January 2007

# Ontologies and Methods for Interoperability of Engineering Analysis Models (eams) in an E-Design Environment

Neelima Kanuri
*University of Massachusetts Amherst*

# ONTOLOGIES AND METHODS FOR INTEROPERABILITY OF ENGINEERING ANALYSIS MODELS (EAMS) IN AN E-DESIGN ENVIRONMENT

**A Thesis Presented**

**By**

**NEELIMA KANURI**

**Submitted to the Graduate School of the**
**University of Massachusetts Amherst in partial fulfillment**
**of the requirements for the degree of**

**MASTER OF SCIENCE IN MECHANICAL ENGINEERING**

**September 2007**

**Mechanical and Industrial Engineering**

# ONTOLOGIES AND METHODS FOR INTEROPERABILITY OF ENGINEERING ANALYSIS MODELS (EAMS) IN AN E-DESIGN ENVIRONMENT

**A Thesis Presented**

**By**

**NEELIMA KANURI**

**Approved as to Style and content by:**

_____
**Ian R. Grosse, Chair**

_____
**Jack C. Wileden, Member**

_____
**Sundar Krishnamurty, Member**

_____
**Mario Rotea, Department Head**
**Department of Mechanical and Industrial Engineering**

## ACKNOWLEDGEMENTS

I would like to thank my advisors, Prof. Grosse and Prof. Wileden, for their continued support during my Master's tenure. Their guidance has always been appreciated, without which I could have not have overcome many of the difficult obstacles throughout my research. I would also like to thank Prof. Krishnamurty for agreeing to serve as a member on my thesis committee and also providing useful input into my ongoing research.

I would like to thank Wei-Shan Chiang, CTO-FiPER and Ron Hodgin from Engineous Software Inc., for providing all the technical help associated with their products FiPER and I-SightFD.

I would like to extend my gratitude to Raytheon, Engineous, and the National Science Foundation for making my research possible, as well as all those who are involved with the Center for e-Design. I would like to thank undergraduate computer science students John and Scott for being extremely helpful during the course of work.

I would also like to thank my labmates- Tiefu, Brian, Paul, and Lieselle for their companionships during these last two years. A special thanks to my family for their love and support.

## ABSTRACT

### ONTOLOGIES AND METHODS FOR INTEROPERABILITY OF ENGINEERING ANALYSIS MODELS (EAMS) IN AN E-DESIGN ENVIRONMENT

**SEPTEMBER 2007**

**NEELIMA KANURI, B.S., BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCES PILANI INDIA**

**M.S., UNIVERSITY OF MASSACHUSETTS AMHERST**

**Directed by: Professor Ian Grosse**

Interoperability is the ability of two or more systems to exchange and reuse information efficiently. This thesis presents new techniques for interoperating engineering tools using ontologies as the basis for representing, visualizing, reasoning about, and securely exchanging abstract engineering knowledge between software systems. The specific engineering domain that is the primary focus of this report is the modeling knowledge associated with the development of engineering analysis models (**EAMs**). This abstract modeling knowledge has been used to support integration of analysis and optimization tools in iSIGHT FD[1], a commercial engineering environment. ANSYS[2], a commercial FEA tool, has been wrapped as an analysis service available inside of iSIGHT-FD.


[1] **iSIGHT-FD is developed and distributed by Engineous Software Inc., Cary, NC**
[2] **ANSYS is developed and distributed by ANSYS, Inc., Canonsburg, PA**

Engineering analysis modeling (**EAM**) ontology has been developed and instantiated to form a knowledge base for representing analysis modeling knowledge. The instances of the knowledge base are the analysis models of real world applications. To illustrate how abstract modeling knowledge can be exploited for useful purposes, a cantilever I-Beam design optimization problem has been used as a test bed proof-of-concept application. Two distinct finite element models of the I-beam are available to analyze a given beam design- a beam-element finite element model with potentially lower accuracy but significantly reduced computational costs and a high fidelity, high cost, shell-element finite element model. The goal is to obtain an optimized I-beam design at minimum computational expense. An intelligent KB tool was developed and implemented in FiPER[3]. This tool reasons about the modeling knowledge to intelligently shift between the beam and the shell element models during an optimization process to select the best analysis model for a given optimization design state.

In addition to improved interoperability and design optimization, methods are developed and presented that demonstrate the ability to operate on ontological knowledge bases to perform important engineering tasks. One such method is the automatic technical report generation method which converts the modeling knowledge associated with an analysis model to a flat technical report. The second method is a secure knowledge sharing method which allocates permissions to portions of knowledge to control knowledge access and sharing. Both the methods acting together enable recipient specific fine grain controlled knowledge viewing and sharing in an engineering workflow integration environment, such as iSIGHT-FD.

[3] **FiPER is developed and distributed by Engineous Software Inc., Cary, NC**

These methods together play a very efficient role in reducing the large scale inefficiencies existing in current product design and development cycles due to poor knowledge sharing and reuse between people and software engineering tools. This work is a significant advance in both understanding and application of integration of knowledge in a distributed engineering design framework.

## TABLE OF CONTENTS

**Page**

ACKNOWLEDGEMENTS ..... iv

ABSTRACT ..... v

LIST OF TABLES ..... xi

LIST OF FIGURES ..... xii

CHAPTER

1. INTRODUCTION AND OBJECTIVES ..... 1

   1.1 Introduction ..... 1
   1.2 Objective ..... 4

2. REVIEW OF RELATED WORK ..... 6

3. ONTOLOGIES FOR REPRESENTING ENGINEERING ANALYSIS MODELING KNOWLEDGE ..... 11

4. IMPLEMENTATION OF THE OBJECT ORIENTED METHODS IN PRODUCT DEVELOPMENT ENVIRONMENT ..... 19

5. SECURED SHARING OF ANALYSIS MODELING KNOWLEDGE ..... 36

6. INTEROPERABILITY OF MODELING KNOWLEDGE BETWEEN ANALYSIS OPTIMIZATION AND DECISION MAKING TOOLS IN A PRODUCT DEVELOPMENT ENVIRONMENT ..... 43

   6.1 Analysis tool in product development environment ..... 44
   6.2 Optimization tool in product development environment ..... 47
   6.3 Developing Intelligent Decision making tool in product development environment ..... 48

7. TEST BED APPLICATION ..... 50

   7.1 General problem formulation ..... 50
   7.2 Development of the analysis beam element model for I-Beam ..... 51
   7.3 Development of the analysis shell element model for I-Beam ..... 53
   7.4 Tradeoff between the beam element model and the shell element model ..... 55
   7.5 Theoretical calculation of bending and twisting deflections of I-beam ..... 66

8. FORMULATION OF I-BEAM OPTIMIZATION PROBLEM IN ISIGHT-FD ..... 70

8.1 Developing EAM ontology instance for the I-Beam beam and shell element model ..... 70
8.2 Configuring analysis and optimization components in iSIGHT-FD for I-Beam ..... 71

8.2.1 Setting up the analysis component in iSIGHT-FD ..... 71
8.2.2 Setting up the optimization component in iSIGHT-FD ..... 74
8.2.3 Setting up the intelligent decision making tool in iSIGHT-FD ..... 78
8.2.4 Setting up conditional workflows in iSIGHT-FD ..... 81

8.3 Complete integration of the tools ..... 82
8.4 Passing required analysis modeling knowledge for interoperating between analysis and optimization tools ..... 83
8.5 Integrating I-beam modeling knowledge to a CAD-integrated environment,ENCAPTA ..... 87
8.6 Overall setup of the optimization process ..... 90

9. RESULTS ..... 93

9.1 Summary of the results: ..... 107

10. VALIDATION OF TEST BED APPLICATION RESULTS ..... 109

11. VALIDATION OF OWL EAM ONTOLOGY, TECHNICAL REPORT COMPONENT AND SECURED KNOWLEDGE SHARING COMPONENT WITH INDUSTRIAL APPLICATION ..... 114

11.1 Catheter clamp analysis model ..... 115
11.2 Developing EAM ontology instance for the catheter open clamp model ..... 118
11.3 Generation of Technical report in iSIGHT-FD using catheter clamp analysis modeling knowledge ..... 119
11.4 Secured sharing of catheter clamp analysis modeling knowledge using iSIGHT-FD ..... 124
11.5 Validation ..... 125

12. SUMMARY ..... 129

13. FUTURE WORK ..... 133

APPENDICES

A. CUSTOMIZED GENERATION OF TECHNICAL REPORTS ..... 135
B. ANSYS COMMAND FLIES FOR BEAM AND SHELL ELEMENT MODELS ..... 143

BIBLIOGRAPHY ..... 148

## LIST OF TABLES

**Table** **Page**

1. Properties of Engineering Analysis Models ..... 14

2. Results for von Mises stress and deflection for varying length/width ratios for a constant length/height ratio of 25 and thickness of 0.5 ..... 57

3. Results for von Mises stress and deflection for varying length/height ratios for a constant length/width ratio of 25 and thickness of 0.5 ..... 58

4. Required model based on accuracy expectation and eccentricity of the load for varying l/h ratios, l/w=25 and thickness=0.5 ..... 63

5. Required model based on the accuracy expectation and eccentricity of the load for varying l/w ratios, l/h=25 and thickness=0.5 ..... 65

6. Results obtained for bending and total deflections using theoretical calculations ..... 68

7. Optimization results for example problem 1. ..... 95

8. Optimization results for sample problem 2 ..... 98

9. Optimization results for sample problem 3 ..... 100

10. Optimization results for sample problem 4 ..... 102

11. Optimization results for sample problem 5 ..... 105

12. Optimization results for all the sample problems ..... 107

13. Formulation of sample problems for the shell element optimization problem ..... 110

14. Optimization results obtained for an I-beam optimization problem ..... 111

15. Percentage of shell element model used vs. error in the processes ..... 112

## LIST OF FIGURES


**Figure** **Page**

1. Class hierarchy of some engineering analysis model types. The proposed ontology is applicable to all physics-based engineering analysis models. ..... 13
2. Implementation of ontology for representing Analysis Modeling Knowledge ..... 15
3. Class tree of EAM ontology ..... 16
4. Instances of the EAM ontology class generated are CCSB-2-D model, lateral elevation model and simple shelf bracket. ..... 17
5. Flat technical report generated for the CCSB-2-D model ..... 18
6. Protégé component working in FiPER environment. ..... 19
7. Protégé editor accessed from within FiPER environment ..... 20
8. Selecting an ontology from the Protégé editor in the FiPER environment..... 20
9. Parallel class of "Tech report EAM" created for technical report generation method……..……22
10. Accessing knowledge using "InteroperateOWLKB" component..... 28
11. Selecting instance and inspecting the knowledge ..... 30
12. Generating technical reports for multiple instances..... 33
13. Creating multiple technical reports using the Tech-report method ..... 34
14. Parallel class of "Permissions" and corresponding slot permissions which define the permission level of each slot of the knowledge base. ..... 38
15. Access to instances before including knowledge access permission code ..... 39
16. Access to instance knowledge after including knowledge access level control ..... 40
17. Implementation of permission based knowledge control for generating documentation of instance knowledge ..... 41

18. SIMCODE component for running ANSYS in product development environment, iSIGHT-FD ..... 45

19. Editor of the ANSYS simcode component in iSIGHT-FD ..... 46

20. Editor of Script component in iSIGHT-FD ..... 49

21. Cross section of I-Beam Showing Eccentric Load ..... 51

22. Orientation of elements and nodes for I-Beam model using element BEAM188 ..... 52

23. Orientation of elements and nodes for I-Beam model using element SHELL181 ... 54

24. Results for accuracy percentage of beam element model vs. (length/height) ratio for constant length/width ratios of 25 and 35 for the I-beam model shown ..... 60

25. Results for accuracy percentage of beam element model vs. (length/width) ratio for constant length/height ratios of 20 and 30 ..... 61

26. Cross section of I-beam, deflections due to bending and eccentricity ..... 68

27. Instance for beam and shell element analysis models of I-beam ..... 71

28. Editor of SIMCODE component ..... 73

29. Output file and mapping output parameters ..... 73

30. Parameters for the SIMCODE component ..... 74

31. Editor of the optimization component ..... 76

32. Design variables for the optimization component ..... 77

33. Constraints for the optimization component ..... 77

34. Objective for the optimization component ..... 78

35. Script component editor ..... 79

36. Parameters for the “Intelligent EAM selector”component-Configuring the “Intelligent EAM selector” component for optimization of I-beam in iSIGHT-FD ..... 80

37. Setting up conditional workflow in iSIGHT-FD ..... 81

38. Set up of the optimization process ..... 82

39. Mapping of parameters-Setup of the optimization component using the ANSYS SIMCODE component and intelligent decision making tool ..... 82

40. Integration of components ..... 85

41. Accessing the I-beam modeling instance- Passing ontological modeling knowledge as parameters in a product development environment ..... 85

42. Selecting parameters ..... 86

43. Input and output parameters from the knowledge base ..... 86

44. Version of EAM ontology within the EnCapta framework ..... 88

45. Creation of a geometrical model for I-beam ..... 89

46. Adding geometry links from CAD model to EAM ontology ..... 89

47. Encapta knowledge represented in Protégé OWL ..... 90

48. Flowchart for optimization process of I-beam using optimization component, ANSYS SIMCODE component and intelligent decision making tool ..... 91

49. Optimization process for I-beam switching between the shell and the beam element models based on intelligent tool decision……..……96

50. Setup of shell element model optimization process..... 109

51. Simple catheter clamp..... 115

52. Analysis model of the catheter open clamp ..... 116

53. Nodes in orange show the edges held fixed during the analysis..... 116

54. Two different insertion positions ..... 117

55. Analysis modeling knowledge of catheter clamp instance within Protégé environment ..... 119

56. Inspecting instance knowledge of the Catheter clamping-open clamp analysis model……..120

57. Generating technical reports for multiple instances..... 123

58. Accessing instance knowledge with secured knowledge sharing..... 125

59. Permission for the instances for the generation of technical reports ..... 135

60. Permission levels 2,3 and 2 selected by the user who has permission level of 1 ... 136

61. Technical report for CCSB_model instance generated with permission level 3 .... 137

62. Technical report for EAM-ontology_instance_1000 instance generated with permission level 2 ..... 137

63. Technical report for EAM-version_B_00440 instance generated with permission level 2 ..... 138

64. Permission levels 3, 1 and 3 selected by the user who has permission level of 1 ..... 139

65. Technical report for CCSB_model_1 instance generated with permission level 1..... 139

66. Technical report for EAM-ontology_instance_1000 instance generated with permission level 2 ..... 140

67. Multiple instances for generating technical reports ..... 141

68. Selecting the ccsb-model-1 instance for generating technical report ..... 141

69. Appending multiple instances for the technical report ..... 142

# CHAPTER 1

# INTRODUCTION AND OBJECTIVES

## 1.1 Introduction

Engineering product design is the process of developing a system, component or process to satisfy required objectives. During this process engineers communicate with various types of information like form, behavior and function. It is a decision making process in which engineering and mathematical sciences are applied to convert resources optimally to meet a stated objective. During the design of an engineered product, engineers rely heavily on engineering analysis models to inform design decisions. With the increasing ubiquity and relative low cost of computer hardware and software, manufacturing companies are relying more and more on engineering analysis models (EAMs) of engineered products to inform product development. Various modeling assumptions, idealizations and justifications or supporting rationale are made during the development of analysis products. Thus, the development of an EAM is a cognitive, knowledge-based activity. Unfortunately, there are no current systems that provide efficient mechanisms to capture and share this knowledge in a computational environment. Since product development is evolutionary by nature, EAMs are evolutionary and ought to be reused and adapted for similar applications as much as possible to reduce model development time and costs.

Product evolution might require changes in the design requirement, test conditions or a change in the process of manufacturing. If designers can evaluate the

effect of change in one of the above mentioned criteria, and notify the end user in an appropriate way, then reusability and adaptation of previously developed EAM's can reduce product development time and cost. The exchange of only the input and output files of a model is not sufficient for reusability of the product or analysis model in a new environment. The transfer of knowledge between different tools has been difficult for engineers during the past because of the fact that engineers should enter the information about the processes separately in the format that the tools can understand. This has often led to inefficient exchange of knowledge.

Interoperability is defined as the ability of two or more systems to exchange information and use the information that has been exchanged [IEEE, 1990]. For organizations to collaborate productively, they must be able to produce and use interoperable knowledge. Interoperable knowledge is the knowledge which can be easily exchanged between different software packages. One major problem with the emergence of various heterogeneous CAD systems is the lack of interoperability among these systems. A 1999 study by RTI International [NIST 1999] estimated that imperfect interoperability imposes costs of at least $1 billion per year on the U.S. automotive supply chain alone; other industries may have similar costs associated with lack of interoperability among CAD/CAE/CAM applications. As the complexity of products increase and gets distributed, rather than the traditional CAD and analysis tools, new software tools with a suitable environment for supporting interoperability between existing tools are required.

Engineering knowledge that is brought to bear on engineering tasks has been identified as knowledge regarding geometry, time, engineering principles, mathematical

formulations, experimental knowledge, standards, constraints, goals, objectives, qualitative behavior etc.[Dym et al, 1991]. Since there have been no current systems that provide efficient mechanisms to capture and share modeling knowledge effectively, a system for describing such abstract knowledge about models has to be developed. This system should provide support for the representation and reasoning of design and engineering knowledge for rapidly evolving products and processes.

One of the most promising ways to separate and represent domain knowledge from operational or problem solving knowledge is through the use of ontologies. We have taken the definition of an ontology as a formal explicit description of domain concepts. Ontologies are used to facilitate knowledge sharing, reuse, agent interoperability and knowledge acquisition. [Noy et al, 2001] elaborate to define an ontology as a formal explicit description of domain concepts. These domain concepts include classes, their properties called slots, and the instances of classes, which have knowledge associated with the properties of a class. An ontology for representing and sharing engineering analysis modeling (EAM) knowledge in a web-based environment has been developed by Grosse et al, 2005 and implemented into a computational knowledge base system, called ON-TEAM, using Protégé [Protégé, 2005]. Methods can be developed to operate on the knowledge base systems to facilitate reuse and interoperability of engineering analysis models. These methods when integrated in a product development environment like iSIGHT-FD, help in developing intelligent decision making tools for automating the processes of analysis and optimization.

## 1.2 Objectives

The main objective of this work is to develop methods that facilitate reuse, adaptation and interoperability of engineering analysis models through the use of computational knowledge base ON-TEAM. To achieve this objective, firstly, methods that operate on the knowledge base to extract knowledge associated with ON-TEAM have been developed. Later, methods that use knowledge associated with ON-TEAM to intelligently drive tools for developing automated systems have been developed.

One method that operates on knowledge associated with ON-TEAM is the flat technical report generation method. This method describes the properties or class relationships of an engineering modeling analysis class or the modeling knowledge involved in the development of a specific engineering analysis model. It is a JAVA application that accesses the EAM knowledge base application using the Protégé application programming interface (API). It presents the user a graphical user interface for selecting the EAM class or specific analysis model instance and then exports the appropriate knowledge to a text file to form the basis of a technical report.

Secondly, a method controlling knowledge access and sharing is developed which allocates permissions to portions of the knowledge base according to accessibility permissions. This method controls as efficiently as possible fine grain knowledge sharing. Both the methods acting together enable automatic generation of recipient-specific technical reports based on the recipient's security permissions and customized knowledge viewing.

Thirdly, implementation of these methods and our EAM knowledge base application as components within a product development environment is presented.

Methods have been developed to represent knowledge in the form of customized technical reports with secured sharing of the knowledge.

Finally, a method which uses knowledge from the EAM knowledge base to interoperate between analysis and optimization tools is demonstrated. This method operates on the EAM ontological knowledge grabbing specific parameter values and meta knowledge to intelligently run analysis and optimization tool in a commercial workflow environment. For this method the I-beam and its corresponding finite element models is used as test bed application to optimize the model based on design variables and objective functions. Optimization of I-beam is a problem which minimizes the volume of the I-beam based on the stress constraint. The design variables are the height, width and thickness of the beam. The EAM knowledge for the beam and the shell analysis models will be exploited to intelligently shift between the models during the optimization process based on eccentricity and accuracy expectation required. The modeling knowledge from the EAM ontology is used as design parameters for the optimization process. The design parameters for the I-beam optimization problem are the values for input parameters for the I-beam instance of the EAM ontology.

Sharing of knowledge in a product development environment and interoperability between analysis and optimization tools will be the first step in developing methods which operate on an ontological knowledge base. Methods can also be developed to componentize customized tools used for model design. These tools when componentized in the product development environment help in effective exchange, reuse, adaptation and interoperability with people, tools, or agents.

# CHAPTER 2

## REVIEW OF RELATED WORK

The first knowledge-based or expert system, Dendral [Feigenbaum and Lederberg], was developed in 1965 by Edward Feigenbaum and Joshua Lederberg of Stanford University in California and was used to analyze chemical compounds. Since 1965, knowledge-based systems have enhanced productivity in business, science, engineering, and the military. They also attempt to predict the weather, stock market values, and mineral deposit locations. Knowledge-based systems appear to have a great deal of potential, but they also face some challenges. These include the shortage of knowledge engineers with necessary skills and the relative immaturity of many of the available tools.

Most knowledge based systems deal with very specific problem domains, and therefore do not undertake or support a complete activity The benefit that such software offers is not necessarily to automate the process completely and cut costs drastically, but to assist the user to complete the activity faster, somewhat more cheaply, and probably more accurately.

Researchers from the field of Knowledge Engineering (KE) have proposed systematic guidelines for the development of knowledge based systems since the mid-eighties [Studer et al, 1999]. CommonKADS is a methodology to support structured knowledge engineering. CommonKADS provides the tools required to analyze knowledge-intensive tasks at different grain-size levels [Schreiber et al, 1994]. Moka [Moka, 1998] was developed in 1998 for collecting, structuring and formalizing the

engineering knowledge associated with designs. This made it easy to plan and organise the process of building KBE applications, updating them and re-using modules. Though CommonKADS is a powerful tool to support knowledge management within a highly secured environment, it does not provide a free source API (Application programming Interface) for developing and sharing of object oriented methods which support interoperability between different tools. Moka has a design oriented knowledge organization, but does not support development of new object oriented methods by modifying the API.

The MRA (multi-representation architecture) routinization process proposed by [Peak et al, 1998] is a knowledge capture technique for transforming physical behavior research and design standards into catalogs of ready-to-use analysis modules. It is particularly aimed at capturing reusable analysis knowledge at the preliminary and detailed design stages.

STEP [SCRA, 2001], Standard for the Exchange of Product Model Data, provides a representation of product information along with the necessary mechanisms and definitions to enable product data to be exchanged. The exchange is among different computer systems and environments associated with the complete product lifecycle including design, manufacture, utilization, maintenance, and disposal. STEP uses application protocols (AP's) to specify the representation of product information for one or more applications. For example STEP AP 209 supports finite element analysis model, finite element analysis model control and results but fails to support alternate representation of the information by tools outside of design and analysis, and the explicit graphical presentations derivable from design or analysis product representations. The

ultimate goal of STEP is an integrated product information database that is accessible and useful to all the resources necessary to support a product over its lifecycle. Commercially available STEP translators address only geometry, analysis and configuration management data. As of yet, they do not support the meta-knowledge associated with an engineering analysis model and its secured sharing.

Though there have been formal representation schemes for modeling knowledge, these systems cannot be easily extended by methods that would operate on this knowledge or exchange this knowledge with other people or tools. There is a need to separate out operational knowledge, i.e. problem solving methods, from the domain knowledge upon which the problem solving methods operate. One of the most promising ways to share and to separate and represent domain knowledge in an object-oriented manner from operational or problem solving knowledge is through the use of ontologies.

An ontology is an explicit specification of conceptualization of a domain [Gruber, 1993]. Thus, by the definition, an ontology is a set of concepts and their relationships. The Protégé [Protégé, 2005] view of an ontology consists of a defined class as an abstract representation of a concept in a domain. Each class has properties called slots which describe the various features and attributes of the class. The property constraints of a slot are described by facets. The classes are arranged hierarchically with subclasses inheriting slots from their super classes and adding further specification with additional properties and facets. Ontologies modeled in such object oriented nature are well suited for implementation and developing methods which operate on them using object-oriented languages like JAVA and C++.

[Borst et al.,1994] and [Borst et al.,1995] have developed a set of formal engineering ontologies called PhysSys for dynamic physical systems. During the development of the PhysSys ontology it has been found that constructing an ontology from smaller ontologies leads to an ontology that because of its structure is easy to understand and well suited for reuse. Meta-level information has to be included as attributes of engineering analysis models and information like modeling assumptions has to be validated as examples of such knowledge. Gruber has listed [Gruber, 1995] a number of design principles for ontologies, such as clarity, coherence and extendibility.

A knowledge base will be created with information that fills in the properties of the classes, i.e. instances of the class object. This ontology then represents a knowledge base for the particular domain. It has an object hierarchy which represents the knowledge.

The distinction between an ontology-based knowledge tool and pure object-oriented architecture is that, in the ontological knowledge based systems problem-solving methods are independent on the knowledge structure, i.e. class and instance objects [Musen, 2000]. This has advantage in terms of permitting a good bit of flexibility for developers to include new functionality into the code.

Java Based frameworks have become a recent trend to support engineering analyses. Onyx [Read et al, 1998] is a Java based object-oriented application framework for aerospace propulsion system simulation that is capable of integrating advanced multidisciplinary and multifidelity analysis methods, dynamically constructing arbitrary simulation models, and distributing computationally complex tasks. But this system requires a representation of these models to preexist in the database of components and is

restricted to aerospace propulsion systems. MOB-FEM [Shanbhag et al, 2002a, 2002b], and FiPER [Wujek et al, 2002] are also Java based frameworks for supporting engineering analyses. MOB-FEM has been developed for automatically reusing finite element modeling knowledge. A separate Visual Basic tool, called TEK-FEM, with a graphical user interface was developed to extract modeling knowledge from a domain expert. This knowledge is then stored in a Microsoft Access backend database. Although the modeling knowledge extracted from the expert is based on a taxonomy similar to the one proposed by [Finn et al, 1992], the knowledge was not represented or stored using a formal information structure.

One promising platform for exploiting object-orientation to support interoperability of engineering analyses is FiPER [Fiper, 2000]. FiPER is a component based web centric framework with a common protocol to enable companies to 'wrap' their engineering tools, data and processes into the FiPER environment. Phoenix Integration [Phoenix Integration, 2006] is a framework that automates and manages the analysis tools used for product development, accelerating design innovation. But, this tool does not support optimization process yet. The application of optimization tool for real world problems is still under development by Phoenix Integration.

# CHAPTER 3

## ONTOLOGIES FOR REPRESENTING ENGINEERING ANALYSIS MODELING KNOWLEDGE

In the recent years, development of ontologies for a wide range of applications has been an active area of research. Ontology libraries can be found at Stanford's Knowledge Systems Laboratory [Stanford, 2005] and at DARPA's DAML ontology library. As most designers can be considered to be experts, knowledge-based systems design can be considered to be expert systems design. Design of knowledge base systems has become an important area of study, with a rapidly growing literature [Gero, 1985; Hong, 1986; Brown et al, 1989].

For many years, researchers in knowledge acquisition and knowledge-based systems have described problem-solving methods as components that are re-usable in a plug-and-play manner [Chandrasekaran, 1986; Walther et al, 1992; Wielinga et al., 1993]. The engineering knowledge represented by the knowledge base systems include fundamental knowledge like phenomenological knowledge, analytical model knowledge, numerical model knowledge, and meta knowledge, such as goals, objectives, and intentions of the analysis model.

Design and analysis integration is the seamless integration between design and analysis perspectives by capturing the relationships between computer based design and analysis models [Peak et al,1998]. The design/analysis integration problem is typified by the requirement to share geometric shape and analysis information in an iterative environment [Hunten, 1997]. Efficient methods and tools are needed for extracting

analysis modeling knowledge from engineers and incorporating this knowledge into a computational environment to improve interoperability, reusability, and adaptability of analysis models.

Ontologies provide a clear specification of the concepts used in a domain and the relationships between them [Borst, 1997]. [Grosse et al, 2005] have developed an ontology that would be applicable to all engineering analysis models. Figure 1 shows a class hierarchy of some engineering analysis model types included in our ontology. The analysis models have been classified as physics based and non physics based. The proposed ontology is applicable to all physics-based engineering analysis models. Models that consist of one or more connected model components with individual model component parameters lacking spatial dependency are called lumped parameter or discrete models. Continuum based models, which have variables or parameters that are dependent upon one or more independent spatial variables, are classified as analytical or numerical. Boundary element methods, finite element methods, finite difference methods, wavelet methods, and hybrid methods are the subclasses corresponding to continuum based models.

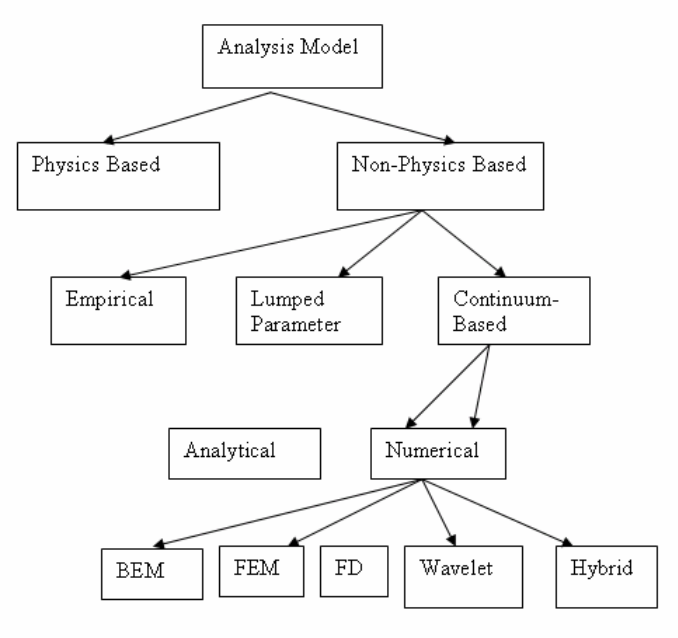



**Figure 1. Class hierarchy of some engineering analysis model types. The proposed ontology is applicable to all physics-based engineering analysis models.**

The engineering analysis model class has the following properties: name, creator, creation date, modifier, modification date, model version number, and model documentation. More abstract modeling knowledge such as model limitations, model assumptions or idealizations, modeling assumption justifications, etc. are important for reusability of the model. Table 1 shows the abstract, or meta knowledge needed to support engineering analysis modeling. [Grosse et al, 2005] describes further details on diverse properties that have been covered by the ontology for its diverse applications.

**Table 1. Properties of Engineering Analysis Models**

| EAM Term | Type | Cardinality | Notes |
|---|---|---|---|
| Model Name | String | Required single | Unique identifier |
| Description | String | Required single | Short and concise |
| Version No. | Float | Required single | Recommend x.x format |
| Documentation | String | Multiple | References to technical report(s)/presentations |
| Creator or Developer | String | Required multiple | |
| Creation Date | Date | Single | < current date |
| Modifiers | String | Multiple | |
| Modification Dates | String | Multiple | < current date |
| Development Time | Float | Single | In hours |
| Purpose | String | Required multiple | |
| Primary Objective | String | Required single | Engineering quantity predicted |
| Physical system basis | String | Single | product or process |
| Graphic of Model | Instance | Multiple | image files of model |
| Accuracy expectation | String | Single | Qualitative multiple choice |
| Resolution expectation | String | Single | Qualitative multiple choice |
| Software requirements | String | Multiple | |
| Robustness | String | Single | Qualitative multiple choice |
| Model input | String | Required multiple | List of input parameters |
| Model output | String | Required multiple | List of output parameters |
| Model Limitations | Instance | Required multiple | List of model limitations |
| Model Idealization | Instance | Required multiple | List of model idealizations |
| Model Components | Instance | Multiple | List of model components |
| Model Development Steps | String | Multiple | List of major model development steps |
| Intended user/user group | String | Required multiple | Employees, customers, etc. |
| Instance permission | Float | Single | Permission level 1,2 and 3 |

The object classes of the EAM ontology have been instantiated to form an engineering analysis modeling knowledge base and implemented into a prototype software tool called ON-TEAM as shown in Figure 2. ON-TEAM is an implementation of the EAM ontology and the instance knowledge. The Protégé environment has been used to develop the ontology. Protégé helps users build other tools that are custom-tailored to assist with knowledge-acquisition for expert systems in specific application areas [Musen, 1989]. The EAM ontology can be interfaced with the current COTS (commercial-off-the-shelf) tools in the engineering analysis domain using the input/output command files and the application program interfaces (API's) of Protégé.

The Protégé GUI can be customized to present diagrammatic output of knowledge in an ontology using the graph widget as a plug-in for Protégé. It is also particularly useful for visualizing networks of instances and relationships between instances. Thus, ON-TEAM can be extended with object-oriented methods that operate on the knowledge base to perform various tasks.

As a first step toward creating such sophisticated methods for exploiting the ON-TEAM knowledge base, we have implemented some simple JAVA-based methods that operate on the knowledge base [Kanuri et al, 2005].

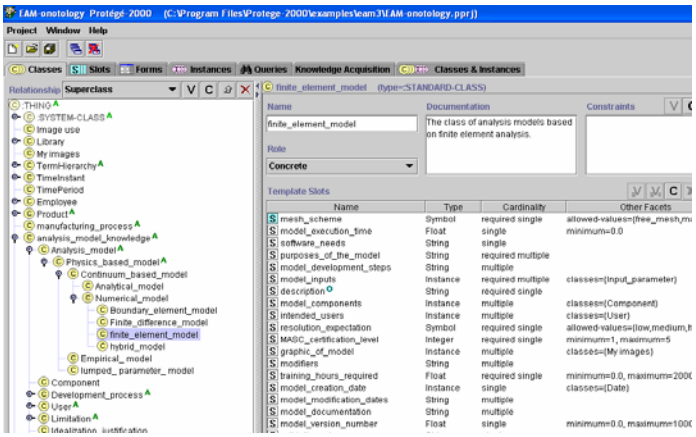



**Figure 2. Implementation of ontology for representing Analysis Modeling Knowledge**

One method that has been developed is a technical report generation method. This method creates a flat technical report that describes either the properties or class relationships of an engineering modeling analysis class or describes the modeling knowledge involved in the development of a specific engineering analysis model. This method can be used to assess the properties of an engineering analysis model stored in ON-TEAM. This method is a simple but representative example of how a Java

application can access the EAM knowledge base application using the Protégé application programming interface.

The user first selects an ontology using the GUI created by the technical report method. Figure 3 shows how the user selects a class from the tree structure of classes in the EAM ontology. This is the class structure generated by the EAM ontology in Protégé. The class hierarchy of the EAM ontology generated in Protégé is viewable and accessible but the property and instance knowledge of the EAM ontology is protected and managed by permission levels and the JAVA programming code. We are able to include the class structure of Protégé in the technical report method by accessing Protégé's application programming interface. The class analysis model contains the classes physics based model, continuum based model, analytical model, numerical model and finite element model in the order as listed. When the finite element model is selected the GUI displays all the instances of this class.

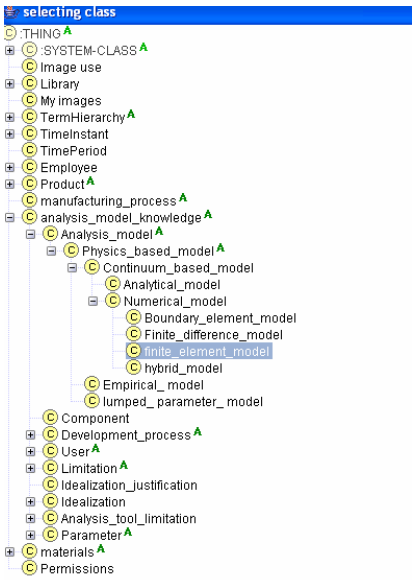



**Figure 3. Class tree of EAM ontology**

Figure 4 shows a list of instances of modeling knowledge associated with three different finite element models found under the finite element model class from which the user can select. The instances are CCSB belt, lateral elevator model and simple shelf bracket model. The instances of a class inherit all the properties of a class, so the various instances each have all the inherited properties such as name, documentation, model inputs, limitations, idealizations, limitations etc, and as instances they have values associated with these properties.

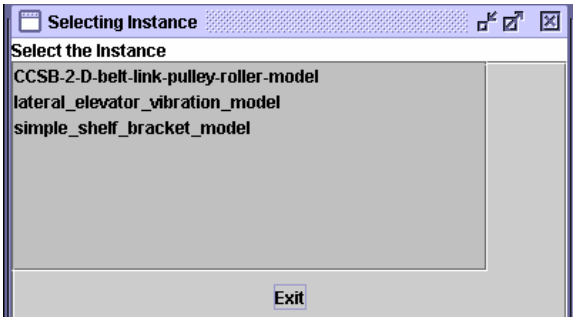



**Figure 4. Instances of the EAM ontology class generated are CCSB-2-D model, lateral elevation model and simple shelf bracket.**

Once the user selects a specific instance of the class, the technical report method is applied to the instance information in the knowledge base to generate the appropriate technical report. Figure 5 shows the technical report generated for the CCSB belt instance.

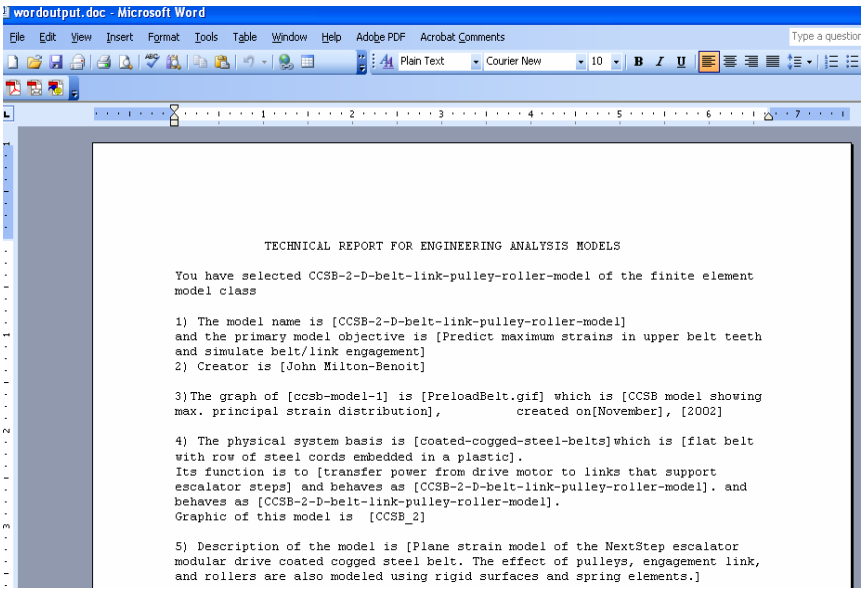



**Figure 5. Flat technical report generated for the CCSB-2-D model**

The technical report method currently writes out the knowledge present in the ontological knowledge base to a file. The corresponding sentences associated with this knowledge have to be manually written and the knowledge placed at the right place in the sentences to make meaningful technical reports. An enhanced technical report method can include a parallel class for technical report sentences to generate dynamic technical reports. This method when componentized in a product development environment can generate multiple technical reports based on the user's permission to access the knowledge present in the ontology. A JAVA programming interface has been developed to include the technical report method in the product development environment. This work is presented in the following section.

## CHAPTER 4

## IMPLEMENTATION OF THE OBJECT ORIENTED METHODS IN PRODUCT DEVELOPMENT ENVIRONMENT

[FiPER, 2005] is a software product that serves two main purposes. It is an internet based distributed framework which supports collaboration among geographically distributed engineering and business partners. It provides a common standard way to model analysis and design processes in conjunction with the product data. Fundamental to the FiPER project is its web-based distributed software architecture. This product development distributed environment enables data exchange across the web and access to various engineering tools that support product development. For purposes of our current experiments, the Protégé technical report method is wrapped in FiPER as an activity component. Figure 6 shows how the technical report component in the FiPER environment can drive the FiPER wrapped MS Word to create a MS-Word file instead of a text file. Figure 7 shows the editor of the technical report component in FiPER.

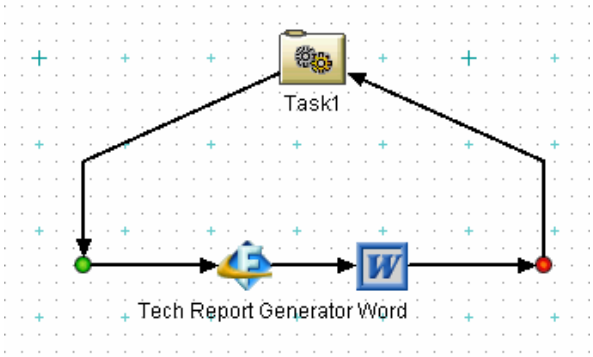



**Figure 6. Protégé component working in FiPER environment.**

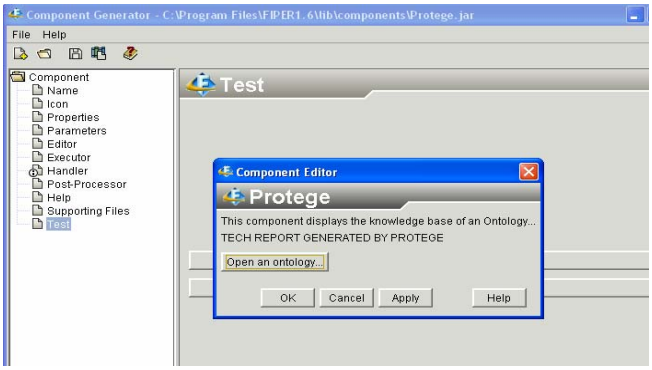



**Figure 7. Protégé editor accessed from within FiPER environment**

Figure 8 shows how the user uses the Protégé editor's GUI componentized within FiPER to select an ontology.

