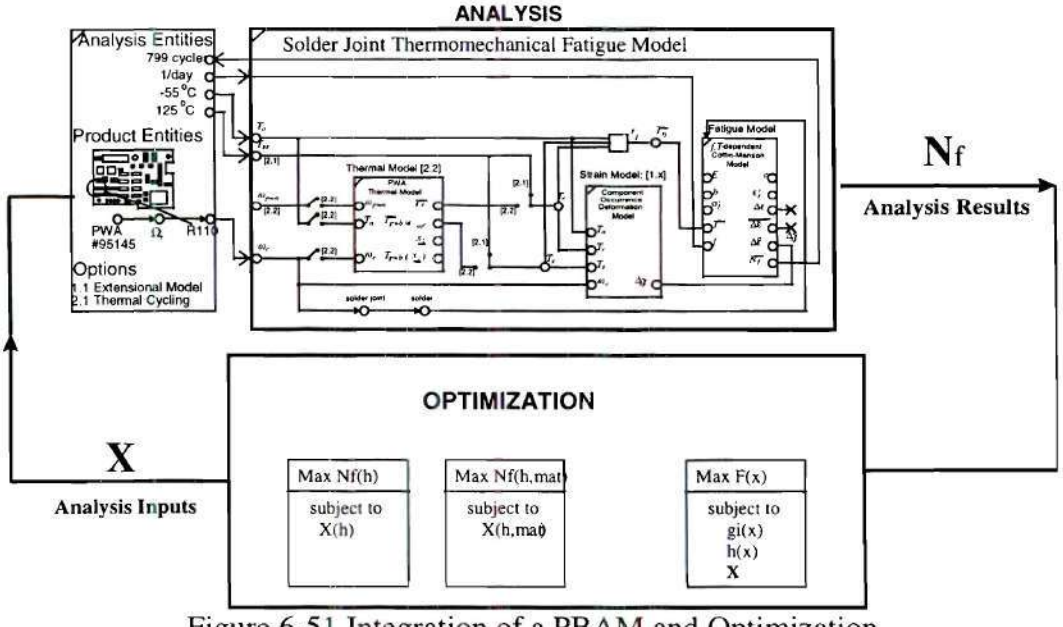


Figure 6-51 Integration of a PBAM and Optimization

6.3.3.2 Optimization Models for Analysis Model 2a

Below, the engineering optimization model of the 2D rectangular solder joint has been shown. There are two design variables, h3 and l3. The constraint on the system is the solder volume with its upper and lower limits.

Engineering Optimization Models

Find

Design variable	Notation
solder joint height	**(h3)**
solder joint base length	**(l3)**

Objective

Solder Fatigue life:

(Objective Function)

$$\overline{N}_f = \frac{1}{2}\left(\frac{\Delta\varepsilon^p}{2\varepsilon_f'}\right)^{1/c}$$

Constraints

$0.0152 <$ Solder Volume < 0.0155 in^2 (allowable)

Bounds on variables

$0.001 \le h3(in) \le 0.024$

$0.01 \le l3(in) \le 0.0625$

Enhancement of Optimization Model for Analysis Model 2a

The above engineering optimization model for solder joint rectangle geometry has been taken as a base. Then, the following optimization models have been developed by modifying the above optimization model to generate new eoms. Two examples are illustrated below.

Eom3: One design variable, height, h3 is used in this model ($0.001 < h3 < 0.024$). This model also helps to compare the results with the earlier model where only the formula based approach has been used in calculation of fatigue life.

Eom4: Two design variables, solder height (0.001< h3 <0.024 in) and solder base length (0.01< l3 < 0.05 in), have been used in this model. The constraint on the system is the solder volume as seen in the base model.

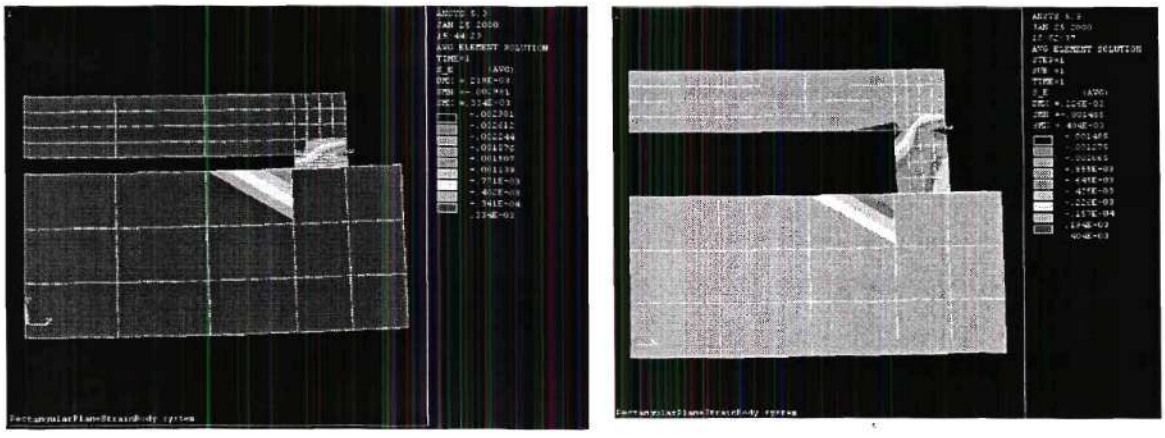
Results

Table 6-7 below shows the results obtained for initial design, eom3 and eom4.

	Original	eom 3	eom 4
H (in)	0.0050	0.0240	0.024
LL(in)	0.0200	0.0200	0.03245
LR(In)	0.0200	0.0200	0.0200
Area of solder joint(in2)	1.47E-02	1.57E-02	1.54E-02
max shear strain	2.98E-03	1.48E-02	1.79E-03
Nf	5.69E+04	2.57E+05	1.71E+05

Table 6-7 Enhanced Optimization Model Results

Finite element post processing' graphical views are shown in the Figure 6-52. The graph in the left of the figure shows the initial configuration with its shear strain view. The right side of the figure shows the optimum configuration for the eom3 model. As expected, height reaches its allowable upper limit to ensure that the maximum fatigue life has been achieved.



Initial Design Final Design

Figure 6-52 Initial and Optimized Design of eom4 Model for Rectangular Solder Joint

Finally, Figure 6-53 below shows how the height of the solder (Design variable) and the Fatigue Life change during the optimization process for eom4.

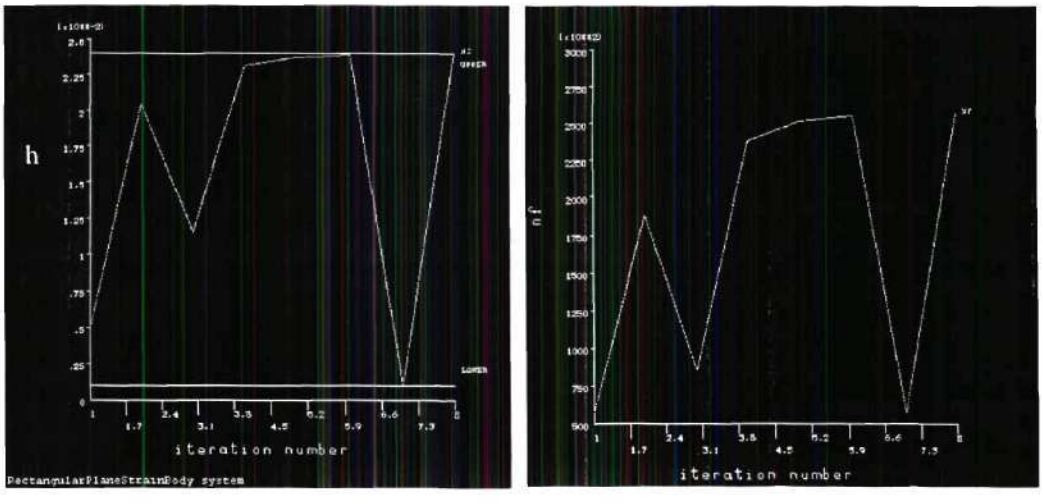


Figure 6-53 Left :Height vs Iteration History Right: Fatigue Life vs Iteration History

6.3.3.3 Optimization Models for Analysis Model 2b

Below, the engineering optimization model of the 2D detailed solder joint has been shown. There are three design variables, h3, l3 and l4 (see Figure 6-47). The constraint on the system is the solder volume with its upper limit.

Engineering Optimization Models

Find

Design variable	Notation
Solder joint height	h3 (in)
Solder base length3	l3 (in)
Solder base length4	l4 (in)

Constraints:

0.001 <h3 (in) <0.024
0.001 < l3 (in) < 0.05625 ; (0.9* LA)
0.001 < l4 (in) < 0.03625 ; (0.9*(LA-LB))
Area of solder joint (in^2) <0.00044 (original area)

Enhancement of Optimization Model for Analysis Model 2b

Similar to Section 6.3.3.2, the above engineering model for the detailed geometry of the solder joint has been taken as a base. Then, the following optimization models have been developed by modifying the above optimization model, helping to generate another set of eoms. Three examples have been illustrated (eom5, eom6 and eom7).

Eom5: Two design variables, solder height (0.001< h3 <0.024 in) and solder base length (0.01< l3 < 0.05525 in), have been used in this model. The constraint on the system is the solder volume as seen in the base model.

Eom6: Three design variables, solder height (0.001< h3 <0.024 in), solder base length (0.01< l3 < 0.05525 in) and solder base length (0.01< l4 < 0.03625 in), have been used in this model. There is no additional constraint on the system except the side constraints on the design variables.

Eom7: Three design variables, solder height (0.001< h3 <0.024 in), solder base length (0.01< l3 < 0.05525 in) and solder base length (0.01< l4 < 0.03625 in), have been used in this model. The constraint on the system is solder area as seen in the base model.

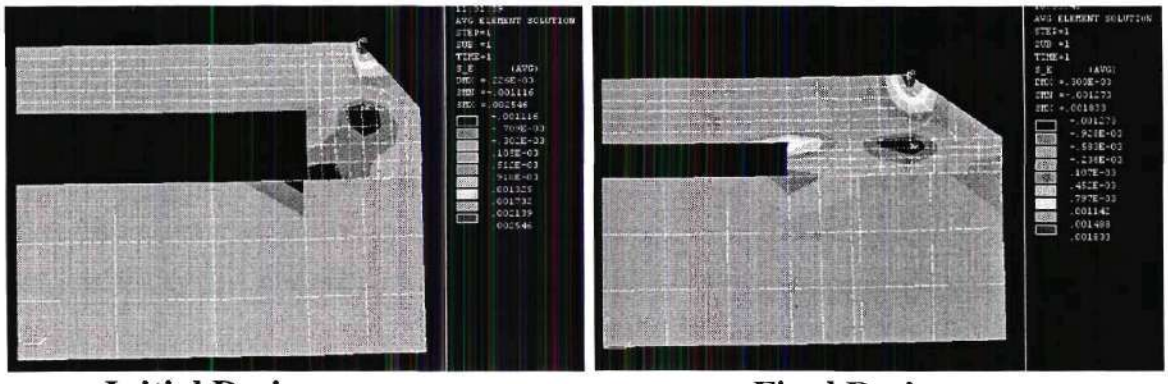
Result

The results obtained for each optimization model are shown in Table 6-8

	Original	eom 5	eom6	eom7
FE results				
H (in)	0.0050	0.0012	0.024	0.001215
LL(in)	0.0200	0.0500	0.05	0.05
LR(In)	0.0200	0.0200	0.036	0.03623
Area of solder joint(in2)	1.47E-02	1.06E-02	2.50E-02	0.00148
max shear strain	2.98E-03	2.40E-02	1.80E-03	1.83E-03
Nf	5.69E+04	9.12E+04	1.70E+05	1.63E+05

Table 6-8 Design Variables, Shear Strain and Nf Final Results

Finite element post processing graphical views for eom7 are shown in Figure 6-54 below. The graph in the left of Figure 6-54 shows the initial configuration with its shear strain view. The right side of Figure 6-54 shows the optimum configuration for the eom7 model.



Initial Design **Final Design**

Figure 6-54 Initial and Optimized Design of eom7 Model for Detailed Solder Joint

Finally, Figure 6-55 below shows how the fatigue life (objective) and height of the solder, and the base lengths (Design variables) change during the optimization process for eom7.

