

# **A VIRTUAL ENVIRONMENT FOR LEGACY MECHANICAL SYSTEMS ENGINEERING**

by

Suraj R. Musuvathy

A thesis submitted to the faculty of  
The University of Utah  
in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Science

School of Computing

The University of Utah

December 2006

Copyright © Suraj R. Musuvathy 2006

All Rights Reserved

THE UNIVERSITY OF UTAH GRADUATE SCHOOL

## SUPERVISORY COMMITTEE APPROVAL

of a thesis submitted by

Suraj R. Musuvathy

This thesis has been read by each member of the following supervisory committee and by majority vote has been found to be satisfactory.

---

---

Chair: Elaine Cohen

---

---

Richard F. Riesenfeld

---

---

Peter Willemsen

THE UNIVERSITY OF UTAH GRADUATE SCHOOL

**FINAL READING APPROVAL**

To the Graduate Council of the University of Utah:

I have read the thesis of Suraj R. Musuvathy in its final form and have found that (1) its format, citations, and bibliographic style are consistent and acceptable; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the Supervisory Committee and is ready for submission to The Graduate School.

\_\_\_\_\_  
Date

\_\_\_\_\_  
Elaine Cohen  
Chair, Supervisory Committee

Approved for the Major Department

\_\_\_\_\_  
Martin Berzins  
Director

Approved for the Graduate Council

\_\_\_\_\_  
David S. Chapman  
Dean of The Graduate School

## ABSTRACT

Legacy systems engineering (LSE), the design of replacement parts and assemblies for sustaining and enabling modernization of legacy mechanical systems, is increasingly becoming essential due to excessive costs of replacing entire systems. LSE scenarios presume availability of multiple, possibly incomplete or inconsistent, sources of information, lack of digital descriptions of the parts, need for modern CAD models that can drive modern CAM processes, and hard time constraints. Previous approaches to designing replacement parts have not effectively addressed such diverse scenarios, and hence present challenges in trying to adapt them to the LSE problem.

The proposed *Multiple Engineering Resources aGent Environment* (MERGE) system enables a new interactive approach toward LSE by providing a unified environment for simultaneously comparing, inspecting, manipulating and modifying multiple heterogeneous resources in order to obtain accurate feature-based CAD models of mechanical parts and assemblies. By enabling intuitive interaction and coherent visualization of information, the MERGE system enables quick identification and resolution of inconsistencies among available resources, thus making the LSE process more effective. The MERGE system also aims at simplifying the LSE process by integrating various computational *agents* with available resources to assist the engineer in processing information and in synthesizing the desired CAD models.

The MERGE system is being implemented in the context of a feature-based geometric modeling system. A framework for effective assembly modeling and analysis has also been integrated with the modeling system. The LSE capabilities of the system are demonstrated using a rocker mount part and a FormulaSAE engine mount assembly as examples.

To my parents

# CONTENTS

|                                                                            |             |
|----------------------------------------------------------------------------|-------------|
| <b>ABSTRACT</b> .....                                                      | <b>iv</b>   |
| <b>LIST OF FIGURES</b> .....                                               | <b>viii</b> |
| <b>ACKNOWLEDGEMENTS</b> .....                                              | <b>xi</b>   |
| <b>CHAPTERS</b>                                                            |             |
| <b>1. INTRODUCTION</b> .....                                               | <b>1</b>    |
| 1.1 Strategies for LSE .....                                               | 1           |
| 1.2 Challenges to LSE .....                                                | 2           |
| 1.3 Reverse Engineering .....                                              | 4           |
| 1.4 Thesis Premise and Goals .....                                         | 5           |
| 1.5 Contributions .....                                                    | 6           |
| 1.6 Scope of the Thesis .....                                              | 7           |
| 1.7 Thesis Outline .....                                                   | 7           |
| <b>2. BACKGROUND</b> .....                                                 | <b>8</b>    |
| 2.1 CAD-based Reverse Engineering .....                                    | 8           |
| 2.2 Immersive Environments for CAD Applications .....                      | 9           |
| 2.3 Engineering Drawing Analysis .....                                     | 10          |
| 2.4 Feature-based CAD Model Reconstruction .....                           | 11          |
| 2.5 CAD Model Structure Visualization .....                                | 13          |
| 2.6 Annotation Layout Management .....                                     | 15          |
| 2.7 Registration of CAD Models,<br>Point Cloud Scans<br>and Drawings ..... | 16          |
| 2.8 Assembly Modeling .....                                                | 19          |
| 2.9 Interactive Focus + Context Visualization<br>using Magic Lenses .....  | 20          |
| <b>3. SYSTEM OVERVIEW</b> .....                                            | <b>22</b>   |
| 3.1 Agents in the MERGE System .....                                       | 23          |
| 3.2 Human Computer Interface to<br>the MERGE System .....                  | 23          |
| <b>4. SYNTHESIS OF CAD MODELS OF INDIVIDUAL PARTS</b> ..                   | <b>25</b>   |
| 4.1 Importing Multiple Resources into the<br>MERGE Environment .....       | 27          |

|           |                                                                                                      |           |
|-----------|------------------------------------------------------------------------------------------------------|-----------|
| 4.2       | Fitting an Initial Feature-based CAD Model . . . . .                                                 | 27        |
| 4.3       | Visualization of the Dependency Graph . . . . .                                                      | 29        |
| 4.4       | Direct Comparison of Features . . . . .                                                              | 32        |
| 4.4.1     | Registration of Drawing Views with CAD Models,<br>Point Cloud Scans . . . . .                        | 32        |
| 4.5       | Visual Presentation of Feature Parameters . . . . .                                                  | 36        |
| 4.6       | Modification of Feature Parameters by Direct Interaction with Draw-<br>ings and CAD Models . . . . . | 40        |
| 4.7       | Summary . . . . .                                                                                    | 42        |
| <b>5.</b> | <b>MODELING AND ANALYSIS OF ASSEMBLIES . . . . .</b>                                                 | <b>43</b> |
| 5.1       | Assembly Interfaces . . . . .                                                                        | 43        |
| 5.1.1     | Why Assembly Interfaces? . . . . .                                                                   | 44        |
| 5.1.2     | Implementation of Interfaces . . . . .                                                               | 45        |
| 5.1.3     | Interfaces in the Context of LSE . . . . .                                                           | 47        |
| 5.2       | Design and Analysis of the FormulaSAE<br>Engine Mount Assembly . . . . .                             | 48        |
| 5.2.1     | Components of the Assembly . . . . .                                                                 | 48        |
| 5.2.2     | Importing the FormulaSAE Engine Mount Assembly Compo-<br>nents into the MERGE System . . . . .       | 52        |
| 5.2.3     | Specification of Interfaces for the FormulaSAE<br>Engine Mount Assembly . . . . .                    | 54        |
| 5.2.4     | Visual Analysis of Assembly Relationships . . . . .                                                  | 57        |
| 5.3       | Summary . . . . .                                                                                    | 58        |
| <b>6.</b> | <b>CONCLUSIONS AND FUTURE WORK . . . . .</b>                                                         | <b>60</b> |
| 6.1       | Conclusions . . . . .                                                                                | 60        |
| 6.2       | Future Work . . . . .                                                                                | 61        |
|           | <b>REFERENCES . . . . .</b>                                                                          | <b>63</b> |

## LIST OF FIGURES

|                                                                                                                                                                                                                                         |    |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.1 A CAD model of a rocker mount. . . . .                                                                                                                                                                                              | 9  |
| 2.2 A typical engineering drawing. The dimensions outlined with boxes<br>show the results of dimension interpretation using the NDAS system. . . . .                                                                                    | 11 |
| 2.3 Overview of the NDAS system. . . . .                                                                                                                                                                                                | 12 |
| 2.4 Feature-based model reconstruction process. . . . .                                                                                                                                                                                 | 14 |
| 2.5 Registration of resources when the CAD model is derived from the<br>drawing. . . . .                                                                                                                                                | 18 |
| 2.6 Registration of resources when the CAD model is derived from the<br>point cloud. . . . .                                                                                                                                            | 18 |
| 3.1 User interacting with the MERGE system. . . . .                                                                                                                                                                                     | 22 |
| 3.2 MERGE as a multiagent system. . . . .                                                                                                                                                                                               | 24 |
| 4.1 A rocker mount, outlined by the yellow frame, in its associated as-<br>sembly context. . . . .                                                                                                                                      | 25 |
| 4.2 Multiple engineering resources in a unified environment. Starting from<br>left, an engineering drawing, a visualization of CAD model structure,<br>a derived CAD model and a point cloud scan. . . . .                              | 26 |
| 4.3 Importing physical resources into MERGE. . . . .                                                                                                                                                                                    | 28 |
| 4.4 Partial model of rocker mount derived from features sketched on the<br>front view of the drawing. . . . .                                                                                                                           | 29 |
| 4.5 The dependency graph. For simplicity, only a single set of represen-<br>tative entities (lines, arcs) are shown for each feature. . . . .                                                                                           | 30 |
| 4.6 The dependency graph visualization. The upper node shows the<br>features comprising the partial CAD model and the lower node shows<br>the various entities comprising each feature. . . . .                                         | 31 |
| 4.7 An engineering drawing directly compared with a reconstructed CAD<br>model and a point cloud. The point cloud is registered with the CAD<br>model. Also shown is a gray cylinder that represents a tracked input<br>device. . . . . | 33 |
| 4.8 The CAD model registered with the point cloud for direct comparison<br>of features. . . . .                                                                                                                                         | 33 |
| 4.9 Side view of model from drawing being registered with 3D model. . . .                                                                                                                                                               | 34 |

|      |                                                                                                                                                                                                                                            |    |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 4.10 | 2D views from the engineering drawing superimposed on the 3D model.                                                                                                                                                                        |    |
|      | 35                                                                                                                                                                                                                                         |    |
| 4.11 | Interactive registration of drawing views with CAD models/scans. . .                                                                                                                                                                       | 36 |
| 4.12 | Layout of labels around the model in an arbitrary orientation. Red lines represent the symbolic lines to which labels are attached. . . . .                                                                                                | 37 |
| 4.13 | The label layout generation algorithm. . . . .                                                                                                                                                                                             | 39 |
| 4.14 | Layout of labels around model in another orientation. . . . .                                                                                                                                                                              | 40 |
| 4.15 | Selecting dimension from drawing. A dimension is highlighted when the tracked input device is pointed toward it. . . . .                                                                                                                   | 41 |
| 4.16 | Selecting target feature parameters via appropriate labels. The label is highlighted when the tracked device is pointed toward it. . . . .                                                                                                 | 41 |
| 5.1  | A FormulaSAE automobile prototype. . . . .                                                                                                                                                                                                 | 49 |
| 5.2  | Closeup view of engine mounted on FormulaSAE automobile. . . . .                                                                                                                                                                           | 49 |
| 5.3  | Reconstructed polygonal surface of an engine to be mounted on the FormulaSAE automobile. . . . .                                                                                                                                           | 50 |
| 5.4  | FormulaSAE automobile frame. . . . .                                                                                                                                                                                                       | 51 |
| 5.5  | Prototype model of the side-plate. Hole features are shown as cylinders. A subset of the hole features match other hole features on the engine and the rest match with hole features on the frame. . . . .                                 | 51 |
| 5.6  | Components of the FormulaSAE engine mount assembly. The links between various components indicate interfaces. . . . .                                                                                                                      | 52 |
| 5.7  | Pattern of hole features measured from engine. . . . .                                                                                                                                                                                     | 53 |
| 5.8  | Compound model consisting of engine surface and hole pattern. The hole features on one side of the engine are outlined. The same pattern of hole features are present on the other side of the engine also. . . . .                        | 53 |
| 5.9  | Right side view of the FormulaSAE engine mount assembly. . . . .                                                                                                                                                                           | 55 |
| 5.10 | Back view of the FormulaSAE engine mount assembly. . . . .                                                                                                                                                                                 | 55 |
| 5.11 | Left side view of the FormulaSAE engine mount assembly. . . . .                                                                                                                                                                            | 56 |
| 5.12 | Enhanced lens visualization of hole feature pairs from an interface. The translucent spheres show the mapping of the focus of interest in the model. The relative size of the spheres give an estimate of the magnification scale. . . . . | 58 |
| 5.13 | Lens visualization of hole feature pairs with entire assembly context. Surrounding engine surface geometry is also magnified for presenting context. . . . .                                                                               | 59 |

5.14 Lens visualization of a pair of inconsistent hole features. From the magnified visualization, it is possible to isolate the inconsistency to this particular pair of hole features and their locations in the assembly. By visually presenting the hole feature axes, analysis of the inconsistency is further simplified. . . . . 59

## **ACKNOWLEDGEMENTS**

I would like to thank my advisor, Elaine Cohen, for guiding me through this thesis. I would like to thank my committee members, Rich Riesenfeld and Pete Willemsen, for their help and support. Special thanks to David Johnson, Jim de St. Germain and Sam Drake, without whose help this thesis would not have been successful.

I would also like to thank Chimiao Xu for providing drawing analysis data. I appreciate the assistance provided by Saurav Basu for the FormulaSAE engine mount assembly project. I would also like to thank other members of the VIPER project, GDC and the School of Computing for providing a rich and competitive environment. I would like to thank Karen Feinaeur for helping me out with the administrative issues. I am grateful to my parents, brother, and to many other family members and friends for their support and encouragement.

This work was supported in part by DAAD19-01-1-0013. All opinions, findings, conclusions or recommendations expressed in this document are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

# CHAPTER 1

## INTRODUCTION

Complex mechanical systems, such as aircrafts, automobiles, and military equipment, are typically utilized for extended periods of time in order to avoid excessive costs of replacing them with new systems. Such systems may no longer be commercially supported nor properly documented because of their enduring service beyond any originally planned or projected scope of specialized function. A classic example is that of the Boeing 707 aircraft. As reported by Wikipedia [44], the 707s are in service to date in non-US airlines, charter services and air cargo operations even though production of the 707s was terminated in the late 1970s. Such *legacy systems* need to be sustained to ensure operability and proper service. Hence, it would be necessary to enable provision of new replacements of subsystems for maintenance in case of wear or failure. With improvements in engineering practices and technology, it may also be desirable to modernize legacy systems. The process of designing replacement parts and assemblies for sustaining and enabling modernization of legacy systems is termed legacy systems engineering (LSE) [1]. Due to its diverse nature, LSE is a time consuming and technically formidable process and hence, developing more effective tools for it is increasingly becoming an industrial imperative.

### 1.1 Strategies for LSE

As described in [1], alternative strategies for LSE include reverse engineering, re-engineering and re-design. Reverse engineering is the process of creating a replica of a physical object using information from existing resources. The process not only involves recreating geometry, but also inferring functionality, design intent, materials, manufacturing processes, etc. Reverse engineering is critical to an LSE

task and initiates re-engineering and re-design. Section 1.3 discusses the reverse engineering process in further detail and gives an account of previous approaches toward reverse engineering of mechanical systems.

Re-engineering is the process of modernizing an object by creating a replacement with improved design and manufacturing methods. Re-engineering involves extensive analysis of form, materials, associated manufacturing processes and a subsequent re-specification of the object. The re-engineered object may be cheaper, more reliable or more effective.

In certain situations, the original objects may not have been suitable for the target application. In such cases, it may be desirable to ignore the original object and re-design it completely or replace it with a different object that provides the same functionality. In order to achieve this, one must infer the functionality and the design intent of the original objects from accessible resources.

The decision of which strategy, or combination of strategies, to pursue, is determined by various factors including availability of proper documentation, turnaround time restrictions, costs, etc. These factors are further discussed in Section 1.2.

## **1.2 Challenges to LSE**

LSE poses a significant engineering challenge due to its multifarious nature. Some of the factors influencing the complexity of a LSE task include:

1. Original replacement parts may no longer be available due to the antiqueness of the system. For the same reason, the original designers may not be available for information.
2. Sufficient documentation in the form of engineering drawings may not be available. If available, the documentation is likely to be in non-digital form (drawings on paper, or in some cases mylar material), since such systems were designed at a time when digital documentation technology was not yet mature. On the contrary, most modern design systems involve digital descriptions of parts and assemblies. In addition, available documentation typically has errors, inconsistencies and legibility problems.