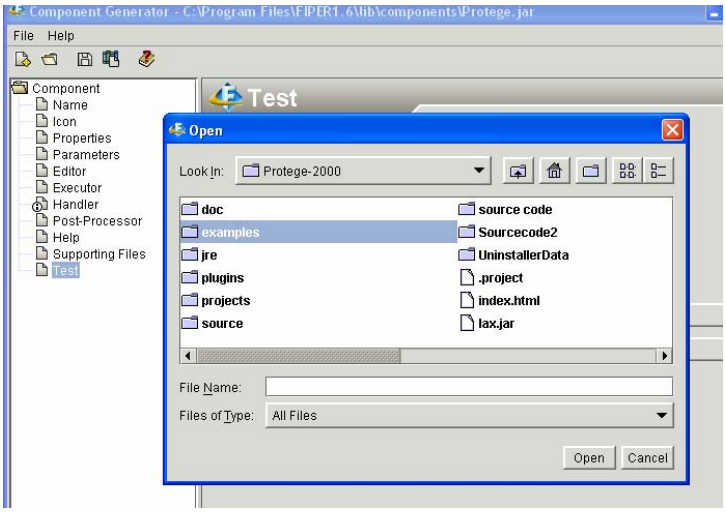



**Figure 8. Selecting an ontology from the Protégé editor in the FiPER environment**

Typically, engineers will not be interested in creating or modifying a component's activities inside the FIPER environment. The objective is to use the component within the work flow of FIPER to take input parameters, perform activities external to FIPER using the native application and return to FIPER work flow new values obtained from this application as output parameters. The technical report component is designed to perform end functionality, typically invoking and interacting with an external application,

Protégé, which is completely external and unknown to FiPER. This is done by developing the technical report method as a custom component which involves writing the JAVA adapter that implements specific interfaces provided in the FiPER software development kit. Also, this way of creating a custom component provides access to the FiPER system from across a network without any additional software installation on the client machine.

The Protégé knowledge base contains only slots and knowledge associated with the slots in a hierarchical manner. The technical report component generates meaningful sentences within the product development environment. This is done by creating a parallel class of "Technical report" in the knowledge base to write meaningful sentences associated with the slots and their values. This parallel class of "Tech report EAM" contains all the classes and the slots associated with these classes. These slots have been defaulted to be of type string, so that meaningful sentences associated with these slots can be written. The default value facet of the slot has been used to create the technical report sentences. Previously, for developing components the knowledge format that has been used for representing the ontology got stored as project files in Protégé. These file formats do not support efficient exchange/reusability of knowledge with other standard available tools. The knowledge base format has been changed to standard OWL representation. Like Protégé, OWL is used to store ontologies, and its API allows for retrieval of the information of the ontology. Ontology files using OWL format are more shareable, reusable and also support product development environment for accessing and working with them. But this format does not support the concept of a default facet. Hence, the default instances have been used to create technical report sentences instead of

default facets. Figure 9 shows the parallel technical report class with all the classes of the EAM ontology knowledge base and the corresponding slots. It can be seen that the default values of the slots are technical report sentences. The default slot values can be overridden by the instance values of the slots. That is, when an instance for the "Tech report EAM" is created, the user can write new sentences corresponding to each slot value. This enables generation of dynamic technical reports using different sentences to describe a slot. The instances for "Tech Report EAM" class will be created by the user when dynamic technical reports are needed.

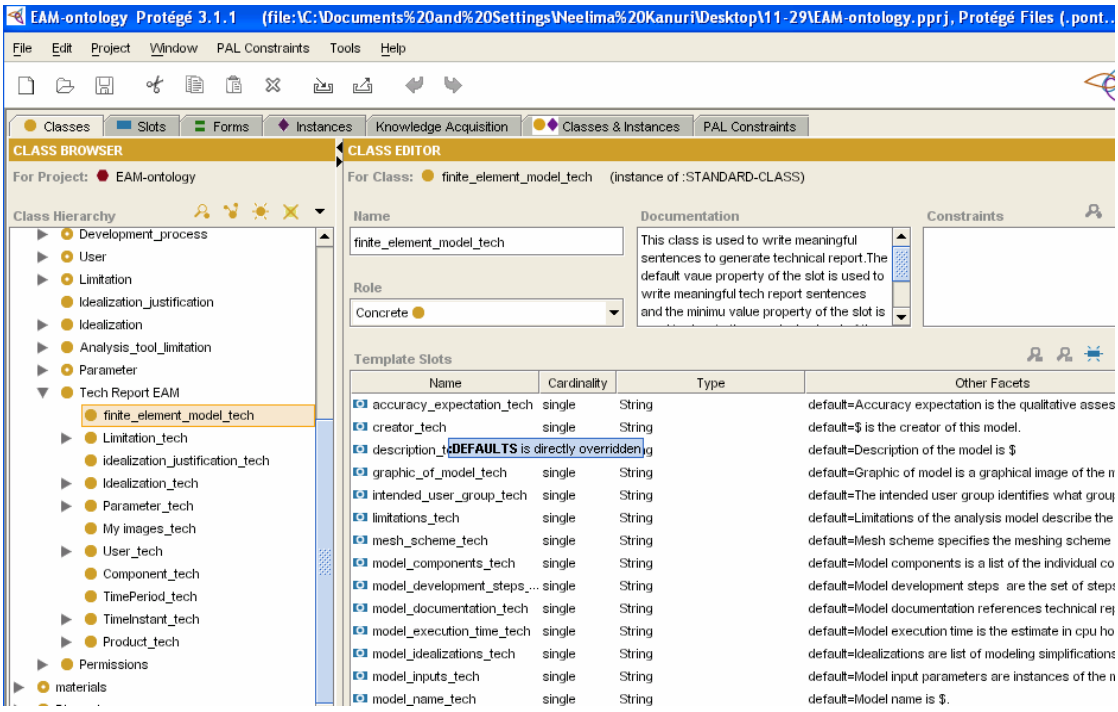



**Figure 9. Parallel class of "Tech report EAM" created for technical report generation method.**

A JAVA programming code for developing parallel class of "Tech report EAM" has been developed by using the Protégé API. This code automatically creates all the classes and slots associated with the knowledge base for writing meaningful sentences for the generation of technical report. This code has also helped in creating a parallel class of

"Permissions" which is used to control the sharing of knowledge in a product development environment. A detailed description about controlling the shared knowledge is found in Section 5.

When the user selects an ontology and wants to view the knowledge associated with it in the form of a technical report, the "Tech Report Generator" programming code accesses the Protégé API to set the technical report sentences from the parallel class of "Tech report EAM" and appends the name and value(s) of the slots accordingly, to make meaningful sentences. When a slot value corresponding to an instance is null, the Tech Report Generator method automatically suppresses the sentences associated with the slot. Hence sentences with slot values as null will be discarded enabling to creation of dynamic technical reports. These sentences written for each slot in the "Tech report EAM" class were written during the creation of the EAM ontology. The sentences have been written in a way to be meaningful when values and names of the slots have been added to the sentence. This technical report works for any ontology provided the creator of the ontology has defined the technical report sentences for each slot.

For viewing the Protégé knowledge, the technical report generation component, "Tech report EAM", has been modified in FiPER to contain two different components, "Interoperate Protégé KB" and "Tech-report". Together these two components create the technical report. The "Interoperate Protégé KB" component grabs all the information from the Protégé knowledge base and makes the knowledge available for inspection and also for passing the knowledge to other components like "Tech-report". For viewing the instance information in the product development environment with a Protégé look and feel environment, an Instance pointer data type has been developed. This new data type in

FIPER allows a Protégé instance to be a parameter in FiPER. This instance pointer data type allows inspection of the knowledge within FiPER, without being able to modify/change the knowledge. Initially, the instance pointer data type was able to handle only a single instance at a time to be viewed or passed as an instance data type. This functionality has been improved by allowing the user to select multiple instances for inspection within the product development environment. The code has been modified to include the functionality of selecting/ passing multiple Instance pointer data types by generating an aggregate of instance pointer data types. With this new functionality, any instance in a Protégé knowledge base can be selected and added to FiPER under the aggregate variable instances and documentation, i.e., technical report can generated for each added instance. The instance pointer data type values are the instances of the Protégé knowledge base. These values when passed to the "Tech-report" component generate the technical report corresponding to the instance pointer data type value that has been passed. This promotes generation of multiple technical reports at the same time.

The technical report generation method could have also been implemented by changing the source code of Protégé to include a new facet for each slot which defines the meaningful technical report sentence associated with the slot. But the disadvantage of using this method is that, each time a new build of Protégé is released, the source code has to be modified to include the new facet for each slot.

Initially, parameters to FiPER have been passed by grabbing all the slot information from the EAM ontology knowledge base. An enhanced version of passing parameters to FiPER can be done by passing only the parameters required for analysis and optimization processes. These parameters later drive the analysis and the

optimization tools. The “INSPECT-IT” method has been developed to enable the user to fetch only required parameters, making other knowledge unseen in the FiPER environment. This enhanced method improves the accessibility of the knowledge base by easily passing the parameters to the FiPER environment.

The “INSPECT-IT” method has been developed by extending the previously developed “Interoperate Protégé KB” method to grab values of input parameters from the ontology and make them as parameters in FiPER so that they can be reused and interoperated between different FiPER components. These grabbed values are mostly of the type integer, real and string so that they can be used to interoperate knowledge between different FiPER components. This component works by allowing the user to look at the slot values within an instance in Protégé and also add individual slot values of the instance to the workflow. Each slot value gets added to the workflow under an aggregate, FiPER parameter with the parameter having the name of the instance. The type of the parameter added matches the type of the slot value. For example, if the slot value is an integer, the value is added as an integer in the workflow. The name of the parameter that accompanies the value matches the name of the slot. If more than one value of the same slot is added, an appropriate number (_2, _3 etc.) is placed after the name. ”INSPECT-IT” gives the user the ability to pass the individual slot values from a Protégé ontology to the workflow, as FiPER parameters, in addition to getting Protégé instances, as done by the “InteroperateProtégéKB” method.

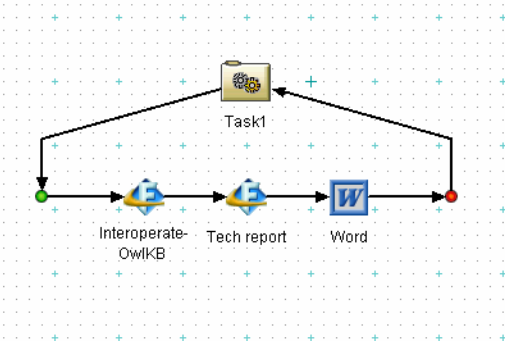
iSIGHT software integrates key steps in the product design process, automates and executes these steps through design exploration tools like optimization, DOE techniques and DFSS tools like reliability, analysis and robust design. iSIGHT is linked

to FiPER to develop iSIGHT-FD. Through FiPER infrastructure models, applications and processes related to iSIGHT are also made easily shareable, accessible with other engineers and groups using iSIGHT-FD. When FiPER received an update, iSIGHT-FD (version 1.0), in February, 2006, the "Interoperate Protégé KB" component and the "Tech-report" component that worked with FiPER failed to work. The problem was with opening the windows for displaying the classes and instances tree structure. These components have been made to work successfully by making each window of the components have an editor dialog as its parent. This editor dialog box was not modal before and has been made modal with the component editor dialog box as the parent, i.e., the user has to first dismiss the dialog box before clicking elsewhere.

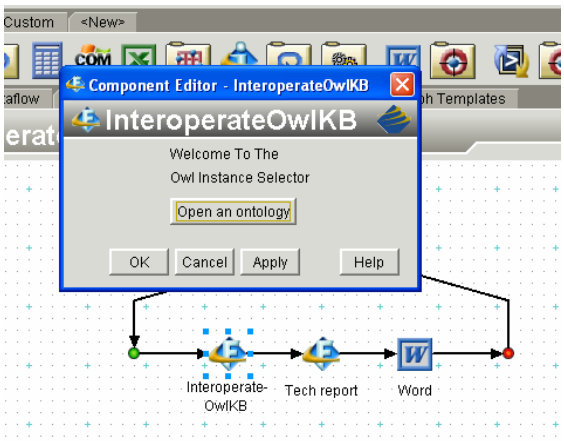
"Interoperate Protégé KB" and "Tech-report" components do not work the same way with OWL files as they have been programmed to work with Protégé standard ontology project files. Hence, these components were redeveloped to work with OWL format ontology files. Also, the data type instance pointer which is being used in a product development environment to inspect the knowledge with Protégé look and feel environment, has to be recreated. The components have been modified to work using the OWL knowledge base format. Hence, "InteroperateOWLKB" and "OWLTech-report" and "OWL pointer data type" components have been developed with the same functionalities that the components had with regular Protégé project files. The basic abilities of the tools are the same. For example, the "InteroperateOWLKB" looks almost similar to "InteroperateProtegeKB" to the end user, other than using ontologies that are represented in the OWL format rather than Protégé standard project files as a source of ontological knowledge. However, there is much terminology difference in OWL. Instead

of having slots and slot values using the regular Protégé project files, OWL ontology has "properties" and "property values" that work similarly. Instances in Protégé standard project file are referred as "Individuals" in Protégé OWL. In addition, OWL ontologies can be stored locally like Protégé regular project format ontologies, but are often stored remotely on the web. So, the "InteroperateOWLKB component" also allows the user to manually enter the URL of a local OWL file to retrieve it.

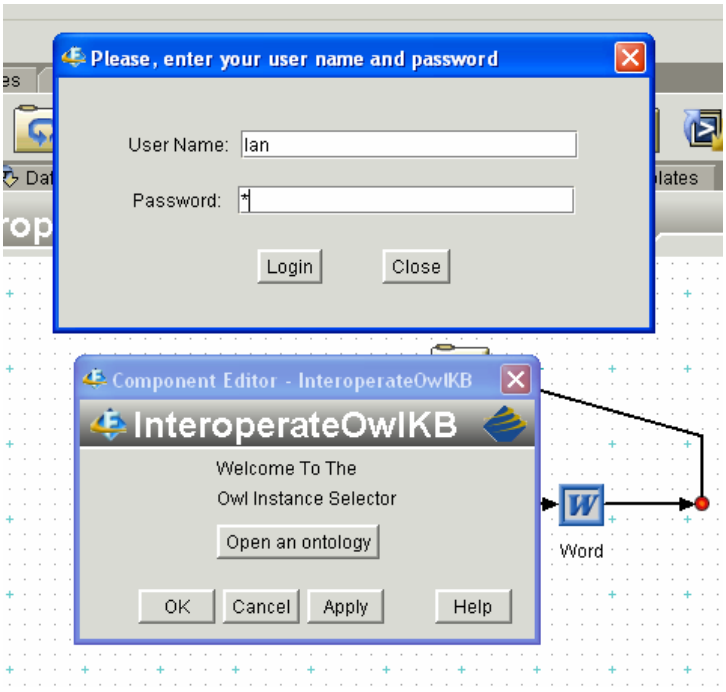
Figure 10 represents the working of "InteroperateOWLKB" and "OWLTech-report" components in iSIGHT-FD. As explained above these components have the same functionalities as "InteroperateProtegeKB" and "Tech-report" components except that the OWL components use OWL representation of ontology rather than the regular project files used as ontology format.



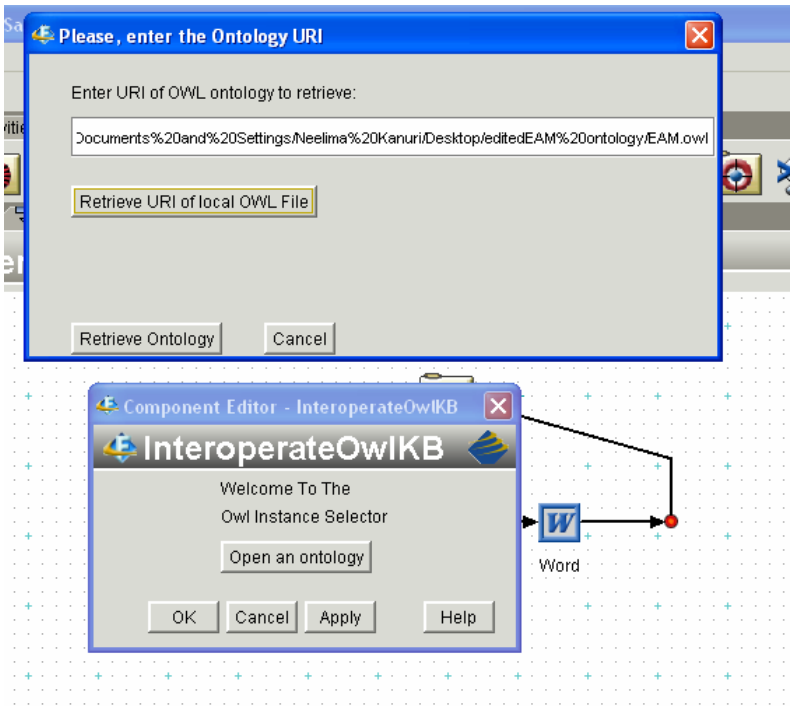

**a) Configuring the components**




**b) Editor of "Interoperate Protégé KB" component**




**C) User Login window**

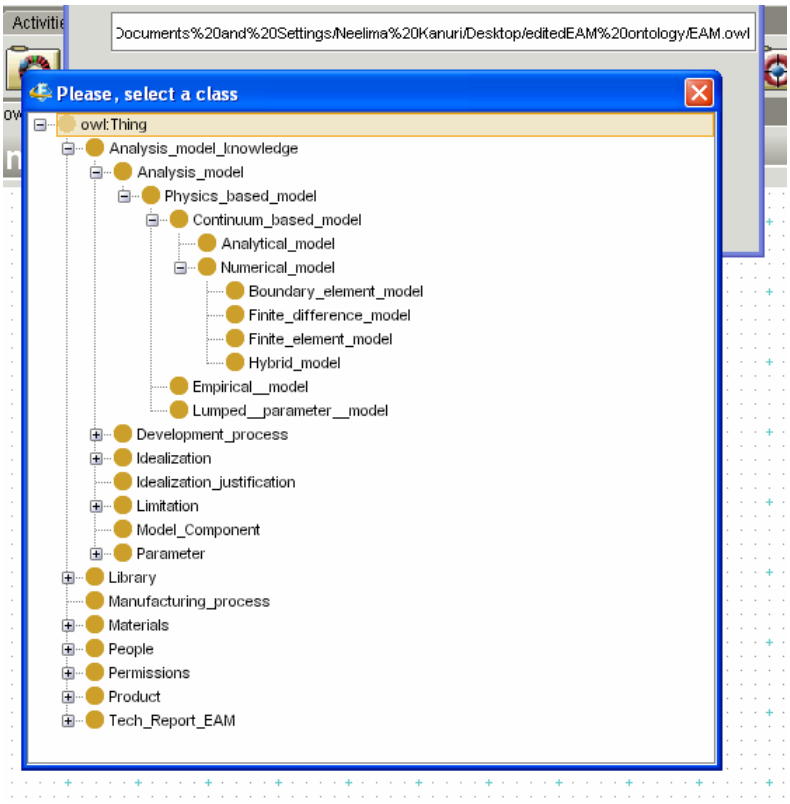



**d) Retrieving the ontology**

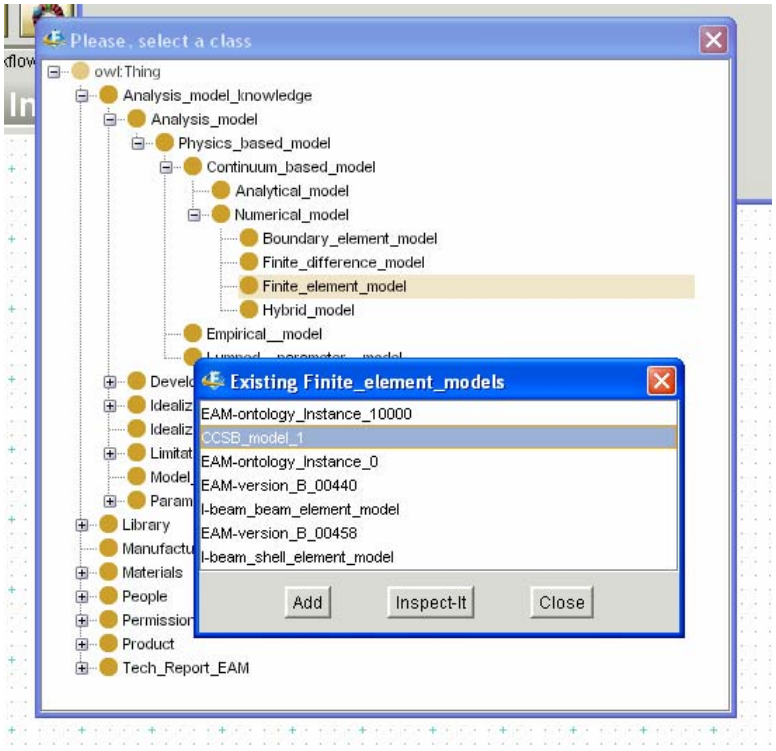
**Figure 10. Accessing knowledge using "InteroperateOWLKB" component**

From Figure 10 we notice that the user has to enter the name and the password, which can be used for generating security permission to access the knowledge. Details about permissions for secured sharing of knowledge are found in Section 5. As explained above, the OWL file can be accessed either remotely or by entering a URL address to fetch the OWL file.

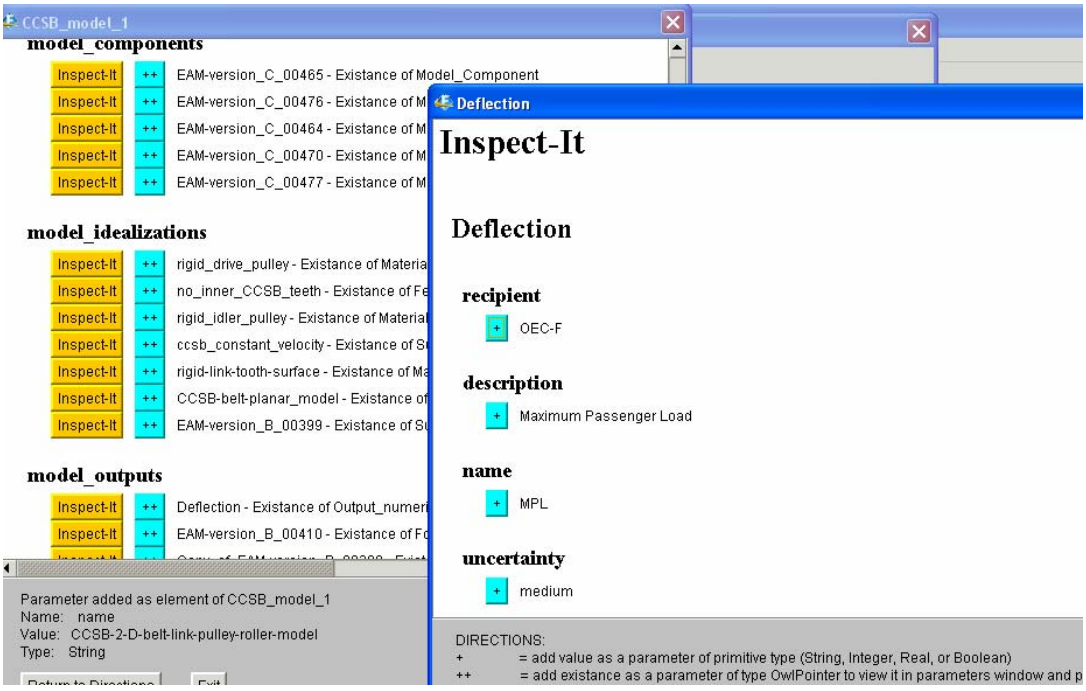
Figure 11 shows the listing of Protégé OWL class files and inspection of the knowledge using the GUI developed by JAVA programming. The "+" button, as shown in Figure 11(c) allows the user to select a slot and its value and make it an iSIGHT-FD parameter just with a button click. The slot and its corresponding value will then be iSIGHT-FD parameters with the same data type as in the OWL knowledge base.




**a) Class structure**



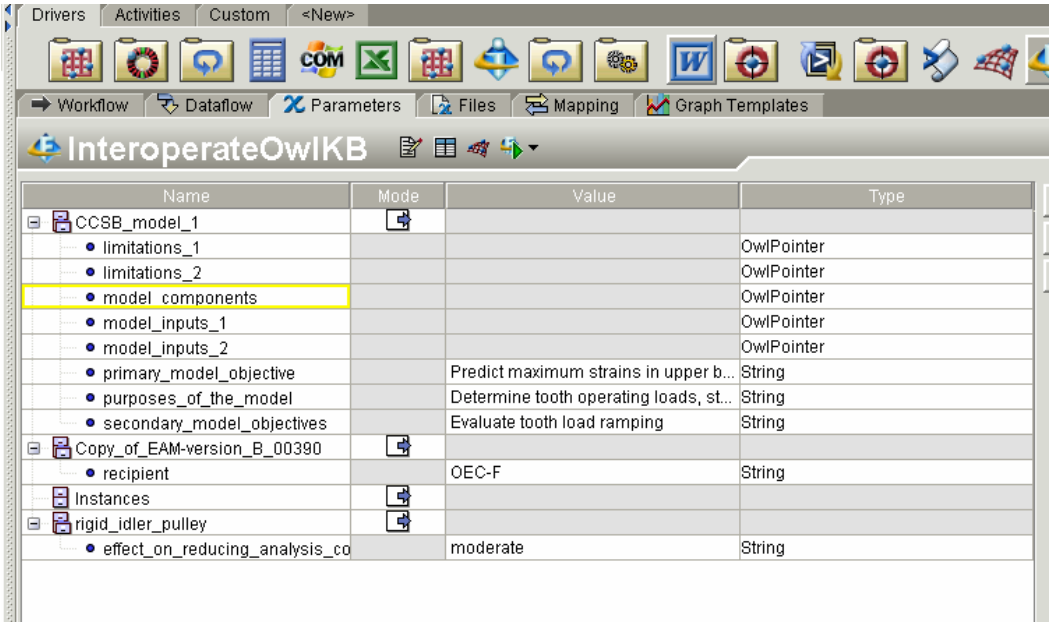

**b) Instances for the selected class**



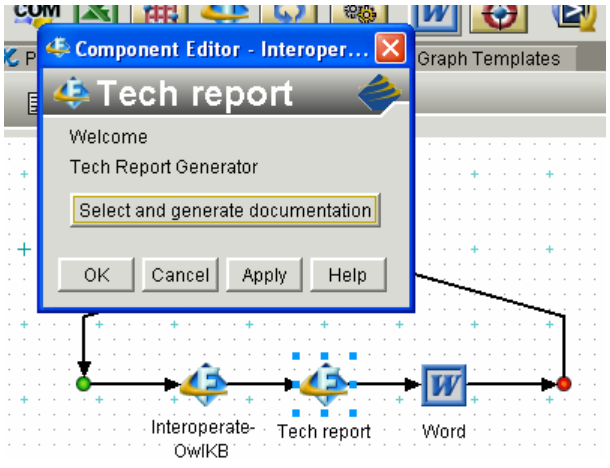

**c) The "Inspect-It" method for viewing knowledge of an Instance**

**Figure 11. Selecting instance and inspecting the knowledge**

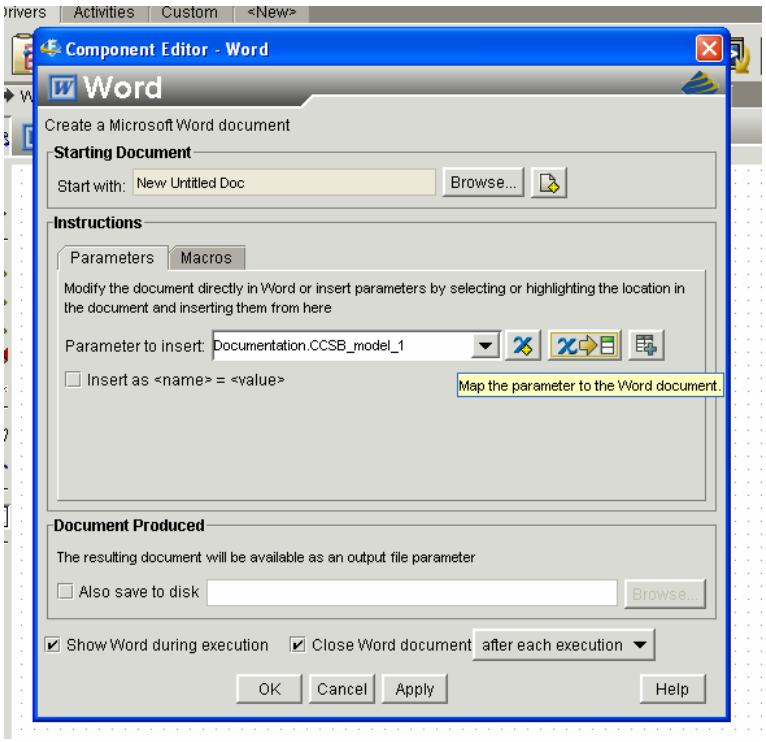
If the selected slot is of type Instance, then the "Inspect-It" button allows the user to look at the knowledge of the instance. It also gives the ability to add the slots of this selected second instance as iSIGHT-FD parameters. The "++" button allows the user to make the instance as an OWL instance pointer so that the user can view the knowledge within product development environment.



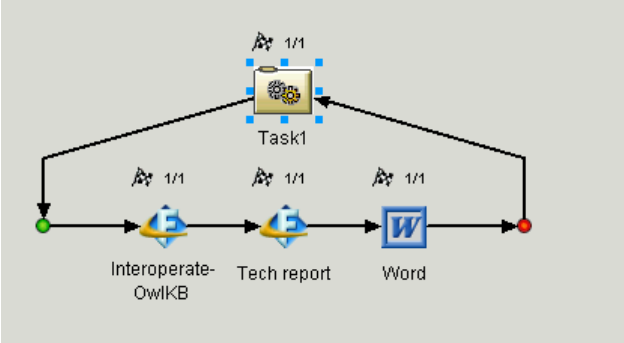

**a) Selected instances represented as "Owl Pointers"**



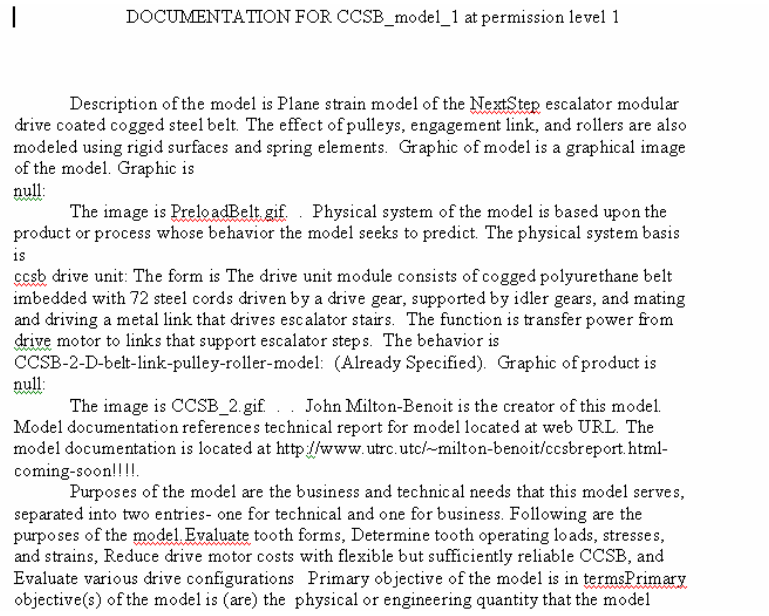

**b) Technical report generation component**




**c) Configuring word component**




**d) Running the components**



DOCUMENTATION FOR CCSB_model_1 at permission level 1

Description of the model is Plane strain model of the NextStep escalator modular drive coated cogged steel belt. The effect of pulleys, engagement link, and rollers are also modeled using rigid surfaces and spring elements. Graphic of model is a graphical image of the model. Graphic is
null:
The image is PreloadBelt.gif. . Physical system of the model is based upon the product or process whose behavior the model seeks to predict. The physical system basis is
ccsb drive unit: The form is The drive unit module consists of cogged polyurethane belt imbedded with 72 steel cords driven by a drive gear, supported by idler gears, and mating and driving a metal link that drives escalator stairs. The function is transfer power from drive motor to links that support escalator steps. The behavior is
CCSB-2-D-belt-link-pulley-roller-model: (Already Specified). Graphic of product is
null:
The image is CCSB_2.gif. . . John Milton-Benoit is the creator of this model. Model documentation references technical report for model located at web URL. The model documentation is located at http://www.utrc.utc/~milton-benoit/ccsbreport.html-coming-soon!!!!.
Purposes of the model are the business and technical needs that this model serves, separated into two entries- one for technical and one for business. Following are the purposes of the model.Evaluate tooth forms, Determine tooth operating loads, stresses, and strains, Reduce drive motor costs with flexible but sufficiently reliable CCSB, and Evaluate various drive configurations Primary objective of the model is in termsPrimary objective(s) of the model is (are) the physical or engineering quantity that the model

**e) Generated Technical report**

**Figure 12. Generating technical reports for multiple instances**

Figure 12 shows how multiple instances can be selected and used to generate multiple technical reports. Also, the OWL instance pointer can be used to inspect the knowledge within the product development environment. As explained above, this new data type can be used for knowledge inspection and multiple instance pointers can be made as iSIGHT-FD parameters using the aggregate parameter of iSIGHT-FD. These instance pointers can be used to inspect the knowledge and also pass the parameters to other iSIGHT-FD components to generate secured technical reports.

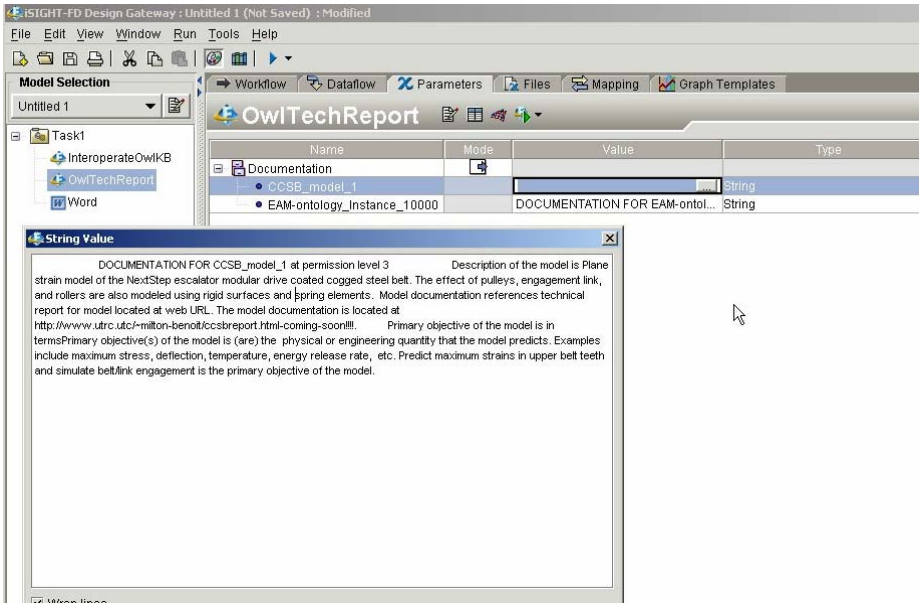



**Figure 13. Creating multiple technical reports using the Tech-report method**

From Figure 13, it can be seen that documentation for multiple technical reports is made available within the "OWL Tech report" component. The user can then select an instance of his choice for which the technical report has to be generated. Also, the user can append the documentation of multiple technical reports to a single technical report using the "wrapped" Word component of iSIGHT-FD. More customized generation of technical reports and creation of single technical report by appending two or more technical reports is detailed in Appendix A.

For secure sharing of analysis modeling knowledge method, permissions have been introduced for accessing various technical reports, while at the same time giving privileges for the user based on his/her permission to access the knowledge. According to this method, the user can see the instances that he/she has permission to generate technical reports for. If the user's permission is greater than the permission required to

generate the technical report, the user will be able to generate it. Details of permission based technical reports are dealt with in Section 5.

# CHAPTER 5

## SECURED SHARING OF ANALYSIS MODELING KNOWLEDGE

For sharing analysis modeling knowledge among different organizations or groups, a common frame of reference is required, where all concepts can be placed and understood. Ontologies are an extremely useful tool for capturing and sharing knowledge, either through rich knowledge representation schema, such as RDF [Klyne et al, 2004] and XML [Walsh, 1997] or through the automatic generation and sharing of flat technical reports as seen above. However, it is important to control this knowledge sharing in order to protect the intellectual property of organizations. Sharing of knowledge based on user permissions is one way to enable secured **EAM** knowledge sharing.

How can access to ontologies be controlled to prevent undesired sharing of analysis modeling knowledge? One solution is an access control method that supports fine granularity control of the knowledge. The method works by including a permission level property for each class at the granularity of slots. This slot controls the accessibility of an individual property of the instance of the class. Now when the user wants to generate a technical report, an enhanced version of the technical report generation method could obtain user permission level and generate a technical report based on the permission property of the class and of all the class slots. Combining this knowledge access method with the technical report generation method enables automatic generation of different technical reports based on security permissions, as well as customized knowledge sharing via RDF or XML exports.

At the class level of an ontology, knowledge sharing is controlled by including the instance permission level as a slot of the class. For example, the **ON-TEAM** engineering analysis modeling ontology contains a class called finite element model which consists of 25 slots and 3 instances. The instances of this class are industry driven applications. Now, to limit the access to the knowledge of this class, a new slot named knowledge access permission of type symbol is added as seen in Table 1. The allowed values for this slot are 1, 2 and 3. The values of this slot are assigned by the ontology developer who has administrative privileges. The ontology developer has the highest accessibility permission level of 1. Accessibility permission level 2 could correspond to managers and analysis engineers of the organization who have sufficient privileges to look at almost all the information present in the ontology but need not necessarily have the permission to modify it. Accessibility permission level 3 might correspond to vendors or suppliers outside of the organization who have some interaction with the organization's activities. They may need access to basic information about the model, but the organization is typically not willing to share the entire analysis modeling information with them. Standard user authorization techniques, such as secure login and database, may be used to determine user identity and assign the appropriate permission level. The access levels may range differently (eg. from 1 to 6) based on the level of security needed.

A programming code which creates parallel class for "Permissions" has been developed. This class contains all the classes and slots of the EAM knowledge base. The slots of this class have been made to be of type real to support the numbering values based on permissions. When the user wants to look, modify or write information to/from the knowledge base, the slot values of the "Permission" class is used to decide the user's

ability to view or modify the knowledge base based on the user's permission level. This gives a fine grained control of the shared knowledge based on the user's identity. Figure 14 shows the permissions class and the corresponding slots of this class which have permission values as the slot values.

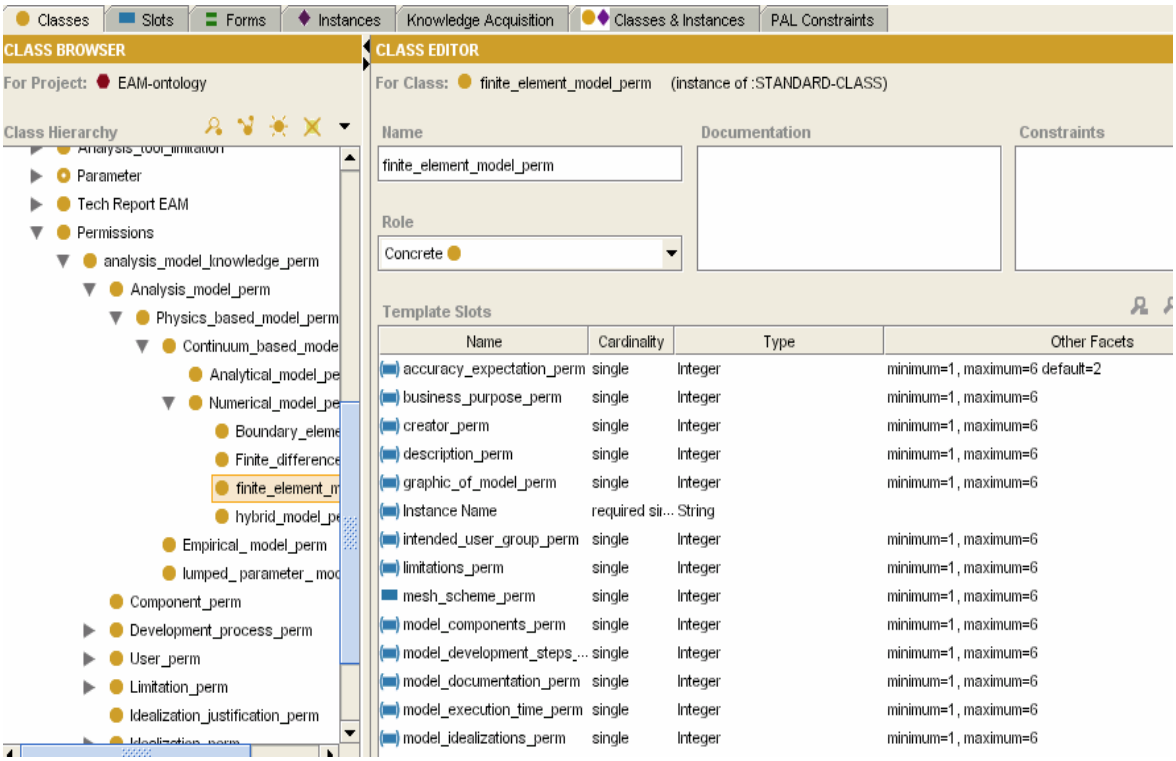



**Figure 14. Parallel class of "Permissions" and corresponding slot permissions which define the permission level of each slot of the knowledge base.**

When the iSIGHT-FD editor for technical report generation component is opened, as shown above in the technical report generation component, the user first enters his username and password. The user then selects an ontology which has to be opened. A Protégé class tree structure shows up which prompts the user to select the class according to his requirements. Later the instances knowledge associated with the selected class will be shown.

Comparing Figure 15 and Figure 16, one can notice that the instance "simple-shelf-bracket-model" does not appear as an accessible **EAM** instance in Figure 16. This is because this **EAM** instance has a higher value for the instance permission level slot compared to the other two instances, and consequently the user had inadequate permission level to access this knowledge instance.