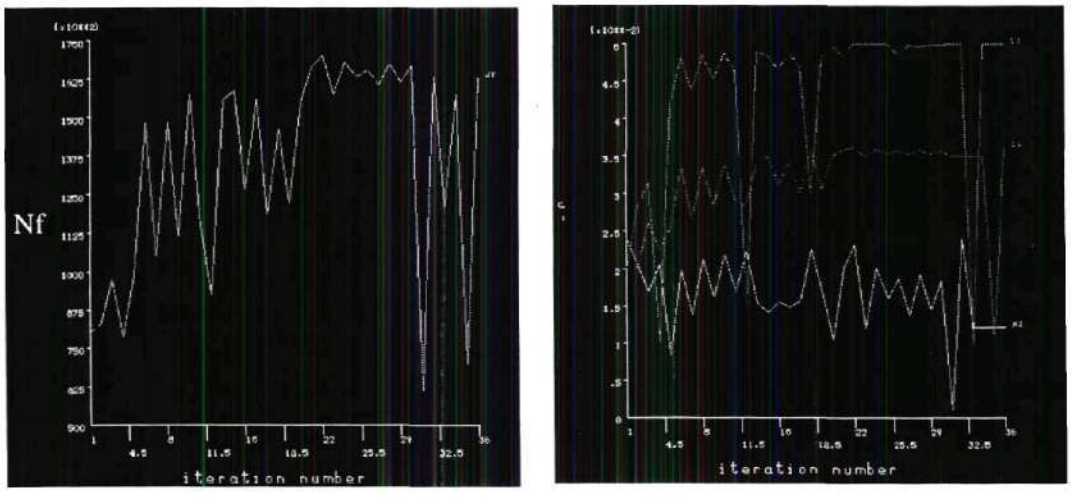


Figure 6-55 Left: Fatigue Life vs Iteration History Right: h3, l3 and l4 vs Iteration History

6.3.4 EOM Representation

The previous section describes the engineering aspects of the analysis and optimization models used. In this section, different aspects of EOM representation are illustrated with the following representative case.

Associativity Linkages for Solder Joint Optimization

Associativity linkages among the optimization, analysis and design entities for the eom1 model are shown in Figure 6-56 below.

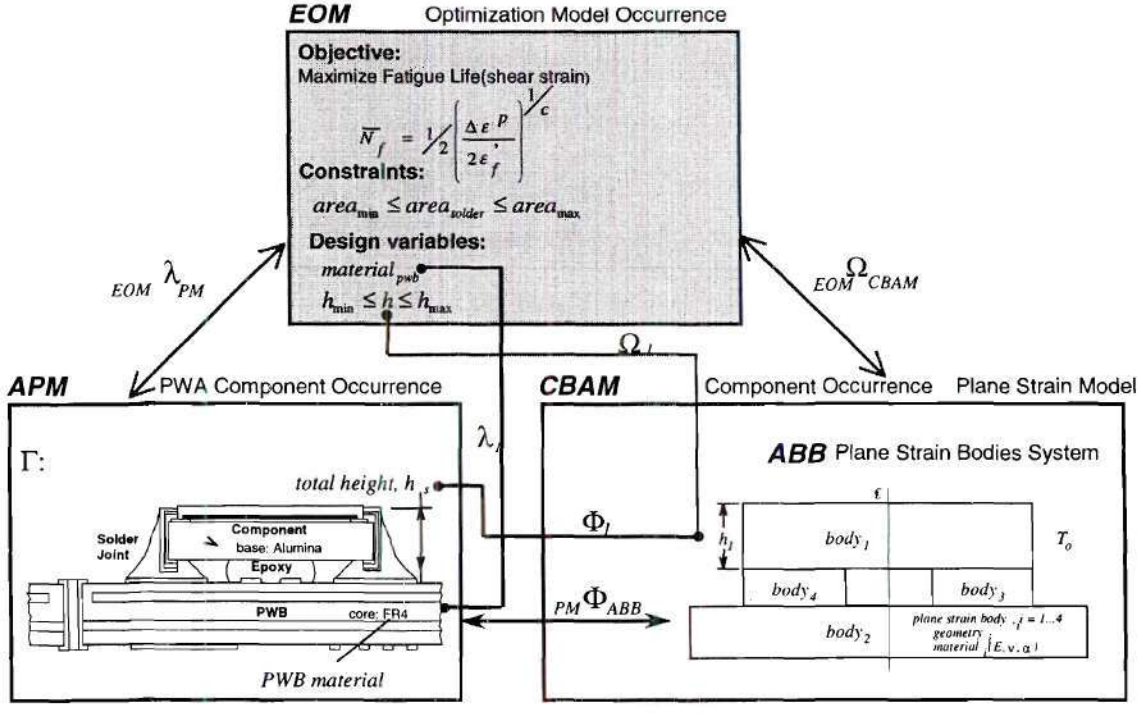


Figure 6-56 Optimization-Design-Analysis Associativity in a Solder Joint Analysis Model

The example for the EOM Schema is shown below for solder joint optimization using eom5.

The first example shows the optimization schema followed by analysis and product model schemas.

```
EOM Solder_Joint_ Fatigue_Life;

/* OBJECTIVE < 0 */

obj: REAL;

/* CONSTRAINTS < 0 */

g1: REAL;
g2: REAL;

/* DESIGN VARIABLES */
xh3 : REAL;
xl3 : REAL;
/* ANALYSIS MODELS */
cbam1 : STRING;
cbam2 : STRING;

RELATIONS
   ala1 : "<xh3> == <cbam1.link.solder.H3>";
   ala2 : "<xl3> == <cbam1.link.solder.L3>";
 /*
   ala3 : "<xl4> == <cbam1.link.solder.L4>";
*/
 /*ala4  :  "<area> ==<cbam2.deformation_model.area>";  */

  ala5 : "<g1> == (1.0/(<cbam1.ms1_mos_model.margin_of_safety> + 1.0))-1";
  ala6 : "<g2> ==(1.52/(100*<cbam1.deformation_model.solder_area>))-1.0";
  ala7: "<obj> == 1.* <cbam1.deformation_model.fatig> / 100000";

END_COB;

END_SOURCE_SET;

USE_FROM ElectronicPackaging\rect\AAcbams.cos AS_SOURCE_SETS;

LINK_DEFINITIONS
  optimization.flap.cbam1 == cbams.pwa_fatigue_model.name;

END_LINK_DEFINITIONS;

END_SCHEMA;
```

```
CBAM  pwa_fatigue_model SUBTYPE_OF link_analysis_model;
   name: STRING; /* 051899 link to optimization */

   deformation_model : pwa_fatigue_abb;
   ms1_mos_model : margin_of_safety_model;
   ms2_mos_model : margin_of_safety_model;

RELATIONS
   al1:"<link.solder.material.stress_strain_model.linear_elastic.youngs_modulus>==
<deformation_model.ex>";
   al2:"<link.solder.material.stress_strain_model.linear_elastic.poissons_ratio>==
<deformation_model.nuxy>";
/*  al3: "<link.solder.material.density> == <deformation_model.density>"; */
   al4: "<link.solder.material.yield_stress> == <deformation_model.ey>";
al1: "<link.solder.material.stress_strain_model.linear_elastic.cte> == <deformation_model.cte3>";
al1: "<link.component.material.stress_strain_model.linear_elastic.cte> == <deformation_model.cte1>";
al1: "<link.pwb.material.stress_strain_model.linear_elastic.cte> == <deformation_model.cte2>";
   al3: "<link.l> == <deformation_model.l>";
al3: "<link.solder.H3> == <deformation_model.xh3>";
al3: "<link.component.LA> == <deformation_model.la>";
al3: "<link.pwb.LB> == <deformation_model.lb>";
al3: "<link.component.H1> == <deformation_model.h1>";
al3: "<link.pwb.H2> == <deformation_model.h2>";
al3: "<link.solder.L3> == <deformation_model.xl3>";
al3: "<link.solder.L4> == <deformation_model.xl4>";
 al6 : "<ms1_mos_model.allowable> == 0.0155";
 al7 : "<ms1_mos_model.determined> == <deformation_model.solder_area>";
/*  al8 : "<ms2_mos_model.allowable> == <link.solder.material.yield_stress>";
 al9 : "<ms2_mos_model.determined> == <deformation_model.sig2>";
*/
END_COB;

CBAM  pwa_fatigue_abb;
l: REAL;/* length */
xh3: REAL;  /* solder height*/
la: REAL
lb:REAL;
xl3:REAL;
xl4:REAL;
h1:REAL;
h2:REAL;
force: REAL;/* total force */
ex: REAL;/* youngs modulus */
ey: REAL;/* yield stress*/
nuxy: REAL; /* poisson's ration */
density:REAL; /*density */
 cte1:REAL;
 cte2:REAL;
 cte3:REAL;
 temperature_s:REAL;
  temperature_c:REAL;
  temperature_ref:REAL;
```

```
 temperature_sj:REAL;
 ssmax: REAL;
 solder_area:REAL.
 coefficient:REAL;
 frequency:REAL;
 fatig:REAL;
 dummy:REAL;
 dk:REAL;
RELATIONS

r1:"<ssmax>==
CobExternalToolFunction[ansys,arfat,ssmax,{<xh3>,<xl3>,<xl4>,<la>,<lb>,<cte1>,<cte2>,<cte3>,<h
1>,<h2>}]" ONEWAY;
r1:"<solder_area>==
CobExternalToolFunction[ansys,arfat,area3,{<xh3>,<xl3>,<xl4>,<la>,<lb>,<cte1>,<cte2>,<cte3>,<h
1>,<h2>}]" ONEWAY;
r2 : "<temperature_sj> == 1/4*(2*<temperature_ref>+<temperature_c>+<temperature_s>)";
r3 :  "<dummy> == (1+<frequency>)";
r3 :  "<coefficient> == -0.442-0.00006*<temperature_sj>+0.0174*N[Log[<dummy>]]";
r3:"<dk>==CobExternalToolFunction[ansys,arfat,nfsl_sxy,{<xh3>,<xl3>,<xl4>,<la>,<lb>,<cte1>,<ct
e2>,<cte3>,<h1>,<h2>}]" ONEWAY;
r3:"<fatig>==CobExternalToolFunction[ansys,arfat,nf,{<xh3>,<xl3>,<xl4>,<la>,<lb>,<cte1>,<cte2>,
<cte3>,<h1>,<h2>}]" ONEWAY;
/* r3 :  "<fatig> == 1/2*((<ssmax>/0.65)**<dk>)";*/

END_CBAM;
```

```
SCHEMA pwa_apm;

SOURCE_SET truss_link_geometric_model ROOT_COB pwa;

/*USE_FROM lib/apm.cos; */

APM  part;
  /*part_number : STRING;*/
  designer : STRING;
END_APM;

APM  material;
   material : STRING;
END_APM;

APM body1;
   material : STRING;
   H1 : REAL;
   LA : REAL;
END_APM;
```

```
APM body2;
   material : STRING;
   H2 : REAL;
   LB : REAL;
END_APM;

APM  body3;
   material : STRING;
   H3 : REAL;
   L3 : REAL;
   L4 : REAL;
END_APM;

APM  pwa SUBTYPE_OF part;
  part_number : STRING;
  l : REAL;
  w : REAL;
  t : REAL;
  area :REAL;
  volume : REAL;
  weight : REAL;
  component:body1;
  solder:body3;
  pwb:body2;
RELATIONS
 PRODUCT_RELATIONS
  pr1 : "<area> == <w>*<t>";
  pr2 : "<volume> == <l>*<w>*<t>";
  pr3 : "<weight> == <volume>*<component.material.density>";

END_SOURCE_SET;

USE_FROM lib/materials.cos AS_SOURCE_SETS;

LINK_DEFINITIONS
   truss_link_geometric_model.pwa.component.material== materials.material.name;
   truss_link_geometric_model.pwa.pwb.material== materials.material.name;
   truss_link_geometric_model.pwa.solder.material== materials.material.name;

END_LINK_DEFINITIONS;

END_SCHEMA;
```

Note that those lexical forms have been implemented with minor modifications for parsing reason in the EOM prototype. In this chapter, test cases have been presented with results to illustrate the different capabilities of the EOM representation.

The next chapter evaluates the objectives of this thesis (Chapter 3) with respect to the test cases presented in this chapter.

CHAPTER 7

EVALUATION

The aim of this chapter is to evaluate the EOM Representation as presented in Chapters 4, 5 and 6. First, Section 7.1 establishes the approach and scope taken for this evaluation with respect to test cases. Section 7.2 analyzes the results obtained and evaluates them against the research objectives.

7.1 Evaluation Approach and Scope

The purpose of this evaluation is to assess the degree to which the EOM Representation has met the objective requirements presented in Chapter 3. The representation has been developed and tested based on the special requirements of optimization-design-analysis integration through prototype and test cases. Consequently, the objectives of this thesis are evaluated by determination of how effective this representation is at integrating optimization with design and analysis applications.

Recall from Chapter 4 that the EOM Representation consists of the following components:

- EOM Information Model;
- EOM Definition Languages;

  EOM Lexical Forms

EOM Graphical Forms

- EOM Protocol;

Evaluation of the EOM Representation is based on evaluating its components against the research objectives. If the components meet a given objective, then the EOM Representation meets the objective.

The objectives presented in Chapter 3 were used as a basis for the evaluation of the EOM Representation. As stated in that chapter, these objectives represent the requirements of a suitable enhanced optimization model representation. Therefore, the extent to which EOM Representation met these objectives indicates how fit it is as an enhanced optimization model representation. This final evaluation of whether or not the EOM Representation meets a given objective is recorded in the columns of Table 7-1. For each objective, "Yes", one of these columns indicates that the objective was satisfactorily met by the corresponding EOM characteristics.

7.2 Evaluation of EOM Representation

Table 7-1 summarizes the results of the evaluation. The reader is referred to Chapter 3 for the descriptions of needs and related objectives. The objectives and the needs that characterize the EOM representation have been grouped as seen in the left column of the Table 7-1. Test cases are presented in the first row of Table 7-1.