3. An exemplar part may not represent original dimensions due to undocumented modifications and wear.
4. Information about the part may be available from its associated assembly and is key to ensuring that the replacement part is compatible with the actual mechanical system. The associated parts in the assembly may not be disassembled for various reasons and hence, information can be inferred only from accessible subparts.
5. The original design rationale may be outdated and must be inferred from existing resources. In addition, the original design may have been flawed or inadequate for the target application scope.
6. Often time is of the essence in manufacturing a replacement part or assembly.
7. The originally used materials may no longer be appropriate, or even available.
8. The originally used manufacturing practices may be outdated and thus, need to be replaced with current practices. Also, LSE scenarios typically involve manufacturing replacements of small batch sizes. This plays an important role in the choice of the manufacturing processes used due to costs involved.
9. Further, an entire assembly of parts may need to be replaced. In this case, relationships between mating parts in the assembly must be inferred and the complex interaction modes between mating parts must be specified. Each individual part may be reverse engineered, re-engineered or re-designed.
10. In addition to replacements, it may be desirable to add on new parts to existing systems for added functionality.

To summarize, LSE scenarios presume availability of partial information from multiple incomplete, inconsistent or even obsolete sources. The LSE task must

yield modern CAD<sup>1</sup> models capable of driving state-of-the-art CAM<sup>2</sup> processes. In addition, significant domain knowledge and design engineering expertise is an essential ingredient in performing an LSE task. This thesis presents an approach that is tuned to better aid a domain expert in performing a given LSE task.

### 1.3 Reverse Engineering

This section provides a brief overview of various approaches toward reverse engineering of mechanical systems. The general approach, as prescribed in [31], involves the following steps:

1. Examine system and infer overall functionality.
2. Identify system features, determine operation and usage of system.
3. Disassemble system, identify components and their relationships, function, materials, etc.
4. Mathematically model system (geometry, function, underlying physical principles, etc).
5. Perform engineering analysis.
6. Re-engineer or re-design based on requirements.
7. Determine manufacturing, assembly and other process plans.

Model reconstruction has been typically performed via manual inspection of physical parts with or without the use of drawings for comparison. This implies that the task is labor intensive and time-consuming. Also, in this case, analysis and comparison of the reconstructed part with the sources (drawings and original parts) is difficult.

---

<sup>1</sup>Computer-aided Design

<sup>2</sup>Computer-aided Manufacturing

More recently, scanners such as co-ordinate measuring machines (CMM) and laser range scanners have been used to speed up the process of inspection. The scanners generate point clouds that can be used to reconstruct models [21]. The point clouds obtained from the scanners tend to be noisy. In addition, they may exhibit important omissions either due to sensor dropout or inaccessibility of certain regions of the part. Thus, CAD models reconstructed from the point clouds tend to be inaccurate, error prone or even incomplete. Feature based fitting techniques for model reconstruction have also recently emerged [10, 11]. These techniques generate feature-based models that more accurately represent form and are more intuitive to modify.

There are a variety of commercial CAD-based reverse engineering software tools that aim at automating the process of model reconstruction. These tools provide functionality to independently reconstruct models from either scans or drawings alone. Such tools are also not very robust [1]. Making assumptions about the quality of available input data for automating the reconstruction process would be quite unreasonable in many cases. Thus, existing systems do not have the flexibility to allow model creation from multiple incomplete, inaccurate or inconsistent data sources.

Although reverse engineering geometry and form of a physical object can be automated to an extent, inferring design intent, materials, etc still involves manual inspection. Thus, the overall theme of this research is to leverage user performance rather than aiming at automating the entire reverse engineering process.

## 1.4 Thesis Premise and Goals

This research addresses the enhancement of LSE by integrating multiple engineering resources in a unified interactive environment. This research exploits the fact that much of the low level processing can be automated. In addition, high level engineering tasks, such as analysis, can be greatly facilitated with interactive computing and visualization environments. This thesis presents a system that provides an interactive environment in which one can visualize and process

information simultaneously from multiple resources, to obtain CAD models that more accurately capture the design intent of the physical parts and assemblies.

In particular, this thesis aims at:

1. Providing intuitive inspection and visual analysis tools to enable identification of errors and inconsistencies.
2. Facilitating accessibility and transfer of data between different resources in an intuitive manner.
3. Incorporating various computational *agents* in the system to simplify the LSE process. The agents automate low level functionality of the system.
4. Enabling effective modeling and analysis of assemblies by allowing specification and visualization of relationships between mating parts.

In summary, this thesis aims at enabling reconstruction, analysis and revision of digital models of parts and assemblies. This thesis also aims at enabling specification of new add-on parts, re-engineered or re-designed parts to legacy systems by allowing the design and integration of new parts with the legacy system as an assembly.

## 1.5 Contributions

The main contribution of this research is a system that enables a new approach toward LSE in particular, and reverse engineering in general, by allowing simultaneous processing of information from multiple heterogeneous resources in a unified environment.

Key components of the system include intuitive interaction with, and visualization of original and derived resources. This thesis presents techniques for visualization of the structure and geometry of feature-based CAD models, interactive registration of drawings with CAD models and scans of physical parts, interactive layout management of textual annotations in the visualization environment, transfer of data via direct intuitive interaction with resources and enhanced visualization

of assembly models via lenses. All of the aforementioned techniques synergistically improve the effectiveness of an engineer performing an LSE task.

In addition, this thesis presents an effective technique for assembly modeling and analysis via the concept of assembly interfaces. This thesis also provides an implementation of assembly interfaces in a feature-based geometric modeling system.

The ideas presented in this research are validated using the rocker mount part and FormulaSAE engine mount assembly examples.

## **1.6 Scope of the Thesis**

This thesis addresses the creation and analysis of accurate models of parts and assemblies that are compatible with existing physical legacy systems. This thesis does not address issues related to materials, engineering analysis (e.g., finite element simulations), manufacturing, assembly sequence generation and planning, etc., albeit the fact that those are also important processes involved in LSE. Model creation initiates other processes in a given LSE task and is hence a necessary and critical component.

## **1.7 Thesis Outline**

This thesis is organized as follows. Chapter 2 discusses previous related work. An overview of the unified LSE environment is presented in Chapter 3. Chapter 4 presents current system capabilities for synthesizing models of individual parts from multiple data sources. Chapter 5 presents techniques for modeling and analysis of assemblies of parts. Chapter 6 presents conclusions and ideas for future work.

## CHAPTER 2

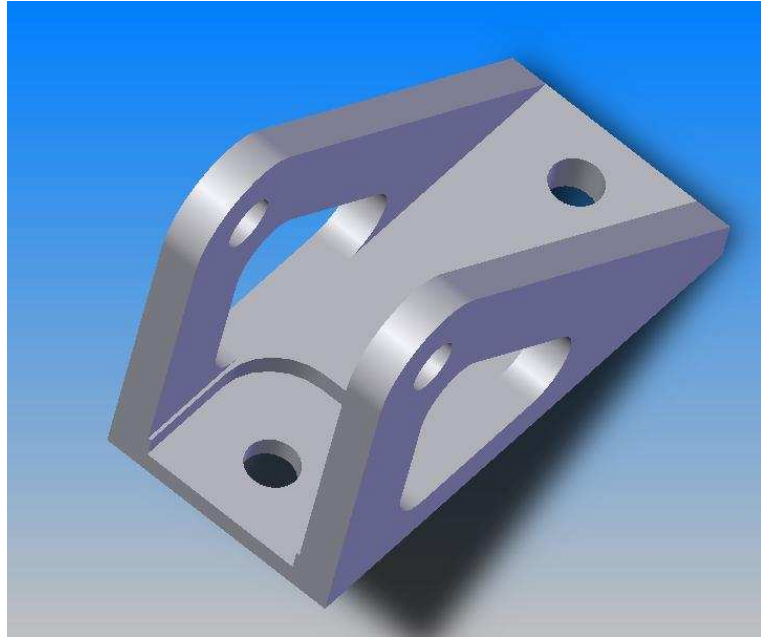
### BACKGROUND

The research presented here draws and incorporates ideas from the following different areas: CAD-based reverse engineering, immersive environments for CAD applications, engineering drawing analysis, feature-based CAD model reconstruction, CAD model structure visualization, annotation layout management, registration of drawings with CAD models/scans, assembly modeling and magic lenses.

#### 2.1 CAD-based Reverse Engineering

With the advent of CAD/CAM technology, product engineering with virtual prototypes has become an industry standard. In addition to cost savings and ease of modification, mathematical models enable analysis via finite element simulations, etc. Virtual prototypes also enable exploration of various *what-if* scenarios in which one can test and evaluate alternative designs, evaluate models for new or extended functionality and analyze interference and motion dynamics of the new system.

The reverse engineering process has since been directed toward reconstruction of CAD models which can then be used to drive CAM processes for manufacturing replacement parts. Previous work has addressed automation of the model reconstruction process [21]. Physical parts are first scanned using sensing devices, such as laser range scanners. The scanners generate sets of 3D points that are then used to fit polygonal meshes or NURBS surfaces. Such generic geometric models are difficult to modify in an intuitive manner. Feature-based modeling technology aims at overcoming such problems and automated feature-based model reconstruction techniques have been developed. These techniques are employed in this research and are further described in Section 2.4. Figure 2.1 shows a feature-based CAD model of a rocker mount part.



**Figure 2.1.** A CAD model of a rocker mount.

## 2.2 Immersive Environments for CAD Applications

Immersive virtual environments allow intuitive interaction with 3D information and hence, these techniques are increasingly being applied to a variety of problems in the mechanical engineering domain. Immersive environments allow users to focus on the higher-level objectives of the problem by making the interface to the system less cumbersome than a traditional WIMP<sup>1</sup> interface.

Immersive environments also enable users to explore large environments (e.g., entire aircrafts) more effectively. At the same time users might be required to focus on certain smaller regions of the mechanical system. Immersive environments allow effective focus+context visualization and interaction with large environments and are hence very powerful.

Immersive environments have been explored for ab initio design [3], distributed design review [8] and analysis [45] of mechanical parts. The DVDS system [3] discusses intuitive interaction techniques in an immersive environment including hand

---

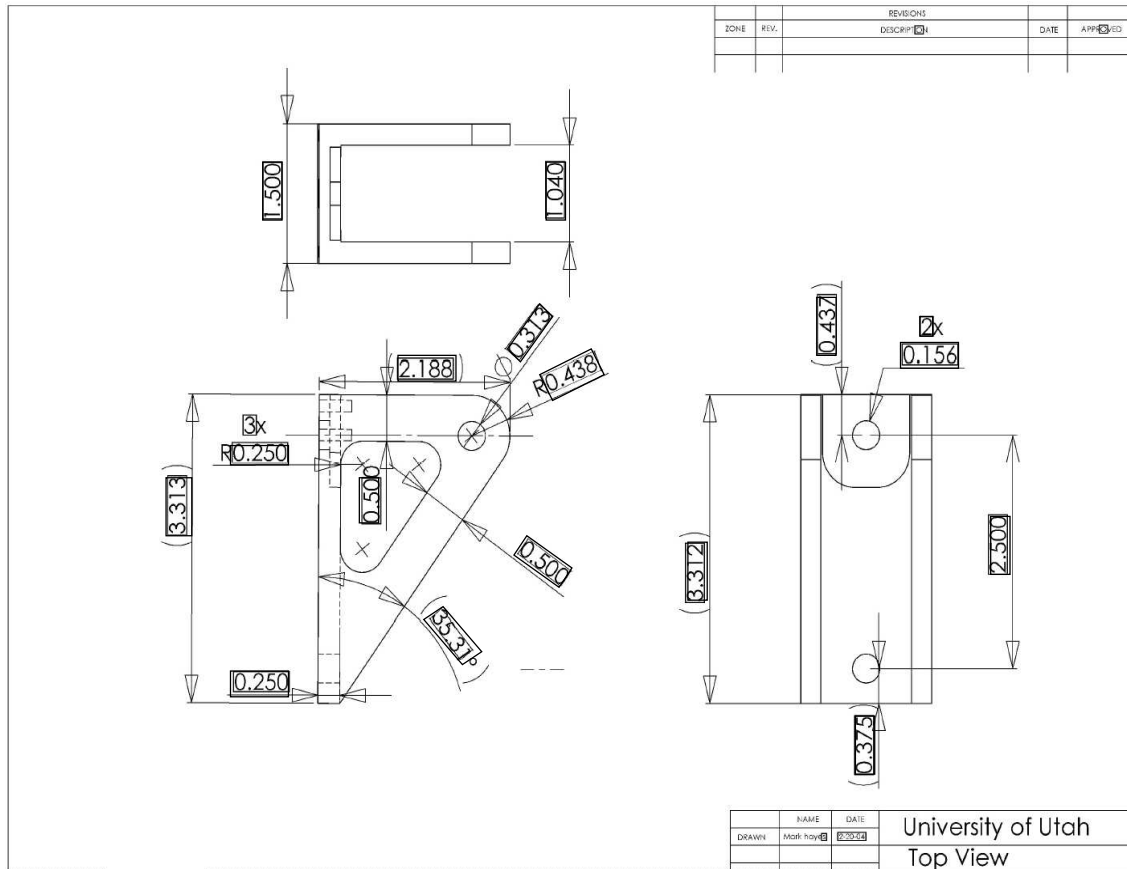
<sup>1</sup>Windows, Icons, Menus, Point and Click

motions, gestures and voice commands for *ab initio* CAD design. The DDRIVE system [8] utilizes an immersive environment for multiparty distributed design review. An immersive environment for interactive simulation and analysis has been developed [45]. Immersive environments have also been previously used for assembly simulation and planning as presented in the VADE [24] and VEGAS [26] systems. A more comprehensive list of immersive environment applications is presented in [25]. We have not come across any previous work on applying immersive environments for reverse engineering.

### 2.3 Engineering Drawing Analysis

Engineering drawings contain geometric dimension and tolerance information for parts in the form of textual annotations. Figure 2.2 is a typical engineering drawing for the 3D model shown in Figure 2.1. Extraction of this information from the drawing can be automated, which proves to be very useful in importing the drawing into a computing environment and transforming it into an interactive resource for accurate model creation.

There has been extensive work on automated interpretation of annotations from scanned images of engineering drawings [37, 39, 46, 12, 18]. Most of these algorithms require noise free conditions which is an unrealistic assumption in LSE scenarios. The engineering drawing interpretation system used in this research is based on the Non-Deterministic Agents System (NDAS) presented in [20] and [37]. The NDAS system tries to overcome noise free requirement constraints. As shown in Figure 2.3, NDAS is an automatic, domain-knowledge-guided multiagent system. It consists of *image analysis agents*, *structure analysis agents* and *evaluation agents*. Image analysis agents extract geometric primitives like lines and arcs, and interpret alpha-numeric characters. Structure analysis agents identify groups of characters as dimensions. Evaluation agents calibrate, monitor and guide analysis agents using stochastic optimization techniques based on explicit and persistent knowledge of the engineering drawing analysis process. The physical process represents the process of scanning paper drawings. NDAS's evaluation agents can request rescanning of



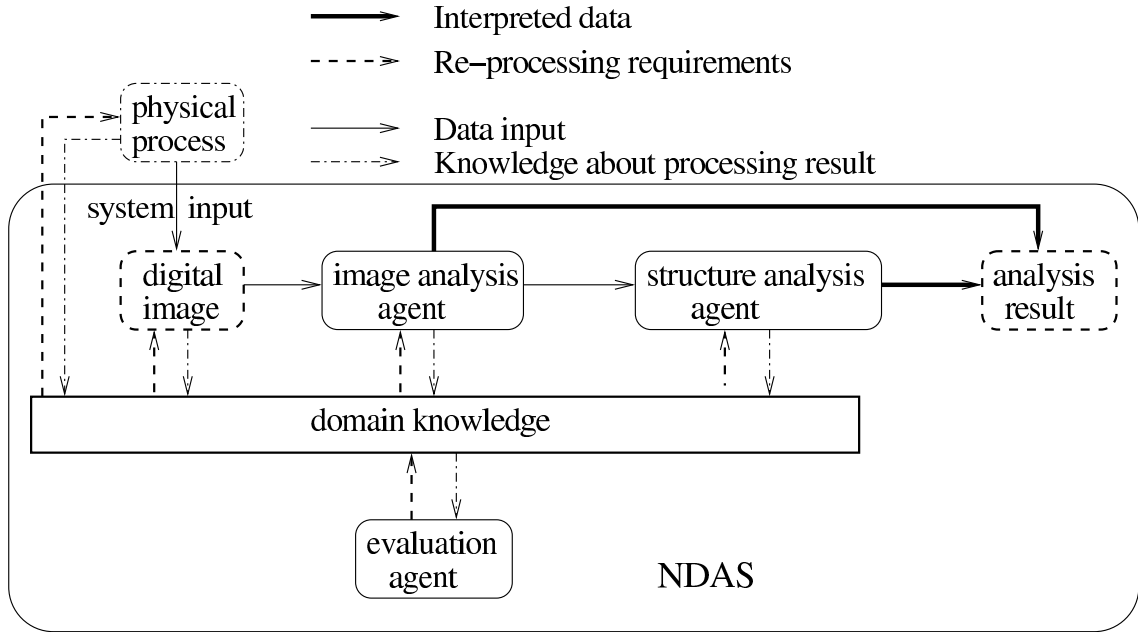
**Figure 2.2.** A typical engineering drawing. The dimensions outlined with boxes show the results of dimension interpretation using the NDAS system.

drawings with certain requirements, such as higher resolution. In Figure 2.2, the dimensions extracted from a sample drawing are outlined.