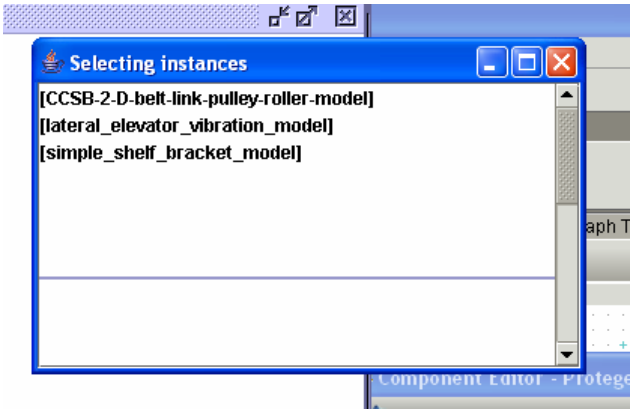



**Figure 15. Access to instances before including knowledge access permission code**

At the slot level of an ontology, knowledge access has been controlled by creating a parallel class called permissions, which has a permission value stored as default slot value for each slot of the finite element model class. Slot level knowledge access control could also have been implemented by including the permission level of the slot as a facet of the slot. However, this would involve changing the Protégé source code, and such changes would have to be re-implemented with each new release of Protégé. To avoid this problem, an alternate class for individual slot access level is created in parallel to finite element model class.

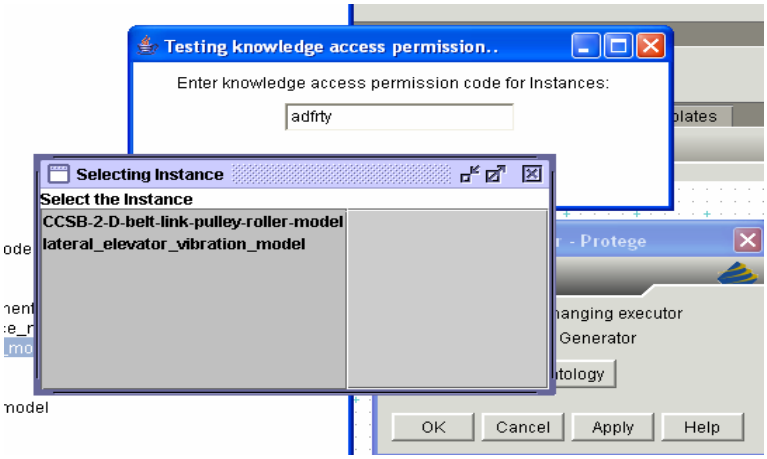



**Figure 16. Access to instance knowledge after including knowledge access level control**

After selecting the instance, the user is only then permitted to access the instance knowledge of each slot for which he or she has sufficient privileges. Furthermore, the method hides the slot, so the user (i.e. knowledge receiver) is unaware of not only the value of the slot but the existence of the slot itself. Such privileges are granted based on the user's permission level compared to the value of the slot's permission level.

As explained in the section four, the "OWL tech report" component generates documentation for multiple technical reports accessing the OWL knowledge base. The permissions based secured knowledge sharing has been extended to write customized secured technical reports using the iSIGHT-FD wrapped MS Word component. Filtering of knowledge has been added for the "InteroperateOWLKB" component and the "OWL pointer data type". When the "Inspect-it" window is brought up, the user can only see the property values that he/she has permission to see according to the user's permission level and the permission settings in the parallel permission class. If no permission level for a property can be obtained, the default is to set the permission level of the most restrictive setting and therefore deny access to the value to all users except those who possess top

level of access or permission. Filtering works the same way when the user looks at the instance/individual from the OWL pointer editor window. Figure 17 shows the working of permission for technical report documentation in iSIGHT-FD. The user can check the boxes of instances whose technical reports have to be generated. Figure 17 shows the implementation of permission to the documentation of technical reports for instance knowledge of an OWL KB.

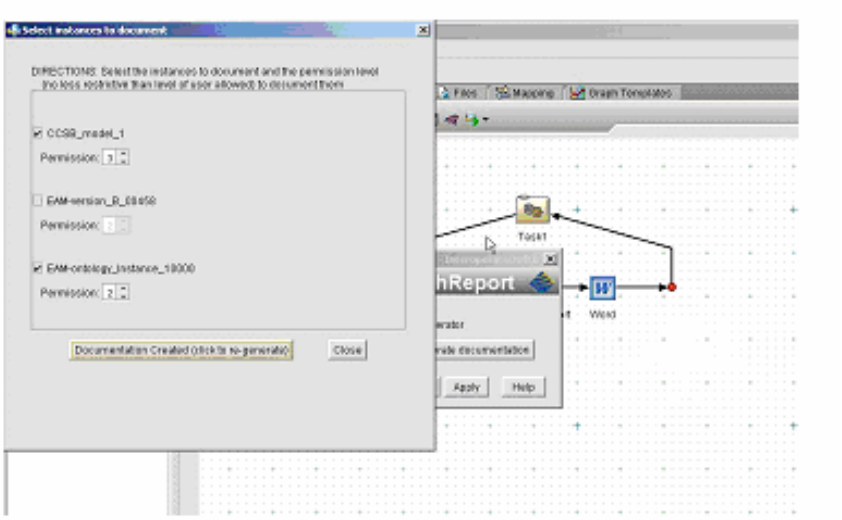



**Figure 17. Implementation of permission based knowledge control for generating documentation of instance knowledge**

This has been implemented in a way that if the user has a permission greater than the permission required to generate the technical report, the documentation of the technical report will be passed to the iSIGHT-FD wrapped Word component. Any user who has less permission than the permission required to generate the technical report will not have the privilege to document and view the technical report for a particular OWL KB instance.

Hence, secure sharing of analysis modeling of knowledge has also been augmented with functionalities for secure viewing and secure technical report documentation of knowledge corresponding to any instance of the OWL ontology knowledge base.

# CHAPTER 6

## INTEROPERABILITY OF MODELING KNOWLEDGE BETWEEN ANALYSIS OPTIMIZATION AND DECISION MAKING TOOLS IN A PRODUCT DEVELOPMENT ENVIRONMENT

Interoperability is the ability of two or more systems to exchange and reuse information efficiently. The EAM knowledge base has higher level abstract knowledge about models that have been developed. This knowledge can be used to run the analysis and optimization tools. In the present work we have continued to integrate design and analysis knowledge and associated tools together to demonstrate the power of such integrated systems. ONTOPT, a Protégé design optimization ontology and knowledge base that has been developed, has been integrated into the iSIGHT-FD environment. OPT EAM has the features of both analysis and optimization ontologies.

An I-Beam design optimization model and its corresponding finite element model are used as a test application to run the analysis and optimization components in a product development environment. The integration of analysis and optimization components in a product development environment needs to be done to interoperate the knowledge between the tools. Though iSIGHT-FD has a built in optimization component, there are no analysis components that have been directly integrated into iSIGHT-FD. This task has been achieved by integrating ANSYS, a commercial FEA tool, into iSIGHT-FD using the SIMCODE component in iSIGHT-FD. Parameters and knowledge obtained from both the **EAM** knowledge base and the ONTOPT knowledge base are passed into iSIGHT-FD as parameters to drive analysis and optimization tools. Hence, integration of knowledge sharing methods and "wrapped" analysis tools in iSIGHT-FD

help in running multiple analysis and optimization models within the product development environment.

The required knowledge from the OPTEAM knowledge base is used as parameters to run the ANSYS and OPTIMIZATION components in iSIGHT-FD. Results from the optimization loop can be written back as values of parameters for the extracted knowledge in the ONTOPT knowledge base system.

## 6.1 Analysis tool in product development environment

ANSYS, a commercial FEA tool, has been componentized in the product development environment, iSIGHT-FD. This is done by using the SIMCODE component of iSIGHT-FD. The data exchanger allows moving data between iSIGHT-FD parameters and text files easily and efficiently. ANSYS in batch mode is run by submitting a file of commands to the ANSYS program. It can be run on any operating systems as a batch job. The SIMCODE component in iSIGHT-FD is an operating system (OS) command component surrounded by two data exchanger components. It is used to wrap an external program that reads and writes files for use in ISIGHT-FD models. The big advantage of using the SIMCODE component over the separate data exchanger and OS command component is that copying of input and output files between machines is avoided; the files are created, used and discarded in a one step. Figure 18 shows the SIMCODE component that has been developed to run ANSYS in batch mode.

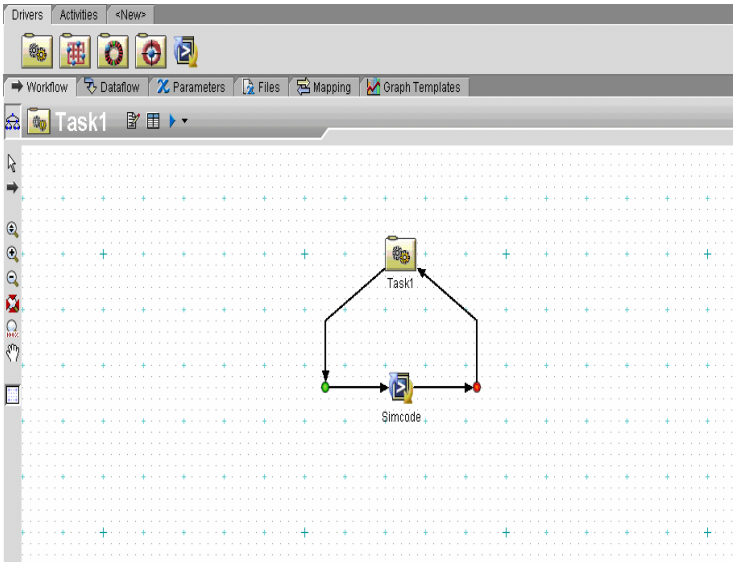



**Figure 18. SIMCODE component for running ANSYS in product development environment, iSIGHT-FD**

The location of the executable file is ["C:\Program Files\Ansys Inc\v90\ANSYS\bin\intel\ansys90"].

The –p option decides the type of ANSYS version being used. Vm1_dat and vm1_out are the input and output files being used. Depending on the type of ANSYS product working on the computer, commands can be issued accordingly as -p product name (this will define which ANSYS product will run during the session). One of the following commands has to be issued based on the ANSYS product running on the computer

-p ansyul ANSYS University Introductory version

-p ansysuh ANSYS University Intermediate version

-p ansysrf ANSYS University Research version

-p an3fl ANSY Multiphysics version

Hence the OS command used to run ANSYS in batch mode is as follows.

“C:\Program Files\Ansys Inc\v90\ANSYS\bin\intel\ansys90" -b -i vm1_dat -o vm1_out -p ansysrf

Figure 19 shows the OS command issued to the SIMCODE component. Vm1.dat is the input file of the component editor of the SIMCODE component. It has the necessary code to run ANSYS in batch mode. In the Vm1.dat file, code has been written to put the output at vm1.out file. So, vm1.out file need not be specified as output file in the component of editor of the SIMCODE component again.

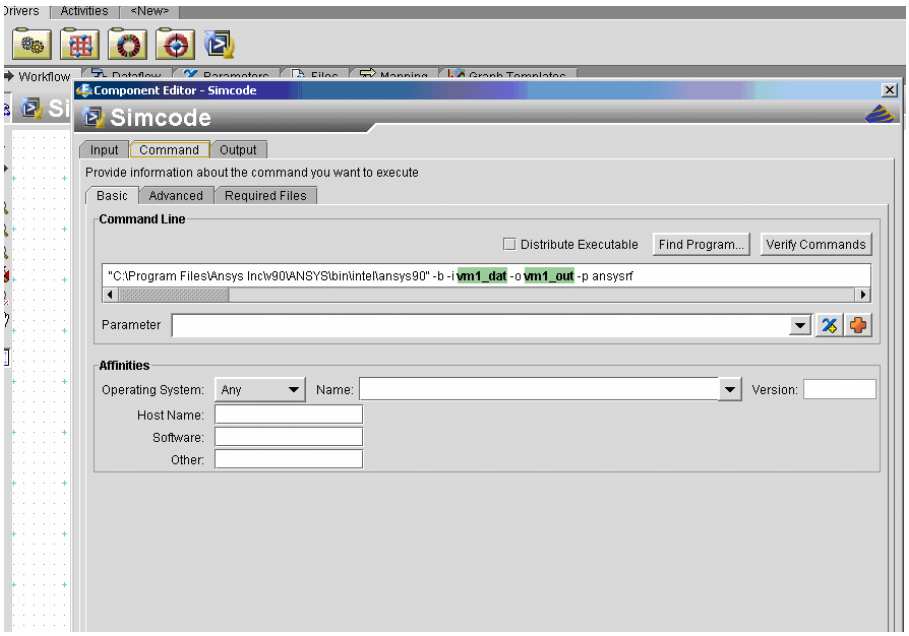



**Figure 19. Editor of the ANSYS SIMCODE component in iSIGHT-FD**

The output file is vm1.out and is saved at the same directory location as input vm1.dat file. The vm1.dat, in general is any input ANSYS command file used to run the analysis of a model. This command file when invoked by the ANSYS batch mode command runs the analysis and writes the results to the output file. The output to this file can be controlled using ANSYS commands just to include the results of the objective functions. These values can later be used as parameters to run the optimization tools.

The Data Exchanger component in iSIGHT-FD is used to prepare input files for external programs and to extract data from program output files. The main operation of the data exchanger is reading or writing a parameter. Parameters from other components can be written to the input file by a "reading" operation of the data exchanger and parameters from the data exchanger can be passed to other components by a "writing" operation of the data exchanger. Parameters from the knowledge base can be read into the input file which is used to run the ANSYS SIMCODE component in batch mode. The results of the analysis, in particular the objectives of the analysis will be written into the output file. These objectives can be written as parameters to other iSIGHT-FD components.

## 6.2 Optimization tool in product development environment

Optimization tool in a product development environment is used to perform the optimization of an analysis model for the given objective functions. This is done by accessing the design parameters and objective functions of the model as parameters of a product development environment.

The optimization component in iSIGHT-FD is a basic optimization design driver which has all required functionality for performing simple optimization studies on problems of various nature. The implementation of the optimization component allows execution of any single optimization algorithm from the following list

- Sequential quadratic programming –NLPQL
- Generalized reduced gradient –LSGRG2
- Multi-Island genetic algorithm
- Hooke-Jeeves direct search method

Most of the needs of the optimization design engineer are covered by this component.

The optimization component in iSIGHT-FD works by selecting the design variables, the constraints, technique and objectives of the model to be optimized. The list of parameters passed to the optimization component will also include the parameters passed to the optimization component by other sub flow components.

## 6.3 Developing Intelligent Decision making tool in product development environment

The intelligent decision making tool, "Intelligenet EAM selector", in a product development environment, iSIGHT-FD has been developed using the "Script" component of iSIGHT-FD. The script component allows execution of Java code in the model. It uses a dynamic interpreter to run the script. Details about using the script component and its execution can be found at FiPER documentation [FIPER, 2005]. The script component is executed statement-by-statement from the top down, meaning that classes, methods and variables must be declared before they are first used. Parameters that can be represented using the script component are Script real as JAVA double, Script Integer as JAVA long, Boolean as Boolean and string as java.lang.string. The value of a parameter is copied into the Java variable before the script starts and the value of the Java variable copied back into the parameter after the script finishes. It performs logical operations on the input and output parameters using JAVA programming. This logic drives decision making tools to perform appropriately. Figure 20 shows script editor working in iSIGHT-FD.

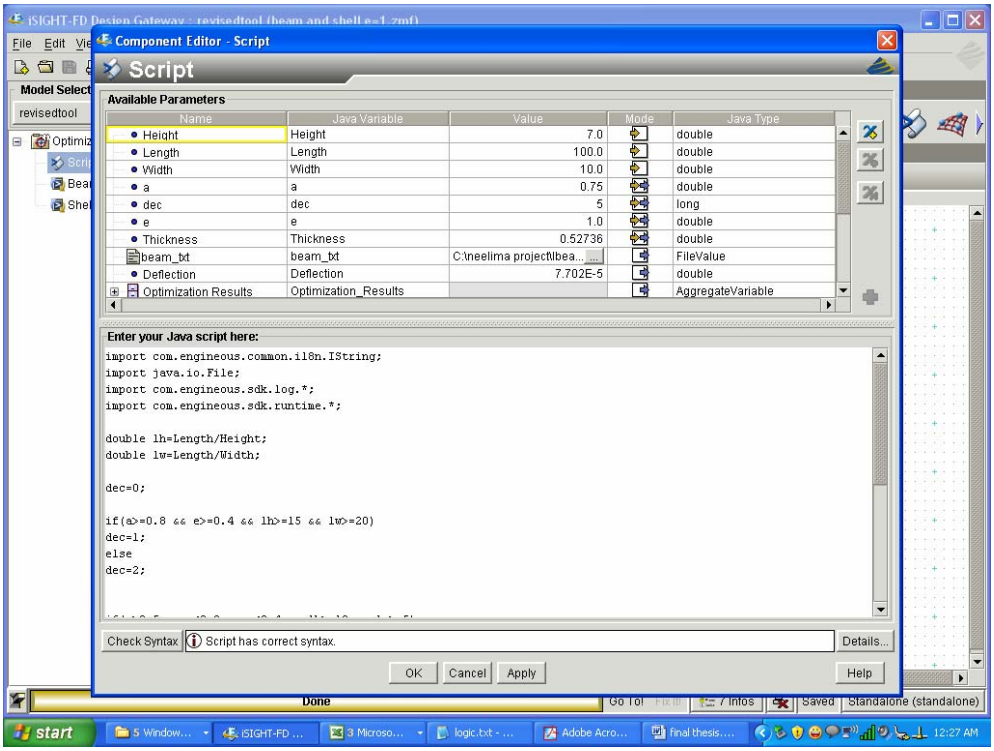



**Figure 20. Editor of Script component in iSIGHT-FD**

The input/output parameters are selected in the programming code for making the component understand that they are the associated input/output parameters declared in iSIGHT-FD environment. Any other parameters can be locally declared inside the JAVA code or created as iSIGHT-FD parameters if needed.

# CHAPTER 7

# TEST BED APPLICATION

Initially the Trapezoidal belt drive system [Kanuri, 2006] was used as a test bed example to test the working of the ANSYS SIMCODE component. Due to the complexity and time constraints associated with the trapezoidal belt drive system, a less complex I-Beam model has been used as a test bed application.

Engineers use beams to support and strengthen structures. An I-Beam is a beam with cross section which looks like the letter "I". The cross-section of a beam determines how a beam reacts to a load, and for this test bed application a uniform beam cross section has been assumed. Beams that strengthen a structure are subject to stresses put upon them by the weight of the structure and by external forces. The strength of a beam is proportional to the amount of force that may be placed upon it before it begins to either excessively deflect or permanently yield. An I-Beam optimization problem has been considered to demonstrate the interoperability of modeling knowledge between analysis and optimization tools in iSIGHT-FD.

## 7.1 General problem formulation

Figure 21 shows a cross section of cantilever beam, which has one end fixed and one free end. A load is applied to the free end. The model is analyzed using beam and shell element models of I-Beam. The cross sectional area of the beam depends on its height, width and thickness. The general objective of the optimization problem for the I-Beam is to minimize the volume of the beam for constant length and varying height, width and thickness based on the eccentricity of the load being applied. A static load of

1000lb is applied on the free end of the I-beam. This load can be applied with an eccentricity e ranging from 0 to the half width of the beam.

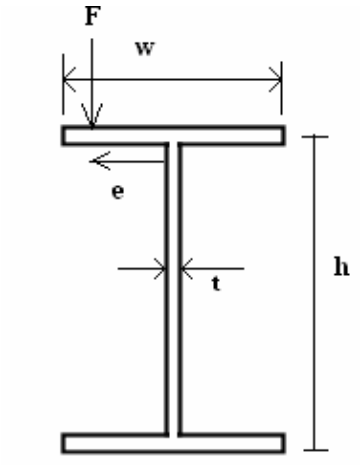



**Figure 21. Cross section of I-Beam Showing Eccentric Load**

## 7.2 Development of the analysis beam element model for I-Beam

Two models for I-Beam, the beam element model and the shell element model have been developed. The advantage of shell element model over the beam element model is that the shell element model captures eccentricity of the load applied on the width of the beam. On the other hand the beam element model is suitable for quickly analyzing the beam structures with no load eccentricity. Loads on beam element models cannot be applied on nodes which are off centric. However, a beam element model can be used to model an eccentrically loaded beam if the error due to ignoring the load eccentricity is within the desired accuracy bounds of the analysis.

The beam element model has been developed in ANSYS using BEAM188 element. This element is commonly used for analyzing slender to moderately stubby/ thick beam structures. BEAM188 is a linear two node or a quadratic beam element in 3D. It has six degrees of freedom at each node. The beam section has been defined using the

common sections of Beam available with ANSYS and has been meshed with BEAM188 elements. This element is well suited for linear, large rotation and large strain non linear applications. This element includes stress stiffness terms by default. The beam elements are one dimensional line elements in space. Hence eccentricity cannot be captured by the beam element model. The beam elements are based on Timoshenko beam theory which is a first order shear deformation theory, i.e. the transverse shear strain is constant through the cross section. The slenderness ratio of a beam structure ($GAL^2$ / (EI)) has been used in judging the applicability of the element where G is shear modulus, A is area of cross section, L is the length of the member and EI is flexural rigidity. Forces are applied at the nodes. If the centroidal axis is not collinear with the element X-axis applied axial force will cause bending. Applied shear force will cause torsional strains and moment if the centroid and shear center of the cross section are different. The nodes should therefore be located at the desired points where the force has to be applied. The model has been built to be symmetric about the Y-axis.

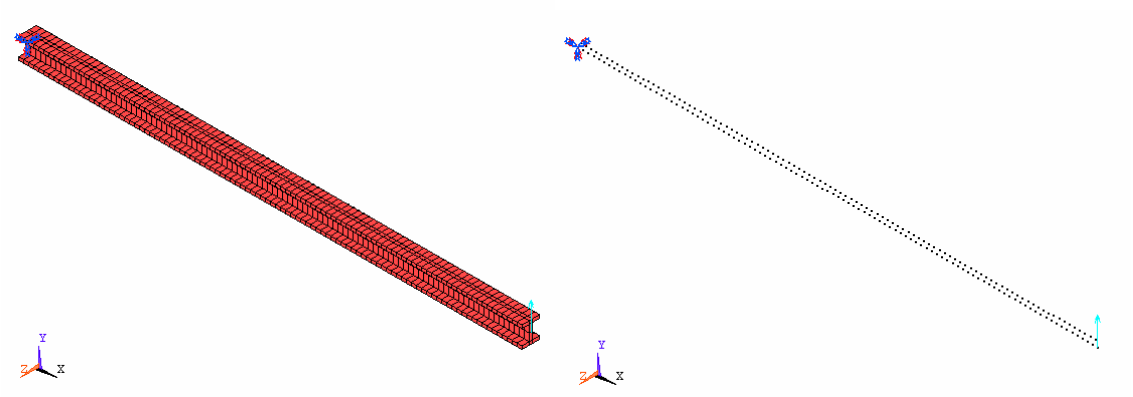


**Figure 22. Orientation of elements and nodes for I-Beam model using element BEAM188**

Figure 22 shows the BEAM188 elements and the orientation of nodes for the beam element. In this application the nodes have been generated on the X-axis and an

axial force is applied in the Y-direction to cause bending. The axial force cannot be applied off centric due to the absence of nodes generated by beam elements. Hence eccentricity cannot be captured by beam elements.

The batch file was created for the I-Beam model to run in iSIGHT-FD using the ANSYS SIMCODE component. The ANSYS command files for the beam and the shell element models are included in Appendix B. This batch file has been parameterized. The design parameters are the length, which is usually constant for a particular optimization problem, height, width and the thickness of the beam. The ANSYS SIMCODE component in iSIGHT-FD runs in batch mode by accessing the input file for beam model and writes the output to the results file specified within the input file. The input file has been parameterized in terms of the design parameters of the I-beam beam element model.

## 7.3 Development of the analysis shell element model for I-Beam

The shell element model was developed in ANSYS using SHELL181 elements. This element is suitable for analyzing thin to moderately-thick shell structures. It is a 4-node element with six degrees of freedom at each node. SHELL181 is well suited for linear, large rotation and large strain non linear applications. The section has been created using the symmetric property about the Y-plane and meshed with SHELL181 elements. The element formulations are based on logarithmic strain and true stress measures. The thickness of the shell is defined using the real constant and can be defined for each of its nodes. For this test bed application the thickness is taken as constant for all the four nodes that define the shell element. The cantilever beam and beam cross section to be modeled with shells are typical examples of bending. This element works best with the full Newton-Raphson solution scheme used in ANSYS.

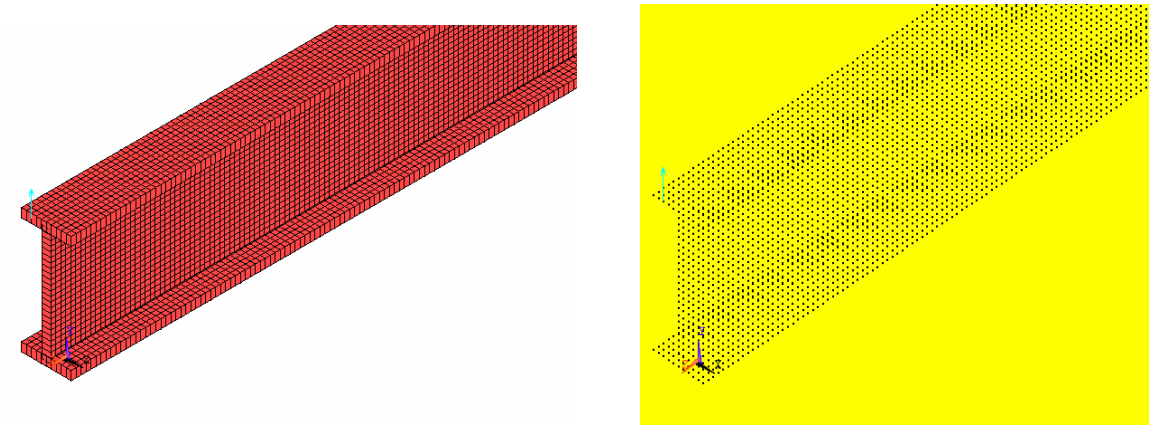


**Figure 23. Orientation of elements and nodes for I-Beam model using element SHELL181**

Figure 23 shows SHELL181 elements and the orientation of nodes for the shell element. Using SHELL181, nodes are generated both on X, Y and Z axes. Hence, unlike beam element, shell element can account for eccentricity of the load on the width of the beam. An element size of (0.2*width of the beam) has been used to mesh the model with shell elements. The dimension of the width of the beam is in inches. As the nodes are generated on the width of the beam, shell elements captures the effect of off centric load from the center of the width of the beam. The eccentricity of the load depends on the width of the beam. Beams with greater widths have more eccentricity from the center of the beam, i.e. as the beams get wider, the off centric loads can get further away. The element size for meshing has been selected in a way so that the eccentricity values as a fraction of the beam's half width that can be used are 0, 0.2, 0.4, 0.6, 0.8 and 1. When applying force on a shell element model, the force has to be applied on the node corresponding to the user specified eccentricity. Since it is tough to apply force by generating elements with varying element size based on the eccentricity, an element size

of (0.2*width of the beam), in inches, has been selected by taking significant effects of different eccentricity values. The off centric loads result in torsion effects on the beam.

## 7.4 Tradeoff between the beam element model and the shell element model

The developed analysis input files for the beam and the shell element models have been used to run tests for deciding the tradeoff criteria between the beam and shell element models. The regular beam elements cannot capture the torsion effect on the beam due to the eccentricity of the load. Shell elements can capture the effects of torsion. Shell elements are computationally expensive

The two models have been run with different length/height and length/width ratios. The length/width ratio remains constant for different values of length/height ratios and vice versa. Point load is applied on the width of the beam. The eccentricity ratio *ER* is defined as the percentage of offset of the point load from the center of the beam with respect to half of beam's width:

$$ER = 100\left(\frac{e}{w/2}\right) \tag{7.1}$$

where *e* is the offset distance from the center of the beam of the load and *w* is the width of the beam.

Since eccentricity of the applied load can be captured only by the shell elements, eccentricity ratio values of 20%, 40% and 100% have been used to run the sample analyses tests to determine the tradeoff between the models. The models have also been analyzed for intermediate eccentricities. But, it has been found that intermediate eccentricity ratios (for example 30%, 50% etc.) show changes in the results of the analyses which do not make much difference in the decision making process of shifting between the models for a design optimization problem. Hence intermediate eccentricity

values have been ignored. This facilitates running fewer numbers of analysis tests and also automatic application of the force on the nodes present on the width of the beam based on the eccentricity values. Hence an element size of (0.2*width of the beam), in inches, has been decided upon. Value of deflection has been used as criteria to differentiate between the beam and the shell element models, as the performance of the beam is being analyzed in terms of the deflection based on the eccentricity of loading. Using stress as differentiation criterion also had similar results. Since shell element can capture the eccentricity effects of the beam, the shell element model is highly accurate, though it needs more computational time. Percent error (PE) has been calculated as the percent difference in the value of the deflection of the beam element relative to the deflection of the shell element.

$$PE = 100\left(\frac{\delta_{shell} - \delta_{beam}}{\delta_{shell}}\right) \tag{7.2}$$

Table 3 lists the results of running analysis models for varying length/width ratios for a constant length/height ratios.

**Table 2. Results for von Mises stress and deflection for varying length/width ratios for a constant length/height ratio of 25 and thickness of 0.5**

| Model (Eccentricity Ratio %) | l/w ratio | von Mises (psi) stress*$10^3$ | Deflection*$10^3$ (inch) | Percentage Error |
|---|---|---|---|---|
| Beam | 5 | 3.16912 | 5.56e-4 | |
| Shell (20) | 5 | 5.8626 | 7.19e-4 | 22.67 |
| 40 | 5 | 10.883 | 9.18e-4 | 35 |
| 100 | 5 | 18.523 | 1.6e-3 | 65 |
| Beam | 7 | 4.4 | 7.6e-4 | |
| Shell (20) | 7 | 7.93 | 8.9e-4 | 14.6 |
| 40 | 7 | 11.03 | 1.04e-3 | 27 |
| 100 | 7 | 18.5 | 1.56e-3 | 51.2 |
| Beam | 10 | 6.2266 | 1.06e-3 | |
| Shell (20) | 10 | 7.466 | 1.1e-3 | 3.6 |
| 40 | 10 | 10.96 | 1.26e-3 | 15 |
| 100 | 10 | 18.3344 | 1.64e-3 | 36 |
| Beam | 12 | 7.422 | 1.26e-3 | |
| Shell (40) | 12 | 10.8219 | 1.42e-3 | 11.26 |
| 100 | 12 | 18.188 | 1.74e-3 | 28 |
| Beam | 15 | 9.187 | 1.55e-3 | |
| Shell (40) | 15 | 11.417 | 1.66e-3 | 6.5 |
| 100 | 15 | 17.963 | 1.92e-3 | 21 |
| Beam | 20 | 12.02 | 2.03e-3 | |
| Shell (40) | 20 | 14.616 | 2.06e-3 | 18 |
| 100 | 20 | 17.64 | 2.20e-3 | 8 |
| Beam | 25 | 14.786 | 2.5e-3 | |
| Shell (40) | 25 | 17.766 | 2.47e-3 | 2 |
| 100 | 25 | 19.616 | 2.61e-3 | 5 |
| Beam | 35 | 20.07 | 3.38e-3 | |
| 40 | 35 | 23.915 | 3.22e-3 | 4.7 |
| 100 | 35 | 25.75 | 3.32e-3 | 1.77 |
| Beam | 45 | 24.967 | 4.2e-3 | |
| Shell (40) | 45 | 29.696 | 3.92e-3 | 6 |
| 100 | 45 | 31.52 | 3.99e-3 | 7 |
| | | | | |
| Beam | 55 | 29.8119 | 5e-3 | |
| Shell (40%) | 55 | 35.454 | 4.592e-3 | 8 |
| Shell (100%) | 55 | 37.268 | 4.64e-3 | 7.2 |

**Table 3. Results for von Mises stress and deflection for varying length/height ratios for a constant length/width ratio of 25 and thickness of 0.5**

| Model (ER) | l/h ratio | Von mises stress(psi) $*10^3$ | Deflection$*10^3$ (inch) | PE |
|---|---|---|---|---|
| Beam | 5 | 1.49366 | 5.28e-5 | |
| 20% | 5 | 9.5704 | 1.45e-4 | 63.6 |
| Shell (40%) | 5 | 17.3809 | 3.24e-4 | 83.7 |
| Shell (100%) | 5 | 8.3018 | 9.08e-5 | 41 |
| Beam | 7 | 2.47 | 1.19e-4 | |
| Shell (20%) | 7 | 9.587 | 1.99e-4 | 40.2 |
| Shell (40%) | 7 | 17.386 | 3.71e-4 | 68 |
| Shell (100%) | 7 | 18.328 | 1.5e-4 | 20.7 |
| Beam | 10 | 4.121 | 2.81e-4 | |
| Shell (20%) | 10 | 8.215 | 3.44e-4 | 7.8 |
| Shell (40%) | 10 | 9.609 | 4.98e-4 | 18.31 |
| Shell (100%) | 10 | 17.392 | 4.03e-4 | 43.57 |
| Beam | 12 | 5.326 | 4.35e-4 | |
| Shell (20%) | 12 | 8.236 | 4.5e-4 | 3 |
| Shell (40%) | 12 | 9.622 | 4.91e-4 | 11.4 |
| Shell (100%) | 12 | 17.392 | 6.34e-4 | 31.4 |
| Beam | 15 | 7.268 | 7.39e-4 | |
| Shell (40%) | 15 | 9.7257 | 7.83e-4 | 2.17 |
| Shell (100%) | 15 | 17.399 | 9.2e-4 | 1.6 |
| Beam | 20 | 10.8149 | 1.46e-3 | |
| Shell (40%) | 20 | 13.576 | 1.48e-3 | 1.35 |
| Shell (100%) | 20 | 17.4 | 1.62e-3 | 9 |
| Beam | 25 | 19.18 | 3.22e-3 | |
| Shell (40%) | 25 | 22.37 | 3.15e-3 | 2.17 |
| Shell (100%) | 25 | 24.094 | 3.17e-3 | 1.6 |
| Beam | 35 | 24.1553 | 5.7e-3 | |
| Shell (40%) | 35 | 27.34 | 5.5e-3 | 11 |
| Shell (100%) | 35 | 28.965 | 5.7e-3 | 15 |
| Beam | 45 | 35.427 | 1.07e-2 | |
| Shell (40%) | 45 | 38.496 | 1.03e-2 | 8 |
| Shell (100%) | 45 | 39.79 | 1.06e-2 | 11 |
| Beam | 55 | 29.8119 | 1.81e-2 | |
| Shell (40%) | 55 | 35.454 | 1.75e-2 | 4.12 |
| Shell (100%) | 55 | 37.268 | 1.77e-2 | 3.56 |

Table 2 and Table 3 it can be observed that as the error in the values of the deflection for the beam and the shell element models increase, the accuracy decreases. Hence a highly accurate model demands using the shell element model if the difference between the values of deflection of beam and shell element models is greater than 20% for a given eccentricity, length/height and length/width ratios. Similarly, for the same eccentricity and ratios of length, height and width, a medium accurate model can use beam or shell element model depending on the difference in the values of deflections of the beam and the shell element model.

Sample analyses runs for varying length/height ratios for constant length/width ratio have also been done. For length/width ratio values of 25 and 35, the variation of accuracy with increasing length/height ratios is shown below in Figure 24.

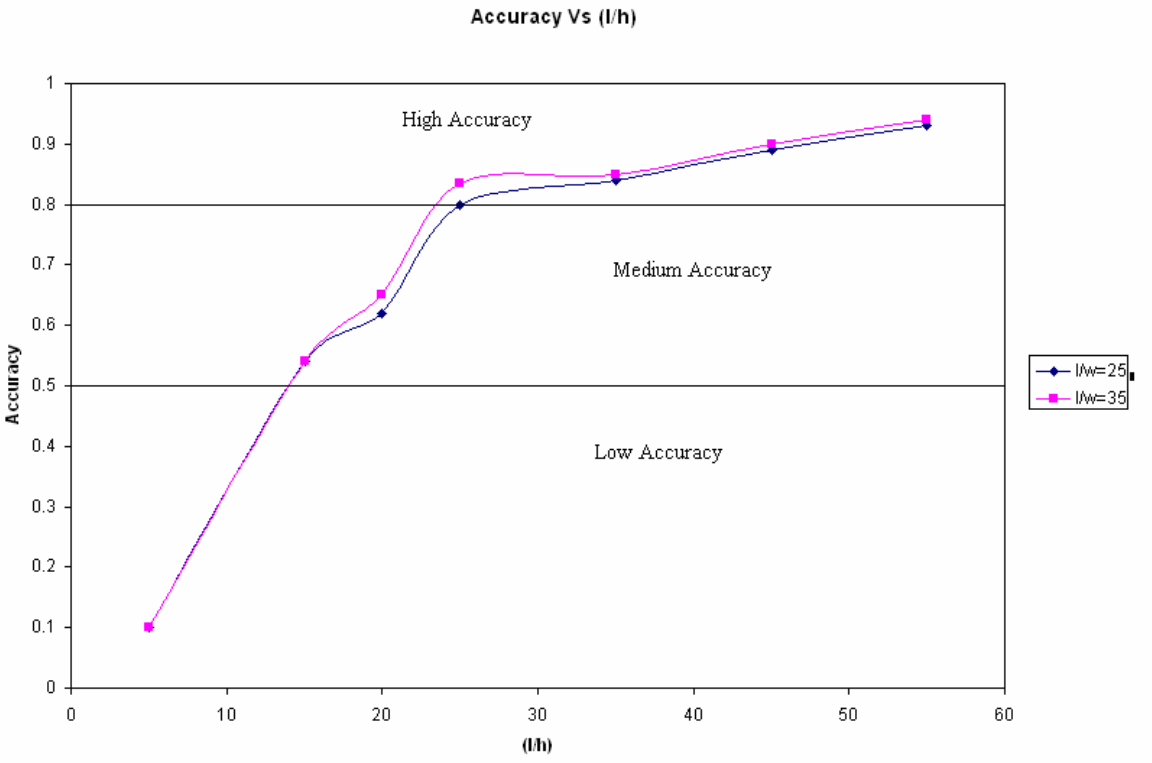



**Figure 24. Results for accuracy percentage of beam element model vs. (length/height) ratio for constant length/width ratios of 25 and 35 for the I-beam model shown**

From the results obtained and the graph plotted in Figure 24, it can be seen that as the length/width ratios keep increasing, the accuracy of the beam element model increases for varying length/height ratios. Hence shell element models are more accurate for lower length/height ratios when length/width ratio is a constant.

Similarly, sample analyses runs for varying length/width ratios for constant length/height ratio have also been done. For length/height ratio values of 20 and 25, the variation of accuracy with increasing length/width ratios is shown in Figure 25.

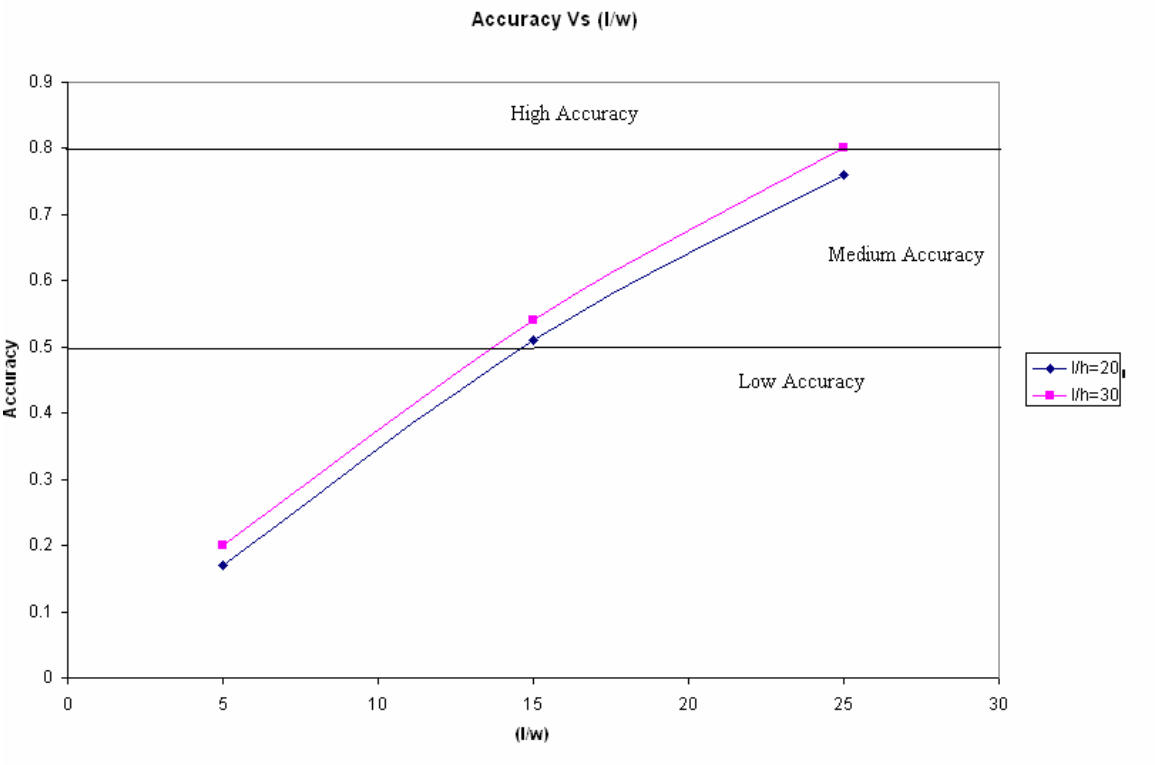



**Figure 25. Results for accuracy percentage of beam element model vs. (length/width) ratio for constant length/height ratios of 20 and 30**

From the results obtained and the graph plotted in Figure 25, it can be seen that as the length/height ratios keep increasing, the accuracy of the beam element model increases for varying length/width ratios. Hence shell element models are more accurate for lower length/width ratios when length/height ratio is a constant.