EOM REPRESENTATION OBJECTIVES		TEST CASES		
		FLAP LINK OPTIMIZATION	TEN BAR TRUSS OPTIMIZATION	SOLDER JOINT FATIGUE LIFE OPTIMIZATION
OPTIMIZATION INTEGRATION		Yes	Yes	Yes
PARTITION	Optimization Models	Yes		Yes
	Analysis Models	Yes		Yes
	Design Models	Yes		Yes
ASSOCIATIVITY	Optimization-Analysis	Yes		Yes
	Optimization-Design	Yes		Yes
ENHANCEMENT	Objectives			
	Constraints	EOM Table 7-2		EOM Table 7-5
	Design Variables	EOM Table 7-2		EOM Table 7-5
	Analysis Models (CBAMs)	CBAM Table 7-3		CBAM Table 7-6
	Product Models (APMs)	APM Table 7-4		APM Table 7-7
AUTOMATED METHOD AND TOOL SELECTION		Yes		Yes
EXTEND ANALYSIS DESIGN INTEGRATION ARCHITECTURE		Yes	Yes	Yes
SUPPORT VARIOUS DISIPLINES		Yes	Yes	Yes

Table 7-1: EOM Representation Evaluation

7.2.1 Optimization Integration

EOM Representation defines the basic building blocks for creating optimization models with their analysis and design associations. One of the main functions of EOM is to present a formal structure for optimization-analysis-design-integration. This representation provides a mechanism to model optimization objects in conjunction with analysis and design objects. It also captures the semantics specific to optimization-analysis-design entities. For that reason, a formal structure, composed with its information model, its languages (lexical and graphical forms) and its protocols, has been developed. Constructs of the representation have been introduced in Chapter 4.

- ***Representation provides a way for partition among optimization, product and analysis models***

A clear distinction among the core attributes and relations that belong to the optimization, product and analysis models was introduced in Subsection 4.1. The flap link prototype example extensively shows the relations among optimization (eom_flap_link_optimization), analysis (cbams) and design entities (apms) in section 6.1.

This partitioning allocates the information to the appropriate engineering domains, allowing reuse and portability. Also, partitioning helps to identify where the information is generated, how to retrieve and transform it.

The optimization attributes include the objective function maximized or minimized, the constraints that are applied to the system and the design variables to achieve the objective functions. Tables 7-2 and 7-5 show the optimization models for minimization of flap link weight and for maximum solder fatigue life in PWA respectively.

The analysis attributes characterize the behavior of the product under specific environmental conditions, and they provide the response of the system under these conditions. Tables 7-3 and 7-6 show the analysis representations of flap link and PWA for different modes and fidelities of analysis models. The intrinsic product attributes and relations refer to geometric and material related attributes. These attributes are independent of the environmental conditions to which the product is subjected. Tables 7-4 and 7-7 show the design representations of flap link and PWA.

- ***Representation provides modeling language for defining optimization models***

EOM lexical and graphical languages are introduced in Subsection 4.2. EOM modeling languages allow engineers to easily create, modify or extend optimization models without requiring extensive effort. The languages are able to capture the main entities of optimization models with their attributes, associativity links and relations. The languages are independent of domain and computer applications that are used.

- ***Representation provides the explicit structure of data integration knowledge***

Generally, the optimization model requires the retrieval of design and analysis information scattered across several repositories and combined for optimization purposes. For that reason, a formal structure for defining the transformations provides explicit description of the data integration process. Generally, these transformations are not explicitly defined. Therefore, they are implicitly placed inside the code of the optimization and analysis applications, making it difficult to reuse or modify them. The formal description of optimization models is defined by the EOM languages. The EOM protocols in Section 4.3 show the methods related to implementation of those associativity links to design and

analysis objects. Specifically, the flap link example of Section 6.1 shows those links to the related objects.

7.2.2 Enhancement of Optimization Models

One of the goals of this thesis is to generate and enhance optimization models based on various fidelities of analysis and product models. This provides modularity and flexibility to multi-fidelity optimization models that are tailored with a varying complexity of analysis and design models. An engineer can easily build and modify optimization model instances with varying complexity. This is one of the major concerns in current approaches to optimization model-design-analysis. The users have a better means for adding or changing optimization models and for retrieving data without a major adaptation of their analysis and product data.

- ***Representation supports for building and enhancing multi-fidelity optimization models***

It has been illustrated in the test cases that EOM representation supports the modification of the optimization models. One class of optimization models can be enhanced by adding new objectives, constraints and design variables while maintaining the same set of design and analysis models for those optimization models. Table 7-2 shows eom1, eom2, eom3 and eom4 for weight minimization of flap link. All of those models use the same analysis (cbam1a_extension) and design (apm flap link preliminary) models. Eom1 has one objective (weight), one design variable (area) and one constraint (stress). Eom2 has one additional design variable (material). Eom3 has one design variable (area) and two

constraints (stress and buckling). Eom4 has both constraints and design variables. Similarly, Table 7-5 shows the optimization model enhancement for solder fatigue life optimization case.

Enhanced Optimization Models (EOMs)	EOM1	EOM2	EOM3	EOM4
Optimization Model Description	Minimize Weight of flap-link_pre based on its area using 1D Normal Stress Model	Minimize Weight of flap-link_pre based on its area and material using 1D Normal Stress Model	Minimize Weight of flap-link_pre based on its area using 2D FE Stress Model	Minimize Weight of flap-link_pre based on its area and material using 2D FE Stress Model
Optimization Model Objectives:	Minimize Weight of Flap Link	Minimize Weight of Flap Link	Minimize Weight of Flap Link	Minimize Weight of Flap Link
Constraints:	limits on area constraint on stress	limits on area list of materials constraint on stress	limits on area constraint on stress constraint on buckling	limits on area list of materials constraint on stress constraint on buckling
Design Variables:	A	A, material	A	A, material
Name: Type:	flap link area continuous	flap link area material continuous discrete	flap link area continuous discrete	flap link area material continuous discrete
CBAM (see APM Table7-3)	cbam1a_extension	cbam1a_extension	cbam1a_extension cbam2_buckling	cbam1a_extension cbam1a_buckling
APM (see APM Table 7-4)	apm_flaplink_pre	apm_flaplink_pre	apm_flaplink_pre	apm_flaplink_pre
Optimization Method	Modified Feasible Direction	Mix Integer_Continuous	Modified Feasible Direction	Mix Integer_Continuous
Optimization Tool	GeneralBolink with CONMIN	GeneralBolink1 with CONMIN	GeneralBolink with CONMIN	GeneralBolink1 with CONMIN

Table 7-2 EOM Models for Flap Link

Enhanced Optimization Models (EOMs)	EOM5	EOM6
Optimization Model Description	Minimize Weight of flap link detailed Based on its flange and wall thickness using FE Plain Strain Model	Minimize Weight of flap link detailed based on its flange and wall thickness, flange width and length using FE Plain Strain Model
Optimization Model Objectives:	Minimize Weight of Flap Link	Minimize Weight of Flap Link
Constraints:	Limits on wall thickness Limits on flange thickness Stress constraint	Limits on wall thickness Limits on flange thickness Limits on flange width Limits on flap link length Stress constraint
Design Variables:	t_f, t_w	t_f, t_w, w_f, l
Name: Type:	wall thickness flange thickness all continuous	wall thickness flange thickness flange width flap link length all continuous
CBAM (see APM Table7-3)	cbaml b_plane_strain_FE	cbaml b_plane_strain_FE
APM (see APM Table 7-4)	apm_flaplink_detailed	apm_flaplink_detailed
Optimization Method	Feasible Direction Method	Feasible Direction Method

Table 7-2 EOM Models for Flap Link (continue)

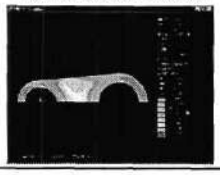
Context Based Analysis Model(CBAMs)	CBAM2	CBAM1	
CBAMS Description Model Description	Buckling Model 	Extensional Model (Formula Based) 	Plane Strain Model 
Model	Buckling	Normal Stress	
APM (see APM Table 7-4)	**Apm_flaplink_1a**	**Apm_flaplink_1a**	**Apm_flaplink_1b**
Solder Joint Analysis Description	**cbam2**	**cbam1a**	**cbam 1b**
	1D Buckling	1D Normal Stress	2D Detailed FE Analysis
Analysis Method	Formula	Formula	FE Element
Analysis Tool	Mathematica	Mathematica	Ansys

Table 7-3 CBAMs: Flap Link Case Study Analyses

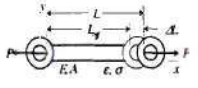
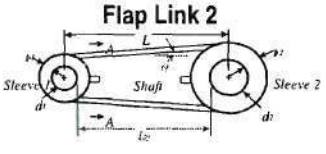
Analyzable Product Models (APMs)	Apm_flaplink_1a	Apm_flaplink_1b
APM Description Flap Link Model Description	Flap Link 1 	Flap Link 2 
Design stage	Preliminary	Detailed
Geometry description	Sleeves Prismatic Link	Sleeves I-Beam Link
Material model description	Linear Elastic : (Steel, Titanium, Aluminum)	
CAD Tool	----------	CATIA IDEAS

Table 7-4 Product Models for Solder Joint Fatigue Case Study

Enhanced Optimization Models (EOMs)	EOM1 1o1d	EOM2 & 2' 1o2d	EOM3 1o1d	EOM4 1o1c2d
Optimization Model Description	Maximize Solder Fatigue Life of solder 1 based on its height using Extensional Model	Maximize Solder Fatigue Life of solder 1 based on its height and PWB material using Extensional Model	Maximize Solder Fatigue Life of solder 2 based on its height using FE Rectangle Model	Maximize Solder Fatigue Life of solder 2 based on its height & length using FE Rectangle Model
Optimization Model Objectives:	Maximize Solder Fatigue Life	Maximize Solder Fatigue Life	Maximize Solder Fatigue Life	Maximize Solder Fatigue Life
Constraints:	limits on height	limits on height list of materials	limits on height	Limits on height Limits on base length Solder Volume
Design Variables:	h	h , α	h_3	h_3 , l_3
Name: Type:	solder height continuous	solder height pwb material continuous discrete	solder height continuous	solder height solder base length3 all continuous
CBAM (see Table 7-6)	cbam1_extension	cbam1_extension	cbam2a_plane_strain	cbam2a_plane_strain
APM (see Table 7-7)	apm_pwa_1	apm_pwa_1	apm_pwa_2	apm_pwa_2
Optimization Method	Variable Metric Method	Mix Integer_Continuous	Sub-problem Approximation Method / M. Feasible Direction	Sub-problem Approximation Method / M. Feasible Direction
Optimization Tool	BFGS Fortran Press at al. & CONMIN	Modified BFGS	Ansys-Opt/ CONMIN	Ansys-Opt/ CONMIN

Table 7-5 EOM Models for Solder Joint

Table 7-5 EOM Models for Solder Joint (continue)

Enhanced Optimization Models (EOMs)	EOM5 1o2d1c	EOM6 1o3d	EOM7 1o3d1c
Optimization Model Description	Maximize Solder Fatigue Life of solder 2 based on its height and base length using FE Detailed Model	Maximize Solder Fatigue Life of solder 2 based on its height and lengths using FE Detailed Model	Maximize Solder Fatigue Life of solder 2 based on its height and lengths using FE Detailed Model
Optimization Model Objectives:	Maximize Solder Fatigue Life	Maximize Solder Fatigue Life	Maximize Solder Fatigue Life
Constraints:	limits on height limits on base length solder volume	limits on height limits on base length3 limits on base length4	limits on height limits on base length3 limits on base length4 solder volume
Design Variables:	h_3 , l_3	h_3 , l_3 , l_4	h_3 , l_3 , l_4
Name: Type:	Solder height solder base length3 all continuous	solder height solder base length3 solder base length4 all continuous	solder height solder base length3 solder base length4 all continuous
CBAM (see Table 7-6)	cbam2b_plane_strain	cbam2b_plane_strain	cbam2b_plane_strain
APM (see Table 7-7)	apm_pwa_2	apm_pwa_2	apm_pwa_2
Optimization Method	Sub-problem Approximation Method	Sub-problem Approximation Method	Sub-problem Approximation Method
Optimization Tool	Ansys-Opt	Ansys-Opt	Ansys-Opt

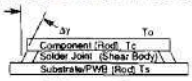
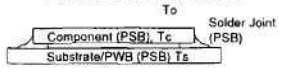
Context Based Analysis Model(CBAMs)	CBAM1	CBAM2	
CBAMS Description Strain Model Description	**Extensional Model**  Deformed State	**Plane Strain Model**  (PSB = Plane Strain Body)	
Solder Model	Viscoplastic Solder Model	Linear Solder Model	
APM (see Table 7-7)	**Apm_pwa_1**	**Apm_pwa_2**	**Apm_pwa_2**
Solder Joint Analysis Description	**cbam1**	**cbam2a**	**cbam2b**
	1D Solder Joint	2D Rectangular Solder Joint	2D Detailed Solder Joint
Analysis Method	Formula	FE Element	FE Element
Analysis Tool	Mathematica	Ansys	Ansys

Table 7-6 CBAMs: Solder Joint Fatigue Case Study Analyses

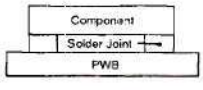
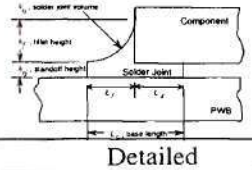
Analyzable Product Models (APMs)	Apm_pwa_1	Apm_pwa_2	
APM Description PWA Model Description	**Printed Wiring Assembly 1** 	**Printed Wiring Assembly B** 	
Design Stage	Preliminary	Detailed	
Geometry description	Resistor R11 **Solder Joint** 2D PWB	ResistorR11 **2D Rectangular Solder Joint** 2D PWB	Resistor R11 **2D Detailed Solder Joint** 2D PWB
Material description	Resistor R11: Alumina Solder Joint : Fr4 PWB: Pb_Sn		
CAD Tool	---------	Menthor Graphics	

Table 7-7 Product Models for Solder Joint Fatigue Case Study

- ***Representation incorporates multi-fidelity analysis and design models into optimization models.***

Above, the enhancement of one class of optimization models can be achieved by adding new objectives, constraints and design variables by using the same analysis and design models. Another class of optimization model has been generated using a higher fidelity of analysis models (cbams) and design models (apms). Tables 7-3 and 7-4 show varying fidelities of analysis and design models which are incorporated into optimization models for the flap link example. Similarly, Tables 7-6 and 7-7 show varying fidelities of analysis and design models for PWA.