Drawings also contain geometric information about a part in the form of figures, or sketches, that are 2D projections of 3D parts from various view directions. These 2D views can be semiautomatically extracted and utilized in the model reconstruction process as presented in Section 2.4.

## 2.4 Feature-based CAD Model Reconstruction

Recently, feature-based part modeling techniques have emerged. Feature-based models are a natural fit in an interactive environment as they allow a user to intuitively modify a model in order to reflect design intent. A detailed account of



**Figure 2.3.** Overview of the NDAS system.

feature-based CAD/CAM can be found in [34]. A *feature* is a high-level modeling entity that links design intent or rationale with the model. Some aspects of a feature include:

1. A feature is a distinct physical constituent of a part or an assembly.
2. A feature relates to a particular shape whose geometry may be defined in terms of dimension parameters and geometric constraints.
3. A feature has functional significance associated with it and has certain properties that can be expected from it when utilized in a mechanical system.

There are various kinds of features such as, form features, assembly features, manufacturing features, material features, etc. based on the phase of the part engineering process in which the model is used. In this thesis, we refer to the class of *form features* that capture the geometry of the features and their associated functionality.

The feature-based model reconstruction techniques used in this research are based on techniques presented in [9, 10, 11]. These techniques work with 2.5D

features that are defined by a feature profile in 2D and an extrusion depth in the third dimension.

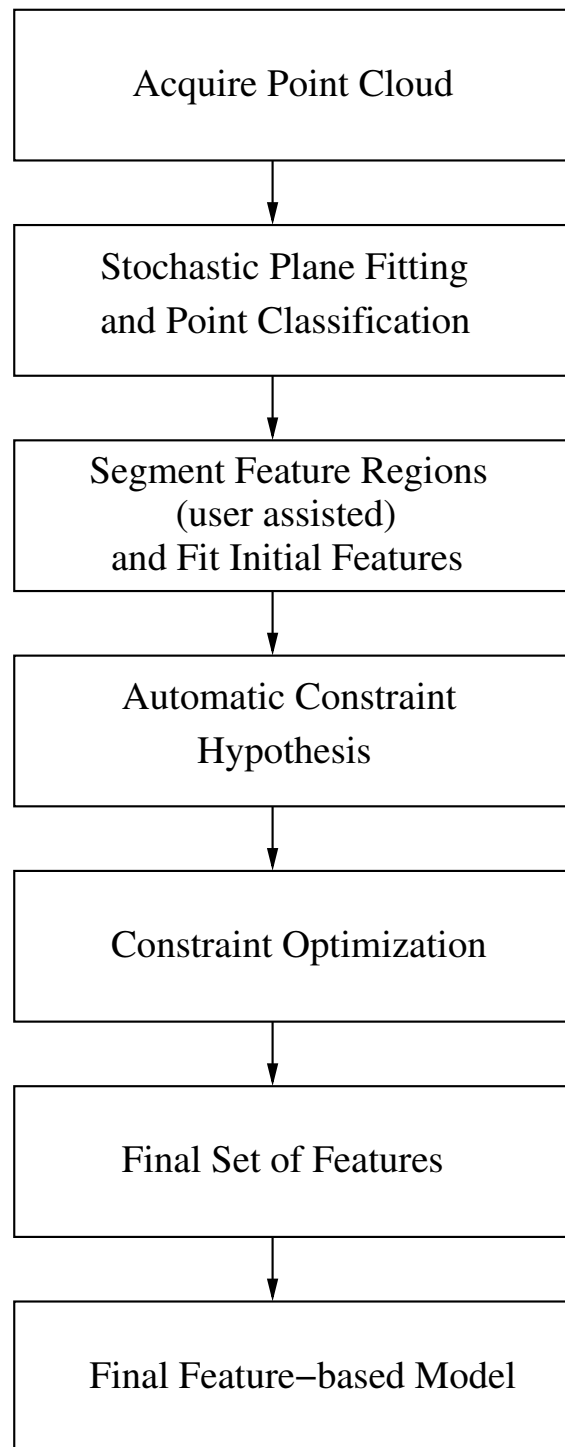
As shown in Figure 2.4, the process begins with the acquisition of a three dimensional point cloud scan of a physical part using scanning devices like laser range scanners. A stochastic plane finding method determines major planar regions of the part and classifies the points in the point cloud according to the fitted planes. These points are then segmented by semiautomatic means to separate out individual planar feature regions. Initial geometric models are fit to all the feature regions with geometric primitives including lines and arcs. After this step, geometric constraints on the primitives such as tangency, perpendicularity, etc are automatically hypothesized. An optimization of the set of hypothesized constraints generates the final feature primitives. The line-arc based geometric models are then converted to 2D NURBS profile contours. Each of the final 3D features are constructed by extruding the 2D profiles to obtain 3D NURBS surfaces, that are then combined using geometric modeling operations (e.g., boolean operators) to obtain a complete CAD model.

This technique can also be used to reconstruct feature-based models from drawings. This is achieved by segmenting out the 2D views of the model from the drawing and sampling the views to generate point clouds. In this case, the plane finding step is unnecessary as the points are already planar. The remaining steps are identical to reconstructing models from scans of 3D parts.

## 2.5 CAD Model Structure Visualization

Understanding the structure of a CAD model is essential for an engineer to make decisions regarding modification of the model, i.e., what parameters to change in order to achieve the desired effect. Previous work presents separate visualization of the geometry and structure of feature-based models [6]. The structure of models is visualized using symbols to represent features.

Our goal is to present an “as is” visualization of the data structure underlying the geometry of the CAD model, thus showing the structure of the CAD model



**Figure 2.4.** Feature-based model reconstruction process.

in addition to its geometry with actual spatial orientation relationships between features. In addition, we also aim at allowing a user to interact directly with the visualization to inspect and modify the geometry of the target CAD model. These ideas are presented in Section 4.3.

## 2.6 Annotation Layout Management

Annotations are commonly used to complement graphical visualizations in order to describe certain important features in more detail. Annotations usually contain text, numerical information and symbols, all of which is basically 2D information. The goal is to find a suitable layout of the annotations in the vicinity of the objects they describe. This problem has been approached in a variety of ways based on the target application. Approaches presented in [2, 17, 16] address situations where the information visualization environment is intrinsically 2D (e.g., geographic maps). For environments with 3D information (e.g., mechanical parts), the general approach is to simplify the layout problem by restricting the location of the annotations to a plane and then finding suitable locations on the plane based on the requirements of the application. Previous approaches include overlaying the annotation on the object of interest in an augmented reality(AR) environment [4], placing the annotations on the shadow of the object [32], rendering the annotation on a polygon derived from the projection of the object in screen space [36], and overlaying the annotations at the extreme corners of the screen in an AR environment [33].

The approach presented in [33] is most directly related to ours. They prevent annotations from occluding the objects of interest by rendering the annotations at the far corners of the screen. This approach will not work directly in our case as the environment is filled with other objects and we do not want the annotations to occlude them. Also, we do not want the annotations to occupy the entire screen even when the object is far away from the user and hence occupies a very small area on the screen.

While we do not aim at replacing the current state-of-the-art in annotation

layout management, we need a variant of these algorithms that best suits our application. Our requirement is a simple, interactive and computationally light solution that does not hinder the performance of other major functions of the system. A major problem with the annotation layout management is dealing with scenarios where a large number of annotations need to be rendered in a small area, and at the same time, avoiding clutter and maintaining a comprehensible layout. Our strategy for avoiding clutter is to allow the user to select a subset of annotations to be displayed. We also aim at allowing users to interact with the annotations to modify the target CAD model. The approach employed in this research is presented in Section 4.5.

## **2.7 Registration of CAD Models, Point Cloud Scans and Drawings**

In order to enable direct visual comparison of CAD models, point cloud scans and drawings, they need to be registered in a common coordinate frame. While automation of the alignment process is desirable, current techniques for registration do not successfully address this problem.

Although the reason for registering the resources is to compare and identify inconsistencies, these inconsistencies make registration a more challenging problem. Another problem is that legacy engineering resources may be incomplete. In particular, the sketches on the drawing may not exactly match the features on the point cloud (and hence, the model derived from either resource) due to various reasons:

1. The physical part may be broken in which case the scan is incomplete.
2. The scan of the physical part may be noisy, and/or incomplete due to sensor dropout or inaccessibility of certain regions of the part.
3. The physical part may not reflect original dimensions due to undocumented modifications or wear.

4. The features in the physical part may be misaligned due to incorrect fixturing or other manufacturing errors.
5. In some cases, the drawings may be damaged, torn or illegible.

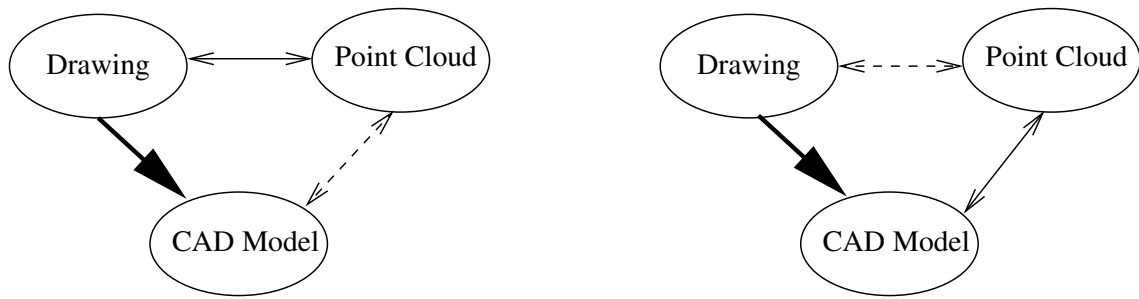
The problem of registering inconsistent data for visual analysis has been studied in the field of medical imaging. Registration of magnetic resonance imaging (MRI) scans of the human body aids cross-subject analysis. Differences in the appearance and location of anatomical entities among various individuals makes accurate registration difficult. Previous solutions for tackling this problem have been directed toward modeling the variations in anatomical entities using statistical techniques and using this knowledge to achieve more accurate registration results [7]. In their approach, a training set of MRI images of the human brain were registered with an *atlas* (one subject's scan chosen as the standard) and the variations were then statistically modeled in terms of image intensity and geometry. This statistical knowledge was then used to guide a deformation process that would then align other images with the atlas.

A significant difference in the legacy engineering scenario that does not allow us to directly incorporate ideas presented in [7] is that we have at most a single exemplar physical part and a single drawing. Thus, we do not have enough data to obtain a statistical distribution of the variations.

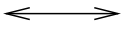
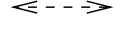
In order to register drawings, CAD models and point clouds, we only need to register any two of them since the CAD models are derived either from the drawings or the point clouds and hence the alignment between the CAD model and the resource from which it was derived is known. Figures 2.5 and 2.6 show all possible cases that we have to consider.

From these figures, it can be inferred that we have three alternatives:

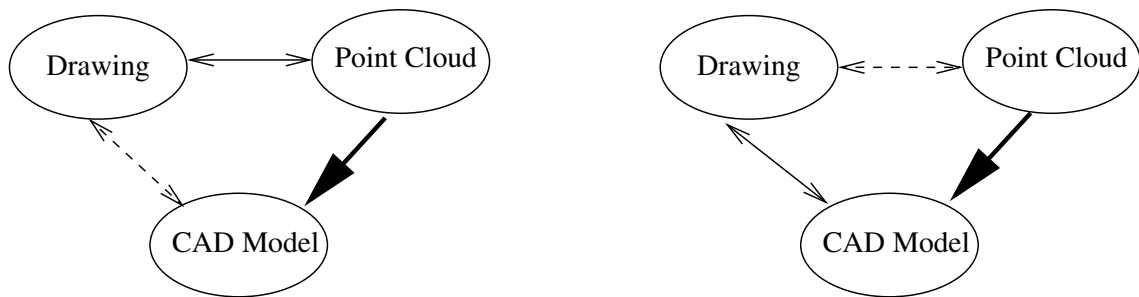
1. Registration of 2D views on drawing with point cloud scans. The 2D views sketched on the drawing are projections of the 3D part on different planes. The 2D views on the drawing can be sampled to obtain 2D point clouds. The



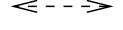
Index:

-  CAD model derivation
-  Alignment performed via computation
-  Alignment derived by transitive property of transformations

**Figure 2.5.** Registration of resources when the CAD model is derived from the drawing.



Index:

-  CAD model derivation
-  Alignment performed via computation
-  Alignment derived by transitive property of transformations

**Figure 2.6.** Registration of resources when the CAD model is derived from the point cloud.

3D point cloud scans can be projected onto planes and then registered with the sampled 2D views.

2. Registration of 2D views on drawing with CAD models. In this case, CAD models can be sampled to obtain 3D point clouds and then projected onto planes for matching and registration as in the previous case.
3. Registration of CAD models with point cloud scans. In this case, the point-sampled CAD models can be used for registration.

Thus, the problem reduces to aligning sets of point clouds in each case. A straightforward approach would be to register the two point clouds using the iterative closest point (ICP) algorithm [28]. However, the two main problems with this approach have been pointed out in [28]. First, it is difficult to find corresponding points and second, the registration process is prone to problems associated with local optima. Hence, accurate registration using ICP is difficult. The techniques used for registration in this research are presented in Section 4.4.1.

A modified approach for automated registration of CAD models with point cloud scans is presented in [28]. They suggest projecting the point cloud on the CAD model and use the projected points as the corresponding points. Then a least-squares optimization process for registering the two point clouds is performed iteratively. At each iteration, the original point cloud is projected onto the CAD model to obtain new corresponding points, thus avoiding the problems associated with local optima.

## 2.8 Assembly Modeling

Mechanical systems rarely exist as individual parts. Instead, they consist of assemblies of interacting parts. Hence, replacement parts have to be compatible with their related assemblies. Also, entire assemblies may need to be replaced. Thus, accurate CAD models for individual parts need to be derived and relationships between features of mating parts in an assembly need to be satisfied. Feature relations impose constraints on feature parameters of related parts.

Previous work on assembly modeling includes techniques that allow specification and analysis of assemblies independent of geometric modeling systems [47]. These assembly modeling systems allow separate design and analysis of assembly structure, positioning, assemblability, kinematics, etc. However, it is desirable to have an integrated framework for design and analysis of both parts and assemblies. Previous work on integrated feature-based assembly modeling systems include [13, 34, 35, 41, 23].<sup>2</sup>

In [13], assembly features are defined as separate features in independent parts, which when combined, form an assembly. An example of peg and hole features is presented. The assembly features can mate only if they satisfy certain relationships. In [35, 34], assembly features are defined as separate entities that specify features of mating parts and their inter-relationships. Both approaches are developed for static assemblies and are not intended for specifying or controlling kinematic interaction between mating parts. They can be used to determine if the individual parts are compatible with each other in the assembly.