From Figure 24 and Figure 25, it can be generalized that shell element models are more accurate for lower length/height or length/width ratios. Beam elements models are accurate for greater length/height and length/width ratios. Also, the model to be selected depends on the required accuracy. As it is obvious, both the models have the same volume for a set of design parameters. The shell element model is highly accurate and takes care of eccentricity of the load in terms of the torsion effect on the beam. The beam elements model is as accurate as the shell element model where there is no eccentricity of

the load. Hence for zero eccentricity, beam and shell model produce the same results, but for loads with eccentricities, the shell element model generates the most accurate results taking both torsion and bending effects into consideration.

A trade off for switching between analysis models has been developed both in terms of accuracy required and the eccentricity of the load. Accuracy expectation can be justified by selecting the right model, beam element model or shell element model, for a given eccentricity of the load. Three levels of accuracy have been considered. A "low" accuracy model is defined as the model that can be used when the difference in the deflections of the beam and the shell elements models is in between 50%-100% for a given eccentricity of the load. A "medium" accurate model has differences ranging from 20% to 50%. A "high" accuracy model is defined as the model that can be used when the difference in the deflections of the beam and the shell elements models is in the 1%-20% range for a given eccentricity of the load.

Table 4 and Table 5 list the selection of the right model, beam element model or shell element model based on the difference in the values of deflections of the models for given eccentricity and length, height, width ratios.

**Table 4. Required model based on accuracy expectation and eccentricity of the load for varying l/h ratios, l/w=25 and thickness=0.5**

| Shell element model Eccentricity (%) | l/h ratio | Percent Error | Accuracy Expectation | | |
|---|---|---|---|---|---|
| | | | Low <=50% | Medium 50-80% | High >80% |
| | | | Required Model | | |
| 20 | 5 | 63.6 | Beam | Shell | Shell |
| 40 | | 83.7 | Beam | Shell | Shell |
| 100 | | 41 | Beam | Shell | Shell |
| | | | | | |
| 20 | 7 | 40.2 | Beam | Shell | Shell |
| 40 | | 68 | Beam | Shell | Shell |
| 100 | | 20.7 | Beam | Shell | Shell |
| | | | | | |
| 20 | 10 | 7.8 | Beam | Beam | Beam |
| 40 | | 18.31 | Beam | Beam | Beam |
| 100 | | 43.57 | Beam | Beam | Shell |
| | | | | | |
| 20 | 12 | 3 | Beam | Beam | Beam |
| 40 | | 11.4 | Beam | Beam | Beam |
| 100 | | 31.4 | Beam | Beam | Shell |
| | | | | | |
| 40 | 15 | 6.5 | Beam | Beam | Beam |
| 100 | 15 | 21 | Beam | Beam | Beam |
| | | | | | |
| 40 | 20 | 1.35 | Beam | Beam | Beam |
| 100 | 20 | 9 | Beam | Beam | Beam |
| | | | | | |
| 40 | 25 | 2.17 | Beam | Beam | Beam |
| 100 | 25 | 1.6 | Beam | Beam | Beam |
| | | | | | |
| 40 | 35 | 11 | Beam | Beam | Beam |
| 100 | 35 | 15 | Beam | Beam | Beam |
| | | | | | |
| 40 | 45 | 8 | Beam | Beam | Beam |
| 100 | 45 | 11 | Beam | Beam | Beam |
| | | | | | |
| 40 | 55 | 4.12 | Beam | Beam | Beam |
| 100 | 55 | 3.56 | Beam | Beam | Beam |

From Table 4 with varying length/height ratios and for constant length/width = 25, it can be observed that as the length/height ratio increases the beam element model becomes as accurate as the shell element model. Depending on the accuracy requirement, the beam or shell element model can be used depending on the ratios of length, width, height and also the eccentricity of the beam. For length/height ratio greater than 15 the beam element model can be used for any accuracy or eccentricity of the load.

From Table 5 with varying length/width ratios and for constant length/height ratio = 25, it can be observed that as the length/width ratio increases the beam element model becomes as accurate as the shell element model. Depending on the accuracy requirement, the beam or shell element model can be used depending on the ratios of length, width, height and also the eccentricity of the beam. For length/width ratio greater than 20 the beam element model can be used for any accuracy or eccentricity of the load.

For developing an intelligent decision making tool which switches between the beam and the shell element models, the logic has been taken depending on the eccentricity of the load and the accuracy of the model. For length/height ratios greater than 15 for a constant length/width ratio, the beam element is as accurate as the shell element model for any eccentricity. For length/width ratios greater than 20, for constant length/height ratio, the beam element model is accurate as the shell element model for any eccentricity. For intermediate accuracy, eccentricity and length, width, height ratios, the tools shifts between the beam and the shell element models as listed in Table 4 and Table 5.

**Table 5. Required model based on the accuracy expectation and eccentricity of the load for varying l/w ratios, l/h=25 and thickness=0.5**

| Shell element model Eccentricity (%) | l/w ratio | Percent Error | Accuracy Expectation | | |
|---|---|---|---|---|---|
| | | | Accuracy Low <=50% | Medium 50-80% | High >80% |
| | | | Selected Model | | |
| 20 | 5 | 22.67 | Beam | Beam | Shell |
| 40 | | 35 | Beam | Beam | Shell |
| 100 | | 65 | Beam | Shell | Shell |
| | | | | | |
| 20 | 7 | 14.6 | Beam | Beam | Beam |
| 40 | | 27 | Beam | Beam | Shell |
| 100 | | 51.2 | Beam | Shell | Shell |
| | | | | | |
| 20 | 10 | 3.6 | Beam | Beam | Beam |
| 40 | | 15 | Beam | Beam | Beam |
| 100 | | 36 | Beam | Beam | Shell |
| | | | | | |
| 40 | 12 | 11.26 | Beam | Beam | Beam |
| 100 | | 28 | Beam | Beam | Beam |
| | | | | | |
| 40 | 15 | 6.5 | Beam | Beam | Beam |
| 100 | | 21 | Beam | Shell | Shell |
| | | | | | |
| 40 | 20 | 18 | Beam | Beam | Beam |
| 100 | | 8 | Beam | Beam | Beam |
| | | | | | |
| 40 | 25 | 2 | Beam | Beam | Beam |
| 100 | | 5 | Beam | Beam | Beam |
| | | | | | |
| 20 | 35 | | | | |
| 40 | | 4.7 | Beam | Beam | Beam |
| 100 | | 1.77 | Beam | Beam | Beam |
| | | | | | |
| 40 | 45 | 6 | Beam | Beam | Beam |
| | | | Beam | Beam | Beam |

## 7.5 Theoretical calculation of bending and twisting deflections of I-beam

The decision to shift between the beam and the shell element models for the I-beam optimization is listed in Table 4 and Table 5. These results have been obtained by running analysis models for beam and shell element models to obtain the deflection and maximum von Mises stress. Theoretically the logic to shift between the models has also been obtained by calculating the total deflection, sum of bending and twisting deflections.

For a given nonzero eccentricity, the accuracy of the beam element model in terms of deflection versus the shell element model in terms of deflection should be analytically computable. For example, for an end loaded cantilever beam of length L, height h, width b, web and flange thickness t, and elastic modulus E, its deflection due to an end load of *F* neglecting eccentricity is

$$\delta_b = \frac{FL^3}{3EI_x} \tag{7.3}$$

where the moment of inertia $I_x$ of the beam's cross section about horizontal x-axis is

$$I_x = \frac{bh^3 - (b-t)(h-t)^3}{12} \tag{7.4}$$

F is 1000 lb and length L = 100 in. for our analyses. The deflection due to torsion only when the load is eccentrically applied can be calculated using the torsional stiffness of an I-shaped beam with width W and height H. This calculation neglects any deflection due to bending of the upper flange. The vertical deflection at the width edge of the beam

section due to pure twisting of the beam due to an applied torque $T = Fe$, where $e$ is the eccentricity of the load, is

$$\delta_e = 2r\sin(\theta/2) \tag{7.5}$$

where

$$r = \frac{1}{2}\sqrt{w^2 + h^2} \tag{7.6}$$

and $\theta$ is the angle of twist given by

$$\theta = \frac{TL}{GJ} \tag{7.7}$$

J is the polar moment of inertia of the section. It is given by

$$J = I_x + I_y \tag{7.8}$$

where for this I-Beam with the y-axis in the vertical direction

$$I_y = \frac{1}{12}\left(2tb^3 + \left(h - 2t\right)t^3\right)\Big) \tag{7.9}$$

G is the shear modulus which for isotropic material is given by

$$G = \frac{E}{2(1+\nu)} \tag{7.10}$$

Note that this calculation neglects any deflection due to bending of the upper flange where the load is applied about the beam's longitudinal axis. From Figure 26 the total vertical deflection is the direct sum of the bending and torsional deflections.

$$\delta_{total} = \sqrt{(\delta_b^2 + \delta e^2 - 2*\delta_b * \delta e*\cos(180-\theta))} \tag{7.11}$$

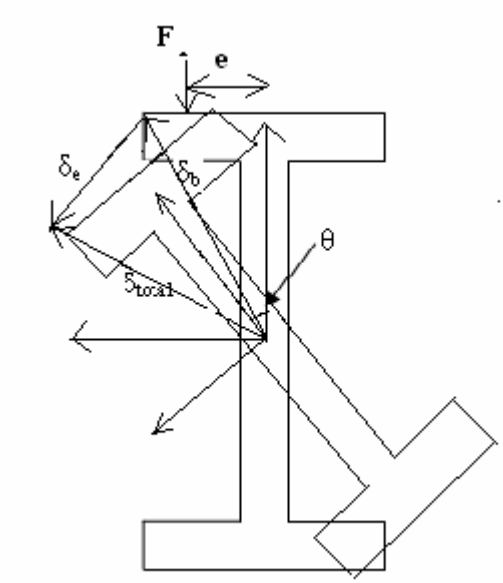



**Figure 26. Cross section of I-beam, deflections due to bending and eccentricity**

Table 6 lists the results obtained for bending deflection and the total deflection using (eqn7.1), (eqn7.3) and (eqn7.5) and for an eccentricity value of 1.

**Table 6. Results obtained for bending and total deflections using theoretical calculations**

| b(in) | d(in) | t(in) | h(in) | I(moment o | I(Abt Y axis) | Bending D | Total Deflec(in) | Difference( | a<50% | 50<a<80 | a>80 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 20 | 4 | 0.5 | 3 | 62.79167 | 682.6667 | 0.530856 | 1.140901 | 53.47046 | beam | shell | shell |
| 14.28 | 4 | 0.5 | 3 | 45.155 | 258.6629 | 0.738198 | 1.632572 | 54.78312 | beam | shell | shell |
| 10 | 4 | 0.5 | 3 | 31.95833 | 99.33333 | 1.043025 | 1.748743 | 40.35574 | beam | shell | shell |
| 8.33 | 4 | 0.5 | 3 | 26.80917 | 64.16746 | 1.243356 | 1.796324 | 30.78332 | beam | shell | shell |
| 6.66 | 4 | 0.5 | 3 | 21.66 | 40.61736 | 1.538935 | 1.9798 | 22.26816 | beam | shell | shell |
| 5 | 4 | 0.5 | 3 | 16.54167 | 26.41667 | 2.015113 | 2.45849 | 18.03451 | beam | beam | beam |
| 4 | 4 | 0.5 | 3 | 13.45833 | 21.33333 | 2.47678 | 2.786 | 11.09906 | beam | beam | beam |
| 4 | 20 | 0.5 | 19 | 666.125 | 12672 | 0.050041 | 0.07968 | 37.19797 | beam | shell | shell |
| 4 | 14.28 | 0.5 | 13.28 | 287.5567 | 3227.897 | 0.115919 | 0.16483 | 29.6735 | beam | shell | shell |
| 4 | 10 | 0.5 | 9 | 120.7083 | 755.3333 | 0.276148 | 0.348123 | 20.67524 | beam | beam | beam |
| 4 | 8.33 | 0.5 | 7.33 | 77.80193 | 358.4008 | 0.428438 | 0.52734 | 18.75481 | beam | beam | beam |
| 4 | 6.66 | 0.5 | 5.66 | 45.584 | 144.6676 | 0.731251 | 0.8356 | 12.48794 | beam | beam | beam |
| 4 | 5 | 0.5 | 4 | 23 | 47 | 1.449275 | 1.6948 | 14.48694 | beam | beam | beam |
| 4 | 2.857 | 0.5 | 1.857 | 5.905606 | 8.942121 | 5.644354 | 5.9346 | 4.890737 | beam | beam | beam |
| 4 | 2.22 | 0.5 | 1.22 | 3.117394 | 6.445673 | 10.69269 | 10.834 | 1.304292 | beam | beam | beam |
| 4 | 1.8 | 0.5 | 0.8 | 1.794667 | 5.722133 | 18.57355 | 18.7324 | 0.847989 | beam | beam | beam |

As observed from Table 6 above the logic for shifting between the beam and the shell element models is almost the same as obtained from ANSYS results. The difference between the values of total deflection and the bending deflection is used to decide the

model to be used based on the required accuracy. An eccentricity ratio of 100% is used to calculate the bending deflection.

Results for other eccentricity ratio values of 20% and 40% have not been tabulated. These values produced minor changes in the results due to the second term in Eqn. (7.11) which is sine of a constant multiplied by the eccentricity. Hence this equation has not been used to verify the deflection values due to eccentricities of 20% and 40%. Whereas the theoretical results obtained using equations 1, 3 and 6 proved to be effective for deciding the logic to shift between the beam and the shell element models for an eccentricity value of 1. These values are in accordance to the results obtained by analysis of I-beam using ANSYS. Hence the theoretical values obtained offer a justification to the values obtained using the analysis tool and the logic to shift between the beam and the shell element models.

# CHAPTER 8

# FORMULATION OF I-BEAM OPTIMIZATION PROBLEM IN ISIGHT-FD

## 8.1 Developing EAM ontology instance for the I-Beam beam and shell element models

Ontological instances for I-Beam beam and shell element analysis models have been developed. The ontology instance contains knowledge about the objectives, input/ output parameters, accuracy expectation, limitations and idealizations of the analysis models. The input parameters of the I-beam beam element model are thickness, length, width and height. The input parameters for I-beam shell element model are the same as the beam element model with an added eccentricity of the load being applied. The limitations of the beam element model are no eccentricity, uniform area of cross section, constant force and no torsional effects.

For shell element model, the limitations are uniform area of cross section, constant force, eccentricity of the load, mesh size of 0.2 and torsional effects included. Figure 27 shows the I-beam instances for the beam element model and the shell element model. The knowledge about input parameters from the instance has been passed as parameters to the product development environment to be used to run automated analysis and optimization processes.

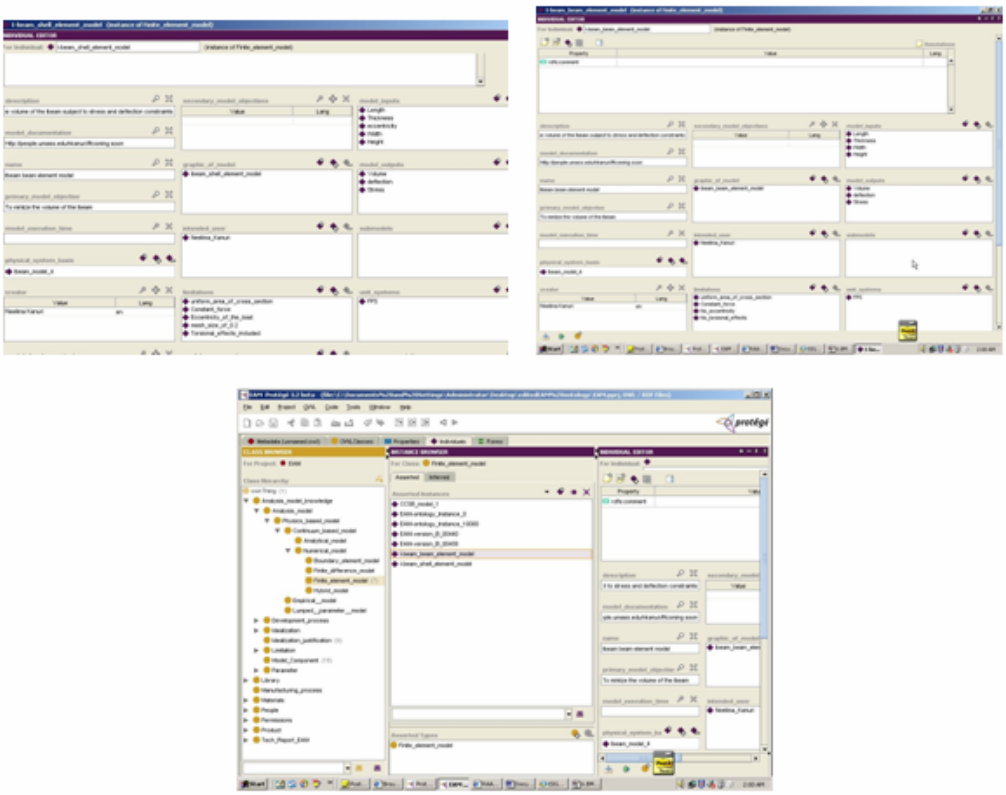


**Figure 27. Instance for beam and shell element analysis models of I-beam**

## 8.2 Configuring analysis and optimization components in iSIGHT-FD for I-Beam

### 8.2.1 Setting up the analysis component in iSIGHT-FD

The SIMCODE wrapped "ANSYS" component has been used as the analysis tool within the product development environment. This component runs an ANSYS finite element analysis for the beam and shell element analysis models of an I-beam. This component takes the input file, runs ANSYS in batch mode and prints the output to an external file. The input file has been parameterized so that the design parameters length, height, width and thickness can be used as parameters in a product development environment, shareable with other components in the optimization workflow. The input

file when run in the batch mode of ANSYS, prints the output to the output file specified. The printing of the output has been configured such that only the von Mises stress, deflection and the volume of the model gets printed to the output file. This has been done by directing the output either to the terminal or the output file depending on what values are required as output parameters. In the optimization of I-Beam problem, the output parameters are von Mises stress, deflection and the volume of the beam. For the shell element model, the eccentricity of the beam has also been considered as an input parameter. As beam element model cannot capture the effect of eccentricity of load on the I-beam, it is not taken as an input parameter for the beam element model.

The input/output parameters in a SIMCODE component can either be read/written to the files using the data exchanger component of SIMCODE. The data exchanger component allows moving data between iSIGHT-FD parameters and text files easily and efficiently. The value of the parameter can be written into the file, or data in the file can be read and used to set the value of the parameter. It is most frequently used to prepare input files for external programs and to extract data from program output files. The reading or writing of a parameter is done by clicking or swiping in the data source area or by entering the details in the swipe details area. For the input files of the shell and the beam components, the input parameters are created by sweeping the area of the input text file which has the values of length, height, width and thickness of the I-beam. For the shell element model the eccentricity of the load being applied is also an input parameter. The output parameters are von Mises stress, deflection and volume which have been created by sweeping the area of the output text file.

Figure 28 demonstrates the setting up of the SIMCODE and the data exchanger component for the I-beam beam and shell element models.

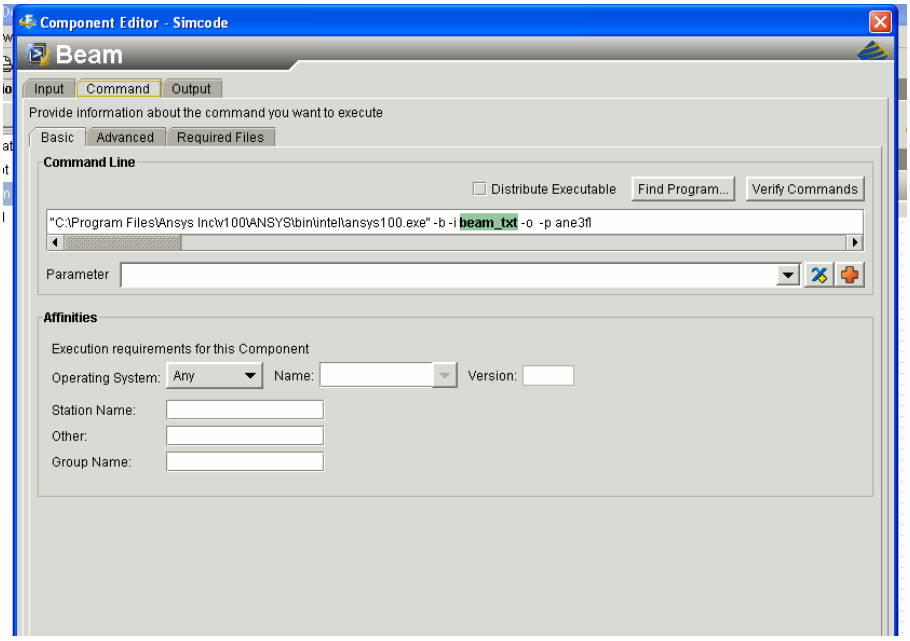



**Figure 28. Editor of SIMCODE component**

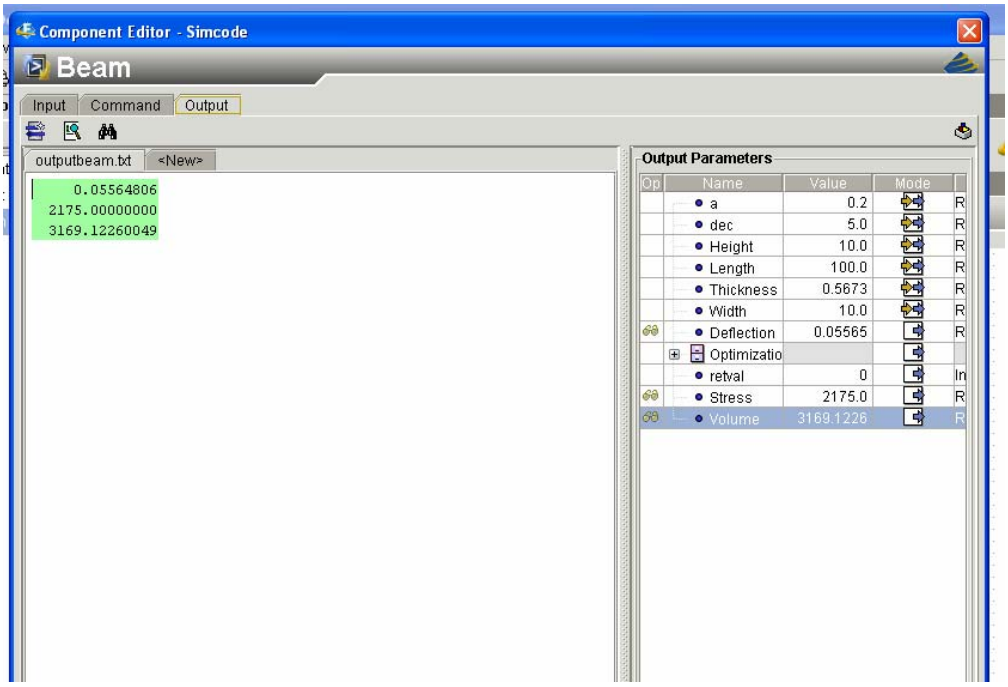



**Figure 29. Output file and mapping output parameters**

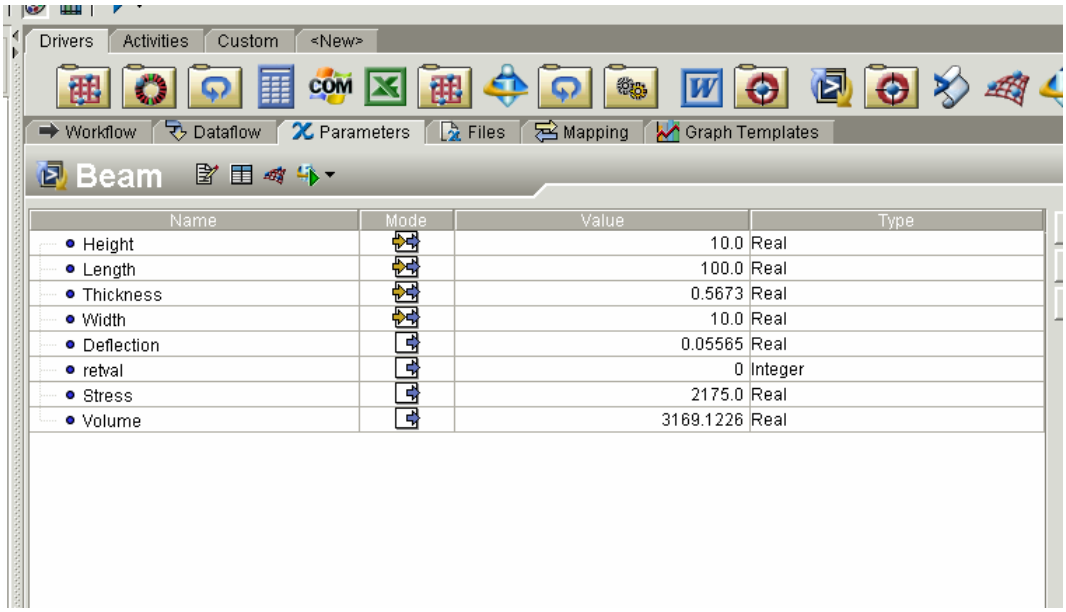



**Figure 30. Parameters for the SIMCODE component**

As seen from Figure 28, Figure 29 and Figure 30, the length, height, width and thickness of the beam have been set up as the input parameters. The von Mises stress, thickness and volume have been setup as the output parameters. The input file beam.txt is read as the input for the batch mode run of ANSYS and the output has been directed to Beamoutput.txt file. Similarly, the shell model input file has been setup with the same read/write parameters as the beam parameters. The eccentricity of the load has been taken as an input parameter for the shell model.

### 8.2.2 Setting up the optimization component in iSIGHT-FD

The optimization component in FiPER has been setup with ANSYS SIMCODE component running the analysis such that for each design state of the optimization, the ANSYS SIMCODE component runs the analysis and prints the output to the output file which will be redirected to the optimization run. As mentioned in section 6.2, the optimization component has to be setup with the design variables, constraints and

objective function. For the I-beam optimization problem, as assumed before, the length of the beam is taken to be constant for a particular optimization problem with varying height, width and thickness which define the area of cross section of the beam. Hence design variables for the I-beam optimization problem are height, width and thickness. Von Mises stress and deflection of the beam are taken as constraints. The upper bound on von Mises stress is 7250lb/in$^2$ which has been obtained as the value of yield strength/ factor of safety. i.e., maximum von Mises stress = Yield strength/ factor of safety. For aluminum, the yield strength =1.0E07Pa = 14500psi. The factor of safety has been taken as 2. Hence, maximum von Mises stress= 14500/2 = 7250psi.

The upper bound on deflection has been taken as 1/100th of the length of the beam. Hence the maximum allowable deflection is taken to be 1/100th of the length of the beam. The objective of the optimization problem is to minimize the volume of the I-beam subjected to constraints by optimizing the design variables. The load being applied on the I-beam is 1000lb which produces deflection and stresses comparable to maximum von Mises stress and maximum allowable deflection. Depending on the problem taken, the maximum allowable von Mises stress has been taken so that the optimization problem does not converge to the minimum values of design variable for minimizing the volume, if the stress constraint is not violated.

If the stress and deflection do not have necessary stress and deflection upper bounds, the optimization problem just runs to minimize the volume by taking the lower limits of the design variables. Hence, maximum allowable stress for each problem has been taken based on the bounds of the design variables.

The method that has been used to run the optimization problem is the Hooke-Jeeves direct search technique. It is a direct search method well suited for linear and continuous design spaces. This is not a gradient based method. This technique begins with a starting guess and searches for a local minimum. This is often used when feasible design has not yet been determined. It uses combined penalty and objective value for optimization. This technique uses a combination of objective and constraints penalty as the objective function f(x). The algorithm examines points near the current point by perturbing design variables, one axis at a time, until an improved point is found. It then follows the favorable direction until no more design improvement is possible.

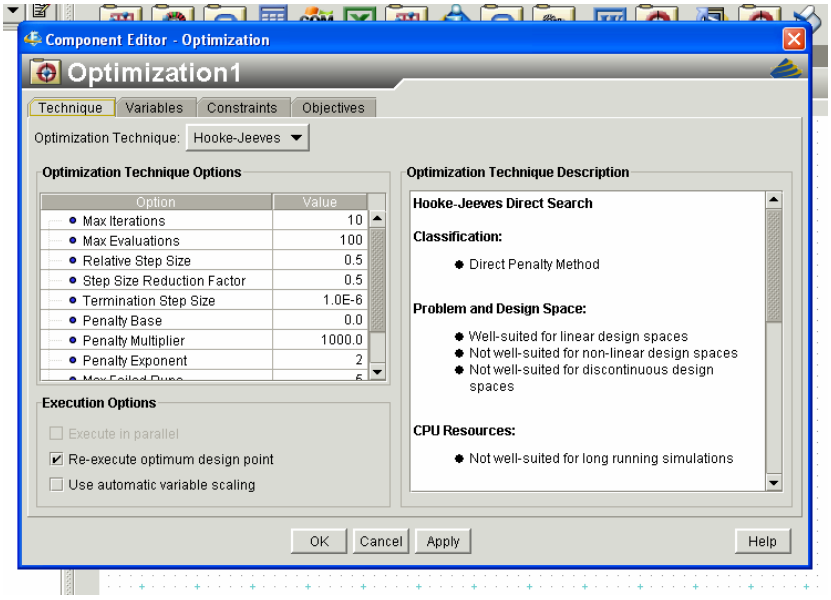



**Figure 31. Editor of the optimization component**

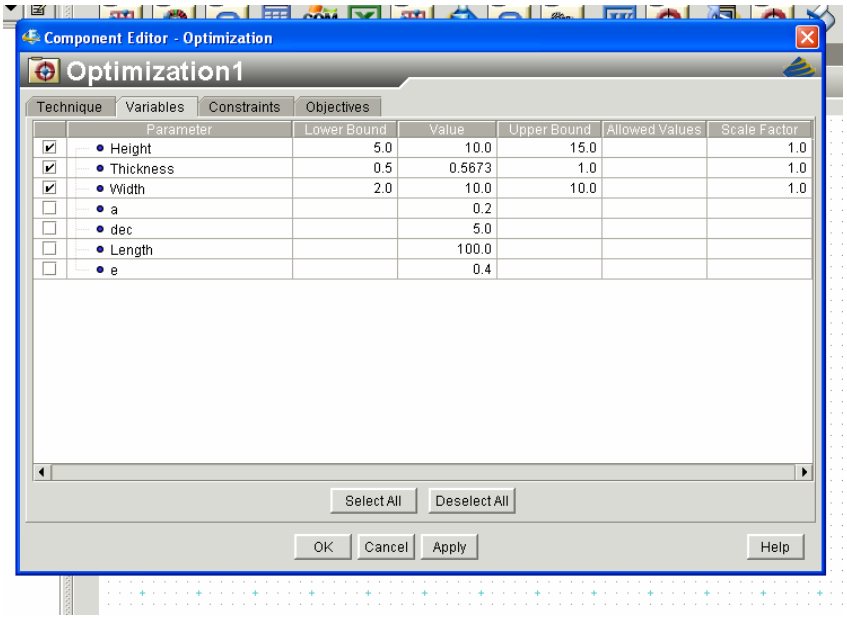



**Figure 32. Design variables for the optimization component**

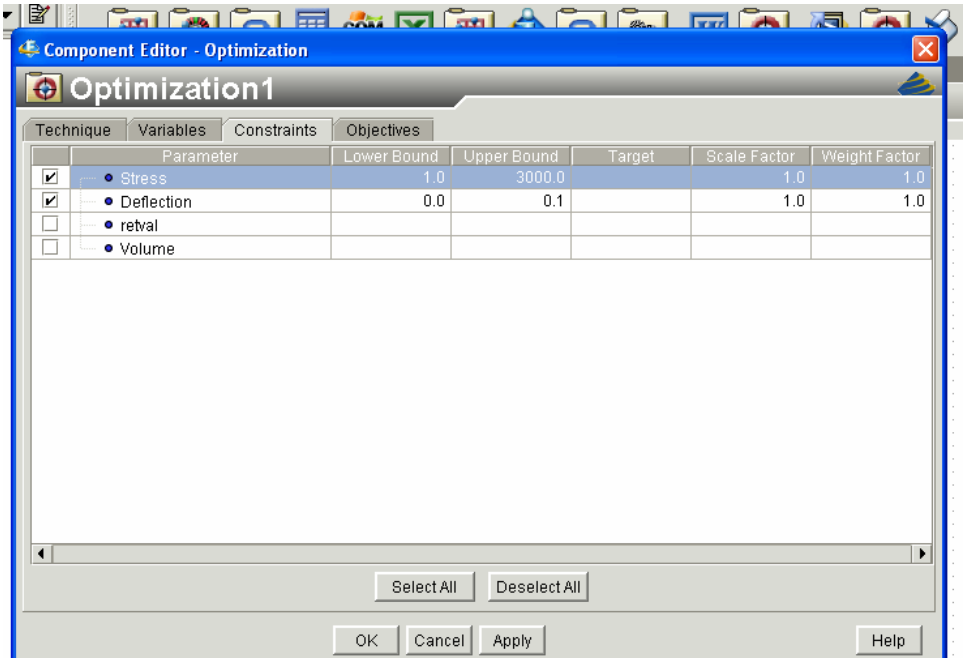



**Figure 33.Constraints for the optimization component**

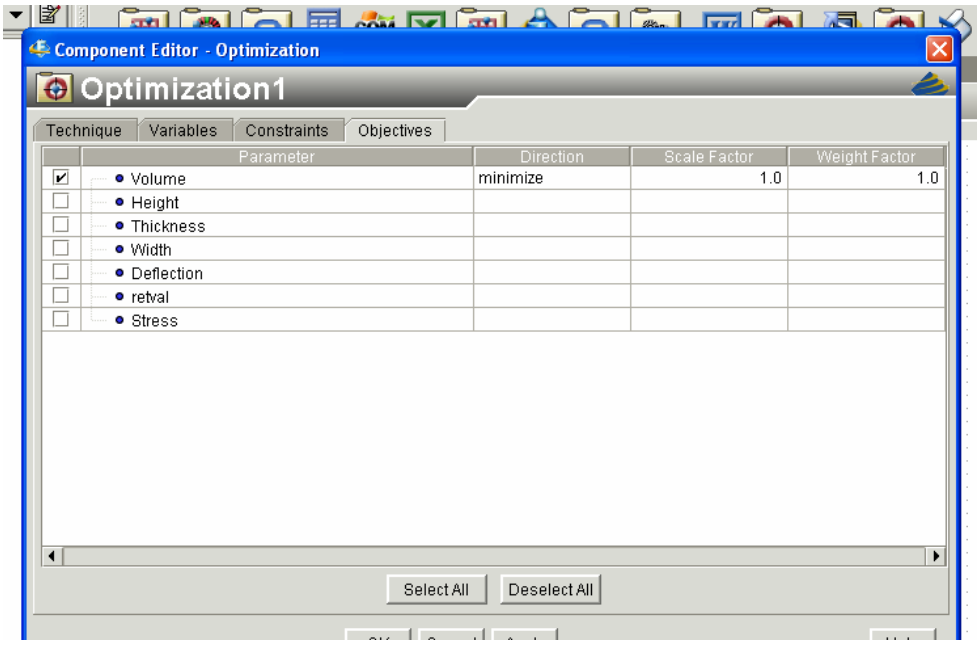



**Figure 34. Objective for the optimization component**

As shown in the Figure 31, Figure 32, Figure 33 and Figure 34, the optimization component considers all the input parameters as design parameters. It takes all the output parameters as constraints or objectives. Hence height, width and thickness are included as input parameters and considered as design parameters for optimization. The output parameters stress, deflection and volume have been taken as constraints and objective for the optimization problem. The input of the optimization component changes for a given design state. The output of the optimization problem, von Mises stress, deflection and volume have been obtained from the output parameters of the ANSYS SIMCODE component after the analysis run. These results will be used by the optimization component to determine the design variables for the next optimization design state.

### 8.2.3 Setting up the intelligent decision making tool in iSIGHT-FD

The Script component of iSIGHT-FD has been used to develop the intelligent decision making tool, “Intelligent EAM selector”, “INTEAM”. The input parameters for

the tool are length, height, width, thickness, accuracy and eccentricity of the load. The output of the tool is the decision about the beam element or shell element model to be used for the design optimization state. The workflow has been setup in a way that the decision from the script component makes the workflow run the ANSYS SIMCODE component of the beam element model or shell element model based on the design optimization state. A JAVA code has been written in the script component. This code contains the logic to shift between the beam and the shell element models. As explained in section 7.3 the tradeoff between beam and shell element models has been logically coded in the script component.

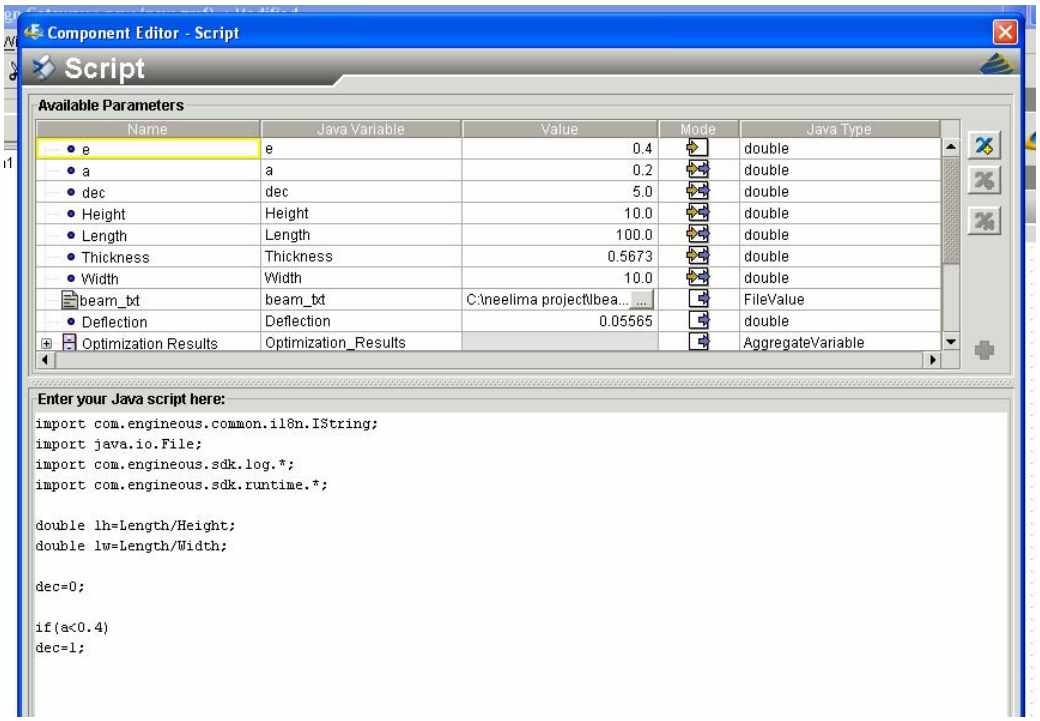



**Figure 35. Script component editor**

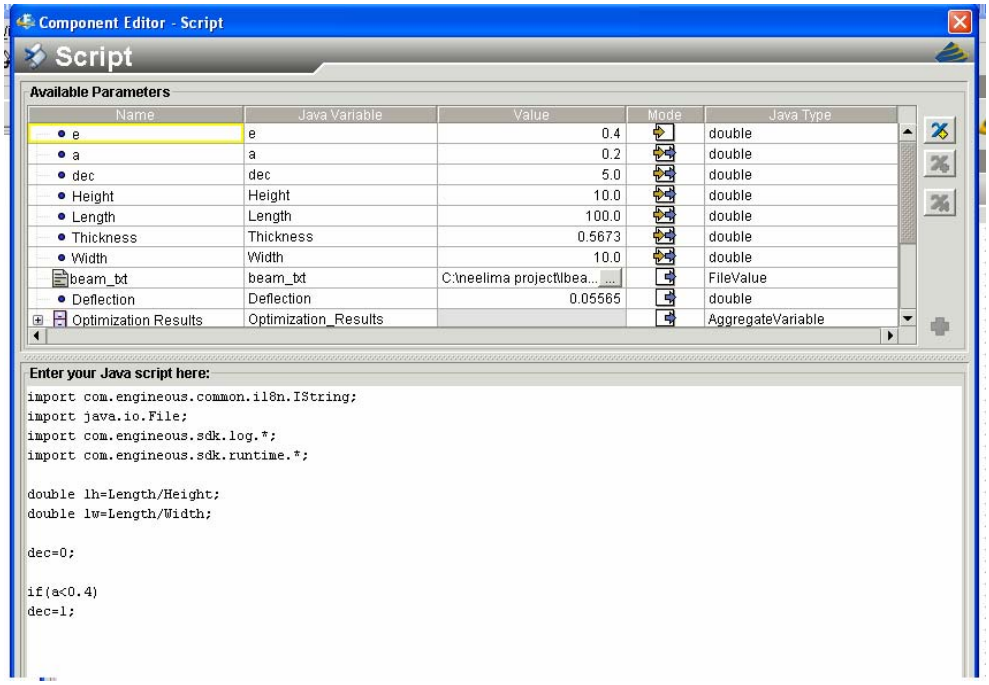



**Figure 36. Parameters for the "Intelligent EAM selector"component-Configuring the "Intelligent EAM selector" component for optimization of I-beam in iSIGHT-FD**

As shown in the Figure 35 and Figure 36 above, the "Intelligent EAM selector" component has a decision variable "dec", which determines the direction of the workflow. For the logic stated in section 7.3 for shifting between the beam and the shell element models, the "Intelligent EAM selector" component assigns values for the decision variable to trigger the ANSYS SIMCODE component of either the beam or the shell element model. The ANSYS SIMCODE component then runs the analysis for the beam or the shell element model and the output parameters from the ANSYS SIMCODE component will be passed to the optimization component for getting new values of design variables based on the results from the previous design state.

### 8.2.4. Setting up conditional workflows in iSIGHT-FD

Conditional workflows allow controlling a section of the workflow. An execute condition can be setup which allows parameter information to determine if the execution should continue or stop. The workflow can be setup to always execute, never execute and conditionally execute. Conditional execution can be controlled by selecting a parameter which decides the workflow.