7.2.3 Representation supports automated Optimization Method and Tool Selection

The determination of automated selection of the optimization method based on the EOM representation has been explained in subsection 6.3. That section also explains how compatible tools for that method can be implemented to find the optimum solution of the optimization model. In Chapter 5, implementation for the flap link case shows that the right method is found by testing the design variables, whether they are continuous or discrete variables. Table 7-2 and Table 7-5 summarize the methods and tools used for eom models for flap link optimization.

7.2.4 Optimization Models from Various Engineering Disciplines

Most of the products require numerous analysis disciplines to test for different behaviors. This is shown in the test case of flap link for the stress and buckling analysis. Also, stress analysis for the ten-bar truss example demonstrates that the results obtained in this thesis are compatible with the results in the literature. Fatigue life analyses of solder joint fatigue life for different fidelities were shown in subsection 6.3. Those analyses are used to obtain the response from the engineering part. Those responses are reflected in the optimization models. The representation supports the approach to optimization where different disciplines are used to construct the models.

7.2.5 Extend and Utilize Multi-Representation Architecture (MRA)

The EOM Representation is compatible with the MRA approach [Peak, 1995]; it utilizes the MRA by interacting with the product information and analysis information. This adds the capability of extending MRA to the optimization domain. In addition, EOM facilitates design representation and analysis representation already developed in EIS Lab. The EOM Representation provides associativity links for interaction among CBAM and APM models. This mechanism also supports the recursive nature of optimization tools during data interaction. Thus, transformation of the information is managed in an efficient manner.

7.3 EOM Representation Versus Classical Optimization Model

Some of the differences and advantages of EOM representation are explained by an example below. Figure 7-1 shows the classical engineering optimization model with its parameters used to define the objective, constraint function and design variables. As shown in Figure 7-1, material has been chosen as one of the design variables. A list of materials has been searched in conjunction with area to find a minimum weight of a flap link.

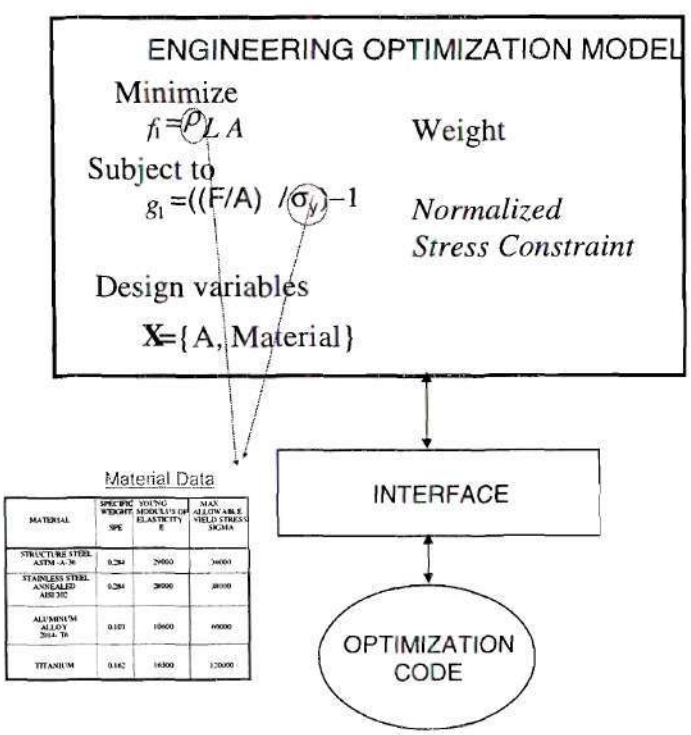


Figure 7-1 Classical Engineering Optimization Model

Selecting a material from the list of materials has an effect on the optimization model during the optimization process. For instance, selection of a particular material from a list

of materials in the above optimization model requires data retrieval of density and yield stress of the material from different sources (material database, catalogs, etc.). Although user interfaces have been developed recently, the information is lumped together in the classical optimization model. The model developer has to retrieve the information with difficulty. There is no explicit relationship between the optimization model and the information sources. If there is a relationship, it is implicit; also, this process is procedural. Figure 7-2 shows the EOM Constraint schematic view of the same model.

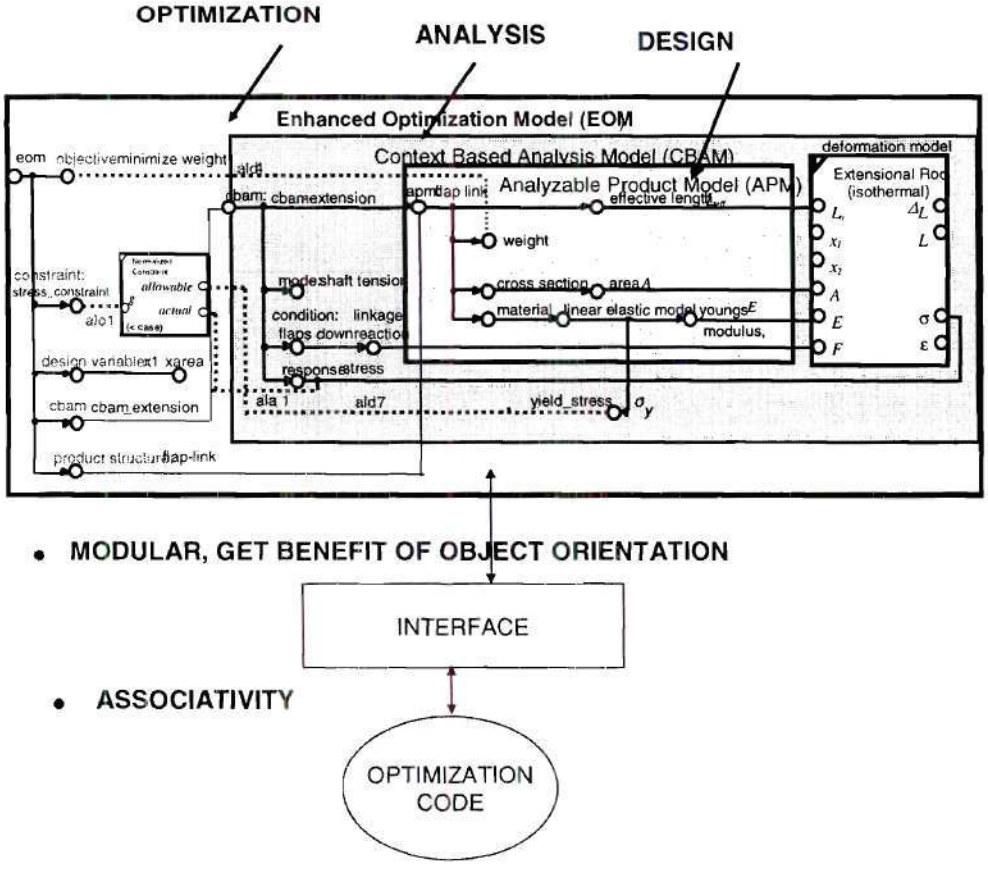


Figure 7-2 EOM Representation-Constraint Schematic Graph

EOM representation utilizes the object oriented and constraint graph concepts. EOM representation partitions the information in the optimization model to its optimization, analysis and design domains. Then, transformation between optimization model objects and analysis/design objects can be formally and explicitly defined by EOM associations. Partition of domains to their attributes, and relationships of association between those attributes, lead to the easier enhancement ability of the optimization model. It becomes more modular and flexible to enhance the model in different levels. This also leads to identification of the optimization model and tools. Thus, automation of the optimization process is achieved for diverse optimization problems based on their explicit information captured in EOM representation. Because information is captured in schema (meta-level), changing the optimization tool does not change the EOM object.

CHAPTER 8

CLOSURE

In this research, information representations of optimization models have been developed to facilitate the design process in a more efficient manner. This information representation named **Enhanced Optimization Model (EOM)** facilitates multiple fidelity optimization, analysis and design models, emphasizing association with product and analysis specific information. One of the main goals of this thesis, optimization-design-analysis integration considering modularity, is achieved by EOM representation. The representation is used for capturing the information in the optimization model with association linkage to the product and analysis specific information. The enhanced optimization model (EOM) is an engineering optimization model which is semantically rich and, where needed, constraints and design variables are defined in terms of product- and analysis-specific terms. These terms are also as flexible to enhance the model during the design progression. In structures, optimization of flap link and ten-bar truss have been demonstrated in test cases. Also, optimization of solder joint fatigue life has been demonstrated in the electronic packaging area. Those test cases, specifically, have demonstrated the ability of EOM Representation by targeting the interaction of optimization with design and analysis to produce an optimized component. The study also demonstrates information techniques for optimization model generation and usage.

8.1 Uniqueness of Solution and Benefits

The primary contributions of this research are representations and methods to improve the design optimization process. The significance of this research is that it supports enhancement in optimization, the explicit representation of optimization problems and modularity, which breaks optimization into natural engineering objects for optimization-analysis-design integration. New optimization capabilities effectively add a highly automated loop that provides inputs to analysis and product data. One contribution of this thesis is the ability to add such optimization loops to different types of analysis models - taking advantage of their modular nature to select appropriate optimization algorithms and to perform optimization at differing levels. Thus, the resulting optimization capabilities could potentially be added to analysis models developed in various disciplines.

Uniqueness of Solution:

a) EOM Representation helps to construct optimization models utilizing product and analysis specific information to drive optimization.

b) Data, with its semantics and values, retrieved from analysis and design representations, are utilized in the optimization model.

c) EOM Representation supports enhancement of optimization models of different complexity and levels by adding optimization to the existing analysis and design models.

d) EOM Representation automates optimization based on product type, analysis technique and optimization models.

Benefits:

a) The representation permits deeper understanding of engineering optimization through new knowledge representations.

b) The representation enables optimization models to retrieve the data they need from design and analysis representations instead of considering all the entities combined as a whole in the optimization model.

c) Engineers can easily understand the representation of an optimization model with its ingredients.

d) Common analysis and design data is shared among several optimization models instead of replicated in the codes of each individual application.

e) The representation is easily defined, constructed and modified.

f) The representation enables improved PWA designs with more balanced consideration of thermo-mechanical issues and other life cycle constraints.

g) The representation offers potential applications to other domains beyond the test cases, taking better advantage of CAE/CAD tools currently deployed by research and industry.

8.2 Recommended Extensions

- **Partitioning by measuring the correlation among the optimization, analysis and design entities**

As design progresses in different phases, the product, analysis and optimization models become more information-intensive. Figure 8-1 above shows a preliminary study of the three main entities of flap link in an orthogonal view with increasing fidelity during design stages.

The design dimension is illustrated with two different fidelities of product model of flap link; flap_link_preliminary(amp1a) and flap_link_detailed(apm1b). The analysis dimension is illustrated with these analysis models: cbam1a and cbam1b for normal_stress analysis, and cbam2 for buckling analysis. The optimization dimension is illustrated with the eom model of different enhanced levels. Marked points in Figure 8-1 show the compatibility among the models. For instance, flap_link_preliminary(amp1a) model is not compatible for cbam2(FE Analysis), which is developed for flap_link_detailed(apm1b). Furthermore, the position in each axis can be a relative measurement for degree of accuracy of the model. That view, which illustrates the relationships among the models and degrees of accuracy, may help to map the models into a mathematical space. This may lead to the use of mathematical operations to better predict and analyze the models.