A more general approach presented in [23], introduces the concept of *interfaces* to manage design complexity and design evolution. Interfaces were used to capture design information relevant to interaction between parts in an assembly including feature descriptions, positioning and kinematic constraints, etc. Interfaces existed as separate entities to which parts were linked to form the assembly.

## 2.9 Interactive Focus + Context Visualization using Magic Lenses

The concept of magic lenses was introduced in [5]. Magic lenses are useful tools that allow a user to selectively investigate only specific regions of interest from a large dataset. They are basically visual filters that render the region inside the lens volume differently from the rest of the space (e.g., magnification lenses [27]). Ideas from technical illustration [29] such as ghosting and “pull-out” views can be

---

<sup>2</sup>In this research, we focus on work related to managing parametric and geometric assembly relationships between part features. Work has been done on integrated management of other kinds of relationships (e.g., functional, interaction) with varied goals [19, 15, 14].

coupled with lenses for enhanced visualization of graphical data. Magic lenses for selective visualization would be useful for enhanced analysis in the reverse engineering context, by allowing users to focus on local feature regions in a model while retaining the overall spatial context in the model.

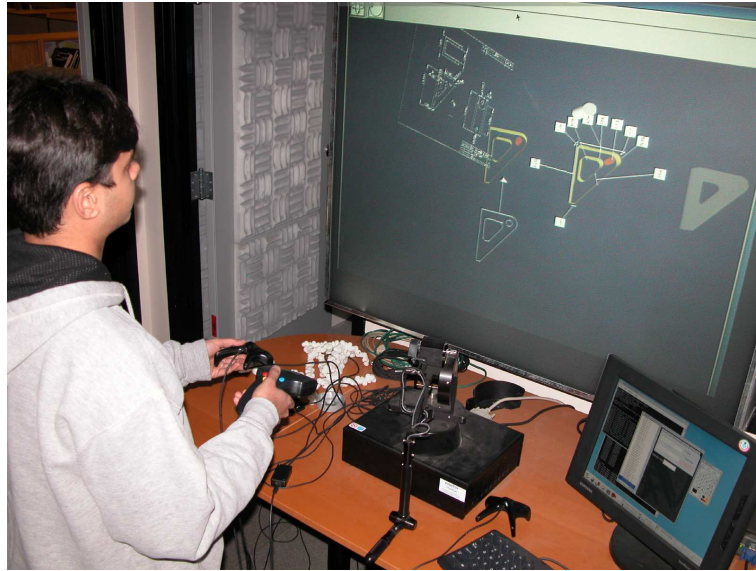
In addition to enhanced rendering of geometry, this thesis integrates semantics with the visualizations in order to better aid analysis as presented in Section 5.2.4.

## CHAPTER 3

### SYSTEM OVERVIEW

This thesis presents an interactive virtual environment called the Multiple Engineering Resources aGent Environment (MERGE) in order to satisfy the requirements of LSE (Fig. 3.1). MERGE aims at providing a unified environment for comparing and manipulating original data sources, such as engineering drawings and laser-scanned physical artifacts, and derived feature based models.

The MERGE system is built as an application layer above the Alpha\_1 advanced CAD research testbed, which provides a feature-based geometric modeling environment. Further information about the Alpha\_1 project can be found in [40].



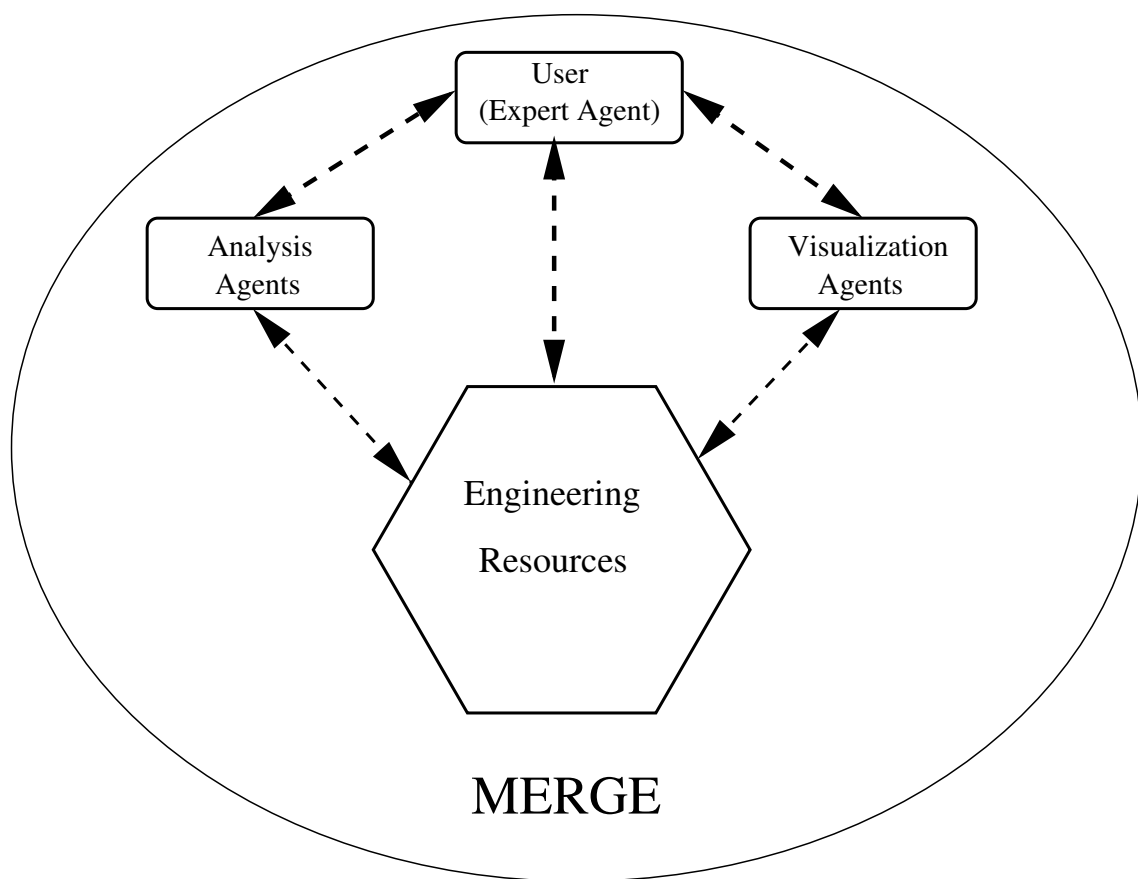
**Figure 3.1.** User interacting with the MERGE system.

### **3.1 Agents in the MERGE System**

Agents in the system co-operate with the engineer performing the LSE task to automate certain low level processes, maintain consistency between different data structures in the system, and present the original data and derived models in a unified and accessible fashion. Figure 3.2 provides a logical overview of agents in the MERGE system. These agents can be classified according to their functionality. The types of agents currently present in the system include analysis agents and visualization agents. Analysis agents analyze the original data sources and extract information that can be used in the LSE process. Visualization agents present information in a coherent and comprehensible manner. The user of the system can be considered an expert agent that controls the behavior of other agents in the system and guides the legacy engineering process.

### **3.2 Human Computer Interface to the MERGE System**

The human computer interface to the MERGE system includes a 3D stereo display and 6DOF tracked input devices in addition to the traditional mouse and keyboard devices. The Intersense IS900 system is used for tracking over a medium-room sized workspace. A rear-projected stereo-wall screen is used for 3D display. The user primarily interacts with the system using tracked devices. In Figure 3.1, the stereo-wall screen is shown with the user interacting with the system using tracked devices in both hands. The tracked devices include buttons and joysticks for input that, along with natural hand motion, can be mapped into operations such as picking and manipulation in the 3D environment.

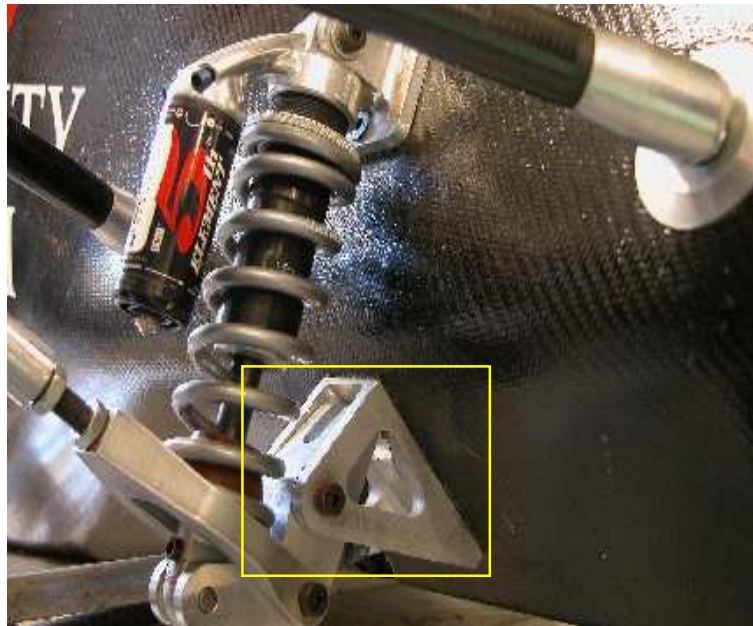


**Figure 3.2.** MERGE as a multiagent system.

## CHAPTER 4

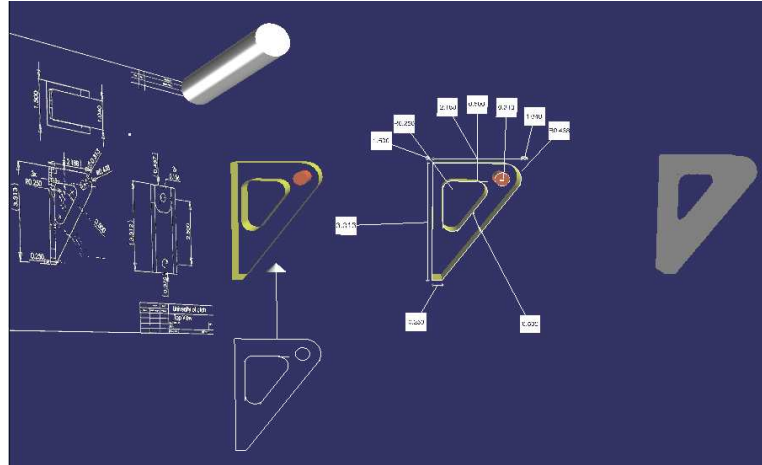
### SYNTHESIS OF CAD MODELS OF INDIVIDUAL PARTS

We first consider the case of synthesizing models for replacing individual parts. Current system functionality is presented via an example scenario of reverse engineering a rocker mount, a typical legacy part (Fig. 4.1). The rocker mount is used to support the suspension system of an automobile such as the FormulaSAE racing car. The part is subjected to large amounts of stress during operation of the automobile, and hence is susceptible to wear and failure.



**Figure 4.1.** A rocker mount, outlined by the yellow frame, in its associated assembly context.

Resources available for reverse engineering the rocker mount include an exemplar physical part and documentation in the form of an engineering drawing. These resources are imported into the MERGE environment. Agents in the system including *drawing analyzer agents*, *visualization manager agents*, *point cloud analyzer agents*, *registration agents* and *layout manager agents*, extract information from these resources and present them to the user in a coherent and comprehensible manner. An initial CAD model is fit from the drawing. Figure 4.2 shows the various engineering resources present in the environment. It presents a snapshot of the system during the process of reverse engineering the rocker mount, wherein a partial feature-based CAD model has been generated using features from the front view of the part sketched on the drawing. The CAD model can be compared directly with available resources in order to interactively identify incorrect fits. These inconsistencies can then be intuitively fixed using information extracted from the resources. These ideas are presented in detail in the following sections.



**Figure 4.2.** Multiple engineering resources in a unified environment. Starting from left, an engineering drawing, a visualization of CAD model structure, a derived CAD model and a point cloud scan.

## 4.1 Importing Multiple Resources into the MERGE Environment

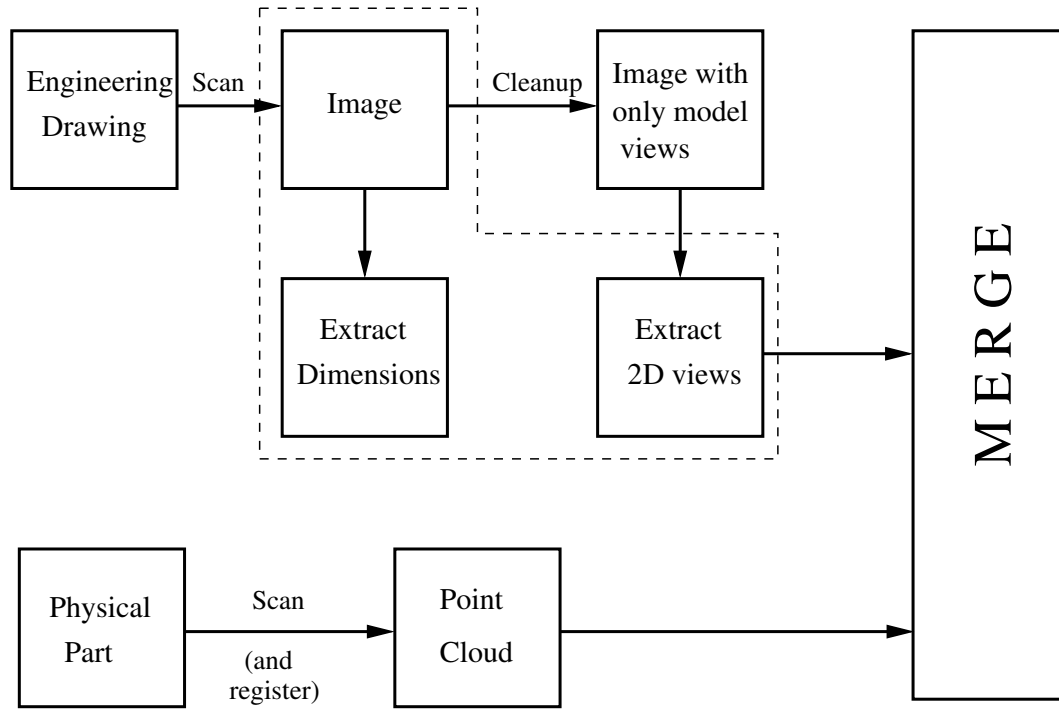
Available resources are imported into the MERGE environment and transformed into interactive higher-level forms that can be used for intuitive inspection, comparison and modification of derived models (Fig. 4.3). The exemplar part is scanned using a laser range scanner to create a point cloud representation. The engineering drawing is scanned and imported as an image file. The image file is then rendered into the alpha channel of a transparent texture map, so that lines and annotations appear solid without occluding other objects in the MERGE environment. Figure 4.2 shows the drawing and the point cloud in the MERGE environment.

As a preprocessing step, drawing analyzer agents extract key data elements from the engineering drawing. In particular, drawing analyzer agents extract the individual 2D views (top, front, right) of the parts and also interpret the geometric dimensions of the part specified on the drawing. Drawing analyzer agents consisting of NDAS systems (Section 2.3) extract dimension information. In order to segment the individual 2D views, drawing analyzer agents require an image of the drawing from which all the annotations and extraneous information has been removed. Currently, image cleanup is performed as a manual preprocessing step. View segmentation is then a straightforward image processing step.

The information extracted from the drawing is embedded into the drawing representation in the MERGE system. This transforms the drawing into a higher-level object with which the user can interact, using the tracked input devices, to query for attributes such as dimensions.