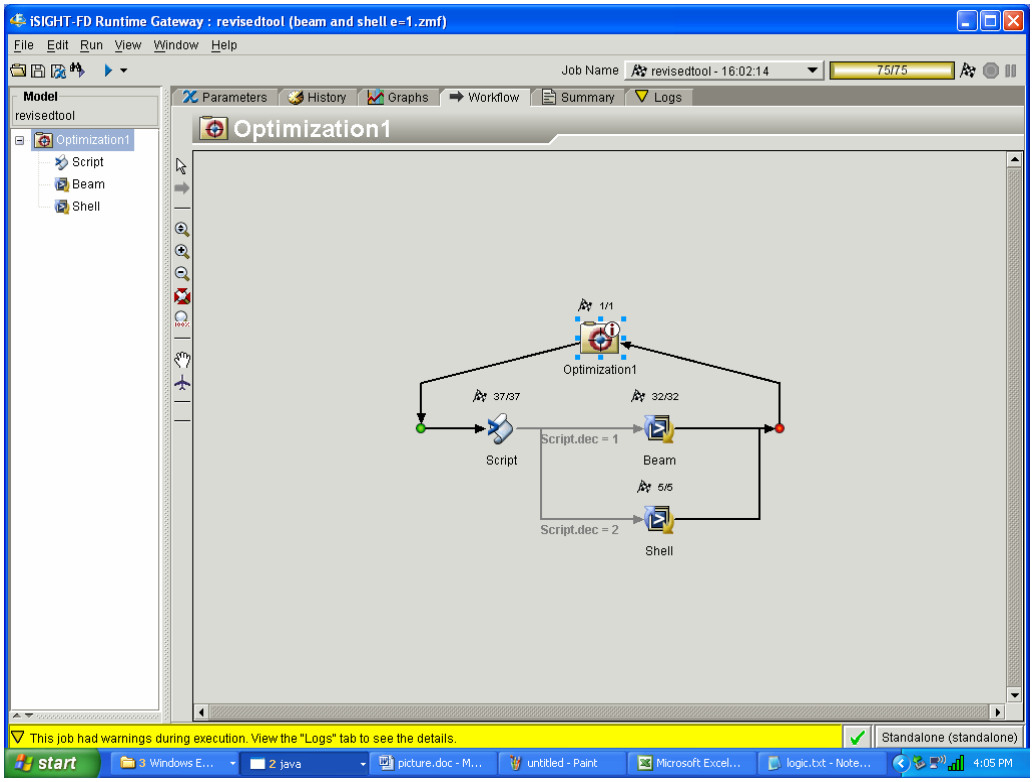



**Figure 37. Setting up conditional workflow in iSIGHT-FD**

Figure 37 shows the setup of conditional work flow using the value of a parameter. When the logic is satisfied the workflow follows the appropriate direction. Hence, the workflows can be customized by creating conditional workflows.

## 8.3 Complete integration of the tools

The optimization component for the I-beam has been setup such that for each design state, the intelligent decision making tool selects the right analysis model based on certain criteria and decides whether the beam or shell element model has to be executed.

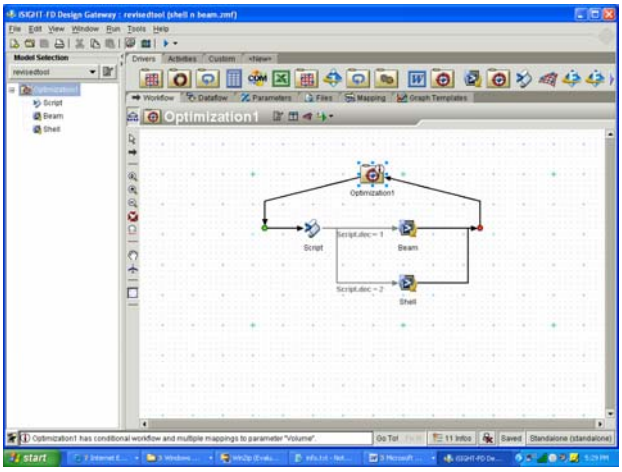



**Figure 38. Set up of the optimization process**

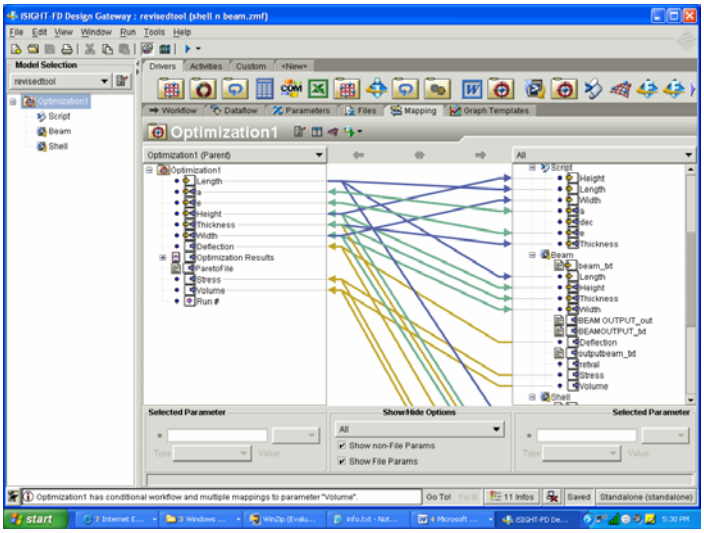



**Figure 39. Mapping of parameters-Setup of the optimization component using the ANSYS SIMCODE component and intelligent decision making tool**

Figure 38 and Figure 39 show the optimization component setup in iSIGHT-FD. Based on the decision taken by the "Intelligent EAM selector" component, the workflow continues either in the direction of the ANSYS SIMCODE shell element model or the beam element model. The results from one of the models that have been selected will be written back to the optimization component. The mapping of parameters in Figure 39 show that each time an optimization method decides the parameters for a design state, the values are passed to the intelligent tool, the "Intelligent EAM selector", "IntEAM" component, to select the right analysis model for the given design state based on certain criteria. The intelligent decision making tool then passes the decision variable which decides the direction of the workflow. The workflow has been setup in a way that a value of 1 for the decision variable makes the workflow follow the direction of analyzing a beam element model and a value of 2 for the decision variable makes the workflow follow the direction of analyzing a shell element model. An arbitrary value for the decision variable is taken initially, for example, a value of 5 has been taken and the intelligent decision making tool changes this value to 1 or 2 according to its decision about the right analysis model to be used for a given design state of the optimization process.

Details of complete optimization workflow setup are in Section 8.6.

## 8.4 Passing required analysis modeling knowledge for interoperating between analysis and optimization tools

The I-beam problem has been instantiated in ONTEAM. The instance knowledge contains idealizations, limitations, inputs, outputs, accuracy expectation, justification

knowledge etc. about the I-beam. Ontological knowledge is required to drive analysis and optimization tools in a product development environment. This supports reusability and interoperability of the knowledge. Also, automated tools can be developed with the ontological knowledge driving analysis and optimization tools. The INSPECT-IT method as explained in section 4 has been used to grab the input/output parameters from the knowledge base and make them as iSIGHT-FD parameters, i.e., making them "fiperized". These input/output parameters are stored in the EAM knowledge base in the form of an instance. Protégé, a knowledge base editor, can be used to open and share the knowledge with other components. Since Protégé is an open source tool, it can be used to develop methods which operate on the knowledge present in the ontology. For formulating and executing an optimization problem in a product development environment, the input/output parameters from the knowledge base can be used to run the optimization/analysis tools.

For the I-beam example, the length, height, width, thickness, accuracy expectation and the eccentricity of the load are the input parameters for the optimization problem. The length, height, width, thickness and eccentricity of the load are present as values of input parameters for the I-Beam instance in the EAM ontology. Accuracy expectation is a property of the I-Beam model which represents the overall expected accuracy of the model. These parameters can be made iSIGHT-FD parameters by writing a method which can grab required information from the ontology and make the input/output parameters as iSIGHT-FD parameters. Figure 40, Figure 41, Figure 42 and Figure 43 shows the passing of ontological modeling knowledge to iSIGHT-FD, using the "INSPECT-IT" method.

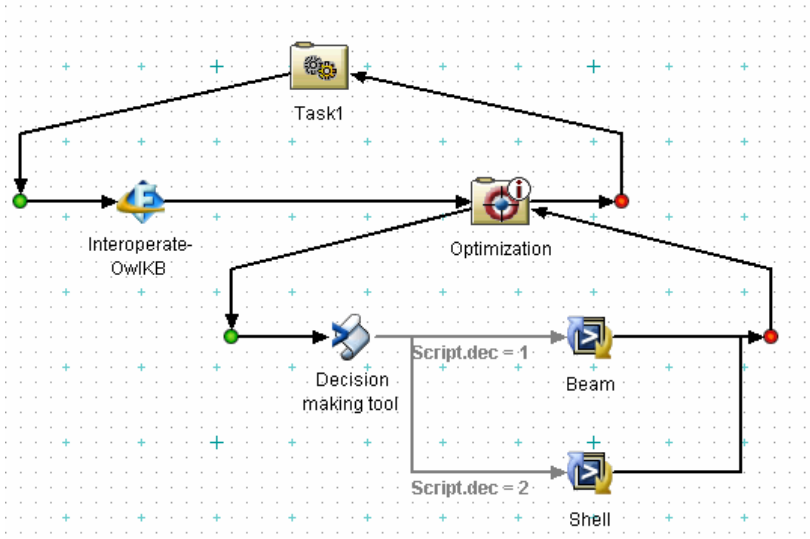



**Figure 40. Integration of components**

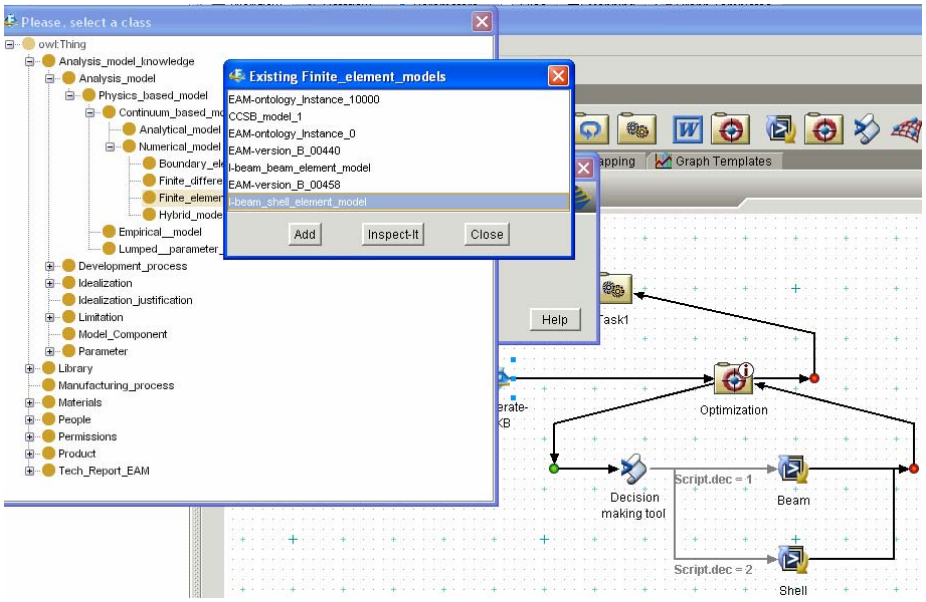



**Figure 41. Accessing the I-beam modeling instance- Passing ontological modeling knowledge as parameters in a product development environment**

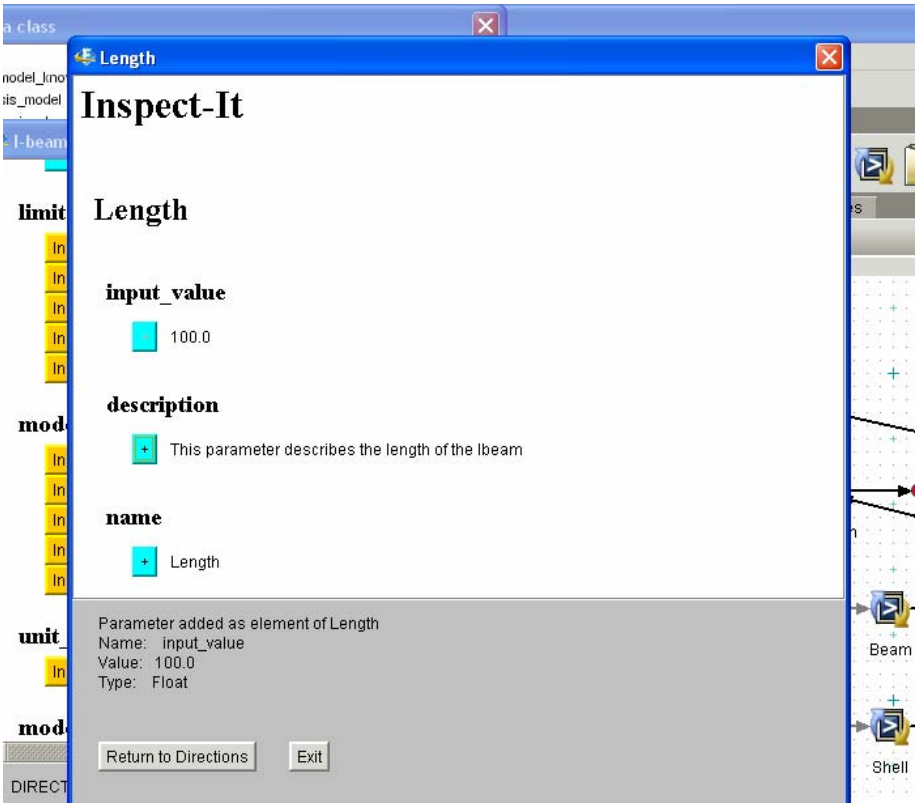



**Figure 42. Selecting parameters**

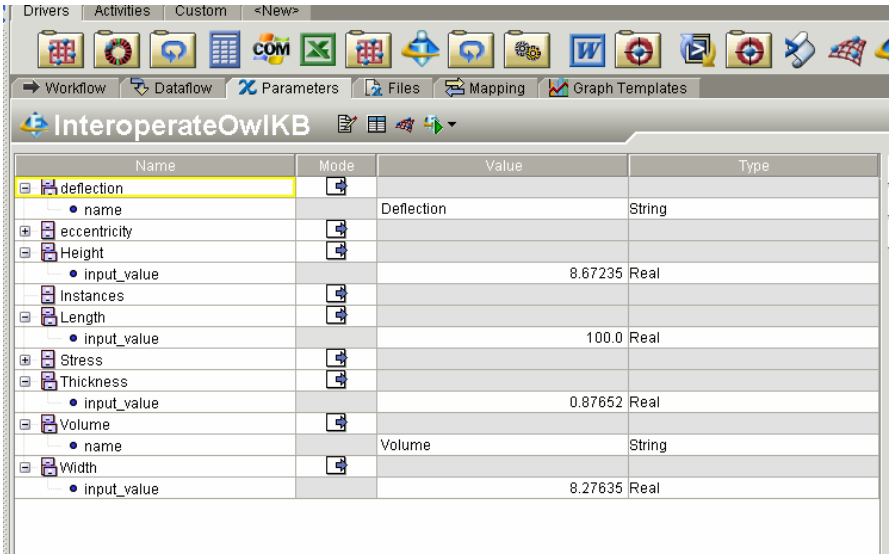



**Figure 43. Input and output parameters from the knowledge base**

These parameters are passed on to the optimization component and are used to run the optimization process using the ANSYS SIMCODE and decision making tool components.

## 8.5. Integrating I-beam modeling knowledge to a CAD-integrated environment, ENCAPTA

EnCapta provides engineers with the capability for building specialized, CAD-integrated design environments that enrich the CAD model as a source of information. The advantage with such integrated systems is that changes in engineering models, material specifications etc. that are stored separately can be captured and managed as objects and linked to the relevant geometry in the CAD models. The XML tools associated with such integrated systems extract design information from the CAD models and report to other systems or databases throughout the enterprise. Hence distributed and collaborative data sharing of CAD models with less effort for updating the changed design revisions can be achieved using CAD integrated systems. The I-beam instance knowledge can be passed to the CAD integrated environment, EnCapta and the design parameters can be linked directly to the CAD model parameters.

Methods have been developed to import EAM ontology into Vistagy's EnCapta framework. Using XSLT, OWL ontology has been translated to the native EnCapta application language.

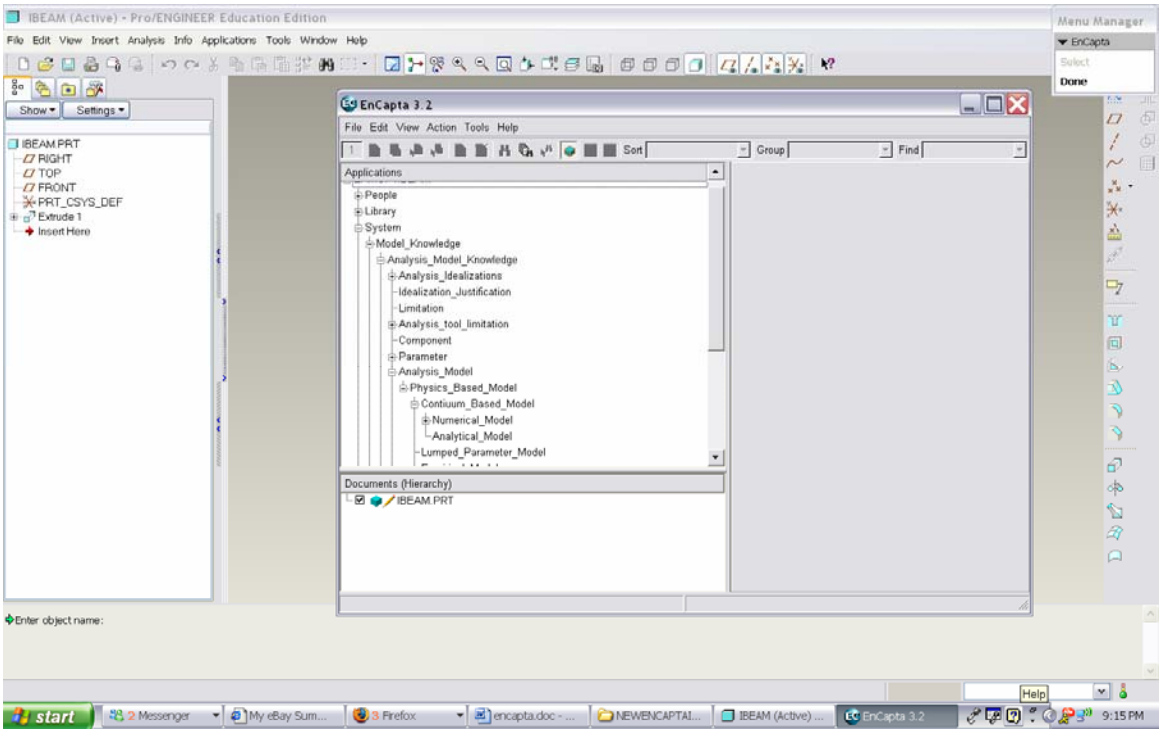



**Figure 44. Version of EAM ontology within the EnCapta framework**

Figure 44 shows the version of the EAM ontology within the EnCapta framework. This method allows the complete EAM ontology to be accessed and navigated within an integrated framework when developing various CAD or Analysis models.

To demonstrate the capabilities and advantages of the EAM ontology within EnCapta, an I-Beam CAD model has been developed within PTC's [PTC, 2005] Pro/ENGINEER. Pro/ENGINEER is one of the multiple CAD development tools EnCapta is able to communicate with. Figure 45 shows the creation of a geometric model while modeling an I-Beam.

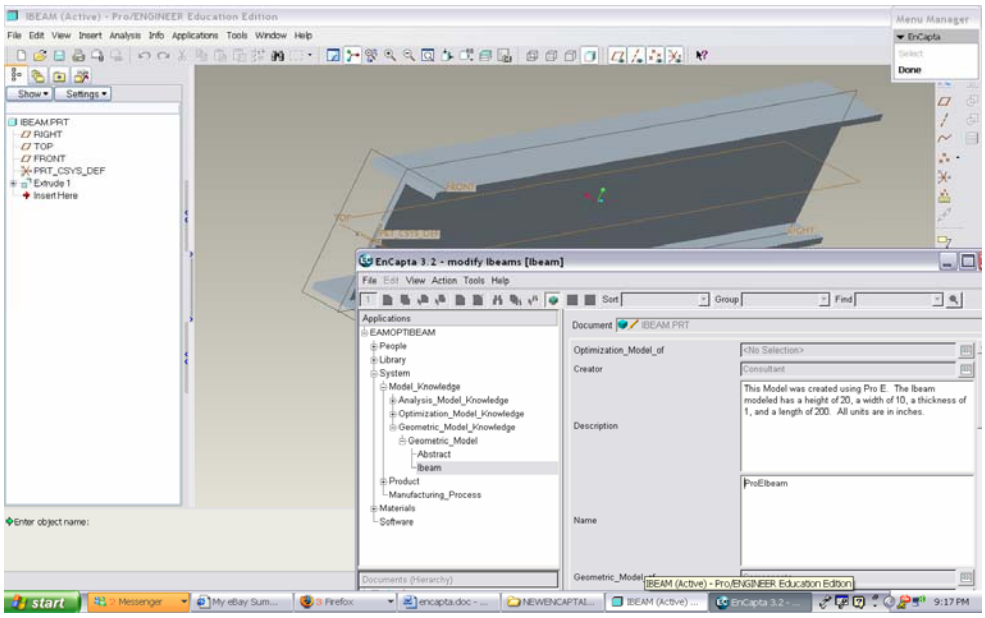



**Figure 45. Creation of a geometrical model for I-beam**

One of the unique capabilities provided by EnCapta is the ability to link actual CAD geometry with instantiated knowledge.

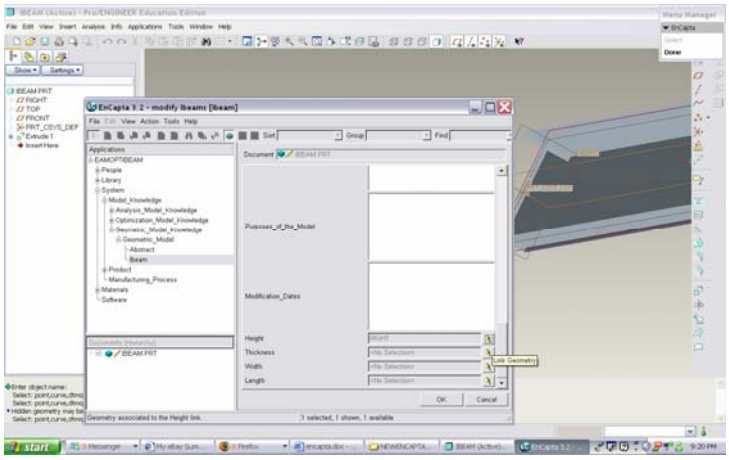


**Figure 46. Adding geometry links from CAD model to EAM ontology**

This ability enables a user to link EAM ontology parameters to geometry parameters within a CAD model. Figure 45 and Figure 46 show how geometry links have been added to the EAM ontology, and the knowledge captured by these links represent

the I-Beam parameters width, length, height, and thickness. Once the instantiation of knowledge within EnCapta has been completed, EnCapta provides the ability to store this knowledge within the actual CAD model.

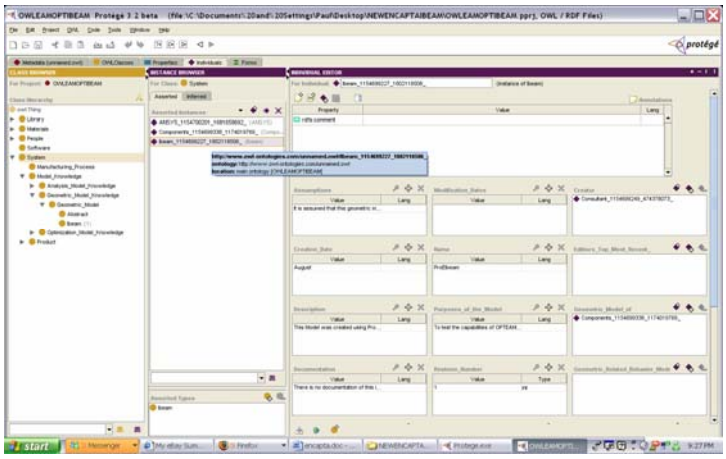


**Figure 47. Encapta knowledge represented in Protégé OWL**

EnCapta also allows this knowledge to be exported as an XML file using the native EnCapta language. Using this advantage, methods have been developed for taking the knowledge and converting it from native EnCapta to the OWL format. This then allows the ability to import any knowledge captured using the EnCapta tool back into an OWL knowledge base. Figure 47 shows the EnCapta knowledge as represented in Protégé OWL. Once the knowledge has been converted to the OWL format, it then becomes accessible to the OWL-enabled iSIGHT-FD methods.

## 8.6. Overall setup of the optimization process

A flow chart which describes the over all setup of the optimization process is shown in Figure 48.

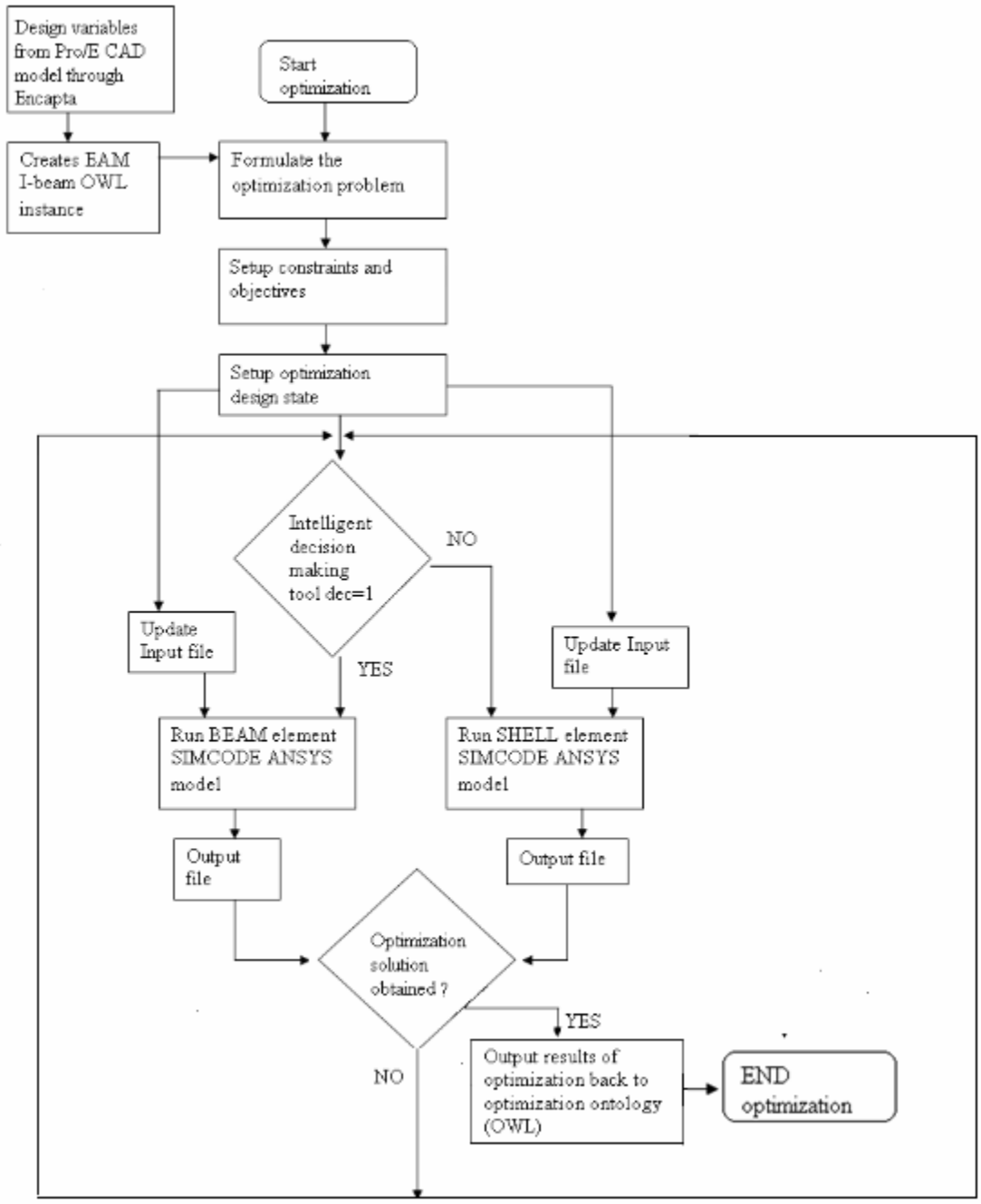



**Figure 48. Flowchart for optimization process of I-beam using optimization component, ANSYS SIMCODE component and intelligent decision making tool**

The design parameters that the optimization technique updates for each run also get updated for the SIMCODE components. The design parameters from the Pro/E CAD model are read into the optimization component, the updates design parameters are then passed to the input files which run ANSYS in batch mode and write the corresponding output to the output file which update the values of outputs in the optimization component. Hence the mapping of parameters shown in Figure 48 takes the values of input parameters from the optimization component, passes to the intelligent decision making tool which decides the direction of workflow and pass the parameters to the appropriate beam or shell element SIMCODE component. These ANSYS SIMCODE components in turn return the results to the output file which gets read as output parameters from the SIMCODE component to the optimization component.

Hence different design states of optimization process are formed by the input parameter values updated by the technique used in the optimization process and the corresponding output values taken from the results of the ANSYS SIMCODE component. The selection of analysis model that has to be run, the beam or the shell element model, is taken by the "Intelligent EAM selector" component based on the criteria and the design state input.

# CHAPTER 9

## RESULTS

The optimization process for the I-beam problem has been setup to run for different values of length, height, width, thickness, accuracy expectation and eccentricity. As explained in the section 7.1, the design variables of the optimization problem are height, width and thickness. The length of the beam is kept constant for a given optimization problem. The constraints are the maximum von Mises stress and the deflection of the I-beam. The objective of the optimization process is to minimize the volume of the beam. Six sample problems have been run using the optimization process and the results have been formulated. A low accuracy model has an accuracy expectation of less than 50%. A medium accuracy model has an accuracy expectation of 50%-80% and a high accuracy model has an accuracy expectation of more than 80%. One problem with low accuracy, one with high accuracy and four problems with medium accuracy have been run to demonstrate the power of automated optimization process. The results have been validated using a highly accurate shell model. The Hooke-Jeeves optimization technique has been used to run the optimization problem. The maximum number of allowable optimization iterations has been taken as 101.

**Sample Problem 1:**

Objective:

Minimize V=lt(2(w-t) +h))

Subject to:

$$(\sigma_V)_{max} \leq 6000\text{psi}$$

$$\delta_{max} \leq 0.5 \text{inch}$$

$$20 \leq l/h \leq 6.66$$

$$200 \leq l/t \leq 100$$

$$20 \leq l/w \leq 6.66$$

With l=100in, Accuracy expectation= 0.7 and eccentricity ratio=0.4

Results:

The model ran for 96 different optimization design states. The optimized value of minimum volume is 1592 $in^3$. Values of design variables at the optimum solution are height =10in, thickness =0.547in and width = 10in. The optimization process ran for 64 design states using the beam element model and 32 times using the shell element model. Table 7 shows the history of the design parameters during the optimization run. Dimensions of the I-beam and deflection are in inches, Stress in $lb/in^2$ and Volume in $in^3$.

As seen, when the height of the model during second design state is equal 15in, then the model shifts to shell element model as the length/height ratio is less than 10 which means that, from Table 4, the shell element model is more accurate for an accuracy expectation of 0.7 and eccentricity of 0.4. For 2nd, 6th, 9th and 12th design states the length/height or the length/width ratios are lesser than 10 and hence the shell model is accurate for an eccentricity of 0.4 and an accuracy expectation of 0.7. Else, for length/height and length/width ratios greater than 10, the beam model is as accurate as the model for the given accuracy expectation and hence can be used according to Table 4 and Table 5.

**Table 7.Optimization results for example problem 1.**

| Height | Length | Thickness | Width | Deflection | Stress | Volume | decision |
|---|---|---|---|---|---|---|---|
| 10 | 100 | 0.52736 | 10 | 0.012988 | 1854.192 | 1551.458 | beam |
| 15 | 100 | 0.52736 | 10 | 0.099389 | 4834.891 | 1817.949 | shell |
| 5 | 100 | 0.52736 | 10 | 0.061175 | 4466.481 | 1287.778 | beam |
| 10 | 100 | 0.79104 | 10 | 0.009255 | 1324.626 | 2272.971 | beam |
| 10 | 100 | 0.79104 | 15 | 0.092181 | 2091.986 | 3101.586 | shell |
| 10 | 100 | 0.79104 | 5 | 0.016428 | 2398.165 | 1481.931 | beam |
| 5 | 100 | 1 | 10 | 0.040437 | 2964.172 | 2325 | beam |
| 15 | 100 | 1 | 10 | 0.040065 | 1216.33 | 3400 | shell |
| 10 | 100 | 0.73632 | 10 | 0.009806 | 1402.742 | 2125.527 | beam |
| 10 | 100 | 0.73632 | 5 | 0.017362 | 2533.667 | 1389.207 | beam |
| 10 | 100 | 0.73632 | 15 | 0.101018 | 2432.767 | 2891.063 | shell |
| 15 | 100 | 0.79104 | 10 | 0.054816 | 2027.419 | 2706.066 | shell |
| 5 | 100 | 0.79104 | 10 | 0.046233 | 3383.12 | 1877.451 | beam |
| 10 | 100 | 0.52736 | 10 | 0.012988 | 1854.192 | 1551.458 | beam |
| 10 | 100 | 1 | 10 | 0.00772 | 1107.165 | 2825 | beam |
| 10 | 100 | 0.79104 | 15 | 0.092181 | 2091.986 | 3101.586 | shell |
| 10 | 100 | 0.79104 | 5 | 0.016428 | 2398.165 | 1481.931 | beam |
| 7.5 | 100 | 0.79104 | 10 | 0.017849 | 1938.142 | 2075.211 | beam |
| 12.5 | 100 | 0.79104 | 10 | 0.07445 | 2023.254 | 2508.306 | shell |
| 10 | 100 | 0.92288 | 10 | 0.008203 | 1175.513 | 2623.299 | beam |
| 10 | 100 | 0.6592 | 10 | 0.010741 | 1535.45 | 1915.691 | beam |
| 10 | 100 | 0.6592 | 7.5 | 0.013667 | 1974.258 | 1586.091 | beam |
| 10 | 100 | 0.6592 | 12.5 | 0.12588 | 3045.133 | 2263.746 | shell |
| 7.5 | 100 | 0.52736 | 10 | 0.024566 | 2661.182 | 1419.618 | beam |
| 12.5 | 100 | 0.52736 | 10 | 0.129355 | 4823.499 | 1686.109 | shell |
| 10 | 100 | 0.5 | 10 | 0.013605 | 1941.723 | 1475 | beam |
| 10 | 100 | 0.6592 | 10 | 0.010741 | 1535.45 | 1915.691 | beam |
| 10 | 100 | 0.52736 | 7.5 | 0.016488 | 2379.272 | 1287.778 | beam |
| 10 | 100 | 0.52736 | 12.5 | 0.16948 | 4865.78 | 1817.949 | shell |
| 11.25 | 100 | 0.6592 | 10 | 0.113154 | 3000.797 | 2016.546 | shell |
| 8.75 | 100 | 0.6592 | 10 | 0.014489 | 1822.989 | 1833.291 | beam |
| 10 | 100 | 0.72512 | 10 | 0.009929 | 1420.223 | 2095.2 | beam |
| 10 | 100 | 0.59328 | 10 | 0.011738 | 1676.839 | 1734.444 | beam |
| 10 | 100 | 0.59328 | 11.25 | 0.151055 | 3790.659 | 1892.962 | shell |
| 10 | 100 | 0.59328 | 8.75 | 0.013129 | 1885.655 | 1586.124 | beam |
| 8.75 | 100 | 0.52736 | 10 | 0.017447 | 2192.4 | 1485.538 | beam |
| 11.25 | 100 | 0.52736 | 10 | 0.152637 | 4828.047 | 1620.189 | shell |
| 10 | 100 | 0.5 | 10 | 0.013605 | 1941.723 | 1475 | beam |
| 10 | 100 | 0.59328 | 10 | 0.011738 | 1676.839 | 1734.444 | beam |
| 10 | 100 | 0.52736 | 8.75 | 0.01452 | 2084.169 | 1419.618 | beam |
| 10 | 100 | 0.52736 | 11.25 | 0.176595 | 4858.397 | 1686.109 | shell |
| 10.625 | 100 | 0.59328 | 10 | 0.143246 | 3761.535 | 1781.722 | shell |
| 9.375 | 100 | 0.59328 | 10 | 0.013549 | 1820.019 | 1697.364 | beam |
| 10 | 100 | 0.62624 | 10 | 0.011213 | 1602.358 | 1825.285 | beam |
| 10 | 100 | 0.56032 | 10 | 0.012326 | 1760.227 | 1643.168 | beam |
| 10 | 100 | 0.56032 | 10.625 | 0.166829 | 4263.441 | 1719.604 | shell |
| 10 | 100 | 0.56032 | 9.375 | 0.013011 | 1863.209 | 1573.128 | beam |
| 9.375 | 100 | 0.52736 | 10 | 0.014978 | 2010.592 | 1518.498 | beam |
| 10.625 | 100 | 0.52736 | 10 | 0.167568 | 4830.683 | 1587.229 | shell |
| 10 | 100 | 0.5 | 10 | 0.013605 | 1941.723 | 1475 | beam |
| 10 | 100 | 0.56032 | 10 | 0.012326 | 1760.227 | 1643.168 | beam |
| 10 | 100 | 0.52736 | 9.375 | 0.013709 | 1962.466 | 1485.538 | beam |
| 10 | 100 | 0.52736 | 10.625 | 0.180773 | 4846.305 | 1620.189 | shell |
| 10.3125 | 100 | 0.56032 | 10 | 0.16246 | 4239.205 | 1667.074 | shell |
| 9.6875 | 100 | 0.56032 | 10 | 0.013224 | 1832.195 | 1625.658 | beam |
| 10 | 100 | 0.5768 | 10 | 0.012023 | 1717.324 | 1688.86 | beam |
| 10 | 100 | 0.54384 | 10 | 0.012647 | 1805.767 | 1597.368 | beam |
| 10 | 100 | 0.54384 | 10.3125 | 0.175737 | 4529.099 | 1635.934 | shell |
| 10 | 100 | 0.54384 | 9.6875 | 0.012988 | 1857.042 | 1563.378 | beam |
| 9.6875 | 100 | 0.52736 | 10 | 0.013931 | 1929.556 | 1534.978 | beam |
| 10.3125 | 100 | 0.52736 | 10 | 0.176068 | 4821.333 | 1570.749 | shell |
| 10 | 100 | 0.51088 | 10 | 0.013351 | 1905.779 | 1505.44 | beam |
| 10 | 100 | 0.54384 | 10 | 0.012647 | 1805.767 | 1597.368 | beam |
| 10 | 100 | 0.52736 | 9.6875 | 0.013338 | 1906.793 | 1518.498 | beam |
| 10 | 100 | 0.52736 | 10.3125 | 0.183039 | 4833.674 | 1587.229 | shell |
| 10.15625 | 100 | 0.54384 | 10 | 0.173426 | 4517.504 | 1610.441 | shell |
| 9.84375 | 100 | 0.54384 | 10 | 0.013095 | 1841.916 | 1588.87 | beam |
| 10 | 100 | 0.55208 | 10 | 0.012484 | 1782.653 | 1620.282 | beam |
| 10 | 100 | 0.5356 | 10 | 0.012814 | 1829.602 | 1574.427 | beam |
| 10 | 100 | 0.54384 | 10.15625 | 0.176888 | 4525.359 | 1618.939 | shell |
| 10 | 100 | 0.54384 | 9.84375 | 0.012815 | 1831.046 | 1580.373 | beam |
| 9.921875 | 100 | 0.54384 | 10 | 0.012868 | 1823.681 | 1593.119 | beam |
| 10.07813 | 100 | 0.54384 | 10 | 0.175721 | 4517.877 | 1606.193 | shell |
| 10 | 100 | 0.53972 | 10 | 0.01273 | 1817.593 | 1585.9 | beam |
| 10 | 100 | 0.54796 | 10 | 0.012564 | 1794.122 | 1608.828 | beam |
| 10 | 100 | 0.54384 | 9.921875 | 0.01273 | 1818.319 | 1588.87 | beam |
| 10 | 100 | 0.54384 | 10.07813 | 0.177476 | 4523.228 | 1610.441 | shell |
| 10.03906 | 100 | 0.54384 | 10 | 0.176889 | 4518.064 | 1604.068 | shell |
| 9.960938 | 100 | 0.54384 | 10 | 0.012756 | 1814.685 | 1595.243 | beam |
| 10 | 100 | 0.5459 | 10 | 0.012605 | 1799.922 | 1603.099 | beam |
| 10 | 100 | 0.54178 | 10 | 0.012688 | 1811.657 | 1591.635 | beam |
| 10 | 100 | 0.54178 | 10.03906 | 0.178661 | 4557.302 | 1600.22 | shell |
| 10 | 100 | 0.54178 | 9.960938 | 0.01273 | 1817.931 | 1587.402 | beam |
| 9.960938 | 100 | 0.53972 | 10 | 0.01284 | 1826.562 | 1583.792 | beam |
| 10.03906 | 100 | 0.53972 | 10 | 0.178666 | 4591.546 | 1592.139 | shell |
| 10 | 100 | 0.53766 | 10 | 0.012772 | 1823.574 | 1580.164 | beam |
| 10 | 100 | 0.54178 | 10 | 0.012688 | 1811.657 | 1591.635 | beam |
| 10 | 100 | 0.53972 | 9.960938 | 0.012772 | 1823.886 | 1581.684 | beam |
| 10 | 100 | 0.53972 | 10.03906 | 0.179557 | 4594.278 | 1594.247 | shell |
| 10.01953 | 100 | 0.54178 | 10 | 0.178365 | 4554.684 | 1597.046 | shell |
| 9.980469 | 100 | 0.54178 | 10 | 0.012743 | 1816.119 | 1590.577 | beam |
| 10 | 100 | 0.54281 | 10 | 0.012667 | 1808.707 | 1594.501 | beam |
| 10 | 100 | 0.54075 | 10 | 0.012709 | 1814.619 | 1588.768 | beam |
| 10 | 100 | 0.54178 | 10.01953 | 0.17881 | 4556.044 | 1598.104 | shell |
| 10 | 100 | 0.54178 | 9.980469 | 0.012709 | 1814.789 | 1589.519 | beam |
| 10 | 100 | 0.54178 | 10 | 0.012688 | 1811.657 | 1591.635 | beam |

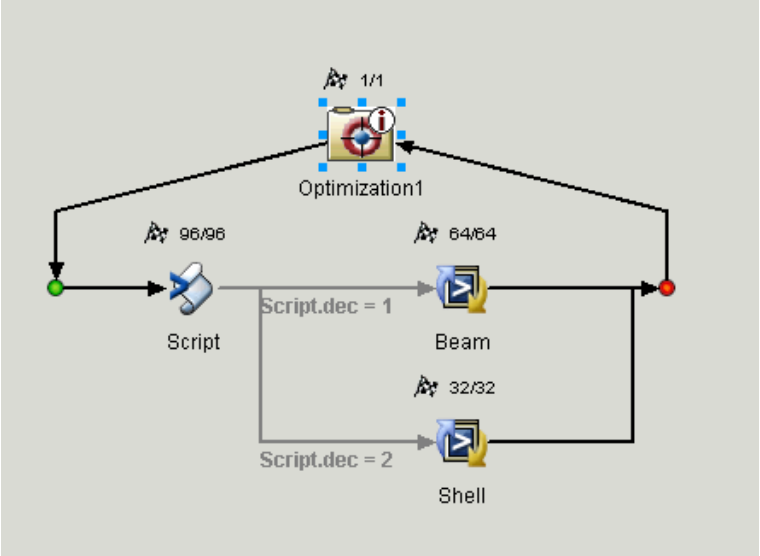



**Figure 49. Optimization process for I-beam switching between the shell and the beam element models based on the intelligent tool decision**

Figure 49 shows the switching of the analysis models for a given optimization state. The beam element analysis model analyzed for 64 times and the shell element analysis model analyzed for 32 times based on the design state of the optimization process. As explained above the length/width and length/height ratios define the logic for switching between the analysis models. Hence the optimization problem switches between the beam and the shell element models based on the results tabulated in Table 4 and Table 5.