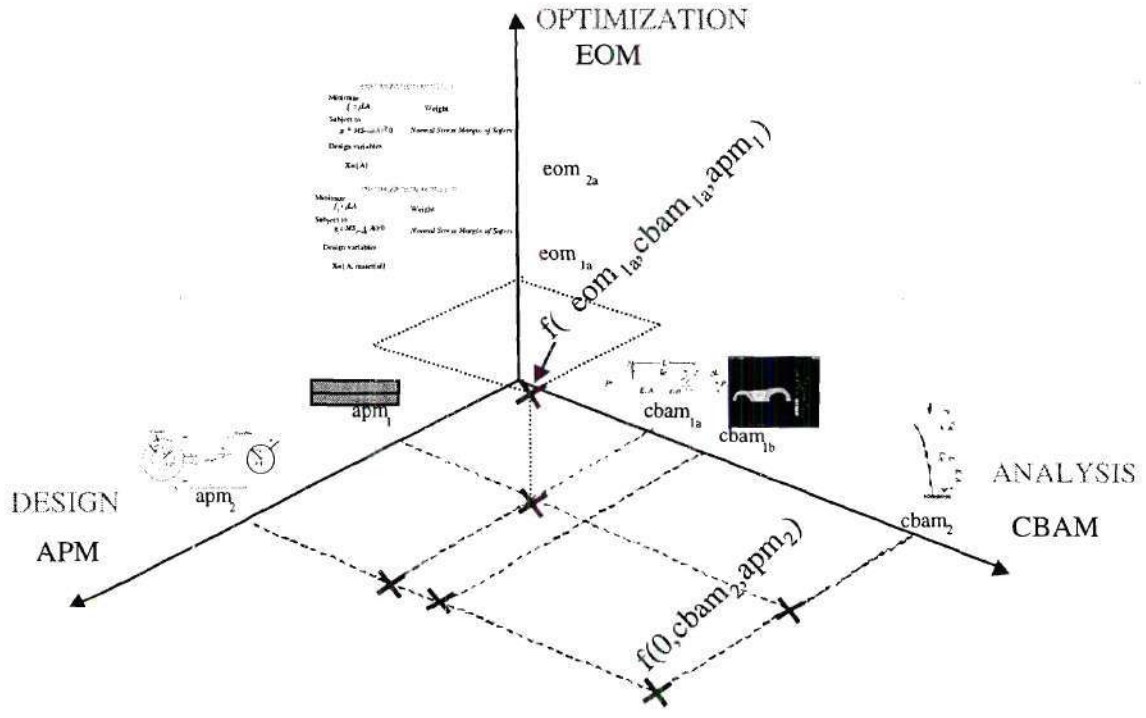


Figure 8-1 Design-Analysis-Optimization Progression Correlation with Cartesian View

- **Measuring the accuracy of the individual optimization models, analysis models and design models independently**

Usage of various fidelities of engineering models of optimization, analysis and design requires research about the degree of accuracy of those models. There is a need for an intelligent way of measuring the accuracy of the engineering models individually. A model is defined with estimated or tested accuracy as a base value; then, measurements of different fidelities of that representative model may be performed with respect to that base value.

- **Measuring the accuracy of the collective optimization model**

The optimization models utilize both the analysis and design models. The analysis models utilize the design models. The accuracy of design and analysis models affects the optimization model downstream. Replacing any model of analysis or design models with another fidelity model of the same type has an impact on the accuracy of the optimization model; further research is required to develop a method to determine the collective accuracy of the whole system.

- **EOM Integrity Check**

It is desirable to extend integrity check and debugging capabilities to the current EOM structure. EOM can facilitate performing automatic checks on optimization models. These include basic checks such as addition, deletion of attributes and relations, as well as syntax and logic checks such as objectives, constraints and design variables, ensuring that they are selected in a consistent manner. Furthermore, the addition of a debugging feature based on integrity checks in the model has an impact on designers who develop the eom models.

- **Developing of better Optimization Solution Method Model**

The analysis solution methods have been explored and developed with their main building blocks in Eislab. An analogy exists between optimization and analysis, and

further exploration in the area of optimization solution method modeling is indicated to increase the performance of EOM implementation. The faster and robust data transfer and execution of the tools increase the performance. The other extension related to OSMM can be solution assistance. EOM objects may facilitate solution assistance to analyze the input and output of solution tools (i.e. local minimum versus global min/max).

Finally, the functionality of automatic identification of optimization method tools based on eom schema can be extended to larger class of methods and tools.

- **Extend EOM capabilities to Multidisciplinary Optimization**

The cases presented in this thesis do not include the coupled interaction for the analyses. The basic building blocks of EOM have been built. Some of those building blocks can be extended to include higher complexity problems requiring multidisciplinary optimization solutions.

APPENDIX A

EOM SCHEMAS FOR FLAP LINK

EOM Schema for Flap Link (Preliminary) (1o2c1d case)

```
/*
Copyright 1999 by Georgia Tech
Engineering Information Systems Lab
eislab.gatech.edu

Author: Selcuk Cimtalay
*/

SCHEMA flap;

SOURCE_SET optimization ROOT_DOMAIN flap;

COB flap;

/* objective */

        obj: REAL;
/* constraints < 0 */

        g1: REAL;
        g2: REAL;
/* design variables */

        area : REAL;
/* analysis models */
        cbam1 : STRING;
        cbam2 : STRING;
RELATIONS
   r1 : "<area> ==<cbam1.deformation_model.area>";
   r2 :  "<area> ==<cbam2.deformation_model.area>";
   r3 : "<g1> == (1.0/(<cbam1.stress_mos_model.margin_of_safety> + 1.0))-1";
   r3 : "<g2> ==(1.0/(<cbam2.buckling_mos_model.margin_of_safety> +1.0))-1";
   r5 : "<obj> == (<area>*<cbam1.link.l>)*<cbam1.link.material.density>";
END_COB;

END_SOURCE_SET;

USE_FROM flap_link/selcuks' flap/1dcase/flap_link_cbams_buckling_mod.cos AS_SOURCE_SETS;

LINK_DEFINITIONS
  optimization.flap.cbam1 == cbams.link_extensional_model.name;
  optimization.flap.cbam2 == cbams.link_buckling_model.name;
END_LINK_DEFINITIONS;

END_SCHEMA;
```

EOM Schema for Flap Link (Preliminary)
(1o1c2d case)

```
/*
Copyright 1999 by Georgia Tech
Engineering Information Systems Lab
eislab.gatech.edu

Author: Selcuk Cimtalay
*/

SCHEMA Omep;

SOURCE_SET optimization ROOT_DOMAIN Omep;

COB Omep;

/* objective */

obj: REAL;

/* constraints < 0 */

g1: REAL;

/* design variables */
t : REAL;
w : REAL;


/* analysis models */
cbam1 : STRING;

RELATIONS
   r1 : "<t> == <cbam1.link.t>";
   r2 : "<w> == <cbam1.link.w>";
   r3 : "<g1> == (1.0/(<cbam1.stress_mos_model.margin_of_safety> + 1.0))-1";
   r4 : "<obj> == (<cbam1.link.area>*<cbam1.link.l>)*<cbam1.link.material.density>";

END_COB;

END_SOURCE_SET;

USE_FROM flap_link/selcuks' flap/2dcase/Ccbams.cos AS_SOURCE_SETS;

LINK_DEFINITIONS
  optimization.Omep.cbam1 == cbams.link_extensional_model.name;

END_LINK_DEFINITIONS;

END_SCHEMA;
```

EOM Schema for Flap Link (Preliminary) (1o2c2d case)

```
/*
Copyright 1999 by Georgia Tech
Engineering Information Systems Lab
eislab.gatech.edu

Author: Selcuk Cimtalay
*/

SCHEMA Omep;

SOURCE_SET optimization ROOT_DOMAIN Omep;

COB Omep;

/* OBJECTIVE < 0 */

obj: REAL;

/* constraints < 0 */

g1: REAL;
g2: REAL;

/* Design Variables */
t : REAL;
w : REAL;

/* analysis models */
    cbam1 : STRING;
    cbam2 : STRING;
RELATIONS
    r1 : "<t> == <cbam1.link.t>";
   r2 : "<w> == <cbam1.link.w>";
   r3 : "<t> == <cbam2.link.t>";
   r4 : "<w> == <cbam2.link.w>";
   r5 : "<g1> == (1.0/(<cbam1.stress_mos_model.margin_of_safety> + 1.0))-1";
   r6 : "<g2> ==(1.0/(<cbam2.buckling_mos_model.margin_of_safety> +1.0))-1";
   r7 : "<obj> == (<cbam1.link.area>*<cbam1.link.l>)*<cbam1.link.material.density>";
END_COB;

END_SOURCE_SET;

USE_FROM flap_link/selcuks' flap/2dcase/Ccbams.cos AS_SOURCE_SETS;

LINK_DEFINITIONS
  optimization.Omep.cbam1 == cbams.link_extensional_model.name;
  optimization.Omep.cbam2 == cbams.link_buckling_model.name;
END_LINK_DEFINITIONS;
END_SCHEMA;
```

EOM Schema for Flap Link (Detailed) with FE Analysis (1o1c1d case)

```
/*
Copyright 1999 by Georgia Tech
Engineering Information Systems Lab
eislab.gatech.edu

Author: Selcuk Cimtalay

*/

SCHEMA flap;

SOURCE_SET optimization ROOT_DOMAIN flap;

COB flap;
/* objectives  */
    obj: REAL;
/* constraints < 0 */
      g1: REAL;
/* analysis models */
     cbam3: STRING;
/* design variables */
      area : REAL;
 RELATIONS
  r1 : "<area> ==<cbam3.link.shaft.critical_cross_section.basic.area>";
  /*r2 : "<g2> == <cbam3.ux_mos_model.margin_of_safety> * -1"; */
   r3 : "<g1> == 1/(1+<cbam3.sx_mos_model.margin_of_safety>)-1";
   r4 : "<obj> == (<area>*<cbam3.link.effective_length>)*<cbam3.link.material.density>";

END_COB;

END_SOURCE_SET;

USE_FROM flap_link/FEoptimization/flap_link_cbams08.cos AS_SOURCE_SETS;

LINK_DEFINITIONS
 /*  optimization.flap.cbam1 == cbams.link_extensional_model.name; */
 /*  optimization.flap.cbam2 == cbams.link_buckling_model.name; */
   optimization.flap.cbam3 == cbams.link_plane_stress_model.name;


END_LINK_DEFINITIONS;

END_SCHEMA;
```

APPENDIX B

CBAM SCHEMA FOR FLAP LINK

SCHEMA flap_link_analysis_models;

Syntax: cob v2.1

Purpose:
The flap link example demonstrating typical CBAM structure described below. It currently has three flap_link-specific cbams to demonstrate supporting diverse modes, fidelity, and analysis tool usage:

- extension mode: 1D extensional rod model
- extension mode: 2D plane stress model (FEA-based)
- buckling mode: 1D Buckling rod model

This example schema uses three source_sets in the currently recommended manner for cbams:

1. cbam source set
   - specifies cbams specific to flap_link (in flap_link_cbams.cos - this file)
   - specifies associativity between apm and generic deformation_models and mos models
   - uses lib/abbs.cos
   - uses apm source set via USE_FROM construct
   - defines links to apm via part_number (see LINK_DEFINITIONS)
2. apm source set
   - specifies apm for flap_link (in flap_link_apm.cos)
   - uses common apm resources from lib/apm.cos etc.
   - uses materials source set via USE_FROM construct
   - defines links to materials via material id (see LINK_DEFINITIONS in the apm schema)
3. materials source set
   - specifies materials property data schema (in lib/materials.cos)
   - these materail cob instances would get populated largely from material databases
   - instances may be included in this directory as copies of those from the lib directory, or they may be read in directly from the lib directory

```
SOURCE_SET cbams ROOT_COB cbam;

USE_FROM lib\Selcuk_abbs.cos;


COB link_analysis_model SUBTYPE_OF cbam;
  /** "Subclasses relate stress mos with deformation models of varying fidelity. Demonstates cbam
concepts."; ==*/
  link : STRING;
  associated_condition : condition;
 END_COB;


COB link_extensional_model SUBTYPE_OF link_analysis_model;
  /*== "Represents 1D formula-based extensional model."; ==*/
  /*== MODE ==*/
  /* "compression"; */
```

```
   name: STRING; /* 032299 link to optimization */
   stress_mos_model : margin_of_safety_model;
   deformation_model : extensional_rod_isothermal;
 RELATIONS
   al1 : "<deformation_model.undeformed_length> == <link.l>";
   al2 : "<deformation_model.area> == <link.area>";
   al3 : "<deformation_model.material_model.youngs_modulus> ==
<link.material.stress_strain_model.linear_elastic.youngs_modulus>";
   al4 : "<deformation_model.material_model.name> == <link.material.name>";
   al5 : "<deformation_model.force> == <associated_condition.reaction>";
   al6 : "<stress_mos_model.allowable> == <link.material.yield_stress>";
   al7 : "<stress_mos_model.determined> == <deformation_model.material_model.stress>";

END_COB;

COB link_buckling_model SUBTYPE_OF link_analysis_model;
      /*== "Represents 1D formula-based extensional model."; ==*/
  /*== MODE ==:*/
      /* "tension"; */
   name: STRING; /* 032299 link to optimization */
   buckling_mos_model : margin_of_safety_model;
   deformation_model : pinned_pinned_column ;
 RELATIONS
   al1 : "<deformation_model.undeformed_length> == <link.l>";
   al2 : "<deformation_model.area> == <link.area>";
   al2 : "<deformation_model.moment_of_inertia> == <link.inertia>";
   al3 : "<deformation_model.material_model.youngs_modulus> ==
<link.material.stress_strain_model.linear_elastic.youngs_modulus>";
   al4 : "<deformation_model.material_model.name> == <link.material.name>";
 /*  al5 : "<deformation_model.inertia> == <associated_condition.reaction>"; */
   al6 : "<buckling_mos_model.allowable> == <deformation_model.critical_force>";
   al7 : "<buckling_mos_model.determined> == <associated_condition.reaction>";
END_COB;


COB link_plane_stress_model SUBTYPE_OF link_analysis_model;
   name: STRING; /* 051899 link to optimization */
  deformation_model : link_plane_stress_abb;
  sx_mos_model : margin_of_safety_model;
  ux_mos_model : margin_of_safety_model;
 /* wt_mos_model :  margin_of_safety_model;*/
RELATIONS
  al1: "<link.material.stress_strain_model.linear_elastic.youngs_modulus> == <deformation_model.ex>";
  al2: "<link.material.stress_strain_model.linear_elastic.poissons_ratio> == <deformation_model.nuxy>";
  al3: "<link.effective_length> == <deformation_model.l>";
  al5: "<link.sleeve1.width> == <deformation_model.ws1>";
  al6: "<link.sleeve1.wall_thickness> == <deformation_model.ts1>";
  al7: "<link.sleeve1.outer_diameter> == <deformation_model.rs1>*2.0";
  al8: "<link.sleeve2.width> == <deformation_model.ws2>";
  al9: "<link.sleeve2.wall_thickness> == <deformation_model.ts2>";
  al10: "<link.sleeve2.outer_diameter> == <deformation_model.rs2>*2.0";
 /* use critical_cross_section."basic" */
  al11: "<link.shaft.critical_cross_section.basic.web_thickness> == <deformation_model.tw>";
```

```
  al12: "<link.shaft.critical_cross_section.basic.flange_thickness> == <deformation_model.tf>";
  al13: "<link.shaft.critical_cross_section.basic.flange_width> == <deformation_model.wf>";
  al14: "<associated_condition.reaction> == <deformation_model.force>";
  al6 : "<sx_mos_model.allowable> == <link.material.yield_stress>";
  al7 : "<sx_mos_model.determined> == <deformation_model.sx>";
  al8 : "<ux_mos_model.allowable> == <link.allowable_inter_axis_length_change>";
  al9 : "<ux_mos_model.determined> == <deformation_model.ux>";
/*al10 :"<wt_mos_model.determined> == <deformation_model.wt>"; */
END_COB;

COB link_plane_stress_abb;
    l: REAL;        /* length */
    ws1: REAL;  /* width at sleeve1 */
    ts1: REAL;  /* thickness at sleeve1 */
    rs1: REAL;     /* radius Of sleeve1 (rs1<rs2) */
    ws2: REAL;    /* widthth at sleeve2 */
     ts2: REAL;    /* thickness of sleeve2 */
     rs2: REAL;    /* radius of sleeve2 */
     tw: REAL;     /* thickness at shaft_web */
     tf: REAL;      /* thickness of shaft */
     wf: REAL;     /* thickness at shaft_flange */
     force: REAL; /* total force */
      ex: REAL;    /* youngs modulus */
     nuxy: REAL; /* poisson's ration */
     ux: REAL;   /* deformation in x dir */
     sx: REAL;   /* stress in x dir */
     wt :REAL;    /*  objective function weight  Selcuk's addition*/
RELATIONS
  r1: "<ux> ==
CobExternalToolFunction[ansys,flaplink,ux,{<ex>,<nuxy>,<l>,<ws1>,<ws2>,<rs1>,<rs2>,<ts1>,<ts2>,<t
w>,<tf>,<wf>,<force>}]" ONEWAY;
  r2: "<sx> ==
CobExternalToolFunction[ansys,flaplink,sx,{<ex>,<nuxy>,<l>,<ws1>,<ws2>,<rs1>,<rs2>,<ts1>,<ts2>,<t
w>,<tf>,<wf>,<force>}]" ONEWAY;
r3: "<wt> ==
CobExternalToolFunction[ansys,flaplink,wt,{<ex>,<nuxy>,<l>,<ws1>,<ws2>,<rs1>,<rs2>,<ts1>,<ts2>,<t
w>,<tf>,<wf>,<force>}]" ONEWAY;
r4: "<tw> ==
CobExternalToolFunction[ansys,flaplink,tw,{<ex>,<nuxy>,<l>,<ws1>,<ws2>,<rs1>,<rs2>,<ts1>,<ts2>,<t
w>,<tf>,<wf>,<force>}]" ONEWAY;
END_COB;
END_SOURCE_SET;

USE_FROM flap_link\flap_link_apm.cos AS_SOURCE_SETS;

LINK_DEFINITIONS
        cbams.link_extensional_model.link == flap_link_geometric_model.flap_link.part_number;
        cbams.link_plane_stress_model.link == flap_link_geometric_model.flap_link.part_number;
        cbams.link_buckling_model.link == flap_link_geometric_model.flap_link.part_number;
END_LINK_DEFINITIONS;

END_SCHEMA
```

APPENDIX C

ABBs

```
SCHEMA abbs;

SOURCE_SET abbs ROOT_COB abb;

/*
Copyright 1998 by Georgia Tech
Engineering Information Systems Lab
eislab.gatech.edu

Description:
- abb library (analytical engineering concepts independent of product type)
- used in cbams, etc. via USE_FROM construct
- has basic structure for major types of abbs (after [Peak, 1993]) with placeholders as examples (see
Express-G view)

Syntax: cob v2.1
Authors:  Selcuk Cimtalay, M. Wilson, R. Peak */

COB abb;
/* description */
/* references */
/* variables */
/* subsystems */
/* RELATIONS */
END_COB;

COB analysis_system SUBTYPE_OF abb;
END_COB;

COB specialized_analysis_system SUBTYPE_OF analysis_system;
END_COB;

COB analysis_primitive SUBTYPE_OF abb;
END_COB;

COB material_model SUBTYPE_OF analysis_primitive;
  name : STRING;
  yield_stress, &sigma;<sub>2</sub> : REAL;
END_COB;

COB solid_material_model SUBTYPE_OF material_model;
END_COB;

COB fatigue_model SUBTYPE_OF solid_material_model;
  average_cycles_to_failure : REAL;
  load_frequency : REAL;
  mean_cyclic_temperature : REAL;
END_COB;
```

```
COB coffin_manson_model SUBTYPE_OF fatigue_model;
  total_cyclic_strain_range : REAL;
  elastic_cyclic_strain_range : REAL;
  plastic_cyclic_strain_range : REAL;
  fatigue_ductility_coefficient : REAL;
  fatigue_ductility_exponent, c : REAL;
  fatigue_strength_coefficient : REAL;
  fatigue_strength_exponent, b : REAL;
END_COB;

COB stress_strain_model SUBTYPE_OF solid_material_model;
END_COB;

COB viscoplastic_model SUBTYPE_OF stress_strain_model;
END_COB;

COB plastic_model SUBTYPE_OF stress_strain_model;
END_COB;

COB bilinear_plastic_model SUBTYPE_OF plastic_model;
END_COB;

COB elastic_model SUBTYPE_OF stress_strain_model;
END_COB;

COB one_D_linear_elastic_model SUBTYPE_OF elastic_model;
  youngs_modulus, E : REAL;
  poissons_ratio, &nu; : REAL;
  cte, &alpha; : REAL;
  shear_modulus, G : REAL;
  strain, &epsilon; : REAL;
  stress, &sigma; : REAL;
  shear_stress, &tau;  : REAL;
  shear_strain, &gamma; : REAL;
  thermal_strain, &epsilon;<sub>t</sub> : REAL;
  elastic_strain, &epsilon;<sub>e</sub>  : REAL;
  temperature_change, &Delta;L : REAL;
 RELATIONS
  r1 : "<shear_modulus> * ( 2 * (1 + <poissons_ratio> ) ) == <youngs_modulus> ";
  r2 : "<strain> == <elastic_strain> + <thermal_strain>";
  r3 : "<elastic_strain> == <stress> / <youngs_modulus>";
  r4 : "<thermal_strain> == <cte> * <temperature_change>";
  r5 : "<shear_strain> == <shear_stress> / <shear_modulus>";
END_COB;

COB one_D_linear_elastic_model_isothermal SUBTYPE_OF one_D_linear_elastic_model;
 RELATIONS
  r6 : "<temperature_change> == 0";
END_COB;

COB one_D_linear_elastic_model_noShear SUBTYPE_OF one_D_linear_elastic_model;
 RELATIONS
```

```
  r6 : "<shear_stress> == 0 ";
END_COB;


COB fluid_model SUBTYPE_OF material_model;
END_COB;

COB newtonian_fluid SUBTYPE_OF fluid_model;
END_COB;

COB analysis_body SUBTYPE_OF analysis_primitive;
END_COB;

COB continuum_body SUBTYPE_OF analysis_body;
  /* matter : material_model; */
  /* temperature : REAL; */
  /* position : REAL; */
  /* geometry : REAL; */
END_COB;

COB solid_continuum_body SUBTYPE_OF continuum_body;
END_COB;

COB fluid_continuum_body SUBTYPE_OF continuum_body;
END_COB;

COB rigid_body SUBTYPE_OF solid_continuum_body;
END_COB;

COB deformable_body SUBTYPE_OF solid_continuum_body;
END_COB;


COB plate SUBTYPE_OF deformable_body;
END_COB;

COB plane_stress SUBTYPE_OF deformable_body;
END_COB;

COB plane_strain SUBTYPE_OF deformable_body;
END_COB;

COB shell SUBTYPE_OF deformable_body;
END_COB;

COB membrane SUBTYPE_OF deformable_body;
END_COB;

COB slender_body SUBTYPE_OF deformable_body;
  undeformed_length, L<sub>0</sub> : REAL;
  reference_temperature, T<sub>0</sub> : REAL;
  temperature, T : REAL;
/* material_model : material_model */
```

```
 RELATIONS
  r1 : "<material_model.temperature_change> == <temperature> - <reference_temperature>";
END_COB;

COB bending_beam SUBTYPE_OF slender_body;
END_COB;

COB euler_beam SUBTYPE_OF bending_beam;
END_COB;

COB timoshenko_beam SUBTYPE_OF bending_beam;
END_COB;

COB thermal_bending_beam SUBTYPE_OF bending_beam;
    total_thickness, t : REAL;
    warpage: REAL;
    material_model: one_D_linear_elastic_model;
 RELATIONS
        r1: "<warpage> * <total_thickness> == <material_model.cte> *
<material_model.temperature_change> *<undeformed_length>**2 ";
END_COB;



COB extensional_rod SUBTYPE_OF slender_body;
/*
  description : "Assumes no distributed load (i.e., torque is uniform along the length)."
*/
  start, x<sub>1</sub> : REAL;
  end, x<sub>2</sub> : REAL;
  length, L : REAL;
  total_elongation, &Delta;L : REAL;
  force, F : REAL;
  area, A : REAL;
  material_model : one_D_linear_elastic_model_noShear;
 RELATIONS
  r1 : "<length> == <end> - <start>";
  r2 : "<total_elongation> == <length> - <undeformed_length>";
  r3 : "<material_model.strain> == <total_elongation> / <undeformed_length>";
  r4 : "<material_model.stress> == <force> / <area>";
END_COB;

COB extensional_rod_isothermal SUBTYPE_OF extensional_rod;
 RELATIONS
  r5 : "<material_model.temperature_change> == 0";
END_COB;

COB extensional_rod_noForce SUBTYPE_OF extensional_rod;
 RELATIONS
  r6 : "<force> == 0";
END_COB;
```

```
/* check to make sure correct */
COB torsional_rod SUBTYPE_OF slender_body;
/*
  description : "Assumes no distributed load (i.e., torque is uniform along the length)."
*/
  theta_start, &theta;<sub>1</sub> : REAL;
  theta_end, &theta;<sub>2</sub> : REAL;
  twist, &phi; : REAL;
  torque, T : REAL;
  radius, r : REAL;
  polar_moment_of_inertia, J : REAL;
  material_model : one_D_linear_elastic_model_isothermal;
 RELATIONS
  r1 : "<twist> == <theta_end> - <theta_start>";
  r2 : "<material_model.shear_strain> == <twist> * <radius> / <undeformed_length>";
  r3 : "<material_model.shear_stress> == <torque> * <radius> / <polar_moment_of_inertia>";
END_COB;

COB buckling_column SUBTYPE_OF slender_body;
/*
  "Represents buckling behavior of linear elastic slender bodies. It has subtypes for common end conditions
with specific end factor values";
*/
  critical_force, P<sub>cr</sub> :REAL;
  moment_of_inertia, I  : REAL;
  effective_length_factor, K :REAL;
  effective_length, L<sub>e</sub> :REAL;
  material_model : one_D_linear_elastic_model_isothermal;
  area : REAL; /*7_30_99*/
 RELATIONS
  r1 : "<critical_force> == PI**2 * <moment_of_inertia>*(<material_model.youngs_modulus> /
(<effective_length_factor>*<undeformed_length>))";
  r2 : "<effective_length> == <effective_length_factor> * <undeformed_length>";
END_COB;

COB pinned_pinned_column SUBTYPE_OF buckling_column ;
 RELATIONS
  r3 : " <effective_length_factor> == 1.0 ";
END_COB;

COB fixed_pinned_column SUBTYPE_OF buckling_column ;
 RELATIONS
  r3 : " <effective_length_factor> == 0.7 ";
END_COB;

COB fixed_fixed_column SUBTYPE_OF buckling_column ;
 RELATIONS
  r4 : " <effective_length_factor> == 0.5 ";
END_COB;

COB shear SUBTYPE_OF deformable_body;
END_COB;
```

```
COB discrete_primitive SUBTYPE_OF analysis_body;
END_COB;

COB energy_storage_primitive SUBTYPE_OF discrete_primitive;
END_COB;

COB multiport_I SUBTYPE_OF energy_storage_primitive;
END_COB;

COB uniport_I SUBTYPE_OF multiport_I;
END_COB;

COB point_mass SUBTYPE_OF uniport_I;
END_COB;

COB multiport_C SUBTYPE_OF energy_storage_primitive;
END_COB;

COB uniport_C SUBTYPE_OF multiport_C;
END_COB;

/* note: 'end' is a reserved word in express */
COB spring SUBTYPE_OF uniport_C;
  undeformed_length, L<sub>0</sub> : REAL;
  spring_constant, k : REAL;
  start, x<sub>1</sub> : REAL;
  end0, x<sub>2</sub> : REAL;
  length, L : REAL;
  total_elongation, &Delta;L : REAL;
  force, F : REAL;
 RELATIONS
  r1 : "<length> == <end0> - <start>";
  r2 : "<total_elongation> == <length> - <undeformed_length>";
  r3 : "<force> == <spring_constant> * <total_elongation>";
END_COB;

COB junction_structure SUBTYPE_OF discrete_primitive;
END_COB;

COB multiport_R SUBTYPE_OF junction_structure;
END_COB;

COB uniport_R SUBTYPE_OF multiport_R;
END_COB;

COB zero_junction SUBTYPE_OF multiport_R;
END_COB;

COB one_junction SUBTYPE_OF multiport_R;
END_COB;

COB rigid_connection SUBTYPE_OF one_junction;
END_COB;
```

```
COB uniport_R SUBTYPE_OF multiport_R;
END_COB;

COB frictional_contact SUBTYPE_OF uniport_R;
END_COB;

COB damper SUBTYPE_OF uniport_R;
END_COB;

COB specialized_analysis_body SUBTYPE_OF analysis_body;
END_COB;

COB analysis_variable SUBTYPE_OF analysis_primitive;
END_COB;

COB generalized_power_variable SUBTYPE_OF analysis_variable;
END_COB;

COB effort_variable SUBTYPE_OF generalized_power_variable;
END_COB;

COB force_variable SUBTYPE_OF effort_variable;
END_COB;

COB torque_variable SUBTYPE_OF effort_variable;
END_COB;

COB voltage_variable SUBTYPE_OF effort_variable;
END_COB;

COB temperature_variable SUBTYPE_OF effort_variable;
END_COB;

COB pressure_variable SUBTYPE_OF effort_variable;
END_COB;

COB stress_variable SUBTYPE_OF effort_variable;
END_COB;

COB power_variable SUBTYPE_OF generalized_power_variable;
END_COB;

COB flow_variable SUBTYPE_OF generalized_power_variable;
END_COB;

COB linear_velocity_variable SUBTYPE_OF flow_variable;
END_COB;

COB angular_velocity_variable SUBTYPE_OF flow_variable;
END_COB;

COB current_variable SUBTYPE_OF flow_variable;
```

```
END_COB;

COB heat_flow_rate_variable SUBTYPE_OF flow_variable;
END_COB;

COB volume_flow_rate_variable SUBTYPE_OF flow_variable;
END_COB;

COB strain_rate_variable SUBTYPE_OF flow_variable;
END_COB;

COB generalized_energy_variable SUBTYPE_OF analysis_variable;
END_COB;

COB generalized_momentum_variable SUBTYPE_OF generalized_energy_variable;
END_COB;

COB linear_momentum_variable SUBTYPE_OF generalized_momentum_variable;
END_COB;

COB angular_momentum_variable SUBTYPE_OF generalized_momentum_variable;
END_COB;

COB flux_linkage_variable SUBTYPE_OF generalized_momentum_variable;
END_COB;

COB pressure_impulse_variable SUBTYPE_OF generalized_momentum_variable;
END_COB;

COB momentum_density_variable SUBTYPE_OF generalized_momentum_variable;
END_COB;

COB energy_variable SUBTYPE_OF generalized_energy_variable;
END_COB;

COB q_type_energy_variable SUBTYPE_OF energy_variable;
END_COB;

COB translational_potential_energy_variable SUBTYPE_OF q_type_energy_variable;
END_COB;

COB p_type_energy_variable SUBTYPE_OF energy_variable;
END_COB;

COB translational_kinetic_energy_variable SUBTYPE_OF p_type_energy_variable;
END_COB;

COB generalized_displacement_variable SUBTYPE_OF generalized_energy_variable;
END_COB;

COB linear_displacement_variable SUBTYPE_OF generalized_displacement_variable;
END_COB;
```

```
COB angular_displacement_variable SUBTYPE_OF generalized_displacement_variable;
END_COB;

COB change_variable SUBTYPE_OF generalized_displacement_variable;
END_COB;

COB heat_energy_variable SUBTYPE_OF generalized_displacement_variable;
END_COB;

COB volume_variable SUBTYPE_OF generalized_displacement_variable;
END_COB;

COB strain_variable SUBTYPE_OF generalized_displacement_variable;
END_COB;

COB definition_variable SUBTYPE_OF analysis_variable;
END_COB;

COB margin_of_safety SUBTYPE_OF definition_variable;
END_COB;

COB factor_of_safety SUBTYPE_OF definition_variable;
END_COB;

COB definition_model SUBTYPE_OF analysis_primitive;
END_COB;

COB margin_of_safety_model SUBTYPE_OF definition_model;
/*
  description : "Represents safety by comparing a determined value vs. an allowable. Note this abb assumes
one wants determined < allowable."
  reference : "Shigley & Mitchell"
*/
/* variables */
  allowable : REAL;
  determined : REAL;
  margin_of_safety, MS : REAL;
 RELATIONS
  r1: "<margin_of_safety> == <allowable>/<determined> - 1 ";
END_COB;

COB temperature_dependent_linear_elastic_model;
END_COB;
/* other abstract classes */

COB condition;
  description: STRING;
/* various terms for similar concepts */
  reaction : REAL;
  load : REAL;
END_COB;
```

```
COB environmental_condition;
  description : STRING;
  temperature : REAL;
  reference_temperature : REAL;
END_COB;

COB cbam;
/* description */
/* references */
/* analysis context */
  /* part_feature */
  /* mode */
  /* conditions */
  /* objectives */
/* abb_subsystems */
/* RELATIONS */
/* associativity linkages */
END_COB;

END_SOURCE_SET;

END_SCHEMA;
```