## 4.2 Fitting an Initial Feature-based CAD Model

A feature-based CAD model of the part can be automatically generated from either the individual views extracted from the drawing or from the point cloud representation as presented in Section 2.4. In our example, the features of the CAD model have been derived from the drawing. These are 2.5D features defined by a 2D profile and an extrusion depth. The feature profiles are generated using



**Figure 4.3.** Importing physical resources into MERGE.

semi-automatic techniques. In the current system, human intervention is required at various stages, namely, 1) removing extraneous information from the drawing (e.g., sketches corresponding to hidden lines represented by broken lines), 2) determining specialized feature types (e.g., classifying a feature as a hole instead of a generic profile) and 3) specifying the extrusion depth. Once 3D features have been constructed from the profiles, the features extracted from orthogonal views can be combined using boolean operators to generate a complete 3D CAD model. This model is imported into the virtual environment, which serves as a starting point from which the user can create a more accurate CAD model of the exemplar part.

In the MERGE system, one can generate partial CAD models from particular views of a part and fix incorrect fits or modify them separately, and then combine the partial models to generate the final model. This aids the user in focusing on a small set of features and hence, simplifies the process of fixing inaccurate fits or modifying the model. As shown in Figure 4.4, a partial feature-based CAD model



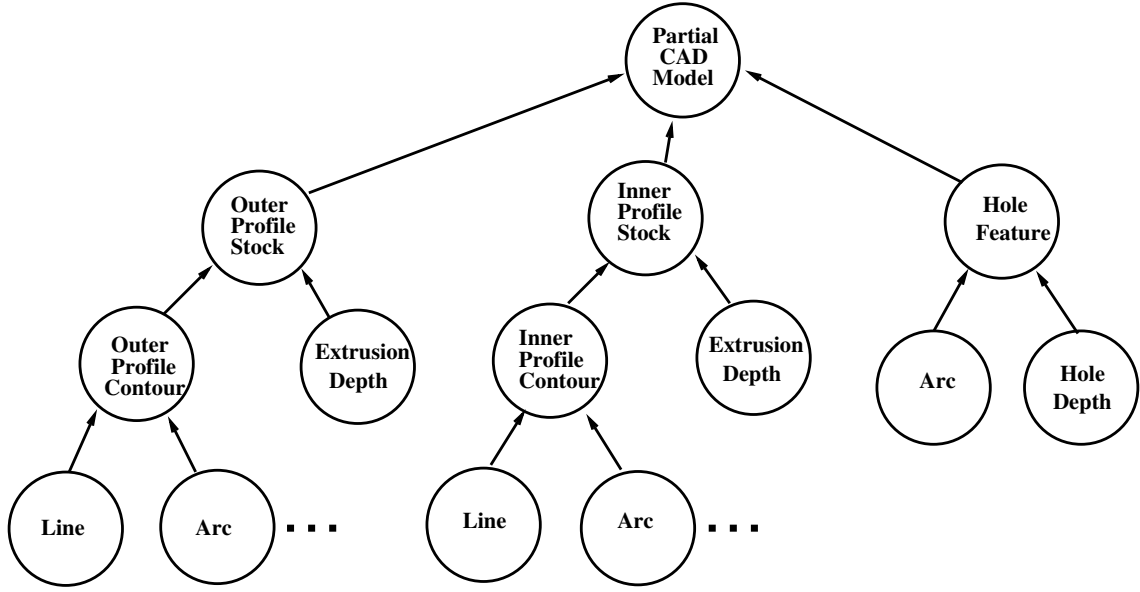
**Figure 4.4.** Partial model of rocker mount derived from features sketched on the front view of the drawing.

has been derived from the features extracted from the front view of the part. This model consists of three features: an outer profile, an inner profile and a hole. This is a much simpler model to work with as compared to the final CAD model shown in Figure 2.1.

### 4.3 Visualization of the Dependency Graph

The results of the process by which the features of the CAD model are derived are stored in a data structure called the “Dependency Graph” [40, 23]. The dependency graph is a directed-acyclic graph whose nodes store information about the formative geometry of the part. Modification of any parameters of the features of a model is achieved by modifying the corresponding nodes of the dependency graph.

The dependency graph for the partial CAD model is shown in Figure 4.5. As it can be seen, the dependency graph for a more complex part can be difficult to understand. We present a more comprehensible visualization of the dependency

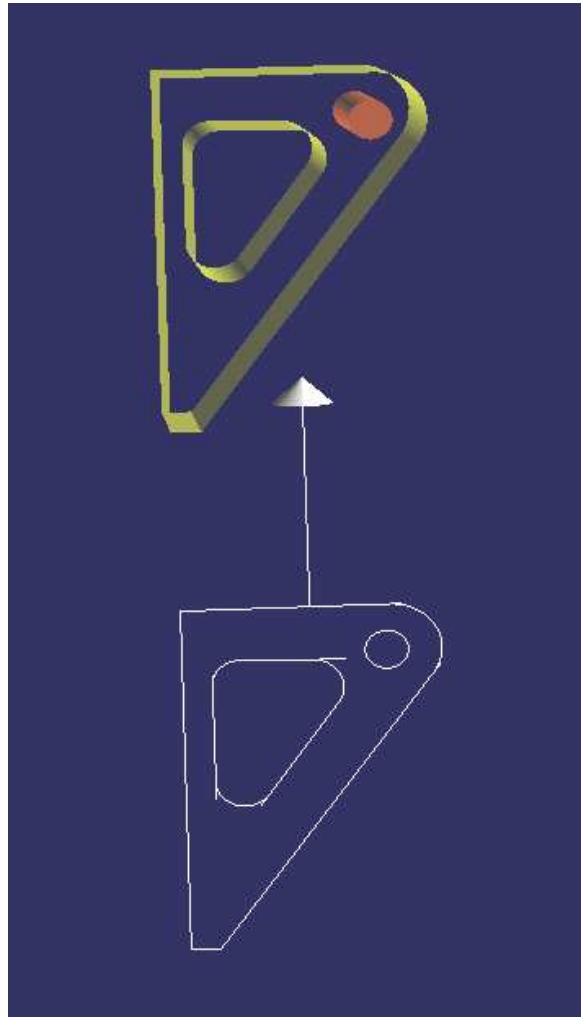


**Figure 4.5.** The dependency graph. For simplicity, only a single set of representative entities (lines, arcs) are shown for each feature.

graph to aid the user in understanding how the final CAD model is generated and thus enabling him to decide how to modify given parameters in order to obtain the desired model. We also allow the user to interact directly with the visualization, thus enabling modification of the CAD model in an intuitive manner.

Nodes in the dependency graph corresponding to features and their entities (lines, arcs) are represented visually with their actual geometry and the actual spatial orientation relationships. Each level in the dependency graph conforms to a node in the dependency graph visualization. Figure 4.6 shows the dependency graph visualization at the point in the algorithm where a partial CAD model has been generated from the front view of the part. In this figure, the upper node shows the hole feature and the inner and outer profile stock features of the partial CAD model. The lower node shows the contours defining these features which consist of entities including lines and arcs. All of the features and their entities are shown with the actual geometry and relative spatial orientations.

Visualization manager agents maintain comprehensible visualization of the dependency graph for each individual part and update them when any of the features



**Figure 4.6.** The dependency graph visualization. The upper node shows the features comprising the partial CAD model and the lower node shows the various entities comprising each feature.

are modified by the user. The MERGE environment allows a user to work directly with the dependency graph data structure via its visualization. The user can select features or entities of interest using the tracked input devices to view information about their parameters or modify them. The parameters of selected entities are displayed around the 3D CAD model as presented in Section 4.5.

## 4.4 Direct Comparison of Features

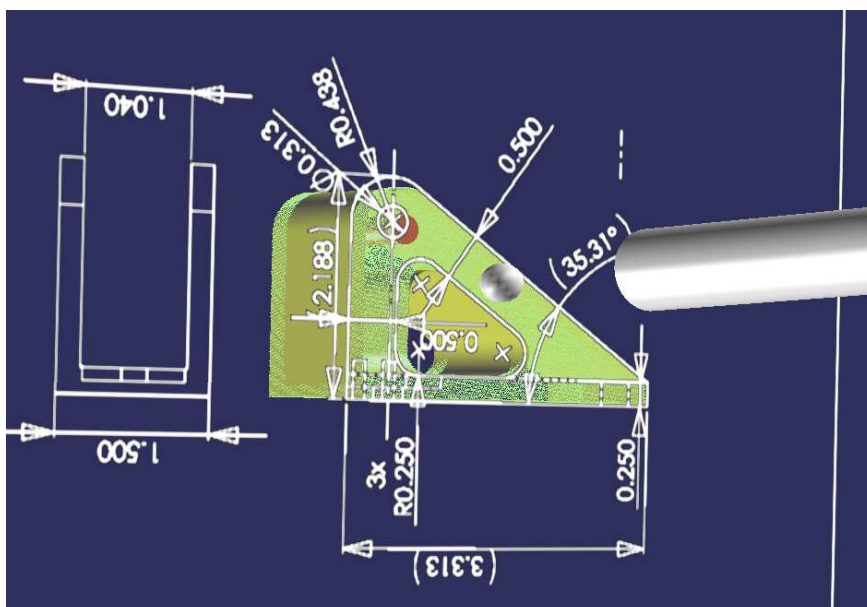
Engineering drawings contain sketches of 3D parts that are basically 2D projections of 3D parts from convenient view directions (e.g., a view direction from which features are fully visible). Typically, drawings contain three orthogonal views that are termed the top, front and side views of the part. In some cases, other views may be included, such as a cross-sectional view, view from a different direction which shows the features of a part more clearly, magnified views of certain regions of the part, etc. These views also contain broken lines that represent projections of features that are hidden from the view direction. Thus the drawing for a 3D part can get complicated and difficult to interpret.

One of the main objectives of building a unified environment for legacy engineering is to enable comparison of original resources and derived models to check for inconsistencies. It is quite difficult to compare paper drawings with physical parts and CAD models directly.

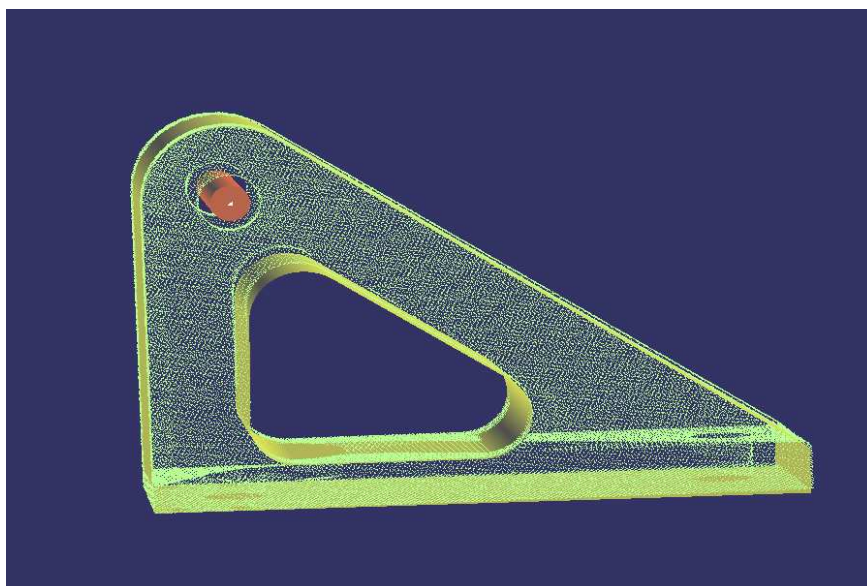
For example, a user can place the transparent drawing over the CAD model and/or the point cloud for direct comparison of features as shown in Figure 4.7. A user can also compare the fitted features with the point cloud as shown in Figure 4.8. In this figure, it is clear that the hole feature in the fitted model does not match the hole present in the point cloud.

### 4.4.1 Registration of Drawing Views with CAD Models, Point Cloud Scans

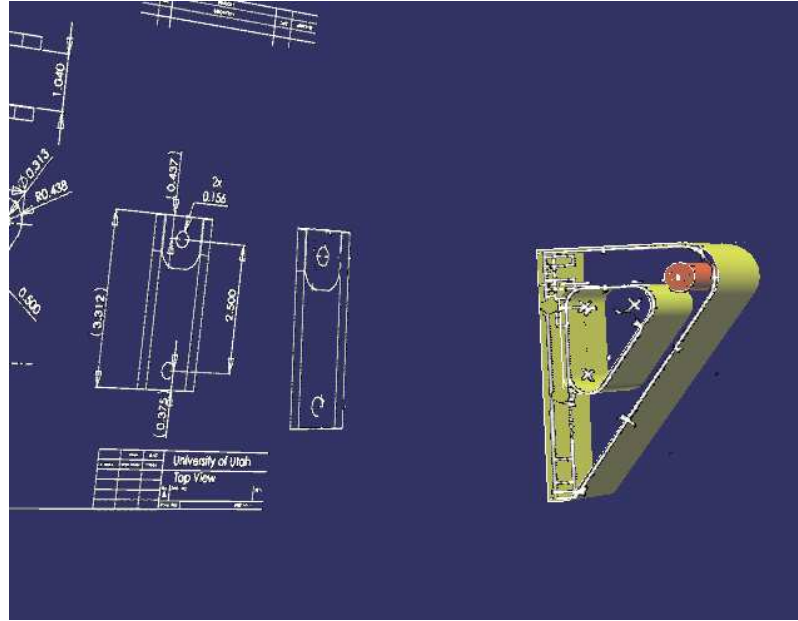
Instead of letting the user manipulate the drawing to align it with the model (or the point cloud), we can register the individual views from the drawing with the model so that they remain fixed relative to model in their correct orientations. The individual views of the part from the drawing can be interactively *folded* over the fitted model and/or the point cloud to aid the reverse engineer in associating and comparing features sketched in 2D and the features of the 3D model. Figure 4.9 shows the front view registered with the model and a snapshot of an animation of the side view registering itself with the model. The interactive animation helps the user in understanding the relation between the current orientation of the part and



**Figure 4.7.** An engineering drawing directly compared with a reconstructed CAD model and a point cloud. The point cloud is registered with the CAD model. Also shown is a gray cylinder that represents a tracked input device.



**Figure 4.8.** The CAD model registered with the point cloud for direct comparison of features.

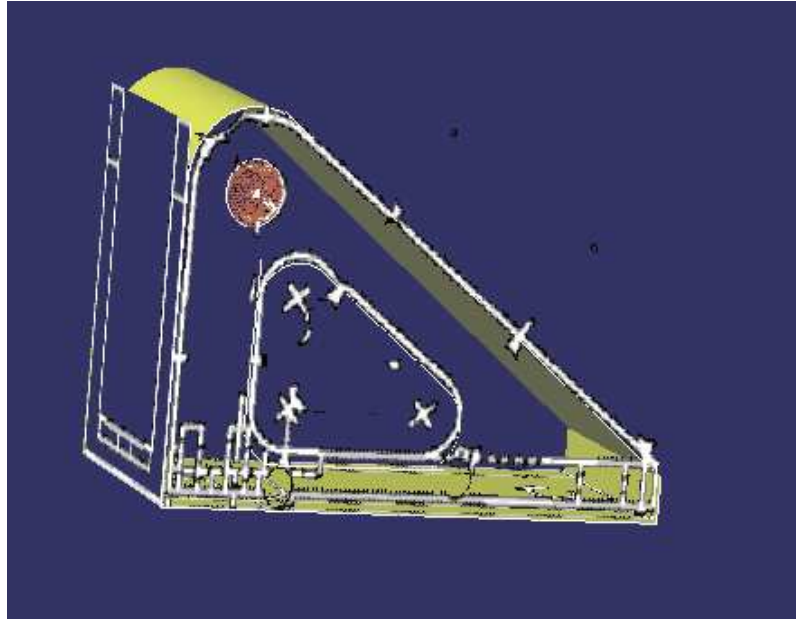


**Figure 4.9.** Side view of model from drawing being registered with 3D model.

the orientation corresponding to the view. Figure 4.10 shows the result of all the views registered with the model.

The current system implements techniques for *soft* registration of the drawing views with CAD models or scans, that provides a reasonable approximation to the actual alignment. Accurate alignment may not be possible due to incompleteness or quality of drawings as well as scans.