**Sample Problem 2:**

Objective:

Minimize V=lt(2(w-t) +h))

Subject to:

$$(\sigma_V)_{max} \leq 3624 psi$$

$$\delta_{max} \leq 0.5 \text{ inch}$$

$$12.5 \le l/h \le 8.33$$

$$200 \le l/t \le 142.85$$

$$12.5 \le l/w \le 8.33$$

with l=100in, Accuracy expectation= 0.7 and eccentricity ratio =0.6

Results:

The model ran for 76 different optimization design states. The beam element model ran for 57 times and the shell element model ran for 19 times.

Minimum volume= 1709.9 $in^3$

Design variables at the optimum solution:

Height=10, thickness =0.684 and width = 8.

As seen from Table 8, the decision to switch between models depends on length, height and width ratios determined by the optimization technique for a given design state. Dimensions of the I-beam and deflection are in inches, Stress in $lb/in^2$ and volume in $in^3$. When the length/width and the length/height ratios are not greater than 10, the shell element has been used for an accuracy of 0.7, in accordance to the results in Table 4 and Table 5. The maximum allowable von Mises stress has been calculated based on the limits of the design variables taken.

**Table 8.Optimization results for sample problem 2**

| Height | Length | Thickness | Width | Deflection | Stress | Volume | decision |
|---|---|---|---|---|---|---|---|
| 10 | 100 | 0.52736 | 10 | 0.012988 | 1854.192 | 1551.458 | beam |
| 12 | 100 | 0.52736 | 10 | 0.137777 | 4825.193 | 1659.741 | shell |
| 8 | 100 | 0.52736 | 10 | 0.021282 | 2453.581 | 1445.986 | beam |
| 10 | 100 | 0.7 | 10 | 0.01022 | 1461.517 | 2027 | beam |
| 10 | 100 | 0.7 | 12 | 0.11876 | 2674.404 | 2331 | shell |
| 10 | 100 | 0.7 | 8 | 0.012333 | 1778.433 | 1747 | beam |
| 8 | 100 | 0.7 | 8 | 0.020564 | 2391.185 | 1607 | beam |
| 12 | 100 | 0.7 | 8 | 0.102821 | 2566.111 | 1911 | shell |
| 10 | 100 | 0.5 | 8 | 0.016374 | 2357.286 | 1275 | beam |
| 10 | 100 | 0.7 | 12 | 0.11876 | 2674.404 | 2331 | shell |
| 12 | 100 | 0.7 | 8 | 0.102821 | 2566.111 | 1911 | shell |
| 8 | 100 | 0.7 | 8 | 0.020564 | 2391.185 | 1607 | beam |
| 10 | 100 | 0.56816 | 8 | 0.014672 | 2113.465 | 1437.655 | beam |
| 10 | 100 | 0.7 | 10.5 | 0.126454 | 2649.121 | 2121 | shell |
| 8.75 | 100 | 0.7 | 8 | 0.016739 | 2122.727 | 1659.5 | beam |
| 11.25 | 100 | 0.7 | 8 | 0.115803 | 2562.298 | 1858.5 | shell |
| 10 | 100 | 0.63408 | 8 | 0.013379 | 1928.196 | 1593.197 | beam |
| 10 | 100 | 0.7 | 9.25 | 0.010917 | 1566.177 | 1922 | beam |
| 10.625 | 100 | 0.7 | 8 | 0.128998 | 2564.444 | 1814.75 | shell |
| 9.375 | 100 | 0.7 | 8 | 0.014293 | 1937.164 | 1703.25 | beam |
| 10 | 100 | 0.66704 | 8 | 0.012829 | 1849.533 | 1670.316 | beam |
| 10 | 100 | 0.7 | 8.625 | 0.01158 | 1665.57 | 1834.5 | beam |
| 9.6875 | 100 | 0.7 | 8 | 0.013261 | 1854.815 | 1725.125 | beam |
| 10.3125 | 100 | 0.7 | 8 | 0.136594 | 2565.377 | 1792.875 | shell |
| 10 | 100 | 0.68352 | 8 | 0.012575 | 1813.106 | 1708.712 | beam |
| 10 | 100 | 0.7 | 8.3125 | 0.011944 | 1720.152 | 1790.75 | beam |
| 10.15625 | 100 | 0.7 | 8 | 0.140663 | 2558.162 | 1781.938 | shell |
| 9.84375 | 100 | 0.7 | 8 | 0.012785 | 1815.918 | 1736.063 | beam |
| 10 | 100 | 0.69176 | 8 | 0.012452 | 1795.558 | 1727.87 | beam |
| 10 | 100 | 0.69176 | 8.15625 | 0.012253 | 1765.663 | 1749.487 | beam |
| 9.84375 | 100 | 0.68352 | 8 | 0.013035 | 1851.215 | 1698.032 | beam |
| 10.15625 | 100 | 0.68352 | 8 | 0.144614 | 2694.43 | 1741.112 | shell |
| 10 | 100 | 0.67528 | 8 | 0.012701 | 1831.093 | 1689.527 | beam |
| 10 | 100 | 0.69176 | 8 | 0.012452 | 1795.558 | 1727.87 | beam |
| 10 | 100 | 0.68352 | 8.15625 | 0.012374 | 1782.935 | 1730.072 | beam |
| 10.07813 | 100 | 0.69176 | 8 | 0.144751 | 2625.266 | 1756.127 | shell |
| 9.921875 | 100 | 0.69176 | 8 | 0.012677 | 1814.282 | 1722.465 | beam |
| 10 | 100 | 0.69588 | 8 | 0.012392 | 1786.944 | 1737.438 | beam |
| 10 | 100 | 0.68764 | 8 | 0.012513 | 1804.279 | 1718.294 | beam |
| 10 | 100 | 0.68764 | 8.078125 | 0.012412 | 1789.136 | 1729.039 | beam |
| 9.921875 | 100 | 0.68352 | 8 | 0.012802 | 1831.987 | 1703.372 | beam |
| 10.07813 | 100 | 0.68352 | 8 | 0.14678 | 2694.689 | 1735.772 | shell |
| 10 | 100 | 0.6794 | 8 | 0.012637 | 1822.044 | 1699.123 | beam |
| 10 | 100 | 0.68764 | 8 | 0.012513 | 1804.279 | 1718.294 | beam |
| 10 | 100 | 0.68352 | 8.078125 | 0.012473 | 1797.894 | 1719.392 | beam |
| 10.03906 | 100 | 0.68764 | 8 | 0.146855 | 2659.774 | 1743.265 | shell |
| 9.960938 | 100 | 0.68764 | 8 | 0.012626 | 1813.636 | 1715.608 | beam |
| 10 | 100 | 0.6897 | 8 | 0.012483 | 1799.905 | 1723.083 | beam |
| 10 | 100 | 0.68558 | 8 | 0.012544 | 1808.679 | 1713.504 | beam |
| 10 | 100 | 0.68558 | 8.039063 | 0.012493 | 1801.058 | 1718.86 | beam |
| 9.960938 | 100 | 0.68352 | 8 | 0.012688 | 1822.504 | 1706.042 | beam |
| 10.03906 | 100 | 0.68352 | 8 | 0.147883 | 2694.816 | 1733.102 | shell |
| 10 | 100 | 0.68146 | 8 | 0.012606 | 1817.561 | 1703.918 | beam |
| 10 | 100 | 0.68558 | 8 | 0.012544 | 1808.679 | 1713.504 | beam |
| 10 | 100 | 0.68352 | 8.039063 | 0.012524 | 1805.468 | 1714.052 | beam |
| 10.01953 | 100 | 0.68558 | 8 | 0.147923 | 2677.274 | 1736.845 | shell |
| 9.980469 | 100 | 0.68558 | 8 | 0.0126 | 1813.357 | 1712.165 | beam |
| 10 | 100 | 0.68661 | 8 | 0.012529 | 1806.475 | 1715.899 | beam |
| 10 | 100 | 0.68455 | 8 | 0.012559 | 1810.889 | 1711.108 | beam |
| 10 | 100 | 0.68455 | 8.019531 | 0.012534 | 1807.066 | 1713.782 | beam |
| 9.980469 | 100 | 0.68352 | 8 | 0.012631 | 1817.794 | 1707.377 | beam |
| 10.01953 | 100 | 0.68352 | 8 | 0.14844 | 2694.878 | 1731.767 | shell |
| 10 | 100 | 0.68249 | 8 | 0.01259 | 1815.33 | 1706.315 | beam |
| 10 | 100 | 0.68455 | 8 | 0.012559 | 1810.889 | 1711.108 | beam |
| 10 | 100 | 0.68352 | 8.019531 | 0.012549 | 1809.279 | 1711.382 | beam |
| 10.00977 | 100 | 0.68455 | 8 | 0.14846 | 2686.086 | 1733.638 | shell |
| 9.990234 | 100 | 0.68455 | 8 | 0.012587 | 1813.228 | 1710.44 | beam |
| 10 | 100 | 0.685065 | 8 | 0.012552 | 1809.783 | 1712.306 | beam |
| 10 | 100 | 0.684035 | 8 | 0.012567 | 1811.997 | 1709.91 | beam |
| 10 | 100 | 0.684035 | 8.009766 | 0.012554 | 1810.082 | 1711.246 | beam |
| 9.990234 | 100 | 0.68352 | 8 | 0.012603 | 1815.448 | 1708.045 | beam |
| 10.00977 | 100 | 0.68352 | 8 | 0.14872 | 2694.909 | 1731.1 | shell |
| 10 | 100 | 0.683005 | 8 | 0.012583 | 1814.217 | 1707.514 | beam |
| 10 | 100 | 0.684035 | 8 | 0.012567 | 1811.997 | 1709.91 | beam |
| 10 | 100 | 0.68352 | 8.009766 | 0.012562 | 1811.191 | 1710.047 | beam |
| 10 | 100 | 0.684035 | 8 | 0.012567 | 1811.997 | 1709.91 | beam |

As explained above the length/width and length/height ratios define the logic for switching between the analysis models. Hence the optimization problem switches between the beam and the shell element models based on the results tabulated in Table 4 and Table 5.

**Sample Problem 3.**

Objective:

Minimize V=lt(2(w-t) +h))

Subject to:

$$(\sigma_V)_{max} \leq 3000 \text{ psi}$$

$$\delta_{max} \leq 0.5 \text{inch}$$

$$11.11 \leq l/h \leq 6.66$$

$$200 \leq l/t \leq 100$$

$$3 \leq l/w \leq 8$$

with l=100in, Accuracy expectation= 0.6 and eccentricity ratio=0.2

Results:

The model ran for 53 different optimization design states. The beam element model ran for 3 times and the shell element model ran for 50 times.

Minimum volume= $1237 in^3$

Design variables at the optimum solution

Height=13.79, thickness =1 and width = 8

As seen from the Table 9 for a low eccentricity of the load, the beam element runs for length/height ratio greater than 10 and length/width ratio greater than 5. Otherwise for an increased eccentricity, the beam element is as accurate as the shell element only for

length/width ratios greater than 10. Dimensions of the I-beam and deflection are in inches, Stress in lb/in2 and Volume in in3.

**Table 9.Optimization results for sample problem 3**

| Height | Length | Thickness | Width | Deflection | Stress | Volume | decision |
|---|---|---|---|---|---|---|---|
| 9 | 100 | 0.5673 | 5.89397 | 0.024024 | 1139.934 | 3151.591 | beam |
| 13.5 | 100 | 0.5673 | 5.89397 | 0.12363 | 1402.402 | 3859.663 | shell |
| 9 | 100 | 0.85095 | 5.89397 | 0.017498 | 1649.127 | 2299.612 | beam |
| 9 | 100 | 0.85095 | 8 | 0.01349 | 2007.552 | 1759.819 | beam |
| 13.5 | 100 | 1 | 8 | 0.053641 | 2850 | 1237.073 | shell |
| 13.5 | 100 | 0.71635 | 8 | 0.080931 | 2061.917 | 2434.932 | shell |
| 13.5 | 100 | 1 | 5.053015 | 0.067715 | 2260.603 | 1715.8 | shell |
| 10.5 | 100 | 1 | 8 | 0.089675 | 2550 | 1616.504 | shell |
| 15 | 100 | 0.71635 | 8 | 0.06775 | 2169.369 | 2435.019 | shell |
| 15 | 100 | 1 | 5.053015 | 0.053745 | 2410.603 | 1502.746 | shell |
| 11.25 | 100 | 1 | 8 | 0.077583 | 2625 | 1491.143 | shell |
| 15 | 100 | 1 | 8 | 0.043818 | 3000 | 1237.788 | shell |
| 13.5 | 100 | 0.858175 | 8 | 0.064287 | 2457.97 | 1653.778 | shell |
| 13.5 | 100 | 1 | 6.526508 | 0.059289 | 2555.302 | 1413.214 | shell |
| 14.625 | 100 | 1 | 8 | 0.045964 | 2962.5 | 1237.858 | shell |
| 12.375 | 100 | 1 | 8 | 0.063801 | 2737.5 | 1335.145 | shell |
| 13.5 | 100 | 0.929088 | 8 | 0.058424 | 2654.488 | 1422.222 | shell |
| 13.5 | 100 | 1 | 7.263254 | 0.05621 | 2702.651 | 1301.453 | shell |
| 12.9375 | 100 | 1 | 8 | 0.058352 | 2793.75 | 1268.506 | shell |
| 14.0625 | 100 | 1 | 8 | 0.049549 | 2906.25 | 1237.913 | shell |
| 13.5 | 100 | 0.964544 | 8 | 0.055916 | 2752.37 | 1324.681 | shell |
| 13.5 | 100 | 1 | 7.631627 | 0.054873 | 2776.325 | 1253.886 | shell |
| 13.78125 | 100 | 1 | 8 | 0.051522 | 2878.125 | 1237.071 | shell |
| 13.78125 | 100 | 0.982272 | 8 | 0.052601 | 2828.843 | 1279.708 | shell |
| 13.78125 | 100 | 1 | 7.815813 | 0.05209 | 2841.288 | 1241.53 | shell |
| 14.34375 | 100 | 1 | 8 | 0.047697 | 2934.375 | 1237.893 | shell |
| 13.78125 | 100 | 1 | 8 | 0.051522 | 2878.125 | 1237.071 | shell |
| 14.0625 | 100 | 0.982272 | 8 | 0.050599 | 2856.469 | 1280.571 | shell |
| 14.0625 | 100 | 1 | 7.815813 | 0.050075 | 2869.413 | 1241.529 | shell |
| 13.64063 | 100 | 1 | 8 | 0.052563 | 2864.063 | 1237.075 | shell |
| 13.92188 | 100 | 1 | 8 | 0.050521 | 2892.188 | 1237.916 | shell |
| 13.78125 | 100 | 0.991136 | 8 | 0.052055 | 2853.492 | 1258.111 | shell |
| 13.78125 | 100 | 1 | 7.907907 | 0.051806 | 2859.706 | 1241.674 | shell |
| 13.85156 | 100 | 1 | 8 | 0.05102 | 2885.156 | 1237.916 | shell |
| 13.71094 | 100 | 1 | 8 | 0.052038 | 2871.094 | 1237.074 | shell |
| 13.78125 | 100 | 0.995568 | 8 | 0.051787 | 2865.81 | 1247.523 | shell |
| 13.78125 | 100 | 1 | 7.953953 | 0.051666 | 2868.916 | 1239.816 | shell |
| 13.74609 | 100 | 1 | 8 | 0.051779 | 2874.609 | 1237.073 | shell |
| 13.81641 | 100 | 1 | 8 | 0.051273 | 2881.641 | 1237.915 | shell |
| 13.78125 | 100 | 0.997784 | 8 | 0.051654 | 2871.968 | 1242.28 | shell |
| 13.78125 | 100 | 1 | 7.976977 | 0.051597 | 2873.52 | 1240.784 | shell |
| 13.79883 | 100 | 1 | 8 | 0.051394 | 2879.883 | 1237.07 | shell |
| 13.79883 | 100 | 0.998892 | 8 | 0.05146 | 2876.803 | 1239.671 | shell |
| 13.79883 | 100 | 1 | 7.988488 | 0.051434 | 2877.58 | 1241.267 | shell |
| 13.83398 | 100 | 1 | 8 | 0.051146 | 2883.398 | 1237.915 | shell |
| 13.79883 | 100 | 1 | 8 | 0.051394 | 2879.883 | 1237.07 | shell |
| 13.81641 | 100 | 0.998892 | 8 | 0.051339 | 2878.558 | 1240.517 | shell |
| 13.81641 | 100 | 1 | 7.988488 | 0.051307 | 2879.338 | 1241.267 | shell |
| 13.79004 | 100 | 1 | 8 | 0.051458 | 2879.004 | 1237.071 | shell |
| 13.80762 | 100 | 1 | 8 | 0.051336 | 2880.762 | 1237.915 | shell |
| 13.79883 | 100 | 0.999446 | 8 | 0.051427 | 2878.343 | 1238.369 | shell |
| 13.79883 | 100 | 1 | 7.994244 | 0.051417 | 2878.732 | 1241.507 | shell |
| 13.79883 | 100 | 1 | 8 | 0.051394 | 2879.883 | 1237.07 | shell |

**Sample Problem 4:**

Objective:

Minimize V=lt(2(w-t) +h))

Subject to:

$(\sigma_V)_{max} \leq 4000psi$

$\delta_{max} \leq 0.5inch$

$5 \leq height \leq 15$

$0.5 \leq thickness \leq 1$

$2 \leq width \leq 10$

with l=100in, Accuracy expectation= 0.9 and eccentricity ratio =0.4

Results:

The model ran for 87 different optimization design states. The shell element model ran for 87 times.

Minimum volume= 1208.8 $in^3$

Design variables at the optimum solution

Height=12.52in, thickness =1in and width = 9.617in

As seen from the values of the Table 10, for an accuracy expectation of 0.9, the beam element models is as accurate as the shell element model only for length/height ratio of 15 and length/width ratio of 20. Dimensions of the I-beam and deflection are in inches, Stress in $lb/in^2$ and Volume in $in^3$.

## Table 10.Optimization results for sample problem 4

| Height | Length | Thickness | Width | Deflection | Stress | Volume | decision |
|---|---|---|---|---|---|---|---|
| 10 | 100 | 0.5673 | 8 | 0.1865 | 1442.797 | 4037.544 | beam |
| 15 | 100 | 0.5673 | 8 | 0.094824 | 1726.447 | 4041.242 | beam |
| 5 | 100 | 0.5673 | 8 | 0.751719 | 1159.147 | 6314.902 | beam |
| 10 | 100 | 0.85095 | 8 | 0.117013 | 2140.058 | 1979.287 | beam |
| 10 | 100 | 0.85095 | 10 | 0.10316 | 2480.438 | 1725.348 | beam |
| 5 | 100 | 1 | 10 | 0.406558 | 2400 | 3302.95 | beam |
| 15 | 100 | 1 | 10 | 0.040065 | 3400 | 1216.33 | beam |
| 15 | 100 | 0.71635 | 10 | 0.063041 | 2455.909 | 2510.536 | beam |
| 15 | 100 | 1 | 6 | 0.049648 | 2600 | 1325.642 | beam |
| 10 | 100 | 1 | 10 | 0.086771 | 2900 | 1420.379 | beam |
| 15 | 100 | 0.71635 | 10 | 0.063041 | 2455.909 | 2510.536 | beam |
| 15 | 100 | 1 | 6 | 0.049648 | 2600 | 1325.642 | beam |
| 12.5 | 100 | 1 | 10 | 0.055899 | 3150 | 1216.181 | beam |
| 12.5 | 100 | 0.858175 | 10 | 0.067183 | 2715.422 | 1696.168 | beam |
| 12.5 | 100 | 1 | 8 | 0.062519 | 2750 | 1319.753 | beam |
| 7.5 | 100 | 1 | 10 | 0.159659 | 2650 | 1983.991 | beam |
| 12.5 | 100 | 1 | 10 | 0.055899 | 3150 | 1216.181 | beam |
| 10 | 100 | 0.858175 | 10 | 0.102194 | 2500.879 | 1693.881 | beam |
| 10 | 100 | 1 | 8 | 0.099507 | 2500 | 1712.326 | beam |
| 13.75 | 100 | 1 | 10 | 0.046818 | 3275 | 1218.421 | beam |
| 11.25 | 100 | 1 | 10 | 0.06853 | 3025 | 1244.404 | beam |
| 12.5 | 100 | 0.929088 | 10 | 0.060972 | 2933.214 | 1427.661 | beam |
| 12.5 | 100 | 1 | 9 | 0.058874 | 2950 | 1222.931 | beam |
| 11.875 | 100 | 1 | 10 | 0.061668 | 3087.5 | 1216.842 | beam |
| 13.125 | 100 | 1 | 10 | 0.05101 | 3212.5 | 1219.57 | beam |
| 12.5 | 100 | 0.964544 | 10 | 0.05831 | 3041.733 | 1316.006 | beam |
| 12.5 | 100 | 1 | 9.5 | 0.057322 | 3050 | 1214.778 | beam |
| 11.875 | 100 | 1 | 9 | 0.065135 | 2887.5 | 1276.107 | beam |
| 13.125 | 100 | 1 | 9 | 0.053569 | 3012.5 | 1223.018 | beam |
| 12.5 | 100 | 0.964544 | 9 | 0.061345 | 2848.824 | 1309.91 | beam |
| 12.5 | 100 | 1 | 8.5 | 0.060595 | 2850 | 1258.794 | beam |
| 12.5 | 100 | 1 | 9.5 | 0.057322 | 3050 | 1214.778 | beam |
| 12.8125 | 100 | 1 | 9.5 | 0.054674 | 3081.25 | 1214.835 | beam |
| 12.1875 | 100 | 1 | 9.5 | 0.060187 | 3018.75 | 1213.598 | beam |
| 12.1875 | 100 | 0.982272 | 9.5 | 0.061422 | 2966.975 | 1255.677 | beam |
| 12.1875 | 100 | 1 | 9.75 | 0.05941 | 3068.75 | 1211.662 | beam |
| 11.5625 | 100 | 1 | 10 | 0.064945 | 3056.25 | 1216.706 | beam |
| 12.1875 | 100 | 1 | 10 | 0.058662 | 3118.75 | 1216.467 | beam |
| 11.875 | 100 | 0.982272 | 10 | 0.06293 | 3034.506 | 1264.934 | beam |
| 11.875 | 100 | 1 | 9.75 | 0.062475 | 3037.5 | 1211.94 | beam |
| 12.34375 | 100 | 1 | 9.75 | 0.057971 | 3084.375 | 1211.503 | beam |
| 12.34375 | 100 | 0.991136 | 9.75 | 0.058567 | 3057.913 | 1235.24 | beam |
| 12.34375 | 100 | 1 | 9.625 | 0.058341 | 3059.375 | 1209.079 | beam |
| 12.65625 | 100 | 1 | 9.5 | 0.055971 | 3065.625 | 1214.811 | beam |
| 12.34375 | 100 | 1 | 9.5 | 0.05872 | 3034.375 | 1213.635 | beam |
| 12.5 | 100 | 0.991136 | 9.5 | 0.057911 | 3023.843 | 1235.555 | beam |
| 12.5 | 100 | 1 | 9.375 | 0.057695 | 3025 | 1214.765 | beam |
| 12.5 | 100 | 1 | 9.625 | 0.056948 | 3075 | 1208.908 | beam |
| 12.8125 | 100 | 1 | 9.625 | 0.054336 | 3106.25 | 1212.654 | beam |
| 12.5 | 100 | 1 | 9.625 | 0.056948 | 3075 | 1208.908 | beam |
| 12.65625 | 100 | 0.991136 | 9.625 | 0.056197 | 3064.108 | 1236.556 | beam |
| 12.65625 | 100 | 1 | 9.5 | 0.055971 | 3065.625 | 1214.811 | beam |
| 12.65625 | 100 | 1 | 9.75 | 0.055268 | 3115.625 | 1211.151 | beam |
| 12.42188 | 100 | 1 | 9.625 | 0.057637 | 3067.188 | 1208.995 | beam |
| 12.57813 | 100 | 1 | 9.625 | 0.056281 | 3082.813 | 1212.891 | beam |
| 12.5 | 100 | 0.995568 | 9.625 | 0.05724 | 3061.813 | 1220.666 | beam |
| 12.5 | 100 | 1 | 9.6875 | 0.056769 | 3087.5 | 1211.157 | beam |
| 12.5 | 100 | 1 | 9.5625 | 0.057138 | 3062.5 | 1212.413 | beam |
| 12.53906 | 100 | 1 | 9.625 | 0.056608 | 3078.906 | 1208.893 | beam |
| 12.53906 | 100 | 0.997784 | 9.625 | 0.056753 | 3072.304 | 1214.721 | beam |
| 12.53906 | 100 | 1 | 9.59375 | 0.056707 | 3072.656 | 1213.562 | beam |
| 12.53906 | 100 | 1 | 9.65625 | 0.05652 | 3085.156 | 1210.031 | beam |
| 12.61719 | 100 | 1 | 9.625 | 0.055949 | 3086.719 | 1212.854 | beam |
| 12.53906 | 100 | 1 | 9.625 | 0.056608 | 3078.906 | 1208.893 | beam |
| 12.57813 | 100 | 0.997784 | 9.625 | 0.056426 | 3076.202 | 1218.765 | beam |
| 12.57813 | 100 | 1 | 9.59375 | 0.05637 | 3076.563 | 1213.571 | beam |
| 12.57813 | 100 | 1 | 9.65625 | 0.056184 | 3089.063 | 1210.036 | beam |
| 12.51953 | 100 | 1 | 9.625 | 0.056778 | 3076.953 | 1208.891 | beam |
| 12.51953 | 100 | 0.998892 | 9.625 | 0.05685 | 3073.655 | 1211.81 | beam |
| 12.51953 | 100 | 1 | 9.640625 | 0.056733 | 3080.078 | 1209.46 | beam |
| 12.51953 | 100 | 1 | 9.609375 | 0.056823 | 3073.828 | 1213.02 | beam |
| 12.48047 | 100 | 1 | 9.625 | 0.057119 | 3073.047 | 1208.93 | beam |
| 12.51953 | 100 | 1 | 9.625 | 0.056778 | 3076.953 | 1208.891 | beam |
| 12.5 | 100 | 0.998892 | 9.625 | 0.057021 | 3071.704 | 1211.832 | beam |
| 12.5 | 100 | 1 | 9.609375 | 0.056993 | 3071.875 | 1213.017 | beam |
| 12.5 | 100 | 1 | 9.640625 | 0.056903 | 3078.125 | 1209.457 | beam |
| 12.5293 | 100 | 1 | 9.625 | 0.056693 | 3077.93 | 1208.892 | beam |
| 12.50977 | 100 | 1 | 9.625 | 0.056863 | 3075.977 | 1208.897 | beam |
| 12.51953 | 100 | 0.999446 | 9.625 | 0.056814 | 3075.304 | 1210.346 | beam |
| 12.51953 | 100 | 1 | 9.632813 | 0.056755 | 3078.516 | 1209.176 | beam |
| 12.51953 | 100 | 1 | 9.617188 | 0.0568 | 3075.391 | 1208.756 | beam |
| 12.50977 | 100 | 1 | 9.609375 | 0.056908 | 3072.852 | 1213.018 | beam |
| 12.5293 | 100 | 1 | 9.609375 | 0.056738 | 3074.805 | 1213.021 | beam |
| 12.51953 | 100 | 0.999446 | 9.609375 | 0.056859 | 3072.181 | 1214.301 | beam |
| 12.51953 | 100 | 1 | 9.601563 | 0.056845 | 3072.266 | 1212.736 | beam |
| 12.51953 | 100 | 1 | 9.617188 | 0.0568 | 3075.391 | 1208.756 | beam |
| 12.51953 | 100 | 1 | 9.617188 | 0.0568 | 3075.391 | 1208.756 | beam |

**Sample Problem 5.**

Objective:

Minimize V=lt(2(w-t) +h))

Subject to:

$$(\sigma_V)_{max} \leq 6000psi$$

$$\delta_{max} \leq 0.5inch$$

$$20 \leq l/h \leq 6.66$$

$$200 \leq l/t \leq 100$$

$$50 \leq l/w \leq 10$$

With l=100in, Accuracy expectation= 0.2 and eccentricity ratio=0.4

Results:

The model ran for 54 different optimization design states.The beam element model ran for 54 times.

Minimum volume= 701.18 $in^3$

Design variables at the optimum solution

Height=5.49, thickness =0.5 and width = 4.28.

As seen from the values of Table 11, for an accuracy expectation of 0.2, the beam element model will be used for all the cases. Dimensions of the I-beam and deflection are in inches, Stress in $lb/in^2$ and volume in $in^3$.We notice that when compared to the shell element model that ran for an accuracy expectation of 0.9, the beam element model shows some difference in the optimization process. This difference is 6%. This is because the shell element model is more accurate for an eccentricity of 0.4, though the beam element model will be used for an accuracy of 0.2. But the shell element model is computationally expensive. In terms of time constraint, the shell model took almost 40%

more time than the beam element model. Hence the tradeoff of selecting the beam element model for low accuracy expectation is well justified.

**Table 11. Optimization results for sample problem 5**

| Height | Length | Thickness | Width | Deflection | Stress | Volume | decision |
|---|---|---|---|---|---|---|---|
| 10 | 100 | 0.5673 | 10 | 0.012195 | 1662.534 | 1741.747 | beam |
| 15 | 100 | 0.5673 | 10 | 0.004982 | 1946.184 | 1040.515 | beam |
| 15 | 100 | 0.85095 | 10 | 0.003493 | 2858.502 | 731.9056 | beam |
| 15 | 100 | 0.85095 | 5 | 0.005839 | 2007.552 | 1258.046 | beam |
| 10 | 100 | 1 | 10 | 0.00772 | 2825 | 1107.165 | beam |
| 15 | 100 | 0.71635 | 10 | 0.004051 | 2429.594 | 847.5509 | beam |
| 15 | 100 | 1 | 5 | 0.005134 | 2325 | 1107.301 | beam |
| 12.5 | 100 | 0.85095 | 10 | 0.005266 | 2645.764 | 931.3696 | beam |
| 15 | 100 | 0.709125 | 10 | 0.004087 | 2406.366 | 855.017 | beam |
| 15 | 100 | 0.992775 | 10 | 0.003071 | 3302.592 | 644.4549 | beam |
| 15 | 100 | 0.85095 | 7.5 | 0.004355 | 2433.027 | 925.4205 | beam |
| 13.75 | 100 | 0.85095 | 10 | 0.004248 | 2752.133 | 821.2076 | beam |
| 15 | 100 | 0.921863 | 10 | 0.003266 | 3081.553 | 684.7506 | beam |
| 15 | 100 | 0.780038 | 10 | 0.003763 | 2633.44 | 787.789 | beam |
| 15 | 100 | 0.85095 | 8.75 | 0.003873 | 2645.764 | 817.3654 | beam |
| 14.375 | 100 | 0.85095 | 10 | 0.003844 | 2805.317 | 774.3289 | beam |
| 15 | 100 | 0.815494 | 10 | 0.003622 | 2746.222 | 758.6141 | beam |
| 15 | 100 | 0.886406 | 10 | 0.003375 | 2970.279 | 707.3679 | beam |
| 15 | 100 | 0.886406 | 9.375 | 0.003549 | 2859.478 | 746.4619 | beam |
| 14.375 | 100 | 0.921863 | 10 | 0.003595 | 3023.936 | 724.7732 | beam |
| 15 | 100 | 0.957319 | 10 | 0.003165 | 3192.324 | 663.8405 | beam |
| 15 | 100 | 0.886406 | 10 | 0.003375 | 2970.279 | 707.3679 | beam |
| 15 | 100 | 0.921863 | 9.375 | 0.003434 | 2966.32 | 722.6669 | beam |
| 14.6875 | 100 | 0.886406 | 10 | 0.003539 | 2942.578 | 727.4502 | beam |
| 15 | 100 | 0.868678 | 10 | 0.003433 | 2914.453 | 719.382 | beam |
| 15 | 100 | 0.904134 | 10 | 0.003319 | 3025.979 | 695.8333 | beam |
| 15 | 100 | 0.886406 | 9.6875 | 0.00346 | 2914.878 | 726.3893 | beam |
| 14.84375 | 100 | 0.886406 | 10 | 0.003456 | 2956.429 | 717.2875 | beam |
| 15 | 100 | 0.89527 | 10 | 0.003347 | 2998.144 | 701.5425 | beam |
| 15 | 100 | 0.89527 | 9.84375 | 0.003388 | 2970.167 | 710.8539 | beam |
| 14.84375 | 100 | 0.904134 | 10 | 0.003399 | 3011.851 | 705.6108 | beam |
| 15 | 100 | 0.912998 | 10 | 0.003292 | 3053.781 | 690.2371 | beam |
| 15 | 100 | 0.89527 | 10 | 0.003347 | 2998.144 | 701.5425 | beam |
| 15 | 100 | 0.904134 | 9.84375 | 0.00336 | 2997.724 | 705.0733 | beam |
| 14.92188 | 100 | 0.89527 | 10 | 0.003387 | 2991.15 | 706.4367 | beam |
| 15 | 100 | 0.890838 | 10 | 0.003361 | 2984.215 | 704.4404 | beam |
| 15 | 100 | 0.899702 | 10 | 0.003333 | 3012.065 | 698.6736 | beam |
| 15 | 100 | 0.89527 | 9.921875 | 0.003367 | 2984.156 | 706.1675 | beam |
| 14.96094 | 100 | 0.89527 | 10 | 0.003367 | 2994.647 | 703.9822 | beam |
| 15 | 100 | 0.897486 | 10 | 0.00334 | 3005.106 | 700.1044 | beam |
| 15 | 100 | 0.893054 | 10 | 0.003354 | 2991.181 | 702.9878 | beam |
| 15 | 100 | 0.89527 | 9.960938 | 0.003357 | 2991.15 | 703.8474 | beam |
| 14.98047 | 100 | 0.89527 | 10 | 0.003357 | 2996.396 | 702.7605 | beam |
| 15 | 100 | 0.894162 | 10 | 0.00335 | 2994.663 | 702.2642 | beam |
| 15 | 100 | 0.896378 | 10 | 0.003343 | 3001.625 | 700.8225 | beam |
| 15 | 100 | 0.89527 | 9.980469 | 0.003352 | 2994.647 | 702.693 | beam |
| 14.99023 | 100 | 0.89527 | 10 | 0.003352 | 2997.27 | 702.151 | beam |
| 15 | 100 | 0.895824 | 10 | 0.003345 | 2999.885 | 701.1823 | beam |
| 15 | 100 | 0.895824 | 9.990234 | 0.003348 | 2998.135 | 701.7568 | beam |
| 14.99023 | 100 | 0.896378 | 10 | 0.003348 | 3000.75 | 701.4305 | beam |
| 15 | 100 | 0.896932 | 10 | 0.003342 | 3003.366 | 700.4632 | beam |
| 15 | 100 | 0.895824 | 10 | 0.003345 | 2999.885 | 701.1823 | beam |
| 15 | 100 | 0.896378 | 9.990234 | 0.003346 | 2999.875 | 701.3968 | beam |
| 15 | 100 | 0.895824 | 10 | 0.003345 | 2999.885 | 701.1823 | beam |

**Sample Problem 6:**

Objective:

Minimize V=lt(2(w-t) +h))

Subject to:

$$(\sigma_V)_{max} \leq 2998psi$$

$$\delta_{max} \leq 0.5inch$$

$$12.5 \leq l/h \leq 8.33$$

$$200 \leq l/t \leq 142.85$$

$$12.5 \leq l/w \leq 8.33$$

with l=100in, Accuracy expectation= 0.7 and eccentricity ratio =1

Results:

The model ran for 66 different optimization design states. The beam element model ran for 57 times and the shell element model ran for 9 times.

Minimum volume= 1189.16 $in^3$

Design variables at the optimum solution:

Height=8.28, thickness =0.5 and width = 8

For an eccentricity of 1 and for the specified bounds on the length, width and height, the shell beam model performs as accurate as the shell model for length/height and length/width ratios greater than 10. As the length/width and length/height ratios increase, the beam model gets more accurate as the shell element model. However the error in the models always exists as shell element model is highly accurate in capturing the eccentricity of the load.

## 9.1. Summary of the results:

The sample six optimization runs demonstrate the capability of automated analysis and optimization tools to reuse and interoperate modeling knowledge. The bounds on the design variables, the accuracy expectation and the eccentricity of the load have been taken for the models in such a way that the shifting of the intelligent tool between shell and beam elements models has been clearly demonstrated. For example, for sample problems 4 and 5 the difference in the optimization problem is just with the accuracy expectation of the model. The optimization results show that for an accuracy expectation of 0.2, the beam model has been used and for an accuracy expectation of 0.9, both shell and beam models have been used based on the length, width and height ratios. From the optimization results of the sample problems 4 and 5 the differences in the minimum volume obtained in both the cases is 6%. This difference is in a well acceptable range for the accuracy expectation of 0.2 and 0.9 as for an accuracy expectation of 0.9, the beam element model ran for 83 times and the shell element model ran for 17 times. The difference is associated with the running of shell element model for 17 times, whereas for an accuracy expectation of 0.2, the process ran only with beam element models. Table 12 lists the results of optimization process for all the sample problems.

**Table 12. Optimization results for all the sample problems**

| | | Height | (in) | Width | (in) | | Thickness | (in) | | Length(in) | eccentricit | Max Vonmises | Min Volum |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Lower | Upper | optimum | Lower | Upper | optimum | Lower | Upper | optimum | | | stress constraint | inch3 |
| | bound | bound | height | bound | bound | width | bound | bound | thickness | | | psi | |
| Problem 1 | 5 | 15 | 10 | 5 | 15 | 10 | 0.5 | 1 | 0.547 | 100 | 0.4 | 6000 | 1592 |
| Problem 2 | 8 | 12 | 10 | 8 | 12 | 8 | 0.5 | 0.7 | 0.684 | 100 | 0.6 | 3624 | 1709.9 |
| Problem 3 | 9 | 15 | 13.79 | 3 | 8 | 8 | 0.5 | 1 | 1 | 100 | 0.2 | 3000 | 1237 |
| Problem 4 | 5 | 15 | 12.519 | 2 | 10 | 9.617 | 0.5 | 1 | 1 | 100 | 0.4 | 4000 | 1208 |
| Problem 5 | 5 | 15 | 5.49 | 2 | 10 | 4.28 | 0.5 | 1 | 0.5 | 100 | 0.4 | 6000 | 701 |
| Problem 6 | 8 | 12 | 8.283 | 8 | 12 | 8 | 0.5 | 0.7 | 0.5 | 100 | 1 | 2998 | 1189 |

For sample problem 6, which has highest eccentricity value, the beam element model ran for 32 times, as the length/width and length/height ratios exceeded 10 as beam element models are as accurate as the shell element models. But for ratios less than 10, shell element model can only be efficient in capturing the effects of eccentricity.

Four sample problems have been run with a "medium" accuracy expectation to clearly show the shifting of the intelligent decision making tool between analysis models. Sample problems 1, 2, 3 and 6 have been formulated in a way to analyze different models with varying eccentricity and accuracy expectation. As seen from the optimization results, as the eccentricity increases for decreasing length/ height and length/width ratios, the shell element model runs more number of times. Both the ratios, length/ height and length/width determine whether to run a beam or a shell element model. For a medium accuracy model, if the length/height and length/width ratios are greater than 10, only then the beam element model runs for the given design optimization state. Else, shell element model works.

Shell element model is taken as the highly accurate model as it can capture the eccentricity effects though it is computationally expensive. The results obtained for the optimization processes using intelligent decision making tool have been validated by running the optimization processes with the shell element model alone without any shifting between the beam and shell element models. If the results obtained by running the shell element model alone during the optimization process are comparable with the results obtained by using the intelligent decision making tool, then the automated optimization process which runs using an intelligent decision making tool shifting between beam and shell analyses models is valid.