APPENDIX D

APM SCHEMA FOR FLAP LINK (PRELIMINARY DESIGN)

```
SCHEMA flap_link_apm;

Copyright 1998 by Georgia Tech
Engineering Information Systems Lab
eislab.gatech.edu

Syntax: cob v2.1

Authors: Selcuk Cimtalay

SOURCE_SET flap_link_geometric_model ROOT_COB flap_link;

COB part;
  /*part_number : STRING;*/
  designer : STRING;
  material : STRING;
END_COB;

COB flap_link SUBTYPE_OF part;
  part_number : STRING;
  l : REAL;
  w : REAL;
  t : REAL;
  area :REAL;
  volume : REAL;
  weight : REAL;
  inertia: REAL;
RELATIONS
 PRODUCT_RELATIONS
  pr1 : "<area> == <w>*<t>";
  pr2 : "<volume> == <l>*<w>*<t>";
  pr3 : "<weight> == <volume>*<material.density>";
  pr3 : "<inertia> == ((<w>*<t>**3)/12.0)";

END_COB;

END_SOURCE_SET;

USE_FROM lib/materials.cos AS_SOURCE_SETS;

LINK_DEFINITIONS
        flap_link_geometric_model.flap_link.material == materials.material.name;

END_LINK_DEFINITIONS;

END_SCHEMA;
```

See [Tamburini, 1999] or [Wilson, 2000] for detailed flap link APM.

APPENDIX E

EOM, CBAM AND APM INSTANCE FORMS
Flap Link (Preliminary)
(1o2c1d case)

```
DATA;

INSTANCE_OF flap;
  /* Eom Data*/
  g1 : ?;
  g2 : ?;
  obj : ?;
  area : 1.0;
  cbam1: "huseyin";
  cbam2: "huseyin";

  cbam1.associated_condition.description : "flaps down" ;
  cbam1.associated_condition.reaction : 10000.0 ;
  cbam1.stress_mos_model.margin_of_safety : ? ;

END_INSTANCE;

END_DATA;


DATA;

/* Cbam Data */

INSTANCE_OF link_torsional_model;

  link : "XYZ-510";

  associated_condition.description : "2G dive" ;
  associated_condition.reaction : 5000 ;
  stress_mos_model.allowable : ? ;
  stress_mos_model.determined : ? ;
  stress_mos_model.margin_of_safety : ? ;
  twist_mos_model.allowable : ? ;
  twist_mos_model.determined : ? ;
  twist_mos_model.margin_of_safety : ? ;

  deformation_model.theta_start : ?;
  deformation_model.theta_end : ?;
  deformation_model.twist : ?;
  deformation_model.torque : ?;
  deformation_model.radius : ?;
```

```
  deformation_model.polar_moment_of_inertia: ?;

  deformation_model.undeformed_length : ?;
  deformation_model.reference_temperature : ?;
  deformation_model.temperature : ?;

  deformation_model.material_model.youngs_modulus : ? ;
  deformation_model.material_model.poissons_ratio : ? ;
  deformation_model.material_model.cte : ? ;
  deformation_model.material_model.strain : ? ;
  deformation_model.material_model.shear_modulus : ? ;
  deformation_model.material_model.stress : ? ;
  deformation_model.material_model.shear_stress : ? ;
  deformation_model.material_model.shear_strain : ? ;
  deformation_model.material_model.thermal_strain : ? ;
  deformation_model.material_model.elastic_strain : ? ;
  deformation_model.material_model.temperature_change : ? ;
  deformation_model.material_model.name : ? ;
  deformation_model.material_model.yield_stress : ? ;

END_INSTANCE;


INSTANCE_OF link_extensional_model;
  name : "huseyin";     /* 032399 for optimization link */
  link : "XYZ-510";

  associated_condition.description : "flaps down" ;
  associated_condition.reaction : 10000 ;
  stress_mos_model.allowable : ? ;
  stress_mos_model.determined : ? ;
  stress_mos_model.margin_of_safety : ? ;
  deformation_model.start : ?;
  deformation_model.end : ?;
  deformation_model.length : ?;
  deformation_model.total_elongation : ?;
  deformation_model.force : ?;
  deformation_model.area : ?;
  deformation_model.undeformed_length : ?;
  deformation_model.reference_temperature : ?;
  deformation_model.temperature : ?;
  deformation_model.material_model.youngs_modulus : ? ;
  deformation_model.material_model.poissons_ratio : ? ;
  deformation_model.material_model.cte : ? ;
  deformation_model.material_model.strain : ? ;
  deformation_model.material_model.shear_modulus : ? ;
  deformation_model.material_model.stress : ? ;
  deformation_model.material_model.shear_stress : ? ;
  deformation_model.material_model.shear_strain : ? ;
  deformation_model.material_model.thermal_strain : ? ;
  deformation_model.material_model.elastic_strain : ? ;
  deformation_model.material_model.temperature_change : ? ;
  deformation_model.material_model.name : ? ;
```

```
  deformation_model.material_model.yield_stress : ? ;

END_INSTANCE;

INSTANCE_OF link_buckling_model;
  name : "huseyin";     /* 032399 for optimization link */
  link : "XYZ-510";
  associated_condition.description : "flaps down" ;
  associated_condition.reaction : 10000 ;
  buckling_mos_model.allowable : ? ;
  buckling_mos_model.determined : ? ;
  buckling_mos_model.margin_of_safety : ? ;
  deformation_model.critical_force : ?;
  deformation_model.effective_length_factor : ?;
  deformation_model.effective_length : ?;
  deformation_model.moment_of_inertia: ?;
  deformation_model.area: ?;
  deformation_model.material_model.youngs_modulus : ? ;
  deformation_model.material_model.poissons_ratio : ? ;
  deformation_model.material_model.cte : ? ;
  deformation_model.material_model.strain : ? ;
  deformation_model.material_model.shear_modulus : ? ;
  deformation_model.material_model.stress : ? ;
  deformation_model.material_model.shear_stress : ? ;
  deformation_model.material_model.shear_strain : ? ;
  deformation_model.material_model.thermal_strain : ? ;
  deformation_model.material_model.elastic_strain : ? ;
  deformation_model.material_model.temperature_change . ? ;
  deformation_model.material_model.name : ? ;
  deformation_model.material_model.yield_stress : ? ;

END_INSTANCE;

END_DATA;


DATA;

/* Apm Data */

INSTANCE_OF flap_link;
part_number : "XYZ-510" ;
designer : "J. Smith" ;
material : "steel" ;
l :17.0;
w :1.0;
t :?;
volume : ?;
weight : ?;
area : ? ;
inertia: ?;
END_INSTANCE;
```

REFERENCES

Bathe, K., Oden, J. T., and Wunderlich, W., eds.,1997, *Formulations and Computational Algorithms in Finite Element Analysis: U.S.-Germany Symposium*, MIT Press, Cambridge, MA.