As mentioned in Section 2.7, CAD models and drawing views can be sampled to obtain point clouds. The registration process is therefore aimed at aligning sets of points. Since the views sketched on the drawing are 2D projections of 3D parts, registration of a 2D view with a 3D point cloud begins with projection of the 3D point cloud on a plane. As presented in Section 2.4, an automated stochastic plane finding method determines major planar regions in a given point cloud and fits planes in these regions. In addition, histogram-based point segmentation [11] techniques can be used to determine another set of planes in the 3D point cloud. The user can interactively select any of these planes for projecting the 3D point cloud. Alternatively, if the aforementioned techniques do not generate satisfactory

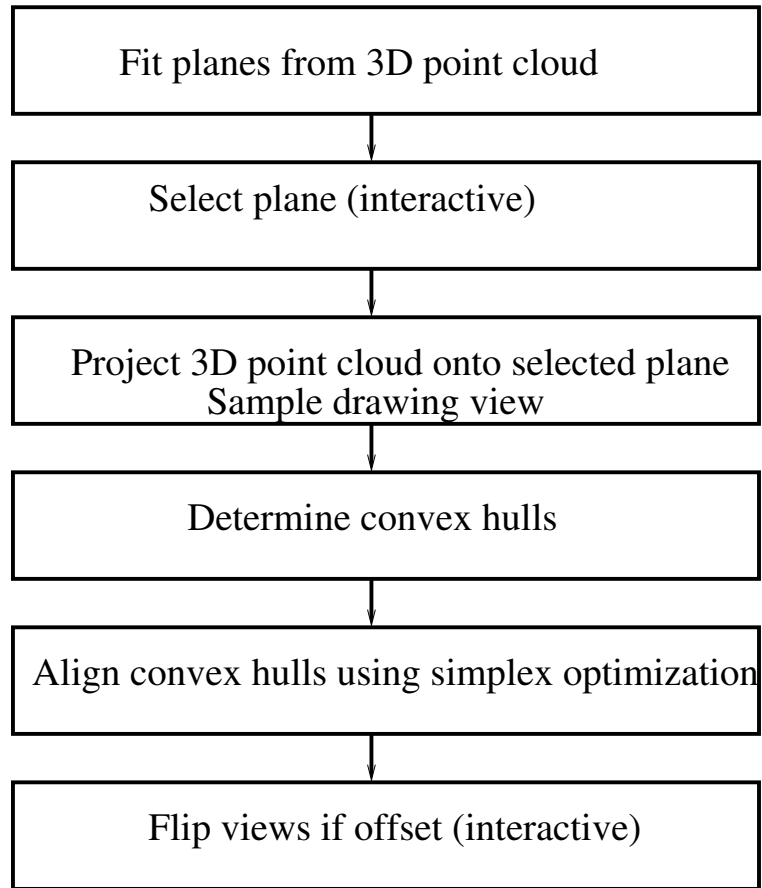


**Figure 4.10.** 2D views from the engineering drawing superimposed on the 3D model.

results, the user can interactively select a set of points from the 3D scan and fit a plane to it. The problem is now simplified to the alignment of sets of 2D points.

In order to speed up the process of alignment, convex hulls of the drawing points and the projected point cloud are determined. Standard optimization techniques such as the simplex [30] method can be used to determine a reasonable alignment between the two sets of convex hull points. In the current system, a 3DOF (two degrees of translational motion, and one degree of rotation) simplex optimization process is used. The simplex optimization method requires an initial alignment estimate. Currently, the center-of-mass of the two convex hulls is used for initializing the translational components. The optimization is then seeded over an arbitrary set of angles (0 to 360 degrees) and the alignment resulting in least error is used for registration. The registration procedure is summarized in Figure 4.11.

One limitation of the above procedure is that the alignment may result in a rotational offset of 90 or 180 degrees. This can be attributed to the fact that convex hulls may be symmetrical about one or two axes. Instead of applying computational

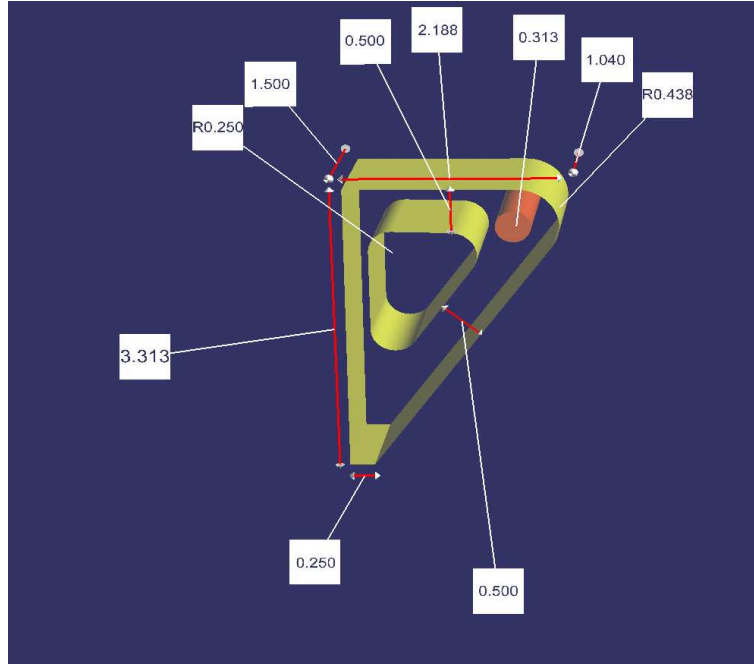


**Figure 4.11.** Interactive registration of drawing views with CAD models/scans.

techniques to resolve this problem, the system allows a user to interactively flip the views to match the scan with appropriate orientation. Thus, the system avoids computational overhead by exploiting the advantages of an interactive environment.

## 4.5 Visual Presentation of Feature Parameters

The current value of feature parameters are displayed using labels connected to the corresponding features in the 3D CAD model for quick verification and correction. Each label is connected to a symbolic line or a point on the 3D model that visually represents the feature parameters. These symbolic primitives help the user in associating dimensions specified on the drawing with the features on the 3D model. Figure 4.12 shows a set of symbolic primitives and labels arranged around



**Figure 4.12.** Layout of labels around the model in an arbitrary orientation. Red lines represent the symbolic lines to which labels are attached.

the CAD model. The labels are active in the sense that the value on the labels can be changed. The new values propagate to the corresponding features and the model is automatically updated.

In the virtual environment, the user can manipulate the CAD model using a 6DOF tracker. Layout manager agents are responsible for maintaining a comprehensible layout of the labels for any orientation of the CAD model relative to the user. The labels are rendered as planes on which the parameter value is texture-mapped and lines connecting the labels to their symbolic primitives are drawn. The layout manager interactively updates the position of the labels based on the orientation of the 3D CAD model relative to the user.

In order to maintain a comprehensible view of all the labels, the layout of the labels must satisfy the following constraints:

1. Labels, ideally must be located as close to the feature or set of features they describe, while, at the same time, not occluding the 3D CAD model geometry.

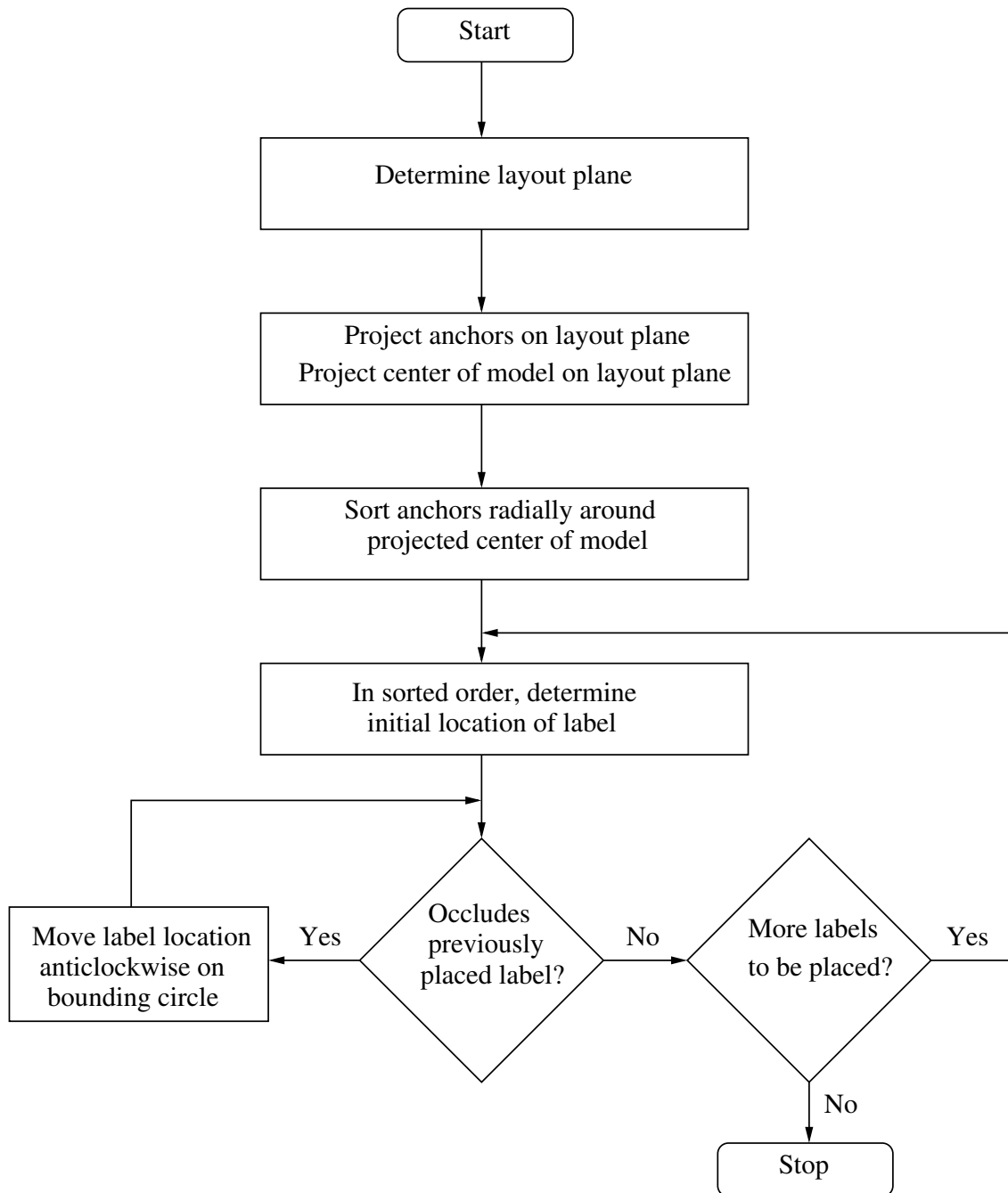
2. Labels must not occlude each other.
3. Connecting lines should not cross each other.
4. Labels should not occlude any other label's connecting line.

The algorithm that is used for generating the label layout is presented in Figure 4.13. We define the following terms that are shown in Figure 4.13:

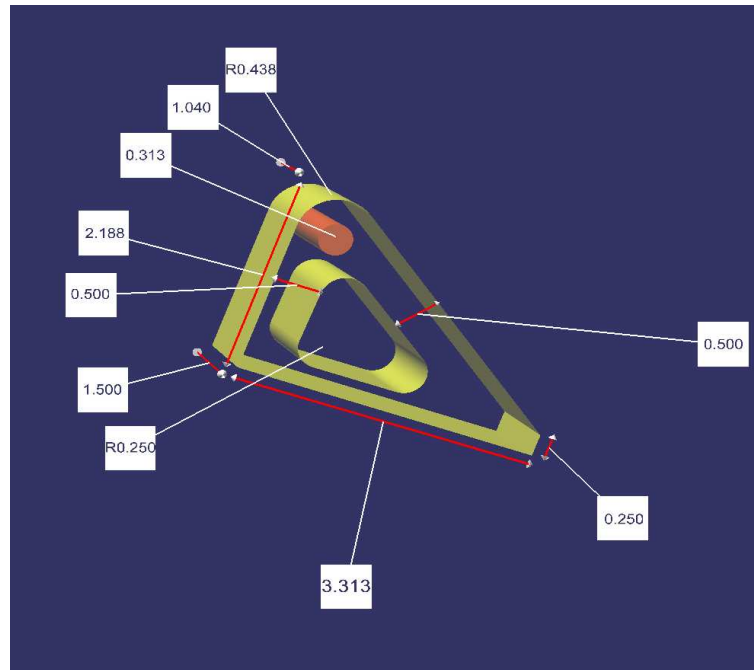
1. Layout plane - In order to simplify the problem, we restrict the location of the labels to a plane whose normal is oriented along the line joining the eye-point of the camera (of the graphical environment) and the center of the 3D model and is located between the user and the model. This plane is called the layout plane. Clearly, the layout plane changes with the user's orientation in the environment.
2. Anchor - In MERGE, an *anchor* is defined as the point on the symbolic primitives to which the label is connected.
3. Initial location of label - The location of the label's centers are restricted to a circle that has a radius greater than that of the bounding sphere of the model to ensure that the labels do not occlude the model. For each label in sorted order, an initial location is determined as the point on the circle that intersects the ray from the projected center to its anchor.

Figures 4.12 and 4.14 show the results of the layout management algorithm for two arbitrary orientations of the part. The algorithm presented here differs from the one presented in [33] in that the labels locations are restricted to a circle instead of the far corners of quadrants. Thus, the labels placed using our approach may be located closer to the actual features than using the approach presented in [33].

As a practical matter, for a large number of labels, there may not be enough room available to retain a comprehensible layout. The user can avoid this problem by requesting a smaller subset of labels to be displayed. This can be done by selecting certain features of interest via the dependency graph visualization or the



**Figure 4.13.** The label layout generation algorithm.



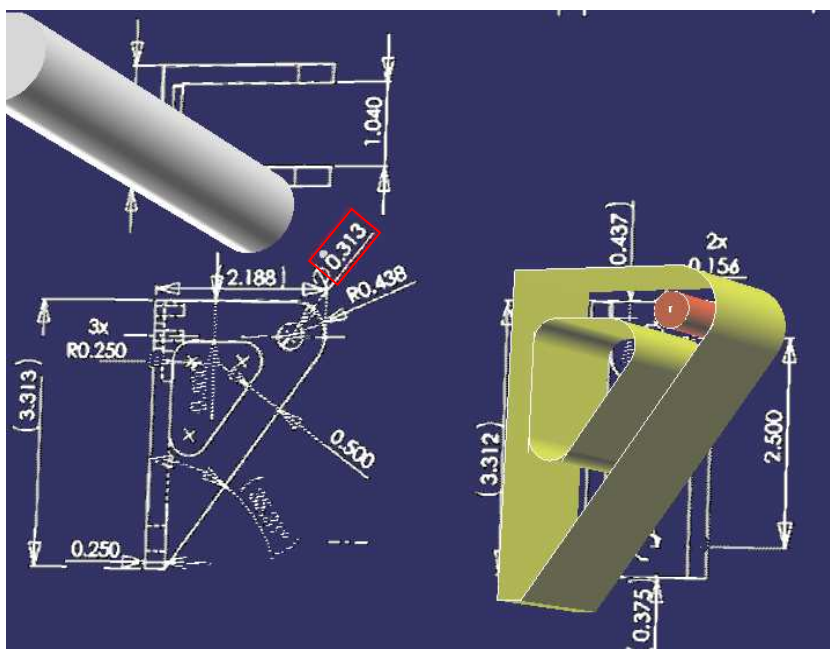
**Figure 4.14.** Layout of labels around model in another orientation.

CAD model using the tracked input devices. Accordingly, labels corresponding only to these selected features are displayed.