# CHAPTER 10

## VALIDATION OF TEST BED APPLICATION RESULTS

An optimization process has been setup for the I-beam optimization using the shell element analysis model. This is done to validate the results obtained by automated optimization process using the intelligent decision making tool. Unlike the automated optimization process, the shell element optimization process does not shift between the analysis models during an optimization design state. It works only with the shell element model for any optimization design state. Since shell element model is the highly accurate model in terms of capturing eccentricity effects, the results have been compared to the shell element model optimization process.

An optimization process using only the shell element model is shown in Figure 50.

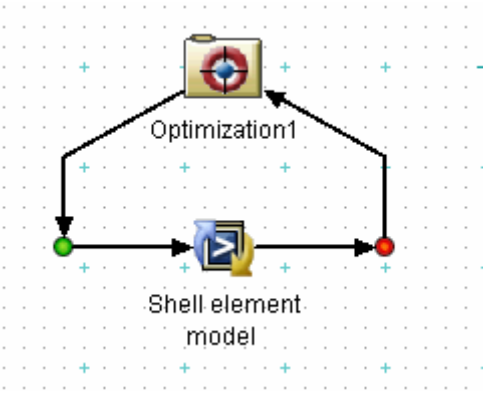



**Figure 50. Setup of shell element model optimization process**

The shell element optimization process works similar to the intelligent decision making tool optimization process in terms of design variables, constraints and objective.

The outputs of shell element model ANSYS SIMCODE component, which are stress, deflection and volume, are mapped back to the optimization process. The shell element model does not have an accuracy expectation as it has the highest accuracy. Accuracy expectation is only used for switching between beam and shell models using the intelligent decision making tool, where beam element models can be used as substitute for shell element models when they satisfy the accuracy condition setup by the optimization process. This saves computational time and cost.

Sample problems 1, 2 and 3 and 6 formulated in section 9 have been used for the optimization process by using the shell element model for checking the validity of automated optimization process using the intelligent decision making tool. Sample problems 4 and 5 have not been used for validation of the results as they ran only with either beam or the shell element model for the whole optimization process without using a combination of beam and shell element models due to their eccentricity, accuracy expectation and length, height, width ratios. Table 13 shows the problem formulation and Table 14 shows comparison of the two models.

**Table 13. Formulation of sample problems for the shell element optimization problem**

| | | Height | | Width | | | Thickness | | | Length | | eccentricit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Lower bound | Upper bound | | Lower bound | Upper bound | | Lower bound | Upper bound | | | |
| Problem1 | | 5 | 15 | | 5 | 15 | | 0.5 | 1 | | 100 | | 0.4 |
| Problem 2 | | 8 | 12 | | 8 | 12 | | 0.5 | 0.7 | | 100 | | 0.6 |
| Problem 3 | | 9 | 15 | | 3 | 8 | | 0.5 | 1 | | 100 | | 0.2 |
| Problem 4 | | 8 | 12 | | 8 | 12 | | 0.5 | 0.7 | | 100 | | 1 |

**Table 14. Optimization results obtained for an I-beam optimization problem**

| | Shell element Volume ($in^3$) | Shell element CPU processing time (minutes) | Automated decision making optimization process Volume($in^3$) | Automated decision making optimization process CPU time(minutes) | %increase in volume | % decrease in CPU time |
|---|---|---|---|---|---|---|
| Problem1 | 1307.846 | 28.6 | 1591.635 | 22.4 | 27.6 | 27.67 |
| Problem2 | 1166.5 | 22.5 | 1709.91 | 20.8 | 46.5 | 8 |
| Problem3 | 795.7621 | 15 | 1237 | 8.25 | 55.44 | 81 |
| Problem4 | 803.158 | 18 | 1189.16 | 12.15 | 48 | 48 |

Table 14 shows the difference in the models in terms of the objective function, i.e., Volume and CPU processing time.

Percentage reduction in CPU time is taken as 100*(Shell element only CPU time - intelligent model selector CPU time)/ (shell element only CPU time)

Percentage increase in volume is taken as 100* (intelligent model selector optimal volume - shell element only optimal design volume) / (shell element only optimal design volume).

The sample problems have been formulated in a way that all the possible eccentricity ratios have been considered to analyze the differences in optimization process of shell element process and intelligent decision making process. Validation of the intelligent decision making tool optimization process with the shell element optimization process has been done in 2 ways.

Firstly, the difference in the optimization objective of the models using only shell element model for optimization process and using intelligent decision making tool to shift between beam and shell element analyses models is between 25-55%. The time required to run the shell element optimization process is almost 30-80% more than the time required to run the intelligent decision making tool optimization process. Hence, using the intelligent decision making tool to shift between the beam and the shell element models based on criteria improves time and memory requirement constraints. When there

is no eccentricity of the load, the beam element works as accurately as the shell element. But when eccentricity exists, shifting between the beam and the shell element models saves time and memory. Since shell element model is computationally expensive, it is more logical to tradeoff using the beam element model depending on certain design criteria to obtain optimization results which vary just 25-55% from the shell element model.

Secondly, percentage of shell analysis models that ran during an optimization process using the intelligent decision making tool have also been tabulated. It can be observed that as the number of shell analysis model runs increase for an automated optimization process, the differences in the results of the shell element model optimization process and the automated decision making optimization process decrease.

**Table 15. Percentage of shell element model used vs. error in the processes**

| | Shell element model usage (%) | eccentricity ratio | % Difference in volume obtained |
|---|---|---|---|
| Problem1 | 66 | 0.4 | 27.6 |
| Problem2 | 75 | 0.6 | 46.5 |
| Problem3 | 5.6 | 0.2 | 55.44 |
| Problem4 | 8.6 | 1 | 48 |

From Table 15 we observe that as the usage of the shell element model keeps increasing during the intelligent decision making tool optimization process, the shell element model optimization process shows less deviation in the results. This is because more usage of the shell element model implies that the model is getting more accurate though the computational costs are high.

An increase in eccentricity value makes the optimization process use a large number of shell elements for the values of the design variables obtained during the design optimization state. The increase in number of shell elements increases the accuracy of the model as the shell element models capture the eccentricity in a more accurate way than beam element models for constant length/height and length/width ratios.

Optimization process using the intelligent decision making tool has been proved to be an efficient way of optimizing the I-beam in terms of reducing the computational time and cost without much compromise on the results of the optimization. None of the tools that exist right now have the functionality of switching between the analysis models during an optimization process for a given design optimization state. For example, ANSYS allows logical statement to be written for analysis and optimization problems. Yet, it doesn't handle shifting between two different analysis models for an optimization problem during its design states.

Hence, importing knowledge from ontological knowledge base and using it to drive analysis and optimization tools to automate the processes is an efficient way of reusing and interoperating knowledge between tools in distributed product development environments. This knowledge can later be used for inspection/sharing and generating technical reports.

# CHAPTER 11

## VALIDATION OF OWL EAM ONTOLOGY, TECHNICAL REPORT COMPONENT AND SECURED KNOWLEDGE SHARING COMPONENT WITH INDUSTRIAL APPLICATION

The EAM ontology, technical report component and secured knowledge sharing components have been validated with an industrial application, Catheter clamping-open clamp. The catheter clamp is a product of BD[4] Medical, which offers infusion therapy products and services such as IV catheters, fluid management components, peripherally inserted catheters and invasive monitoring devices. A catheter is a tube that can be inserted into a body cavity, duct or vessel. Catheters allow drainage or injection of fluids or access by surgical instruments. Placement of a catheter into a particular part of the body may allow drainage of fluids, administration of intravenous fluids, measurement of blood pressure, administration of medications, angioplasty etc.

Catheter clamp is a holder for clamping in place a catheter or other hallow tube, such as a urinary catheter tube, a nasogastric tube or intravenous tube on a patient's body. A simple catheter clamp is as shown in Figure 51. The clamp is adjustable so as to be capable of holding the catheter tube against rotary and longitudinal movement, and also be capable of partially or completely closing the bore of the catheter tube by deforming the tube wall.

[4] BD (Becton, Dickinson and Company), a global medical technology company focused on improving drug therapy, enhancing the diagnosis of infectious diseases and advancing drug discovery.www.bd.com

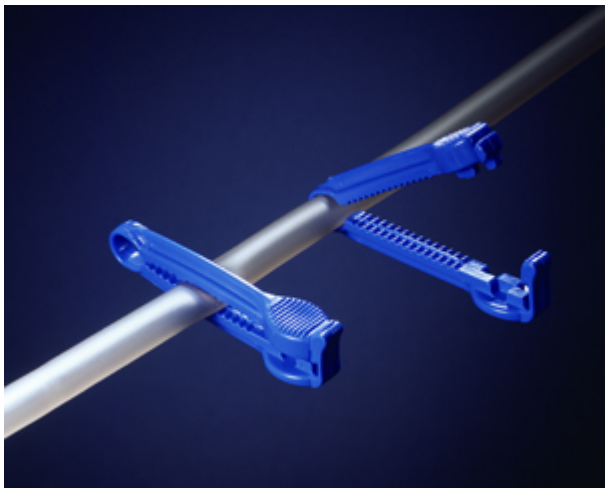


**Figure 51. Simple catheter clamp**

## 11.1 Catheter clamp analysis model

Analysis of catheter open clamp used for BD's products has been done to determine the actuation force of the tube into the clamp for two specified positions, 0.07" and 0.2" from slot end. Figure 52 shows the analysis model of the catheter clamp. As seen from the figure, the distance from the slot end means the distance from the end of the slot to the center point of the tube. The slot end is shown with the arrow. The current open clamp design has also been used to determine the clamping forces acting on the tube and the clamp interface for each tube position (0.07" and 0.2"). The empirical data for the clamp design has been used to decide a coefficient of friction that would closely correlate with the measured results. This calibrated coefficient of friction value which will later be used to predict actuation forces of other clamp models. Solid models for the clamp and the tube have been developed and integrated using ABAQUS[5] analysis tool. The material used for clamp design is polycarbonate Bayer makrolon with an yield strength of 9794psi and yield strain of 0.02887. Tube is made up of nearly incompressible hyper elastic

[5] ABAQUS is a product of SIMULIA. http://www.simulia.com/

material with modulus of 1600psi and poisson's ratio of 0.33. Due to privacy concerns, the actual dimensions of the clamp and the tubing are not included in the documentation.

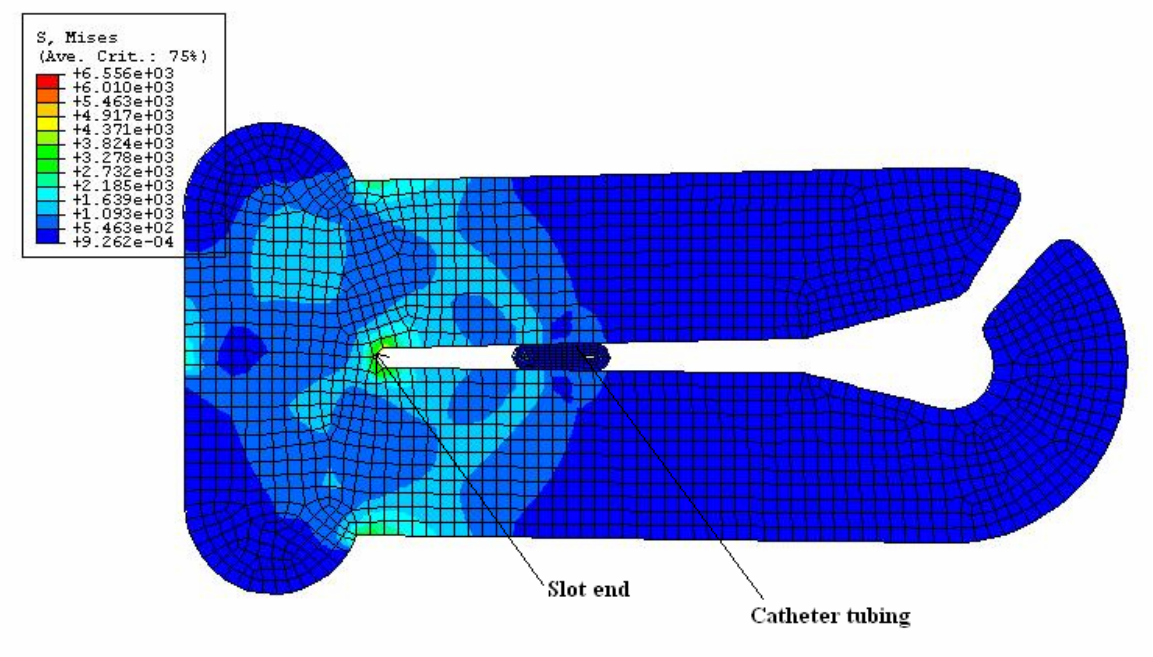



**Figure 52. Analysis model of the catheter open clamp**

The analysis model is a plane strain model with an assumed thickness for the rib on the far left and the main clamp body. Figure 53 shows the nodes of the clamp that were held fixed during the analysis.

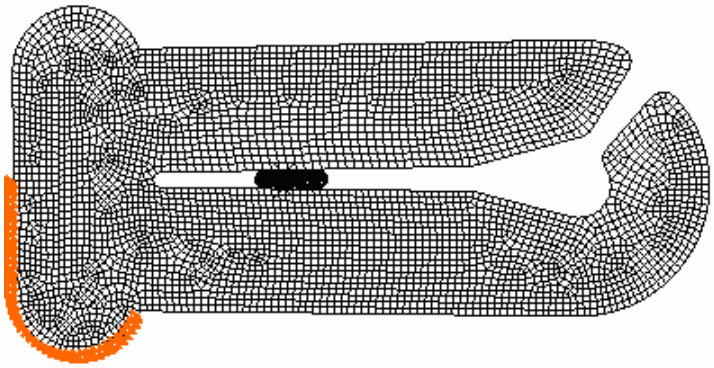


**Figure 53. Nodes in orange show the edges held fixed during the analysis**

Figure 54 shows the two different insertion positions, 0.070” and 0.2” from the slot end.

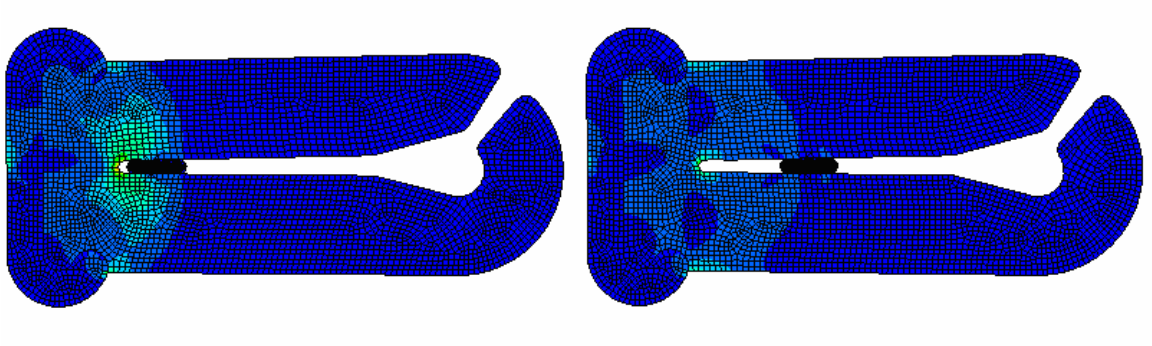


**Figure 54. Two different insertion positions**

The clamp was assumed to have nonlinear plastic material properties of polycarbonate makrolon. The tube component is assumed to be made up of nearly incompressible hyper elastic material. Actuation force predicted for position distances of 0.07” and 0.2” from slot end are 4.15lb and 1.34lb respectively. Clamping force acting on the tube and the clamp interface for each tube position has also been determined. For the tube position of 0.07” from slot end, the clamping force predicted is 28.6 pounds and for the tube position of 0.2” from slot end, the clamping force predicted is 9.24 pounds.

As expected, the actuation and clamping force increased as the tube is further into the slot. Also the calibrated coefficient of friction value of 0.15 best matched the test data for the open clamp. This value will be used in future to predict the actuation forces for other clamp designs. Based on the analysis, the most dramatic effect was caused by the distance of tube insertion for the open clamp geometry. Clamp geometry open clamp or closed clamp also has a significant effect on the actuation force. The Von Mises stress for the open clamp geometry for both the insertion distances have also been obtained. Most

of the cases, the stresses obtained would not exceed the yield strength of the polycarbonate makrolon material.

Additional analysis has been performed to determine the effects of boundary condition assumption on the clamping forces. The boundary condition has been changed such that all the nodes on the base of the clamp have been held fixed for predicting the clamping force. Results obtained show that clamping force decreased with changed boundary conditions. In conclusion, the analyses of the clamp show that the insertion distances play a significant role in the actuation force.

## 11.2 Developing EAM ontology instance for the catheter open clamp model

Ontological instance for catheter open clamp has been developed. The ontological instance contains knowledge about the objectives, input/output parameters, accuracy expectation, limitations and idealizations of the analysis model. The input parameters of the catheter open clamp model are the geometry of the clamp and the insertion distance from the slot end. The output parameters are the clamping and the actuation forces at two different specified positions of 0.07" and 0.2" from the slot end. The limitations of the clamp analysis model are plain strain model, nearly incompressible hyper elastic material used for the tube, non linear plastic material properties assumed for the clamp and the placement of the tube using analysis software, where the placement of tube in the required position has been done by holding two nodes in the x-direction. The model idealizations are altered boundary conditions of the clamp and the prediction of calibrated coefficient of friction so that the analytical predictions closely match the physical test data. The knowledge about catheter clamp will be used to generate customized technical

reports and represent knowledge in a secured environment to be used by other applications as parameters.

Figure 55 shows the instance of analysis modeling knowledge of catheter open clamp within the Protégé environment.

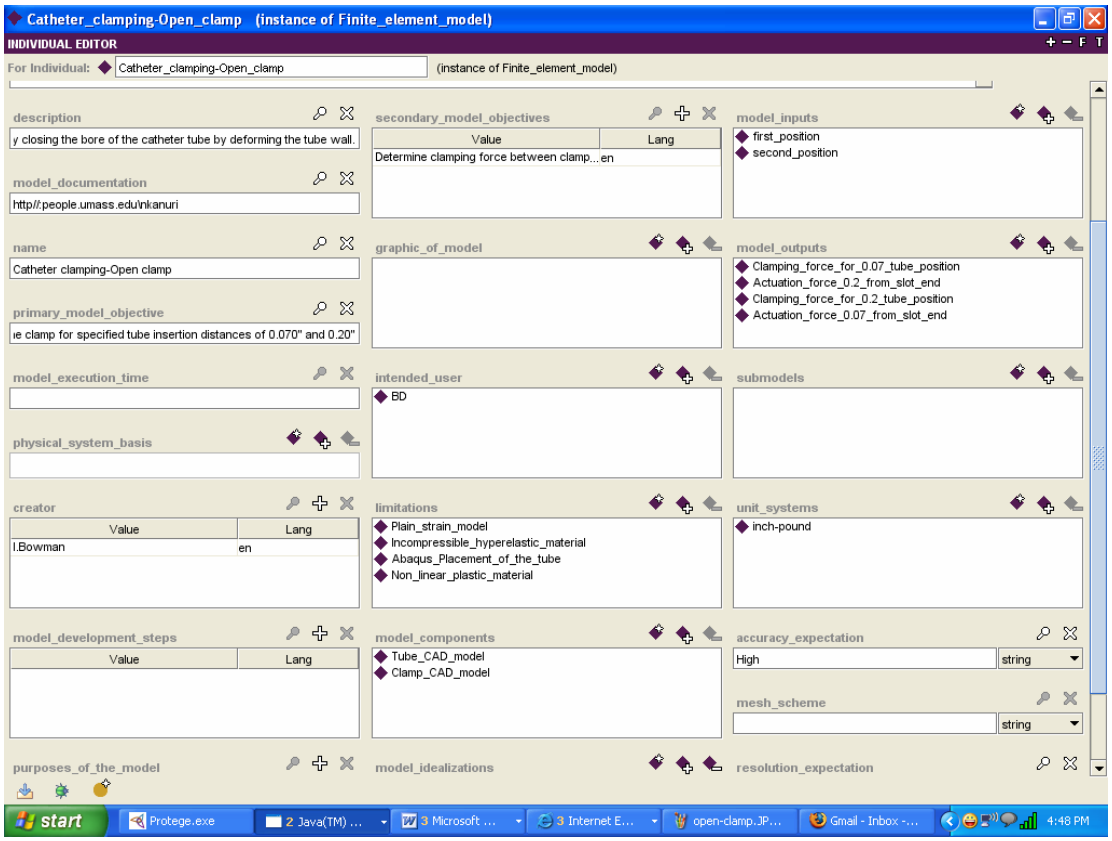



**Figure 55. Analysis modeling knowledge of catheter clamp instance within Protégé environment**

## 11.3 Generation of Technical report in iSIGHT-FD using catheter clamp analysis modeling knowledge

Implementation of Tech report method has been detailed in Section 4. The Tech report method will now be used to generate technical report for the catheter clamp instance. As explained in section 4, the instance can be selected using the Interoperate

OWL KB component and this instance will be passed to the Tech report component for generating the report. Also parallel class of technical report sentences and permissions has been created for the catheter clamping-open clamp instance and will be used to generate the technical reports. Figure 56 shows the "Inspect-It" button which displays the knowledge of an instance in the Protégé instance format. The "+" button, as shown in Figure 56 allows the user to select a slot and its value and make it an iSIGHT-FD parameter just with a button click. The slot and its corresponding value will then be iSIGHT-FD parameters with the same data type as in the OWL knowledge base.

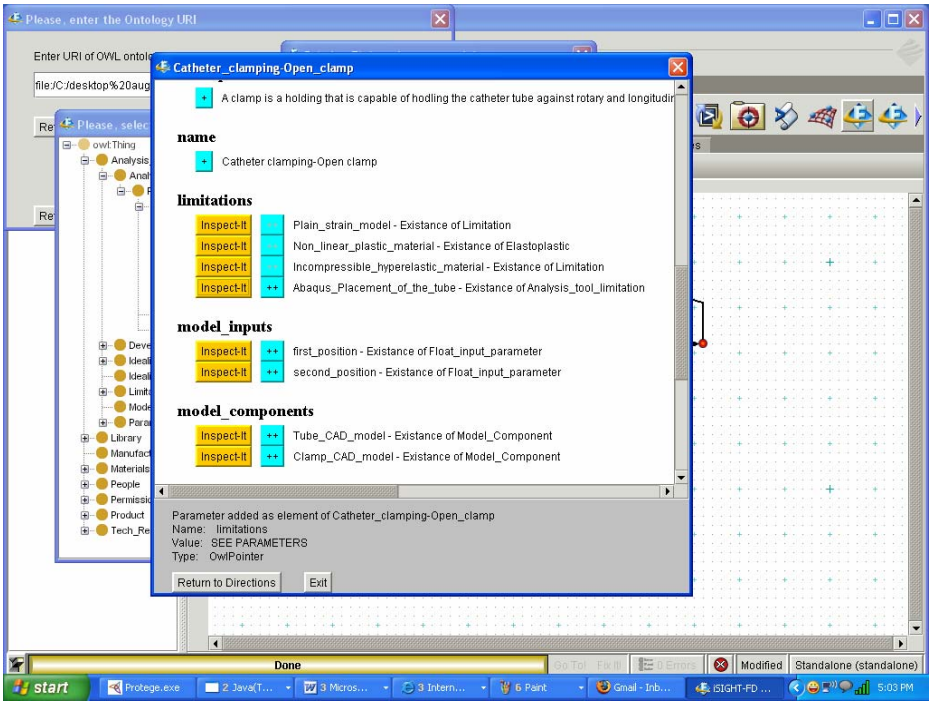
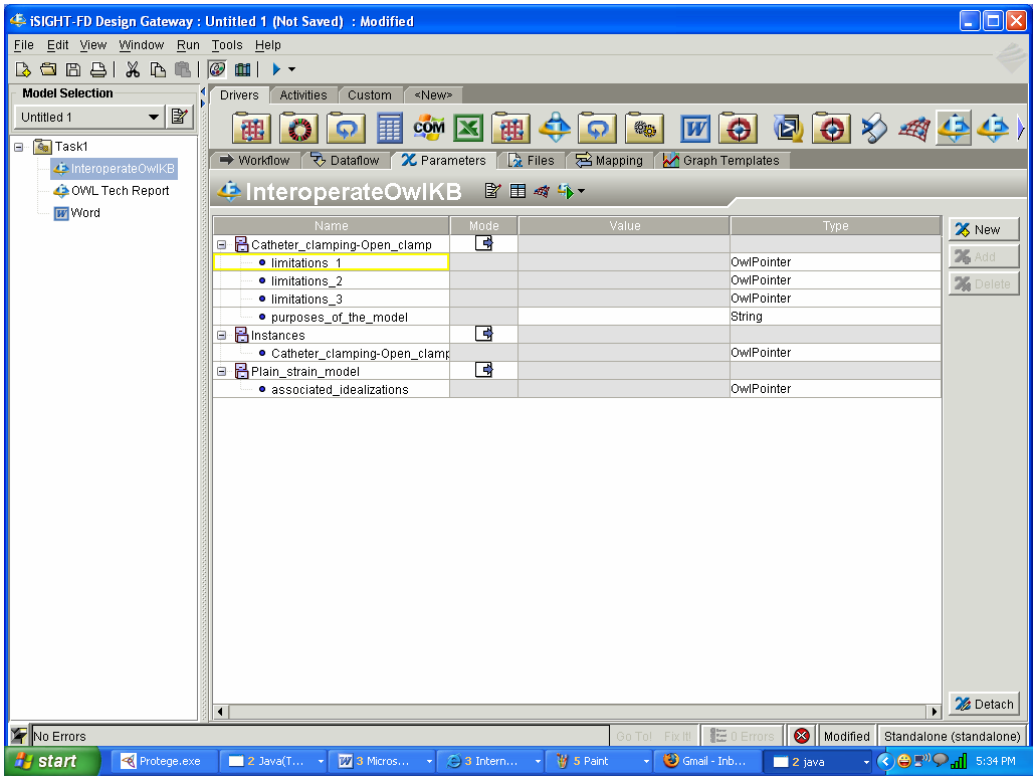


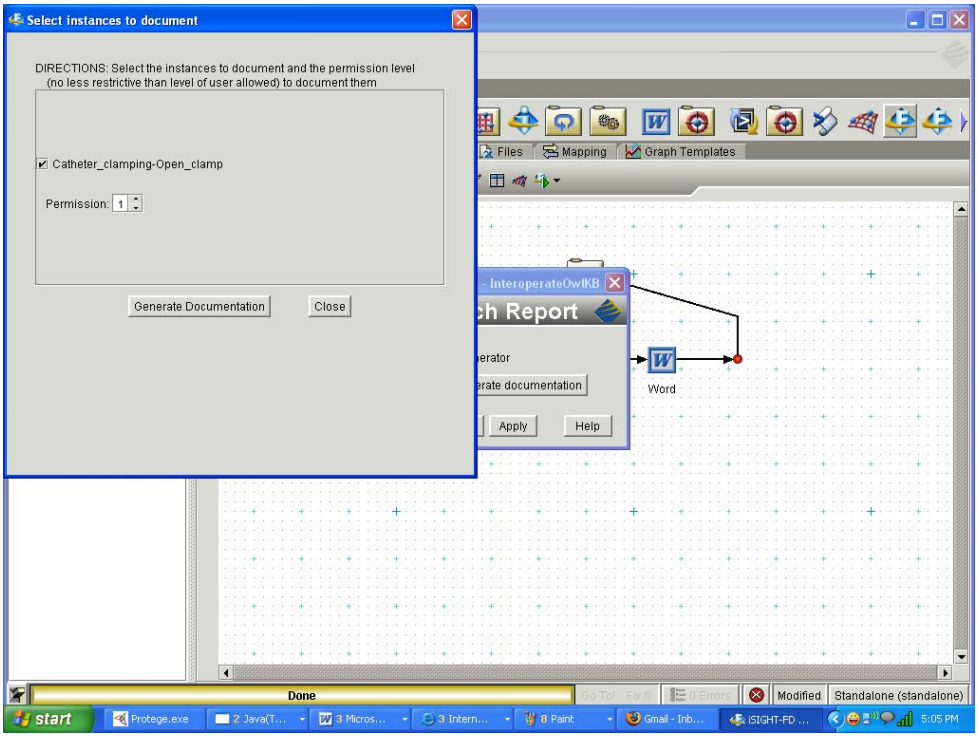

**Figure 56. Inspecting instance knowledge of the Catheter clamping-open clamp analysis model**

If the selected slot is of type Instance, then the "Inspect-It" button allows the user to look at the knowledge of the instance. It also gives the ability to add slots of this selected second instance as iSIGHT-FD parameters. The "++" button allows the user to

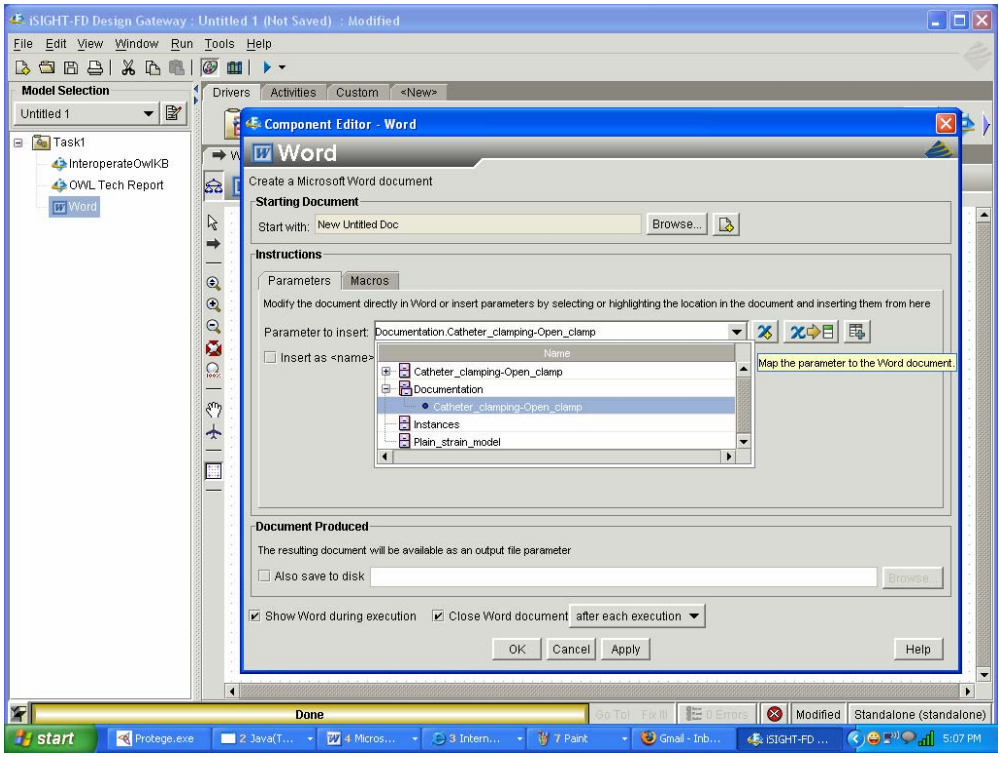
make the instance as an OWL instance pointer so that the user can view the knowledge within product development environment.



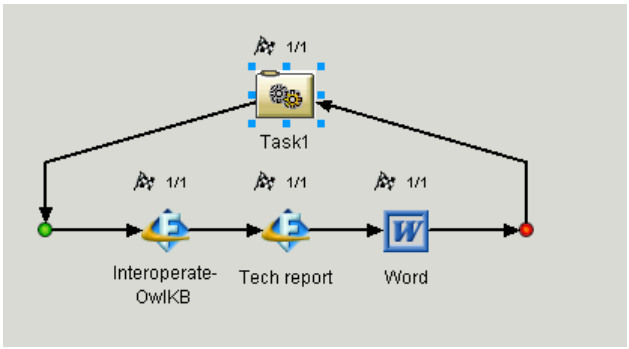

**a) Selected instances represented as "Owl Pointers"**



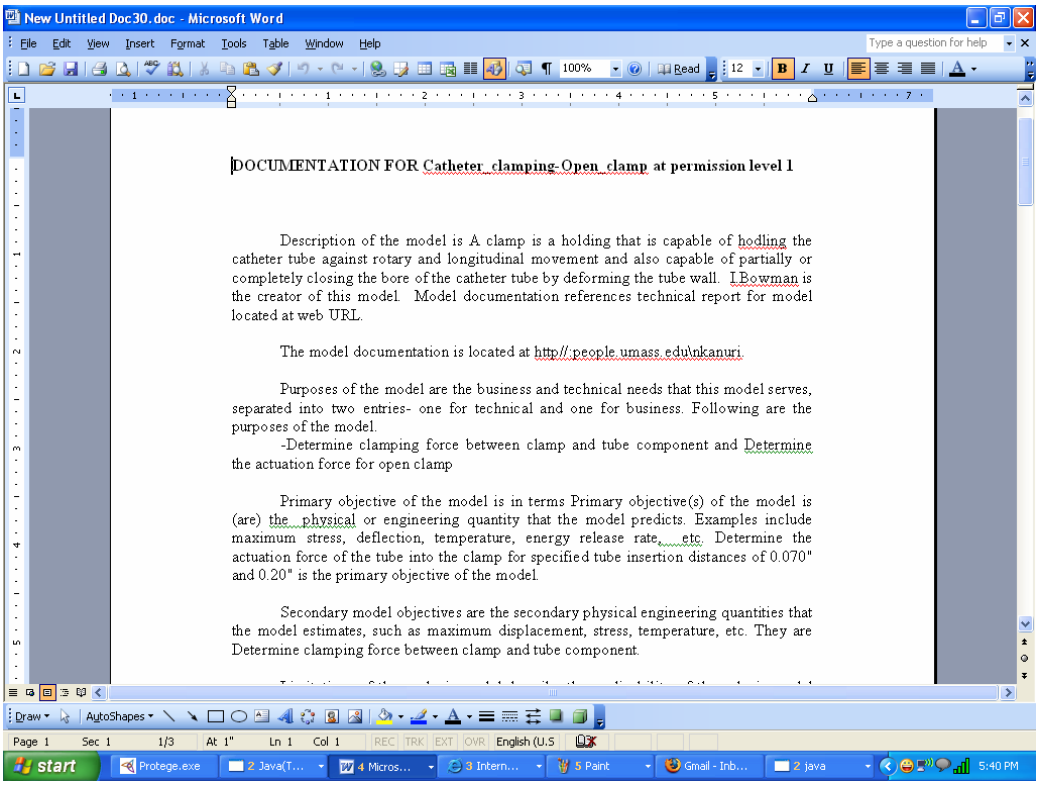

**b) Technical report generation component**




**c) Configuring word component**




**d) Running the components**




**e) Generated Technical report**

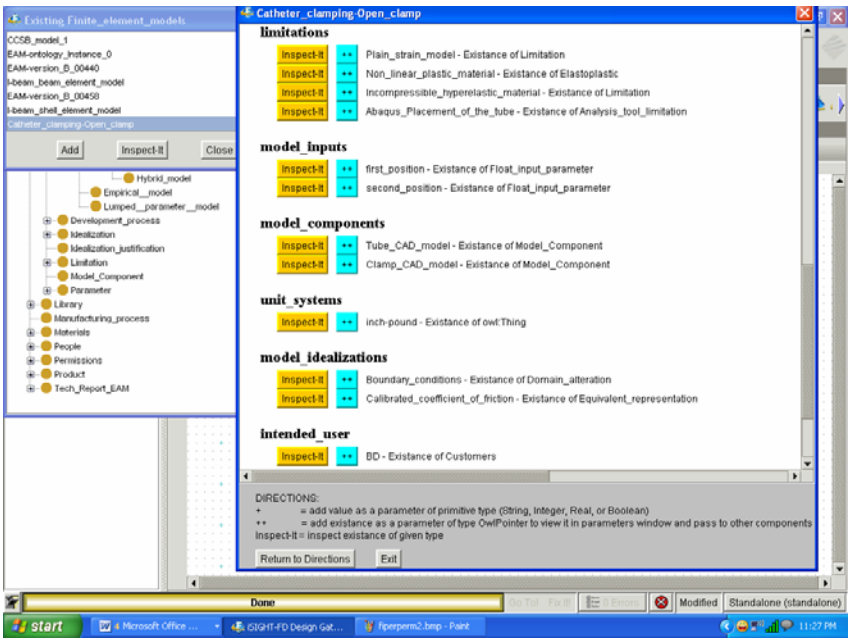
**Figure 57. Generating technical reports for multiple instances**

Figure 57 shows how the instance can be selected and used to generate technical report. The OWL instance pointers can be used to inspect the knowledge and also pass the parameters to other iSIGHT-FD components to generate secured technical reports.

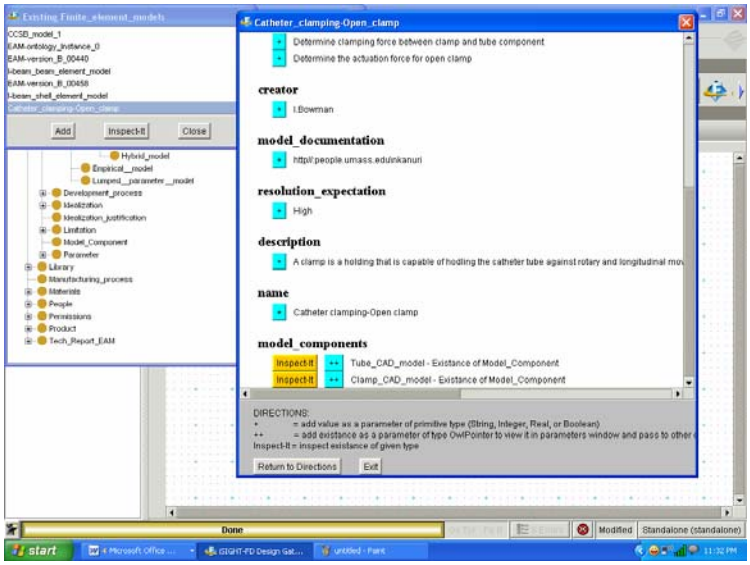
## 11.4 Secured sharing of catheter clamp analysis modeling knowledge using iSIGHT-FD

For secured sharing of analysis modeling knowledge method, permissions have been introduced for accessing various technical reports, while at the same time giving privileges for the user based on his/her permission to access the knowledge. Details have been discussed in Section 5. Permission based technical reports for catheter clamp analysis modeling knowledge will now be discussed.

Figure 58 shows the difference in accessing the instance knowledge based on user permission for the catheter clamping-open clamp instance.




**(a) Accessing instance knowledge without secured knowledge sharing**




**(b). Accessing instance knowledge with secured knowledge sharing**

**Figure 58. Accessing knowledge of catheter clamping-open clamp instance**

From Figure 58, it can be seen that knowledge and shareability is based on the user permission. The user with highest permission (full accessibility to the knowledge base) can see all the knowledge associated with an ontology instance whereas as the user with less accessibility can only see parts of the knowledge base. Hence the user with highest permission can see knowledge associated with the whole catheter clamping- open clamp instance including limitations, idealizations, inputs, outputs etc., whereas the user with low permission can only see parts of the knowledge. Secured technical reports for parts of the knowledge can also be generated for catheter clamping-open clamp instance as discussed in Section 5 and Appendix A.

## 11.5 Validation

Catheter clamping-open clamp instance which has been developed based on an industrial application obtained from BD Medical Systems served as a means to create an

EAM ontology instance and validate the tech report and secured knowledge sharing components. As discussed above, the developed instance can be used to generate technical reports, inspect knowledge associated with the instance and also permit secured knowledge sharing. Also parts of the knowledge can be securely viewed and used to generate partial technical reports based on user permission level. Multiple technical reports of instances can also be generated using secured knowledge sharing in a web based environment like iSIGHT-FD.

iSIGHT-FD provides visual and flexible tools to set up an automated plan to thoroughly explore the design space and find optimum and product solutions. Methods can be developed to grab knowledge from the catheter clamping-open clamp instance as input/output parameters and used in a web based environment like iSIGHT-FD for driving applications like analysis, optimization, design of experiments, design of six sigma etc.

One such method can be modifying the design of the clamp for more effective clamping of the tube. For example, with an alternate material which is blend of Cycoloy and polycarbonate the gap at the inside surface of the tube which causes leakage can be reduced by 10%. But change in material might affect other design parameters like actuation force, clamping force and coefficient of friction. iSIGHT-FD's design exploration tools can help the user decide upon the change of the material used for clamp design . Input parameters for this tool would be the clamp design model, tube properties, the material used for the clamp design and the insertion position. The outputs of the model would be actuation force, clamping force and the gap at the inside surface of the tube. Using the knowledge obtained from the input parameters and the values calculated

by the design exploration tool as the output parameters, the effectiveness of material change for the clamp design can be evaluated. Similarly the design exploration tool can further be extended to modify the clamp design to reduce the potential for yielding and fatigue due to multiple user activations. The design exploration tool would result in deciding the critical parameters that would affect the design of the clamp.

Also, iSIGHT-FD's web based environment helps the user to visualize the design results in real time. A second method would be mapping the knowledge base parameters to iSIGHT integrated VCollab environment where the user can easily share and view the CAD/CAM/CAE simulation results. The CAE simulation knowledge capture of VCollab helps in improved decision making during design reviews, i.e., it helps improving innovation through visual collaboration. For this method, the input parameters can be design model of the clamp (Inner diameter, outer diameter etc.) and the insertion position. The user can then visualize the gap in the inner side of the tube and hence design in a way to reduce the leakage caused by the gap.

Grabbing the knowledge associated with an analysis model and being able to parameterize the knowledge as input/output parameters in a web based environment serves as an excellent way to develop methods which further operate on this knowledge to automate the design process with many design alternatives, resulting in better and more reliable products. The catheter clamping-open clamp instance served as an excellent example to validate the EAM ontology, the tech report and secured knowledge sharing methods. It further demonstrated the effect of interoperability of analysis modeling knowledge in automating the processes while reducing the cost and processing times.

Such advancements help industries reduce the product life-cycle time and cost to gain and maintain a competitive edge.