Balling, R. J., and Wilkinson, C. A., 1997, "Execution of Multidisciplinary Design Optimization Approaches on Common Test Problems," *AIAA Journal*, Vol 35, No. 1, pp 178-186.

Balling, R. J., and Sobieszczanski-Sobieski, J., 1996, "Optimization of Coupled Systems: A Critical Overview of Approaches," *AIAA Journal*, Vol 34, No. 1, pp 6-17.

Balling, R. J. and Sobieski, J., 1994, "An Algorithm for Solving the System-Level Problem in Multilevel Optimization," Paper No. AIAA 94-4333, *AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Panama City, FL, September 7-9, 1994.

Barr, A., and Feigenbaum, E. A., eds., 1981, *Handbook of Artificial Intelligence, Vol I*, William Kaufmann Inc., Los Altos, CA.

Bennet J. A., and Botkin, M. E., 1997, "A Multidisciplinary Framework for Preliminary Vehicle Analysis and Design," *Multidisciplinary Design Optimization: State of the Art,* SIAM.

Bloebaum C. L., Hajela, P., and Sobieszczanski-Sobieski, J., 1991, "Non-Hierarchic System Decomposition In Structural Optimization," *Engineering Optimization*, Vol 19, pp 171-186.

Booch G., 1991, *Object Oriented Design with Applications*, Benjamin/Cummings Pub. Co., Redwood City, California.

Booch, G., Rumbaugh, J., and Jacobson, I., 1999, *The Unified Modeling Language User Guide*, Addison-Wesley Pub. Co., Reading, MA.

Braun, R. D., and Kroo, I. M., 1997, "Development and Application of the Collaborative Optimization Architecture in a Multidisciplinary Design Environment", *Multidisciplinary Design Optimization: State of the Art,* SIAM.

Bravoco, R., and Yadav, S., 1985, "A Methodology to Model the Functional Structure of an Organization," *Computers in Industry*, Vol 6, pp 345-361 ($IDEF_0$).

Bravoco, R., and Yadav, S., 1985, "A Methodology to Model the Information Structure of an Organization," *The Journal of Systems and Software*, Vol 5, pp 59-71 ($IDEF_{1X}$).

Chandrasekhar, A., 1999, *Integrating Analyzable Product Models with Geometric CAD Models*, Masters Thesis, George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA.

Chen, P. P., ed., 1979, *Proc. Intl. Conf. on Entity-Relationship Approach to Systems Analysis and Design*, Los Angeles CA.

Cimtalay, S., Peak, R. S.; and Fulton, R. E., 1996, "Optimization of Solder Joint Fatigue Life Using Product Model-Based Analysis Models," Application of CAE/CAD to Electronic Systems, D. Agonafar, et al., eds., *Proc. 1996 ASME Intl. Mech. Engr. Congress & Expo.*, Atlanta, GA, EEP-Vol 18, pp 47-54.

Cimtalay, S., and Fulton, R. E., 1994, "Parameter Design of Heat Sink: Multiple Trade-Offs," Application of CAE/CAD to Electronic Systems, *Proc. ASME WAM*, November 10, 1994, Chicago, Illinois, EEP-Vol 9, pp 53-58.

Cimtalay, S., and Fadel, G. M., 1993, "Automatic Evaluation of Move-Limits in Structural Optimization", *Structural Optimization*, Vol 6, pp 233-237.

Deitel , H. M., and Deitel, P. J., 1998, *Java: How to Program*, Prentice Hall, Upper Saddle River, New Jersey.

Eldred M. S., Outka, D. E., Fulcher, C. W., and Bohnhoff, W. J., 1995 "Optimization of Complex Mechanics Simulations with Object-Oriented Software Design," *AIAA J*, Sandia National Laboratories, Albuquerque, NM, pp 2406-2412.

Engelmaier, W., 1983, "Fatigue Life of Leadless Chip Carrier Solder Joints During Power Cycling," *IEEE Trans. on Components, Hybrids, and Manufacturing Technology*, Vol CHMT-6, No. 3, Sept. 1983, pp 232-237.

Engelmaier, W., 1989, "Thermal-Mechanical Effects," In M. L. Minges, ed., *Electronics Material Handbook. Vol 1 -Packaging*, ASM International, Materials Park, OH, pp 740-753.

Esping, B., 1995, "Design Optimization as an Engineering Tool", *Structural Optimization*, Vol 10, No. 3-4, pp 137-152.

Fujita, K., Akagi, S., and Tsujimoto, K., 1992, "Automated Optimal Shape Design System of Planar Link Mechanisms using an Object-Oriented Modeling Method," *Proceedings of the 1992 ASME Design Automation Conference*, DE-44-2, (Sept. 1992), pp. 21-28, Scottsdale, Arizona, U.S.A.

Goldberg, N., 1984, "Narrow Channel Forced Air Heat Sink," *IEEE Transactions, Components, Hybrids and Manufacturing Technology*, Vol CHMT, No. 1, pp 154-159.

Haftka, R., Gurdal, Z., and Kamat, M., 1990, *Elements of Structural Optimization*, Kluwer Academic Publishers, Boston.

Hale, M. A., Craig, J. I., Mistree, F., and Schrage, D. P., 1996, "DREAMS & IMAGE: A Model and Computer Implementation for Concurrent, Life-Cycle Design of Complex

Systems," *Concurrent Engineering: Research and Applications*, Vol 4, No. 2, pp 171-186.

Incropera, F. P., and DeWitt, D. P., 1990, *Fundamentals of Heat And Mass Transfer*, Wiley, New York.

ISO 10303-11, 1994, *Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 11: Description Methods: The EXPRESS Language Reference Manual*, International Organization for Standardization.

Kroo, I. M., 1997, "Decomposition and Collaborative Optimization for Large-Scale Aerospace Design Programs", In N. Alexandrov and M. Y. Hussaini, eds., *Multidisciplinary Design Optimization: State of the Art*, SIAM.

Kamat, M., 1993, *Structural Optimization: Status and Promise*, American Institute of Aeronautics and Astronautics, Washington, DC.

Knight, R. W., Gooling, J.S., and Hall, D. J., 1991, "Optimal Design Of Forced Convection Heat Sinks-Analytical", *Journal of Elect. Pack.*, Vol 113, pp 313-321.

Kodiyalam, S., Graichen, C. M., Connel, I. J., and Finnigan, P. M., 1994, "Object-Oriented, Optimization-Based Design of Satellite Structures," *Journal of Spacecraft and Rockets*, Vol 31, pp 312-318.

Kusiak, A., and Larson, N., 1995, " Decomposition and Representation Methods in Mechanical Design", *Journal of Mechanical Design*, Vol 117, pp 17-22.

Mistree, F., Hughes O. F., and Bras, B., 1993, "The Compromise Decision Support Problem And Adaptive Linear Programming Algorithm," In M. P. Kamat, ed., *Structural Optimization : Status and Promise*, AIAA, Washington, D.C., pp 247-286.

More, J., 1994, "Generalizations of the Truss Region Problem," *Optimization Methods and Software*, Vol 2, No. 3-4, 1994, pp 189-209.

Oktay, S., Hanneman, R., and Bar-Cohen, A.,1986, "High Heat From A Small Package", *Mech. Eng.*,Vol 108, No. 3, pp 36-42.

Peak, R. S., 1993, *Product Model-Based Analytical Models (PBAMs): A New Representation of Engineering Analysis Models*, Doctoral Thesis, George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA.

Peak, R. S., and Fulton, R. E., 1993a, "Automating Routine Analysis in Electronic Packaging Using Product Model-Based Analytical Models (PBAMs), Part I: PBAM Overview," *ASME Winter Annual Meeting*, 93-WA/EEP-23.

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

Peak, R. S., and Fulton, R. E., et al., 1998, "Integrating Engineering Design and Analysis Using a Multi-Representation Approach," *Engineering with Computers*, Vol 14, pp 93-114.

Peak, R. S., and Fulton, R. E., et al., 1999, "Phase I Report: Pilot Demonstration of STEP-based Stress Templates," GIT Project Report, CALS Technology Center's Engineering Information Systems Laboratory, Atlanta, GA (Report No. E-15-647).

Prather, M. H., and Amador, R. A., 1997, "Product Simulation Integration for Structures," Internal, The Boeing Commercial Airplane Group (BCAG), Seattle, WA.

Press, et al., ed., 1992, "Numerical Recipes in Fortran 90 : The Art of Parallel Scientific Computing," *Fortran Numerical Recipes, Vol 2*, Cambridge University Press.

Rogers, J. L., 1990, "Knowledge-Based Tool for Decomposing Complex Design Problems," *Journal of Computing in Civil Engineering*, Vol 4, No. 4, pp 298-312.

Rumbaugh J., 1991, *Object Oriented Modeling and Design*, Englewood Cliffs, New Jersey.

Schenck, D., and Wilson, P., 1994, *Information Modeling the EXPRESS Way*, Oxford University Press, New York, NY.

Scholand, A. J., Peak, R. S., et al., 1997, "The Engineering Service Bureau - Empowering SMEs to Improve Collaboratively Developed Products," CALS Expo USA, 1997.

Sobieszczanski-Sobieski, J., 1988, *On the Sensitivity of Complex, Internally Coupled Systems*, NASA Langley Research Center, January 1988, NASA Technical Memorandum 101537.

Sobieszczanski-Sobieski, J., September 1988, *Optimization by Decomposition: A Step from Hierarchic to Non-Hierarchic Systems*, NASA Langley Research Center, NASA Technical Memorandum 101494.

Tamburini, D. R., 1999, *The Analyzable Product Model Representation to Support Design-Analysis Integration*, Doctoral Thesis, George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA.

Tiller M. M., and Dantzig, J. A., 1996, "Implementation of Design Sensitivity Analysis and Numerical Optimization in Engineering Analysis," *Appl. Math. Modelling*, Vol. 20, pp 792-799.

U. S. Air Force, 1981, *Integrated Computer Aided Manufacturing (ICAM) Architecture Part II, Volume IV, Volume -Functional Modeling Manual (IDEF0),* Air Force Materials Laboratory, Wright-Patterson AFB, Ohio 45433 AFWAL-tr-81-4023.

Vanderplaats, G. N., 1978, *CONMIN- A FORTRAN program for Constraint Function Minimization -User's Manual*, NASA TM X-62282.

Vanderplaats, G. N., 1984, *Numerical Optimization Techniques for Engineering Design: With Applications*, McGraw-Hill, New York.

Wilde, D. J., 1978, *Globally Optimal Design*, John Wiley & Sons, New York.

Wilson, M. W., 2000, *The Constrained Object (COB) Representation for Engineering Analysis Integration*, Masters Thesis, George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA.

VITA

Selçuk Cimtalay was born in Izmir, Turkey on February 6, 1964. Mr. Cimtalay spent his entire childhood and adolescence in Izmir, where he attended Kemal Reis for his elementary and Dokuz Eylul for his secondary years. He graduated from Izmir Ataturk High school in 1981. He started his university education in September of the same year at the Middle East Technical University, Ankara, Turkey. He was an intern for two academic terms at EgeMosan and Raks. He received his B.S. degree in Mechanical Engineering in July of 1987.

From January of 1988 until 1990, Mr. Cimtalay worked at Anadolu University as a Research Assistant Engineer in Mechanical Engineering, and received his M.S. degree in Mechanical Engineering from that institution in July of 1990. Afterwards, he received scholarships from Von Karman Institute in Brussels, Belgium, and later from the Turkish Ministry of Education, to pursue a Ph.D. program in Mechanical Engineering.

In September of 1990, Mr. Cimtalay started his graduate studies at the Georgia Institute of Technology in Atlanta, Georgia (United States of America), and received his Masters Degree in Mechanical Engineering in December of 1992. In September of 1992 he began his doctoral studies in Mechanical Engineering, also at the Georgia Institute of Technology, which he completed, with this thesis, in the Fall Semester of 2000. From 1997 until the end of 1999, he worked for Georgia Perimeter College in Atlanta as a Mathematics Instructor.