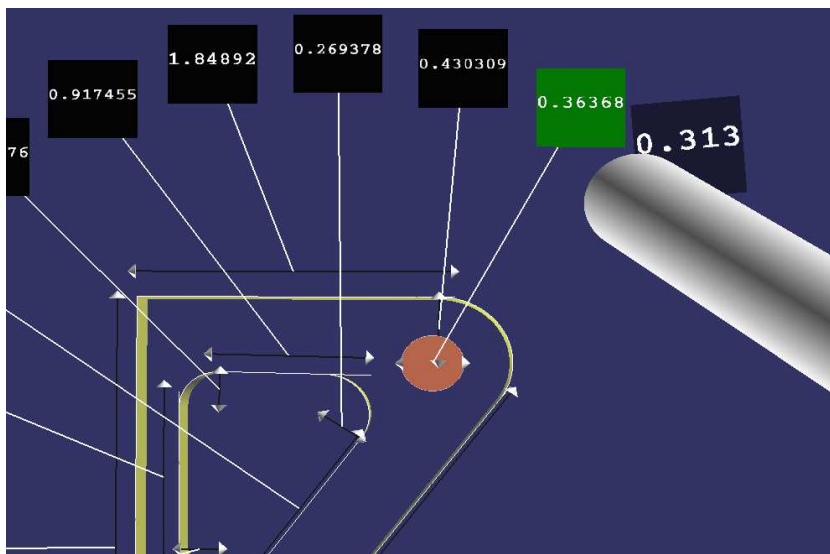
#### 4.6 Modification of Feature Parameters by Direct Interaction with Drawings and CAD Models

When an inconsistency in the parameter value of a feature has been identified (by direct comparison or by comparison of values on the label and the drawing), the user can readily correct it using the dimension information extracted from the drawing. As presented in Section 4.1, the dimensions are embedded in the drawing visualization. The user can select a dimension value from the drawing and drop it on the appropriate label on the 3D model, using the tracked input devices, to modify the parameter with the value selected from the drawing.

For example, in order to fix the diameter of the hole feature of the CAD model, the user selects the corresponding dimension from the drawing as shown in Figure 4.15 and transfers it to the hole feature in the model via its corresponding label as shown in Figure 4.16.



**Figure 4.15.** Selecting dimension from drawing. A dimension is highlighted when the tracked input device is pointed toward it.



**Figure 4.16.** Selecting target feature parameters via appropriate labels. The label is highlighted when the tracked device is pointed toward it.

## 4.7 Summary

Information about legacy parts may be available from multiple incomplete, inaccurate and inconsistent data sources such as drawings and scans. The MERGE system enables model creation from drawings as well as scans in a single environment, thereby enabling direct comparison of derived models with input sources. The MERGE system presents model structure and annotations for helping the user in understanding the current status of the model creation process. The MERGE system also simplifies model revision by enabling intuitive interaction and transfer of data between different resources.

## CHAPTER 5

# MODELING AND ANALYSIS OF ASSEMBLIES

Assemblies typically consist of multiple parts and a web of inter-relationships between part features that dictate overall assembly functionality. This research aims at integrating assembly relationships with part models in a single framework to enable validation and analysis of assembly models. In this thesis, implementation of an assembly modeling technique via the concept of *assembly interfaces* is presented. The ideas presented here build on previous assembly modeling techniques discussed in Section 2.8. Recall that CAD models of individual parts are maintained using dependency graphs in the Alpha\_1 system (Section 4.3). Thus, the goal of this research is to embed assembly relationships within the dependency graph data structure.

While the concept of interfaces presented in [23] was implemented in the Alpha\_1 system, the Alpha\_1 system itself has been redesigned and reimplemented and the implementation of interfaces no longer exists. The ideas presented in [23] were aimed at managing complexity during design evolution. In this thesis, we implement salient features of assembly interfaces that are similar to those presented in [23]. In addition, we extend these ideas and show its effectiveness for LSE.

This chapter first presents assembly interfaces and its implementation in the Alpha\_1 modeling system. The effectiveness of this assembly modeling technique in LSE is demonstrated via the FormulaSAE engine mount assembly project.

### 5.1 Assembly Interfaces

In an assembly, the set of relationships between two or more interacting parts can be grouped into a separate logical entity called the *interface*. Parts designed

or synthesized separately can be linked with interfaces to form the assembly. An interface imposes geometric, parametric, kinematic and other constraints on each individual part to ensure valid assembly configurations.

### 5.1.1 Why Assembly Interfaces?

This section presents a perspective on assembly modeling techniques in the Alpha\_1 system with and without the use of assembly interfaces and presents interfaces as a better alternative.

Parts are modeled, or programmed, using the shape creation language (SCL) [40] in the Alpha\_1 system. SCL is a high-level language for specifying features and spline-based geometric modeling operations. A part model can be considered as a set of SCL statements stored in a common place (typically, a file on disk). Features and related parameters are object variables that are assigned logically related names. On execution of the SCL statements, a dependency graph is generated that keeps track of relationships between *named* objects. A dependency graph node is created for each named object specified in SCL. Edges or links in the graph are created between each node and corresponding nodes of variables that the object explicitly depends on (i.e., the variables specified in the argument list of the corresponding SCL statement). Part modification is achieved by changing the value of the corresponding variables stored in the nodes of the dependency graph. The dependency graph nodes are accessed via the variable names. Changes are propagated through the graph along all edges, thus keeping other dependent variables consistent with the new values.

When an assembly is being designed, one would have to determine interaction between parts and features in advance. We first discuss assembly modeling without the use of interfaces. The set of interacting features would be modeled using SCL and assigned logically related names to aid analysis and future modifications. One would then have to design the remaining geometry of each part in the same coordinate space as that of the interacting features. One obvious limitation of this approach is that the entire design process is constrained to a single coordinate

space, which, in some cases, can become quite cumbersome.<sup>1</sup> More importantly, one has to be aware of the entire assembly detail in advance. This approach can be considered an implicit assembly modeling technique. Typically, the SCL model for the assembly is stored in a common file, leading to a large list of named variables and SCL statements. The complexity of managing assembly relationships would be nontrivial.

The concept of interfaces enables explicit management of assembly relationships. The interfaces, being independent entities, give designers the flexibility to design or synthesize individual parts and their relationships separately and then integrate them together at a later stage. This approach also allows a designer to model each part in separate coordinate spaces and then bring them together into a common assembly coordinate space by specifying position relationships between features of mating parts in the interface object. By explicitly specifying relationships between parts in the interfaces, the complexity of the task is reduced.

In fact, implicit assembly modeling techniques can be used simultaneously with interfaces, thereby giving more flexibility to designers. These observations are summarized in Table 5.1.

Interfaces would simplify modeling and analysis of dynamic or kinematic interaction between parts in the assembly by explicitly specifying constraints on relative motion. The interface modeling approach could potentially be used to automatically infer feature relationships from a set of given parts and also automate the process of part modification to ensure compatibility. Also, interfaces enable distributed design by allowing individual parts to be designed by separate teams that can then be integrated with the interface specification to create the assembly.

### 5.1.2 Implementation of Interfaces

In the current implementation, assembly interface objects are created to encapsulate interacting features and their relationships into a single entity. These objects

---

<sup>1</sup>Even though remaining geometry can be modeled in different coordinate spaces, manually maintaining transformations between the different coordinate spaces can be difficult.

**Table 5.1.** Comparison of assembly modeling techniques in Alpha.1 with and without the use of assembly interfaces.

| Without Interfaces                         | With Interfaces             |
|--------------------------------------------|-----------------------------|
| Implicit                                   | Explicit                    |
| Bottom-up                                  | Top-down                    |
| Base part geometry on interacting features | Design parts independently  |
| Naming conventions for managing complexity | Group into logical entities |

are maintained independently and do not belong to any single part. Assembly interface objects consist of the following attributes:

1. A set of feature definitions for each mating part.
2. A set of geometric constraints that defines the relationship between parameters of mating features.
3. A location anchor that defines the location of the mating parts in a common assembly co-ordinate frame.

Interface objects are created using the following SCL statement:

```
interface_name = assembly_interface('interface_type',
                                   part1_name,
                                   feature_set1,
                                   part2_name,
                                   feature_set2,
                                   location_anchor);

// where,
// interface_name is the name of the interface object.
// interface_type specifies the type of the interface object
// (e.g. hole pattern)
// part1_name, part2_name are the variable names of parts to which
// the interface object will be linked to create the assembly.
```

```
// feature_set1, feature_set2 are the set of interacting features
// in part1 and part2 respectively.
// location_anchor(optional) specifies the location of the assembly
// in 3D space.
```

When the above SCL statement is executed, the system performs the following operations:

1. Copies of the feature sets are created and stored in the interface object.
2. The feature sets are validated based on constraints corresponding to the interface object type.
3. If the feature sets satisfy all constraints, a dependency graph node is created for the interface object. Dependency graph edges are created between participating parts, features and the interface object, thus integrating assembly information within the dependency graph.
4. If any constraint is not satisfied, an exception message is displayed.

In the implementation of interfaces in [23], assembly features were designed independently and stored in interface objects. Parts in the assembly were linked with corresponding features in the interfaces. This approach does not work well for reverse engineering applications since the geometry of the reconstructed features may be dependent on other parameters, implying that it may not be possible to isolate features. The approach presented in this thesis overcomes this problem by allowing parts to be designed/reconstructed independently and linked with interfaces, which contain feature definitions and constraint specifications. Copies of the participating features are created in the interface objects that are then used for constraint validation. This approach is thus more general and works for both forward and reverse engineering scenarios.

### 5.1.3 Interfaces in the Context of LSE

Interfaces can be used to analyze and validate compatibility of replacement parts with the legacy system. Further, analysis of assemblies can be enhanced by

visualization of assembly relationships specified in the interfaces (see Section 5.2.4). In LSE scenarios, even though entire parts in the system may not be accessible, one can still effectively analyze compatibility of replacement parts with the rest of the assembly since only the set of interacting features have to be modeled in order to specify interface objects. The rest of the assembly may be partially scanned and grouped with the features to form compound heterogeneous models. An example of this scenario is presented in Section 5.2.2.

One can also envision scenarios where it would be desirable to replace some parts of an assembly with newer designs or alternative parts themselves. The compatibility of such parts can be validated by specifying interfaces between features of the new part and the assembly.

## **5.2 Design and Analysis of the FormulaSAE Engine Mount Assembly**

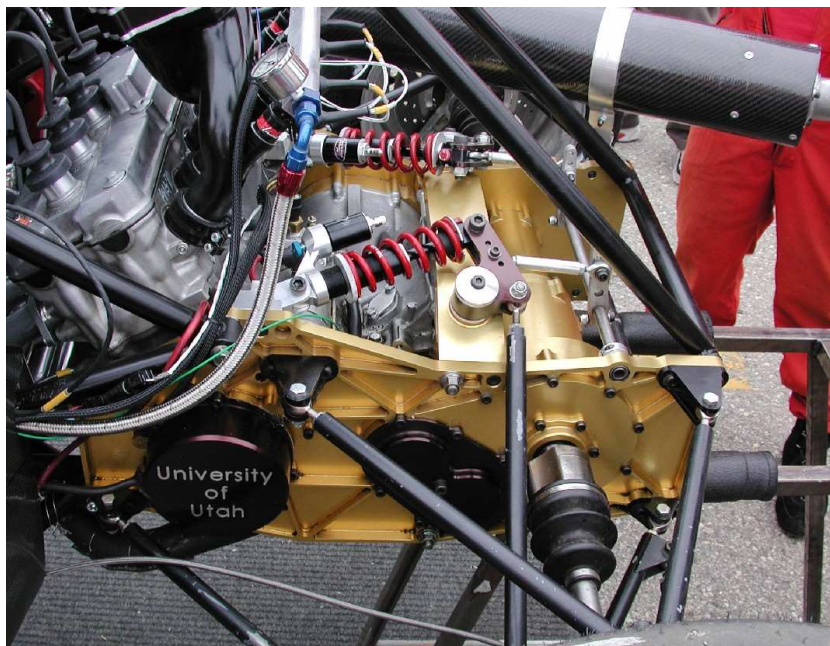
The Society of Automotive Engineers (SAE) sponsors a national competition every year in which, students design, build and race a formula automobile (Fig. 5.1). While majority of the components of the automobile are designed and manufactured in house, some parts including the engine are purchased from vendors. In order to mount the engine on to the automobile, a suitable assembly has to be designed. In this section, we present this problem as an LSE task since prior information regarding the mount assembly is not available directly from documentation and has to be inferred from the physical engine part. We demonstrate the effectiveness of the MERGE system in simplifying and enhancing visual analysis of this task. Once the assembly design has been validated, the required assembly components can be manufactured for mounting the engine on the automobile. Figure 5.2 shows an enlarged view of the engine mounted on the FormulaSAE automobile.

### **5.2.1 Components of the Assembly**

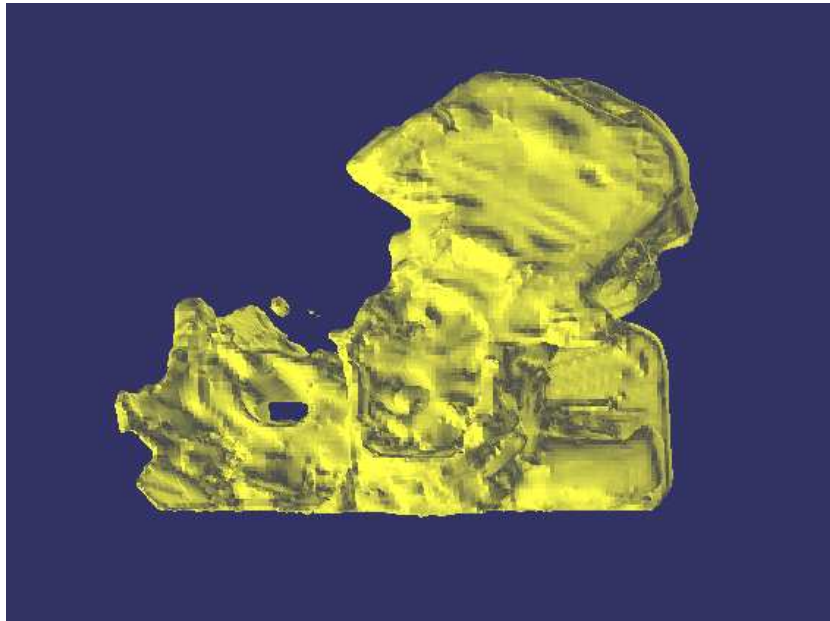
The engine rests on the rear region of the automobile's body frame. Figure 5.3 shows a surface model of the engine reconstructed from 3D scans. The dimensions of rear region of the frame are determined based on the size of the engine via