# CHAPTER 12

## SUMMARY

Within the knowledge modeling community the use of ontologies in the construction of knowledge intensive systems is now widespread. EAM ontology has been developed for representing and sharing engineering analysis modeling (EAM) knowledge. The development of an EAM ontology is knowledge based activity and there are no current systems that provide efficient mechanisms to capture and share this knowledge in a computational environment. The EAM ontology has been instantiated to develop ONTEAM. Methods have been developed that operate on the EAM knowledge stored in ONTEAM to enable sharing, reuse and interoperability of analysis modeling knowledge in a product development environment with fine grained control of knowledge sharing.

One such method is the automatic technical report generation method which allows the user to generate a flat technical report that describes the modeling knowledge associated with an analysis model. The technical report method generates meaningful sentences associated with knowledge instances. A parallel class of "Tech report EAM" has been developed to create meaningful sentences for the generation of technical report. This method has been extended to develop functionality for knowledge inspection, using Protégé look and feel environment, and sharing knowledge in the form of parameters so that pieces of knowledge from the ontological knowledge base can be passed as parameters in a product development environment to be used by other components.

The secured sharing of ontological knowledge has been made possible by implementing "Permissions" on the knowledge. A parallel class of permissions has been created using JAVA programming, where the permissions associated with the knowledge have been stored. The permission associated with the knowledge has been compared to the user permission to grant privilege for the user either for viewing or sharing the knowledge. Permission based knowledge control has been implemented at the slot and the instance level hence maintaining a fine grained control over sharing of analysis modeling knowledge in a product development environment.

The knowledge format for ontological knowledge base has been changed to standard OWL representation. OWL files support format for effective sharing, reuse and interoperability of analysis modeling knowledge. This format extends the functionality of working with standard tools in a product development environment. The catheter clamping-Open clamp analysis modeling knowledge obtained from an industrial application of BD Medical systems has been used test the OWL EAM ontology, technical report component and the secured knowledge sharing component to understand the advantage of using abstract modeling knowledge in secured access/viewing and sharing of analysis modeling knowledge in a distributed and collaborative environment.

ANSYS, a commercial analysis tool has been componentized in a product development environment. This component helps to run analysis models and share the inputs/ outputs associated with the model. The optimization component in iSIGHT-FD has been used to run the optimization process. A test bed application, I-beam model, has been taken to demonstrate the power of knowledge sharing in developing automated analysis and optimization systems. Two analysis models for I-beam, the beam element

model and the shell element model have been developed. The shell element model is the highest accuracy model as it can capture the eccentricity of loading. The beam element model cannot capture the eccentricity of loading. It is very much accurate as the shell element model for certain length/height and length/width ratios depending on the accuracy expectation of the optimization problem. An intelligent decision making tool helps in optimizing the volume of the I-beam using both the beam and the shell element models. This tool intelligently shifts between the beam and the shell element models for optimizing the I-Beam. The decision for shifting between models is based on the length/height, length/width ratios and accuracy expectation.

The results of the optimization process, i.e., variables, constraints and the objective function are written back to the ontology. These results get stored as instances based on various I-beam optimization problems being solved and will be helpful to decide the best model for a new I-beam optimization problem. For example, for modeling a new I-beam, the user can refer to the available I-beam models to gain an understanding of the design parameters required to model it. The user can also query a particular class of models based on criteria like variables, constraints and objective function.

Writing the knowledge associated with the optimization process back to ontology also helps the user to obtain prior understanding of the model and relax the necessary constraints and objective function for obtaining the design parameters and vice-versa. This also reduces the processing cost and time involved in developing new models.

Ontological knowledge has been integrated to EnCapta, a CAD integrated environment which links the parameters of CAD models directly to the knowledge base. Hence changes to engineering models can be captured and managed as objects and linked

to the relevant geometry in the CAD models. This supports distributed and collaborative sharing of CAD integrated knowledge throughout the enterprise.

EAM ontology has been developed to be applicable to all engineering analysis models. To facilitate reuse, adaptation and interoperability of engineering analysis models in a product development environment, a computational knowledge base system ONTEAM has been developed. Methods which work on the knowledge existing in ONTEAM help in knowledge exchange, inspection and sharing in a product development environment. When the knowledge gets integrated to a product development environment using a CAD integrated environment like EnCapta, the power of the developed methods and automated systems can be demonstrated. While the methods aid in secured sharing and reusability of analysis modeling knowledge, the developed automated systems help in solving optimization problems within time and complexity constraints using intelligent decision making tools. The results obtained from an optimization process using are written back to ontology to facilitate reusability of existing models. The uniqueness of such process lies in the fact that the user can use the associated modeling knowledge to query the available models, relax the constraints based on the results obtained from the available models and hence reduce the processing time and cost involved in developing new models. Together the ontological knowledge coupled with the developed methods form a powerful tool for developing knowledge integrated automated systems in a product development environment.

# CHAPTER 13

## FUTURE WORK

Interoperability and reusability of engineering analysis modeling knowledge helps in generating automatic customized technical reports for industrial models. The functionality has been extended for secured sharing and inspection of knowledge in a product development environment. Knowledge from an ontological knowledge base has been brought into a product development environment, iSIGHT-FD to run the processes of analysis and optimization. The intelligent decision making tool that has been developed makes the optimization process automated in the sense of selecting the right analysis model for a given design optimization state.

Currently the modeling knowledge from the ontological knowledge base is brought into a product development environment as input parameters of the data type's real, string and Boolean. Work has been done by one of my colleagues to integrate an I-beam model in a CAD-environment where the input geometric parameters are directly linked to the CAD design parameters using EnCapta. Pro-Engineer has been integrated to EnCapta and the ontological knowledge base has been instantiated in EnCapta. This allows import/export of captured knowledge using the EnCapta tool into/from OWL database. Future work might involve using the knowledge from EnCapta directly in the product development environment so that information can be passed into/ out of the ontological knowledge base through the product development environment. Results from the optimization process, for example, an I-beam optimization process, can then be

directly written to the knowledge base and be used to generate multiple technical reports for the results of an optimization process.

Methods can be developed which include constraints for reasoning on an ontological knowledge base. This means developing a way to include the first order logic in ontologies. This tool can allow putting some restrictions on the knowledge bases and also for validating the acquired knowledge. For example, for the I-beam optimization problem one of the design limitations is that the length/width or the length/height ratio cannot be less than 5 as it makes the computation cost very high and the associated finite element model takes a lot of memory to run the model. This constraint can be included in the ontology as a first order logic statement which checks the validity of the knowledge being instantiated.

Development of intelligent decision making tool is the first step in automating an optimization process in terms of selecting the right analysis model. Intelligent tools can further be developed with functionalities which automate the analysis and optimization processes. For example, if an analysis model does not have a solution for an optimization problem, an intelligent tool can be developed which modifies an optimization problem based on certain constraints for which the analysis model can find a solution. Hence interoperability and reuse of analysis modeling knowledge can support sharing and using of critical modeling knowledge in an efficient way.

# APPENDIX A

## CUSTOMIZED GENERATION OF TECHNICAL REPORTS

This appendix details the customized generation of technical reports based on users permission and also shows how multiple technical reports can be appended to form a single technical report. Figure 59, Figure 60, Figure 61, Figure 62, Figure 63, Figure 64, Figure 65 and Figure 66 show the selection of the appropriate instances for which the technical report has to be generated. For example for the username and the permission specified, the user has a permission of 1 for the generation of technical report for all the instances.

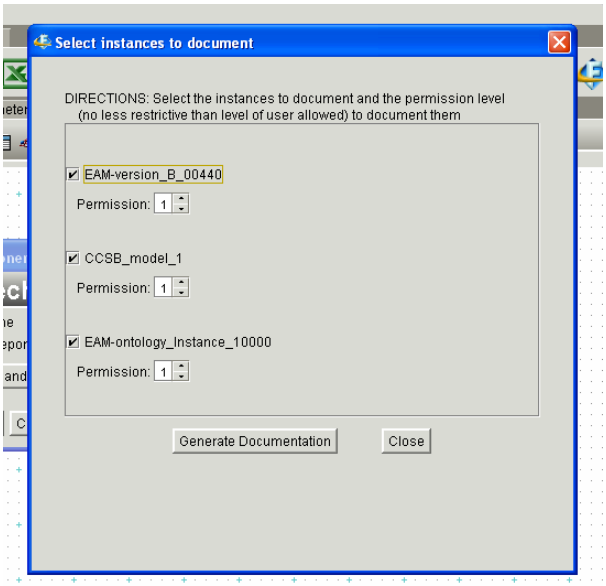



**Figure 59. Permission for the instances for the generation of technical reports**

Permission level of 1 is the highest permission level which gives user the privilege to see all the available knowledge associated with the instance. The user with highest permission level can also generate technical reports for lower permission levels,

like for values 2 and 3. Figure 60 shows the selection of permission level by the user who has the permission level 1. The user can also select the permission level for the technical report for permission level 2 and 3 as he/she has the highest permission level, i.e., 1.

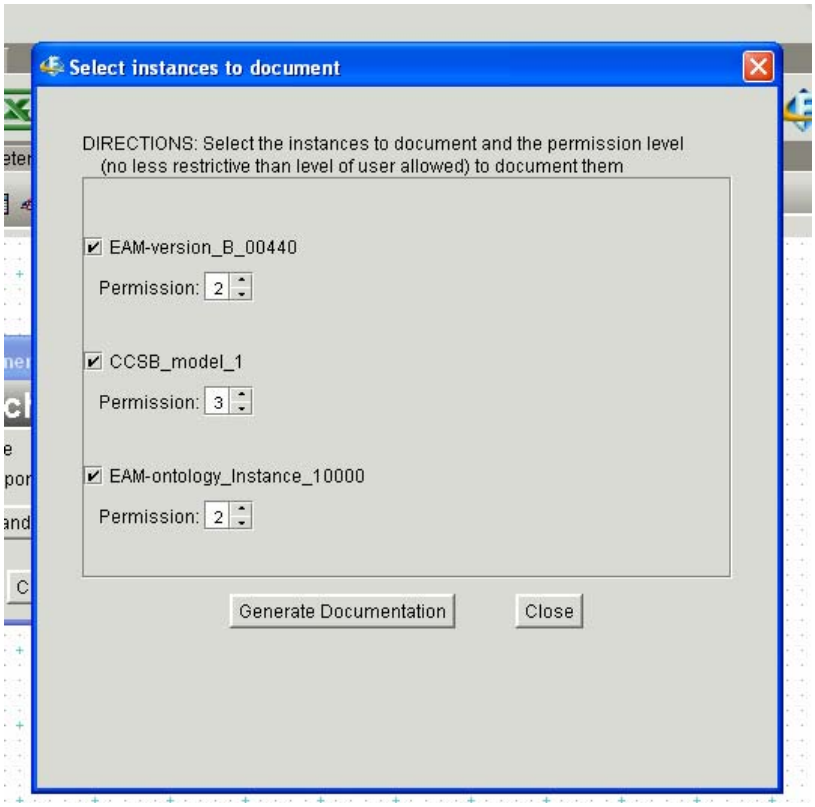



**Figure 60. Permission levels 2,3 and 2 selected by the user who has permission level of 1**

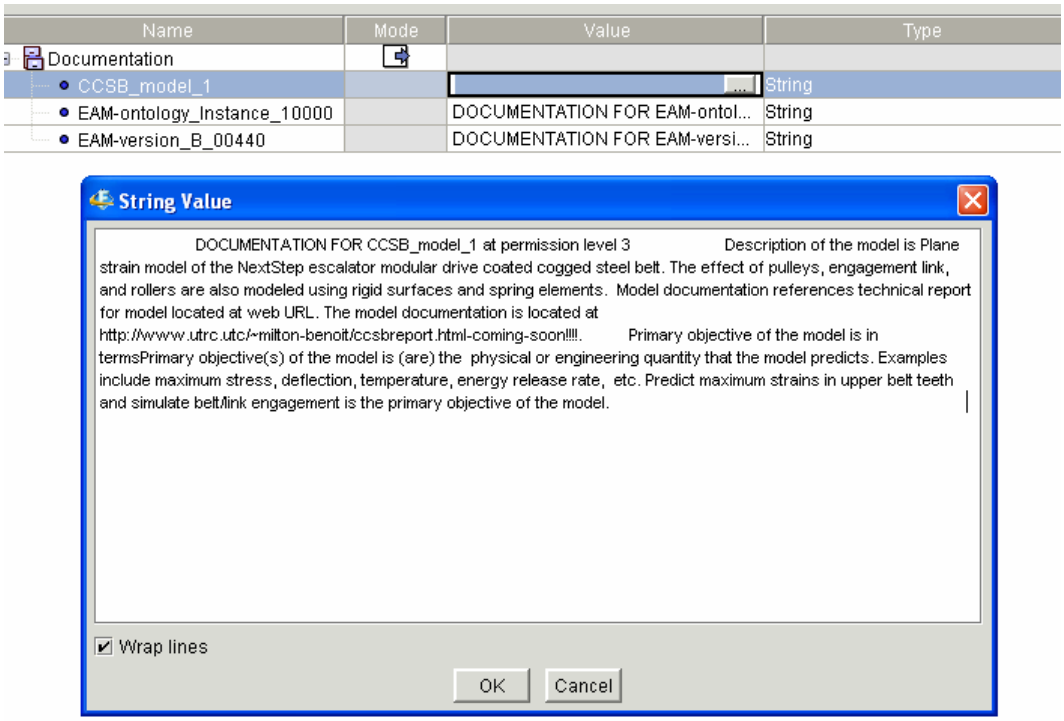



**Figure 61. Technical report for CCSB_model instance generated with permission level 3**

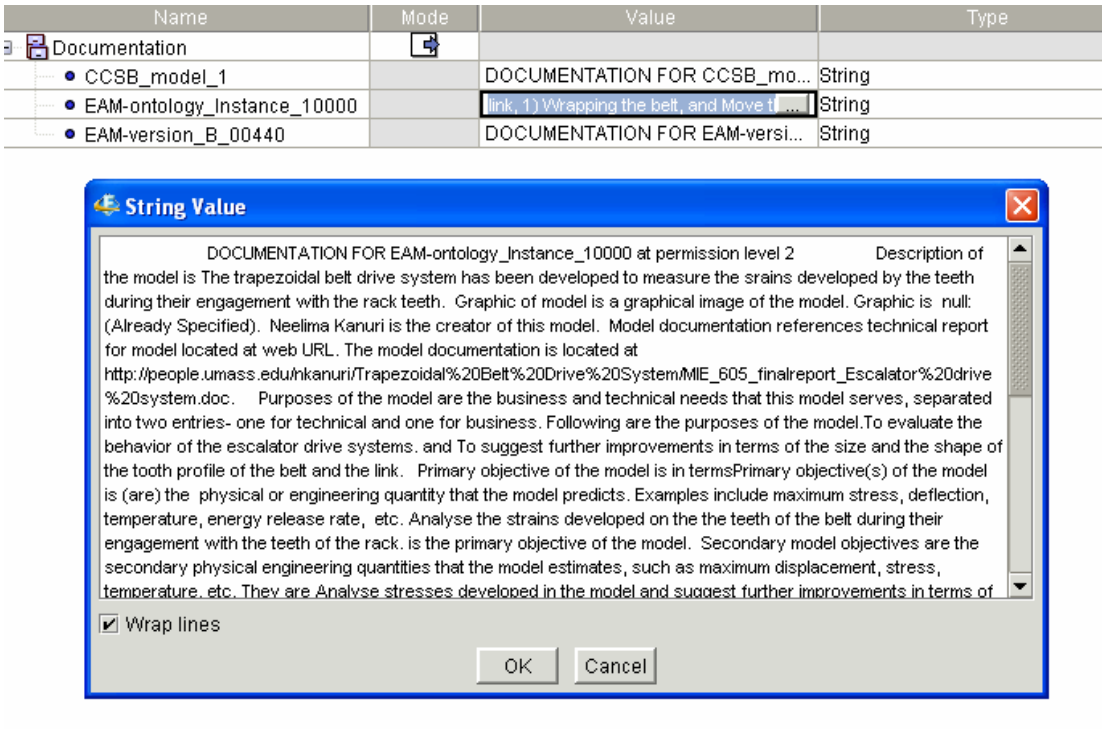



**Figure 62. Technical report for EAM-ontology_instance_1000 instance generated with permission level 2**

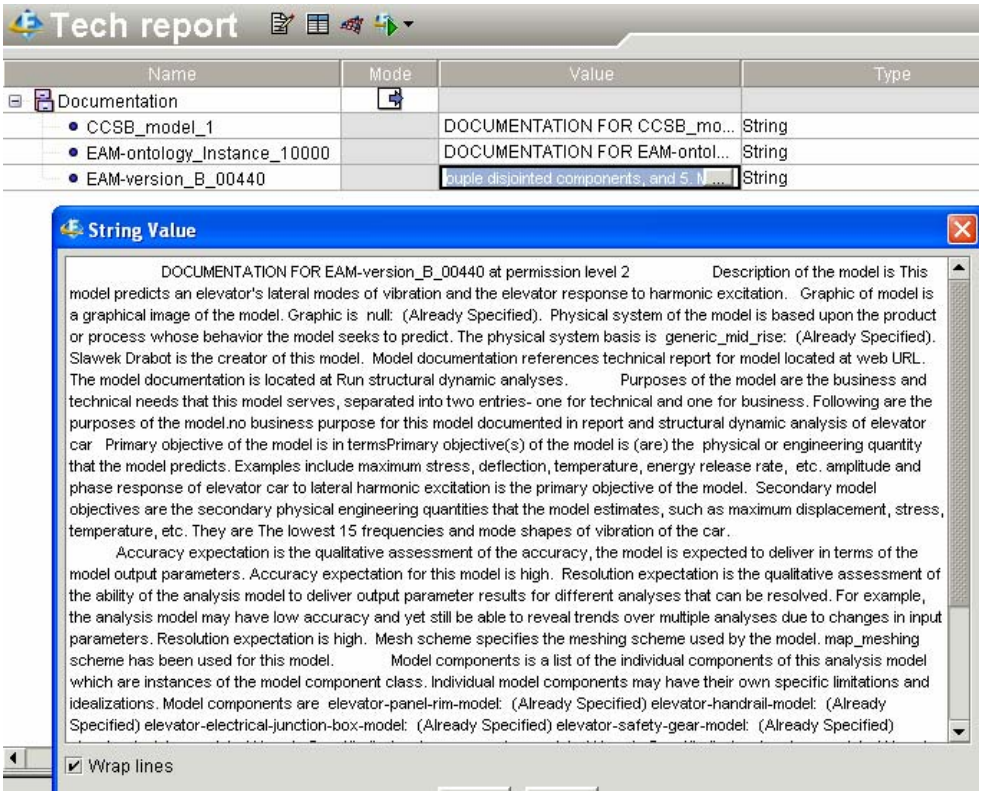



**Figure 63. Technical report for EAM-version_B_00440 instance generated with permission level 2**

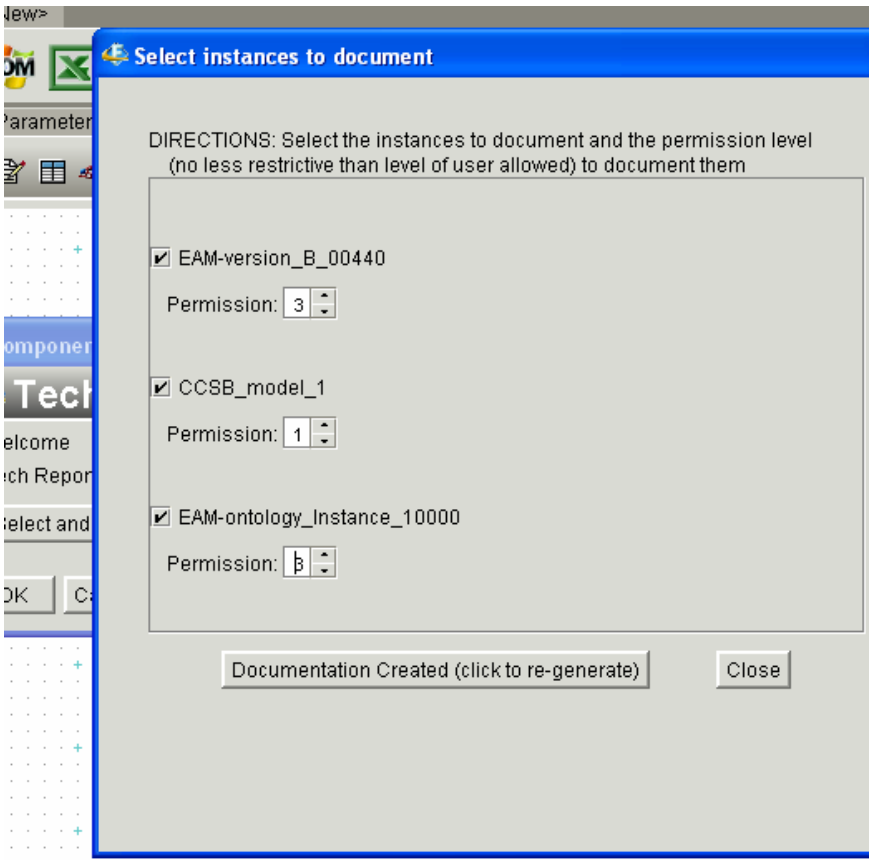



**Figure 64. Permission levels 3, 1 and 3 selected by the user who has permission level of 1**

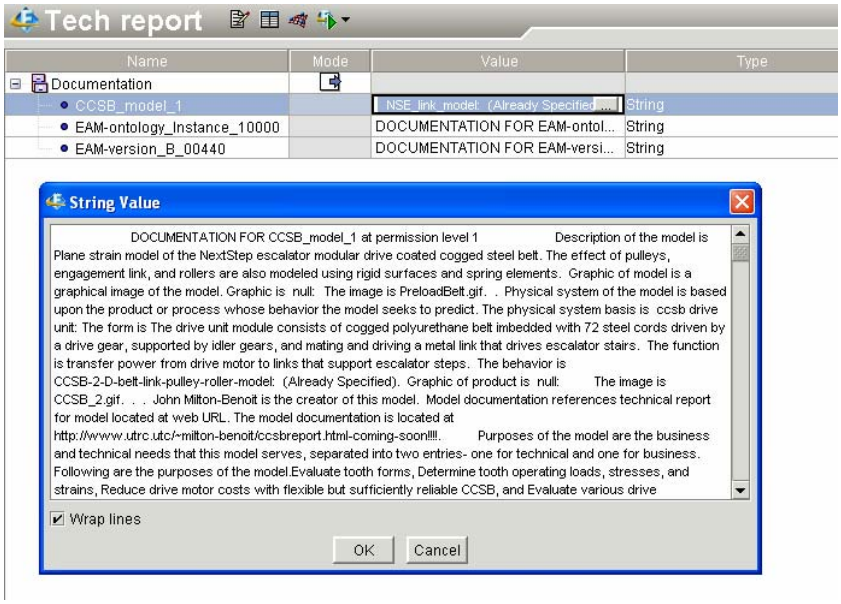



**Figure 65. Technical report for CCSB_model_1 instance generated with permission level 1**

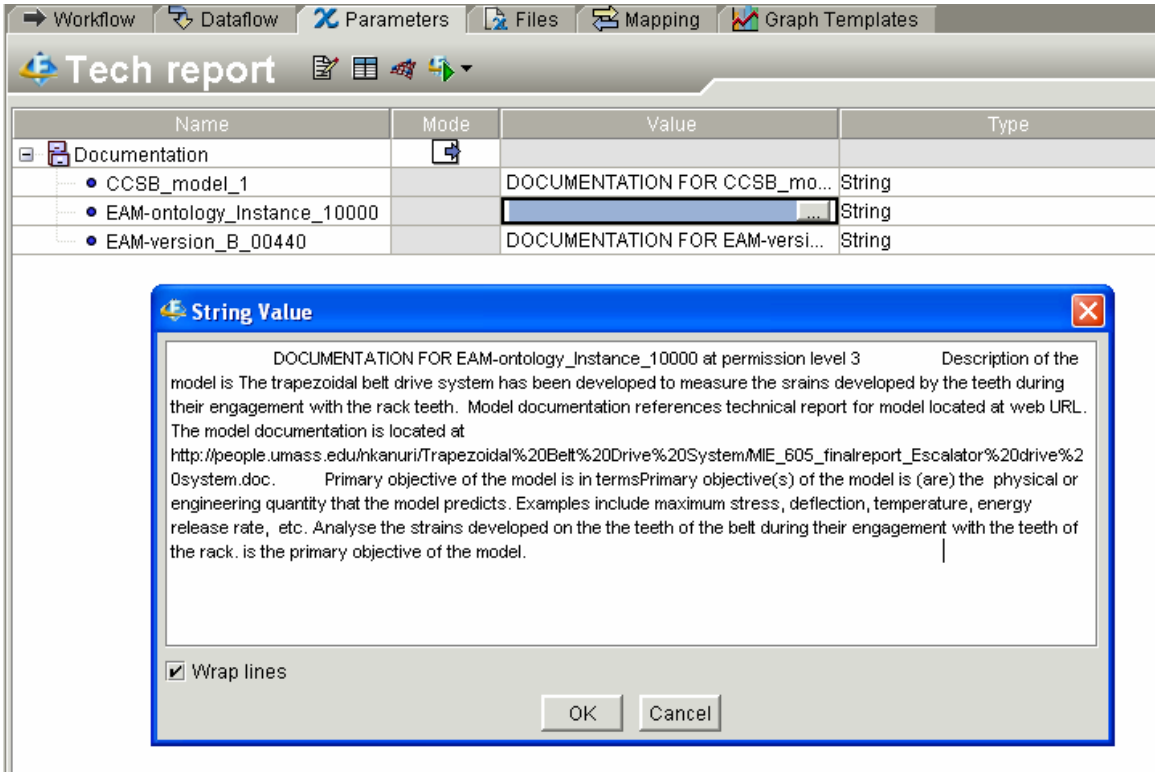



**Figure 66. Technical report for EAM-ontology_instance_1000 instance generated with permission level 2**

Figure 67, Figure 68 and Figure 69 show appending of multiple technical reports to a single report using iSIGHT-FD's Word component.

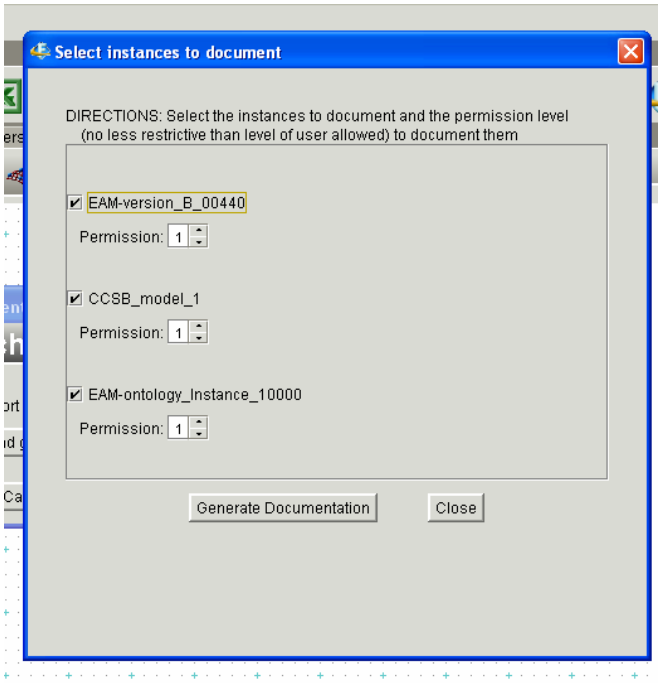



**Figure 67. Multiple instances for generating technical reports**

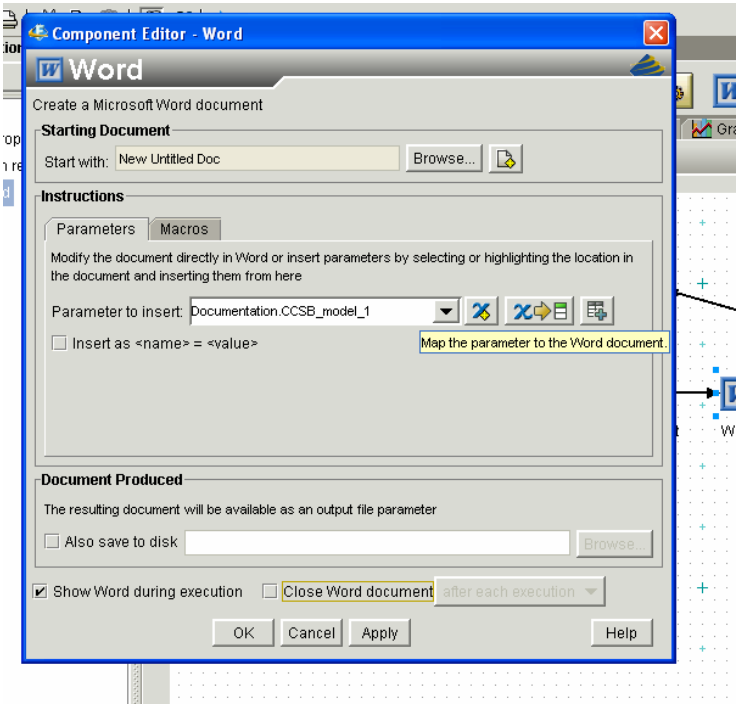



**Figure 68. Selecting the CCSB-model-1 instance for generating technical report**

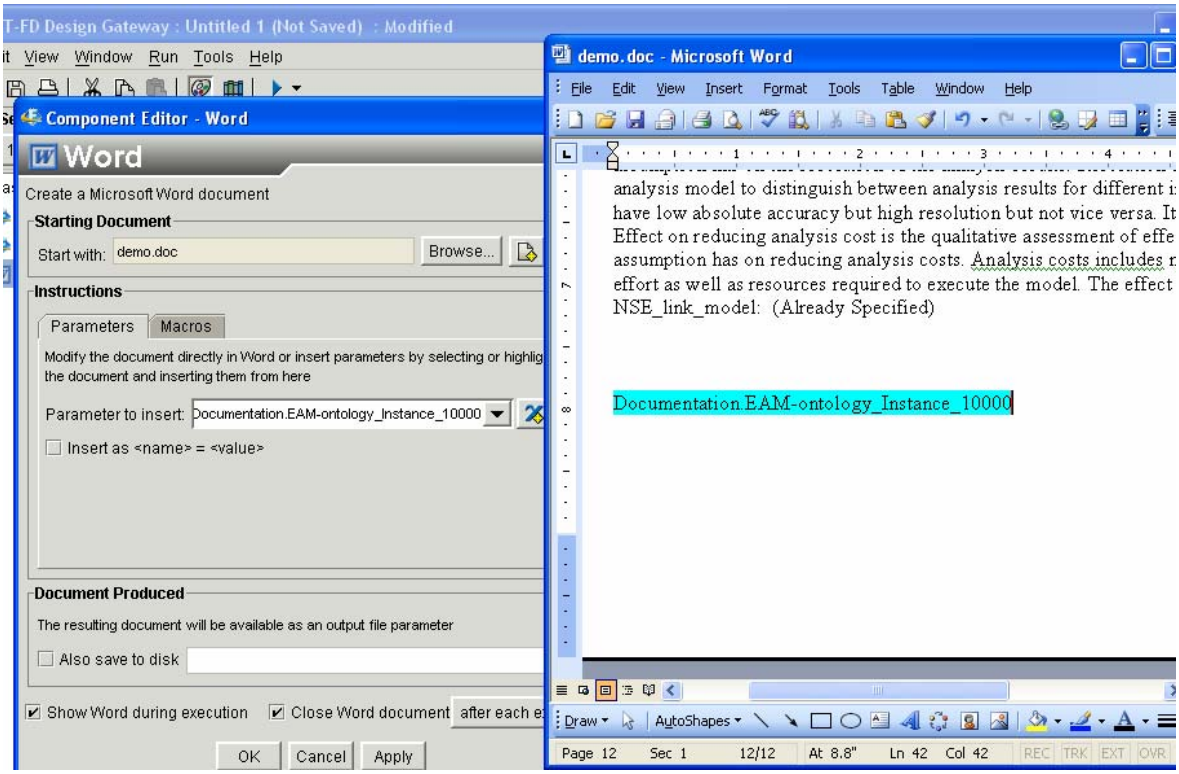



**Figure 69. Appending multiple instances for the technical report**

## APPENDIX B

## ANSYS COMMAND FILES FOR BEAM AND SHELL ELEMENT MODELS

### Beam element model

```
/output, TERM

/graphics,power
/triad,lbot
/auto
/view,1,1,1,1
/eshape,1

/output, TERM
h=  4
w=  20
t=  0.5
l= 100.0

/prep7
et,1,188
keyopt,1,1,1
r,1,
mp,ex  ,1,10e7
mp,nuxy,1,0.3
mp,dens,1,.1/386.1
sectype,1,beam,i,IBEAM
secoffset,cent
secdata,w,w,h,t,t,t

k,1,0,0,0
k,2,l,0,0
k,3,l/2,5,0
l,1,2

latt,1,1,1,,3,,1
esize,1

lmesh,all
finish

/solu
```

```
antype,static
pstres,on

dk,1,all
fk,2,fy,1000

allsel,all
ALLSEL,ALL

SOLVE
FINISH
/POST1

nsort,U,SUM
*get,U,sort,,max

/output, TERM

etable,evol,volu
ssum

*get,vtot,ssum,,item,evol

*GET,smax,SECR,ALL,S,X,MAX
/output, TERM
*DIM,VALUE,,3,3
*VFILL,VALUE(1,1),DATA,U,vtot,smax



/OUTPUT,'outputbeam','txt','C:\neelima project\Ibeam'
*VWRITE,Value(1,1)
(F15.8)

/output, TERM
finish
```

**Shell element model**

```
/output,TERM


THICK= 0.7

Height= 12.0
```

```
Width= 8.0
len=          100.0

h=Height/2
w=Width/2
alpha=0.2!meshing
eccen=  1.0
size=alpha*w
locon=-(eccen*w)

/graphics,power
/triad,lbot
/auto
/view,1,1,1,1
/eshape,1
/output,TERM

/prep7
et,1,181
keyopt,1,3,2
r,1,THICK
mp,ex  ,1,10e6
mp,nuxy,1,0.3
mp,dens,1,.1/386.1

k,1,0,0,0
k,2,0,h-THICK/2,0
k,3,w,h-THICK/2,0
k,4,-w,h-THICK/2,0
l,1,2
l,2,3
l,2,4
lsymm,y,all

k,10,0,0,len

l,1,10
allsel,all
adrag,1,,,,,,7
adrag,2,,,,,,7
adrag,3,,,,,,7
adrag,4,,,,,,7
adrag,5,,,,,,7
adrag,6,,,,,,7

nummrg,kp
```

```
FLST,2,6,5,ORDE,2
FITEM,2,1
FITEM,2,-6
FLST,2,6,5,ORDE,2
FITEM,2,1
FITEM,2,-6
AESIZE,P51X,alpha*w,
FLST,5,6,5,ORDE,2
FITEM,5,1
FITEM,5,-6
CM,_Y,AREA
ASEL, , , ,P51X
CM,_Y1,AREA
CHKMSH,'AREA'
CMSEL,S,_Y
!*
MSHKEY,1
AMESH,_Y1
MSHKEY,0
!*
CMDELE,_Y
CMDELE,_Y1
CMDELE,_Y2
finish

/solu
antype,static
pstres,on
/output,TERM

lsel,s,loc,z,0
dl,all,,all

!added
seltol,0.0002
NSEL,S,LOC,X,locon
NLIST,ALL, , , ,NODE,NODE,NODE
NSEL,R,LOC,Y,(Height-THICK)/2
NSEL,R,LOC,Z,(len)
f,all,fy,1000


ALLSEL,ALL

SOLVE
```

```
FINISH
/POST1

nsort,U,SUM
*get,U,sort,,max

etable,evol,volu
ssum

*get,vtot,ssum,,item,evol

nsort,S,EQV

*GET,smax,SORT,0,MAX

*DIM,VALUE,,3,3
*VFILL,VALUE(1,1),DATA,U,vtot,smax


/OUTPUT,'SHELL OUTPUT','out','C:\neelima project\Ibeam\Static analysis for beam'
*VWRITE,Value(1,1)
(F15.8)

/output, TERM
finish
```

## BIBLIOGRAPHY


Borst et al, 1994. Physical systems ontology, In Working Papers European Conference on Artificial Intelligence ECAI'94 Workshop on Implemented Ontologies, (N.J.I. Mars, Ed.), pp. 47-80. Amsterdam, August: ECCAI.

Borst et al, 1995. The PhysSys ontology for physical systems, in Working Papers of the Ninth International Workshop on Qualitative Reasoning QR'95, (B. Bredeweg, Ed.) pp. 11-21. University of Amsterdam, May.

Borst, 1997. Construction of Engineering Ontologies. PhD report, University of Twente, Enschede.

Brown et al, 1989. Design Problem Solving: Knowledge Structures and Control Strategies". Research Notes in Artificial Intelligence Series, Morgan Kaufmann Publishers, Inc.

Chandrasekaran, 1986. Generic tasks for knowledge-based reasoning: High level building blocks for expert systems design, IEEE Expert, pp-1(3),23-30

Dym et al, 1991. "Toward the integration of knowledge for engineering modeling and computation", Engineering with computers,7, pp.209-224.

Edward Feigenbaum and JoshuaLederberg, http://www.camis.stanford.edu/

http://wwwcamis.stanford.edu/research/history.html

Encapta Software, 2006, http://www.vistagy.com/products/encapta.html

Finn et al, 1992. An Intelligent Mathematical Modeling Assistant for Analysis of Physical Systems, Proc. ASME 1992 Computers in Engineering Conference and Exposition, Vol. 2, pp. 69-74, August: ASME.

Fiper, Documentation Version 1.6,2005,www.engenious.com.Cary,NC

Gennari et al. Reuse for Knowledge-Based Systems and CORBA Components, Section on Medical Informatics, Stanford University School of Medicine Stanford, CA 94305-5479, U.S.A

Gero, 1985. Knowledge Engineering in Computer-Aided Design, North-Holland.

Gruber, 1993. "A Translation approach to portable ontologies, "Knowledge Acquisition, 5(2), pp.199-220

Gruber, 1995. Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies, 43:907-928.

Grosse et al, 2005. Ontologies for supporting engineering analysis models, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 19{1}.

Hong et al, 1986. IEEE Computer,Special issue on Expert Systems in Engineering, Vol. 19, No. 7.

Hunten et al, 1997. CAD/FEA Integration with STEP AP209 Technology and Implementation.

IEEE, 1990. Institute of Electrical and Electronics Engineers IEEE standard computer dictionary: A compilation of IEEE standard computer glossaries, New York, NY.

Kanuri et al, 2005."Ontologies and fine grained control over ssharing of engineering modeling knowledge in a web based engineering environment", American society of Mechanical engineers, November, Florida, US

Kanuri,N., 2006. Trapezoidal belt drive system, MIE 605 final project, Department of Mechanical and Industrial engineering, University of Massachusetts, Amherst, http://people.umass.edu/nkanuri/

Klyne et al, 2004."Composite Capability/Preference Profiles (CC/PP)," W3C consortium.

Moka, 1998. http://www.kbe.coventry.ac.uk/moka/kbe.htm

Musen, 1989. Automated Generation of Model-based knowledge acquisition tools, Pitman Publishing.

Musen, 2000. Ontology-Oriented Design and Programming, in Knowledge Engineering and Agent Technology (Cuena, J., Demazeau, Y., Garcia, A, and Treur, J., Eds.), Amsterdam: IOS Press.

NIST, 1999. Interoperability cosy analysis of the US automotive supply chain, NIST strategic planning and Economic Assessment Office, Planning Report #99-1, available online at http://www.nist.gov/director/prog-ofc/report99-1.pdf.

Noy et al, 2001. "Ontology development 101: A Guide to creating Your First Ontology," Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, Stanford University.

Object Management Group, http://www.omg.org

Paul, 2006. The use of Ontologies to support engineering design optimization, Masters Report, University of Massachusetts, Amherst,MA.

Peak et al, 1998. Integrating engineering design and Analysis using a Multi-represetation Approach. Engineering with Computers, Volume 14 No.2,93-114.

Phoenix integration, 2006, http://www.phoenix-int.com/

Protégé, 2005. Ontology editor and knowledge base frame work, http://protege.stanford.edu/

PTC, 2005. Parametric Testing Corporation, www.ptc.com.

Reed et al, 1998. An object-oriented framework for distributed computational simulation of aerospace propulsion systems, Proc. 4th USENIX Conference on Object-Oriented Technologies and Systems (COOTS), Santa Fe, New Mexico, April 27-30.

Schreiber et al, 1994.CommonKADS: A Comprehensive Methodology for KBS Development. IEEE Expert.

SCRA, 2001."ISO 10303, Step application handbook", Version 2, 5300 International Boulevard, North Charleston, SC, December.19-34.

Shanbhag et al, 2001. Meta-Object Based Finite Element Analysis, Proc. ASME 2001 Design Automation Conference, Paper No. DETC2001/DAC-21062, Pittsburgh, PA, September: ASME.

Shanbhag et al, 2002. Metaobject Based Finite Element Modeling, M.S. Report, Amherst, MA: University of Massachusetts

Stanford KSL network services, 2005. http://www-ksl-svc.stanford.edu:5915/stanford

The DARPA Agent Markup Language, 2005. http://www.daml.org/

Studer et al, 1999. Knowledge Engineering: Survey and Future Directions. In 5th German Conf. on Knowledge-based Systems, Wuerzburg, Germany.

Walsh et al, 1997."A Technical Introduction to XML," Arbor Text, Inc.

Walther et al, 1992. Plug and Play: Construction of task-specific expert-system shells using sharable context ontologies. AAAI Workshop on Knowledge Representation Aspects of Knowledge Acquisition, San Jose, CA, 191-198.

Wielinga et al, 1993. Towards a unification of knowledge modelling approaches. In J.M. David, J.P.Krivine & R. Simmons, Eds. Second Generation Expert Systems, pp. 299-335. Berlin: Springer-Verlag.

Wujek et al, 2002. A Distributed, Component-Based Integration Environment For Multidisciplinary Optimal and Quality Design, 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, September.