**Figure 5.1.** A FormulaSAE automobile prototype.

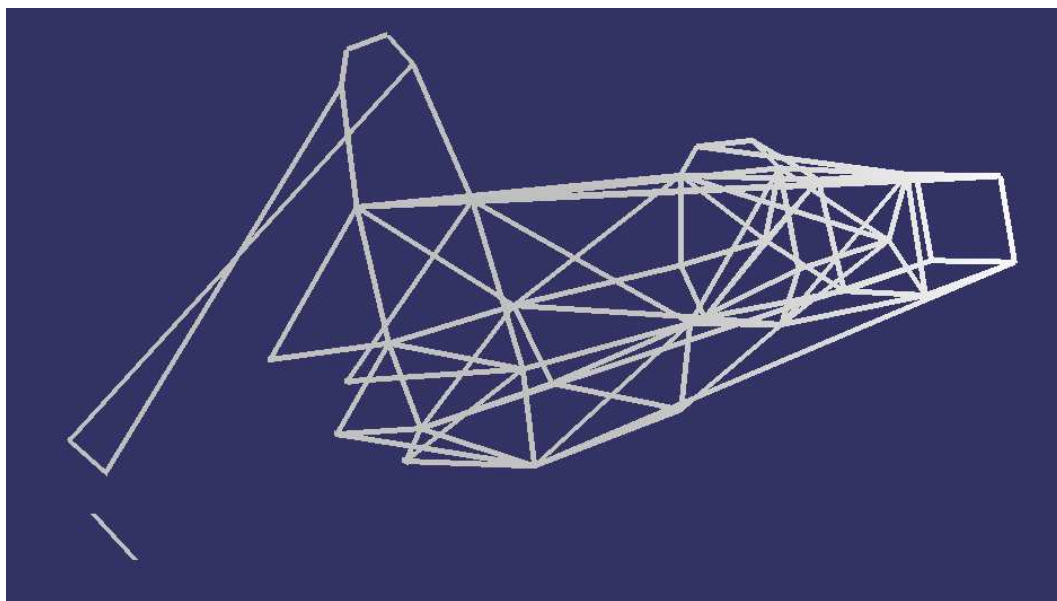


**Figure 5.2.** Closeup view of engine mounted on FormulaSAE automobile.

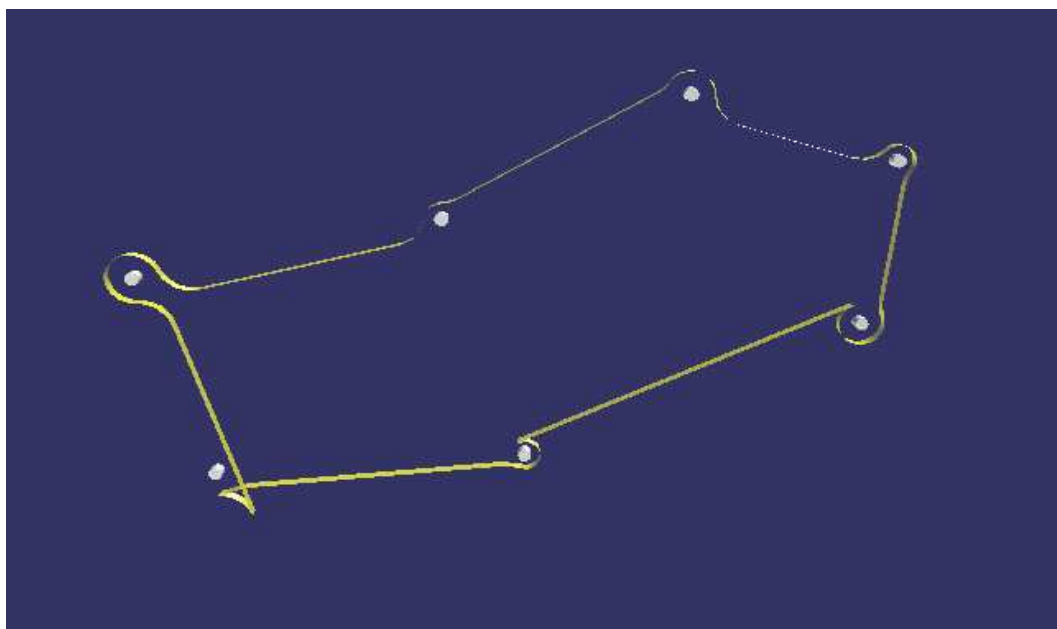


**Figure 5.3.** Reconstructed polygonal surface of an engine to be mounted on the FormulaSAE automobile.

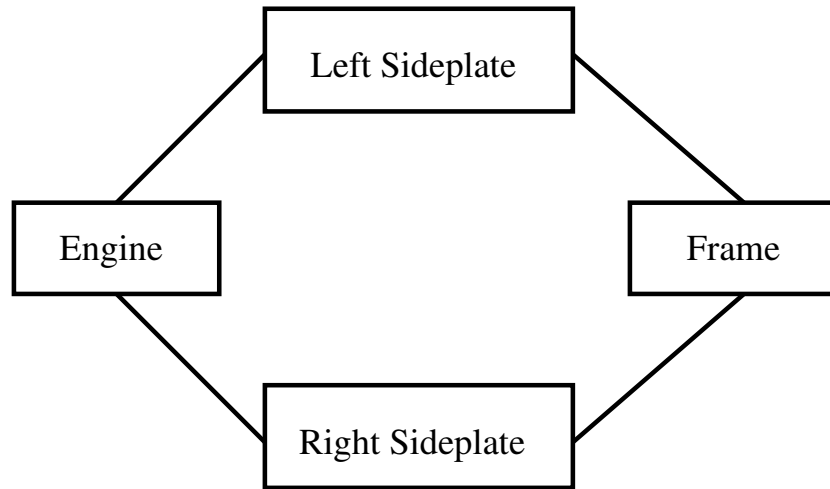
measurements using the coordinate measuring machine (CMM). A CAD model of the frame is shown in Figure 5.4. The engine can be secured in place with the frame by designing a pair of side-plates that fit on two opposite sides of the engine. A prototype model of the side-plate is shown in Figure 5.5. The components of the FormulaSAE engine mount assembly are summarized in Figure 5.6. The side-plates have to be designed based on the location and manner in which they fit on to the frame and the engine. In this example, the side-plates require a pattern of holes that match up with corresponding holes on the frame and the engine. Bolts can then be used to secure the parts in the assembly. The location and size of the hole features on the frame are available from the designed frame CAD model. The location and size of corresponding hole features on the engine have to be reverse engineered (using the CMM). After the side-plates have been designed, the assembly can be modeled in the Alpha\_1 system and analyzed using the interactive MERGE environment.



**Figure 5.4.** FormulaSAE automobile frame.



**Figure 5.5.** Prototype model of the side-plate. Hole features are shown as cylinders. A subset of the hole features match other hole features on the engine and the rest match with hole features on the frame.



**Figure 5.6.** Components of the FormulaSAE engine mount assembly. The links between various components indicate interfaces.

### 5.2.2 Importing the FormulaSAE Engine Mount Assembly Components into the MERGE System

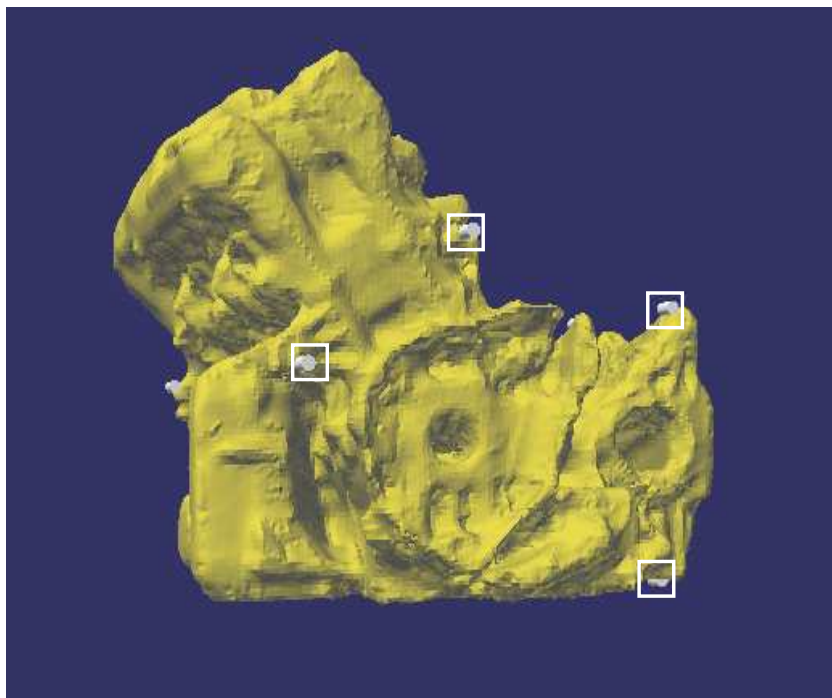
The CAD models (SCL files) of the frame and the side-plate have been developed in an older version of the Alpha.1 system. These models can thus be considered legacy resources that must be manually edited to work with the current Alpha.1 system. The hole pattern on the engine is also modeled in SCL using information obtained via measurements and imported into the Alpha.1 system (Fig. 5.7). The engine is scanned using a laser range scanner and a polygonal surface is reconstructed using techniques presented in [43, 42, 38]. Note that the surface is only a polygonal mesh and no features are derived from this process. The reconstructed engine surface is shown in Figure 5.3. A compound engine model consisting of the scan and the hole pattern model (Fig. 5.8) can be created using the following SCL statement:

```
Engine_Compound_Model: array(Engine_Scan,
                             Engine_Mount_Hole_Pattern);
```

It can be noted that the reconstructed surface is only an approximate model of the engine for a variety of reasons. First, the engine did not have all components



**Figure 5.7.** Pattern of hole features measured from engine.



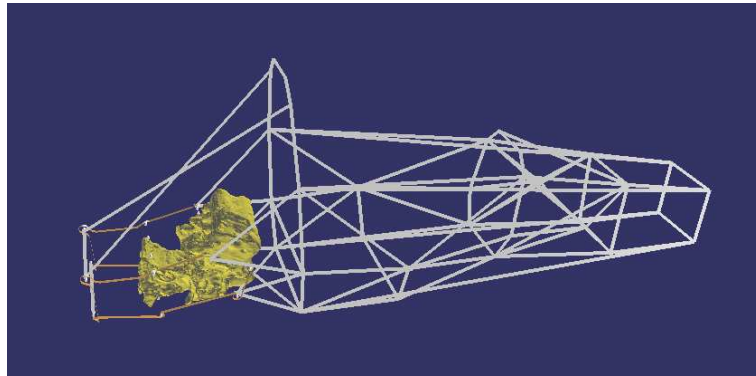
**Figure 5.8.** Compound model consisting of engine surface and hole pattern. The hole features on one side of the engine are outlined. The same pattern of hole features are present on the other side of the engine also.

attached at the time of the scan, so it is incomplete. Missing components include support pipes and bolts. Second, significant noise is introduced during the scanning procedure that results in erroneous artifacts on the surface. Third, the smoothing techniques applied during the surface reconstruction process alter the volume of the model. Nevertheless, the reconstructed surface model is a reasonable approximation for quick visual analysis. In our example, accuracy of the measured hole pattern on the engine is more important as they constitute the interface between other components in the assembly.

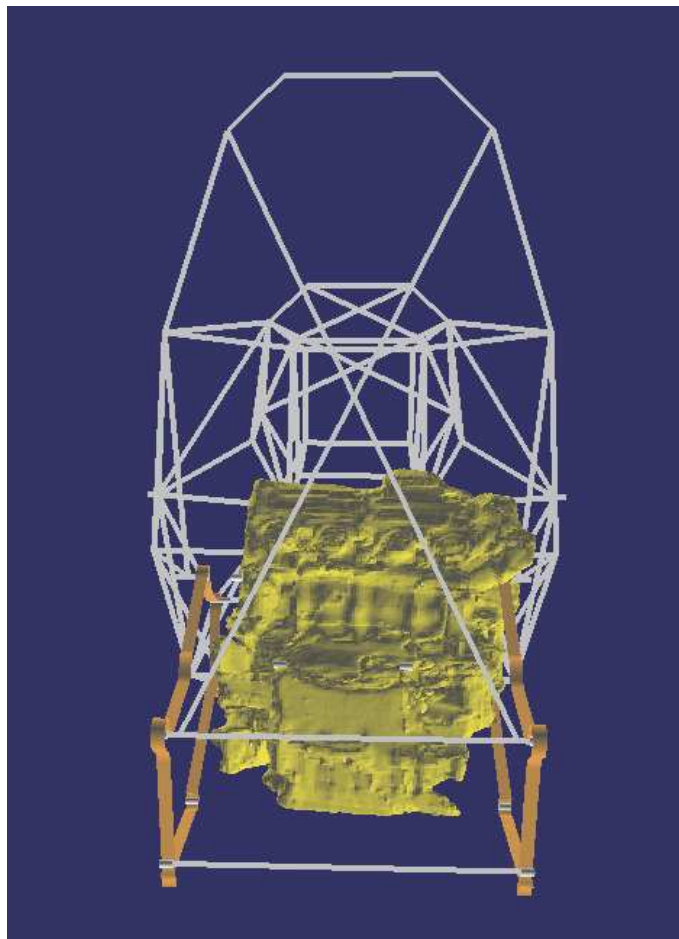
In order to aid visual analysis of the assembly in the MERGE environment, the different components need to be positioned with actual spatial relationships in a common coordinate space. The relative locations of the frame and side-plates are known from design. The location of the engine compound model relative to the side-plates (and hence, relative to the frame also) can be derived using the location of the measured holes. First, a set of points from the engine scan are interactively selected by the user that are matched with the measured hole locations. The selected points are in the engine scan's coordinate space. The relationship between the engine scan's coordinate space and the hole pattern's coordinate space (which has been measured in the frame's coordinate space) can then be derived using a technique such as Horn's closed form solution [22]. Since the engine model is not entirely accurate as noted earlier, the alignment process gives only a reasonable approximation to the true relative location. In the MERGE environment, the user can interactively modify the location of the engine for better visual results. Figures 5.9, 5.10, and 5.11 show different views of the frame, engine and side-plates displayed together in the same co-ordinate space.

### **5.2.3 Specification of Interfaces for the FormulaSAE Engine Mount Assembly**

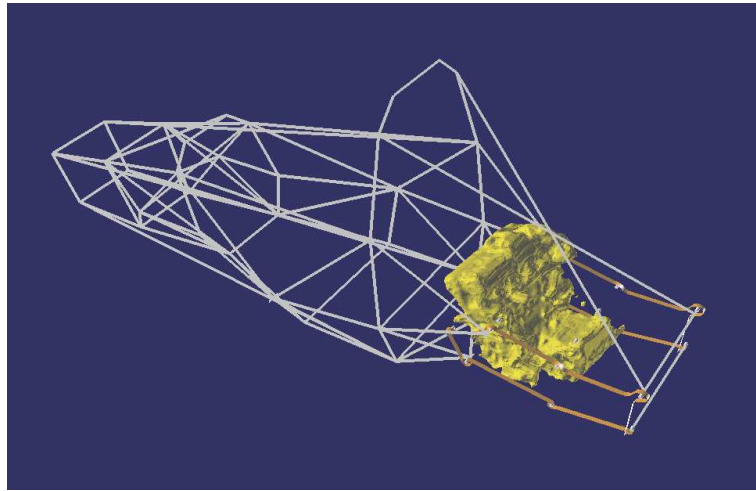
Once the models of the individual components have been imported into the MERGE environment, interface objects can be specified to model the assembly. As shown in Figure 5.6, each link is in fact, an interface between the components. In this example scenario, the assembly consists of four hole pattern type interfaces



**Figure 5.9.** Right side view of the FormulaSAE engine mount assembly.



**Figure 5.10.** Back view of the FormulaSAE engine mount assembly.



**Figure 5.11.** Left side view of the FormulaSAE engine mount assembly.

between the following components:

1. Engine and left side-plate.
2. Engine and right side-plate.
3. Frame and left side-plate.
4. Frame and right side-plate.

These interface objects can be specified using the SCL statement mentioned in Section 5.1.2. The feature sets for all the above mentioned interface objects are sets of hole features. Corresponding hole-pairs are specified in order in the feature set. As implemented in the system, the following parametric and geometric constraints are checked for a hole pattern type interface:

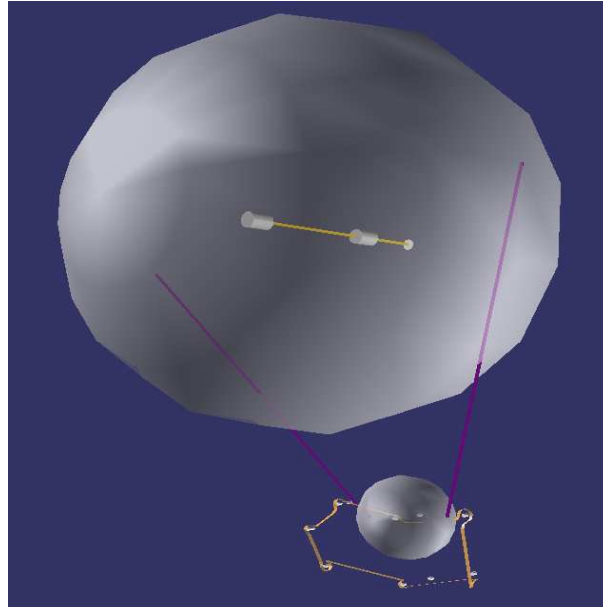
1. Is the number of hole features in both feature sets the same?
2. For each hole pair, are the diameters of the holes equal?
3. For each hole pair, are the axes of the holes co-incident?
4. Are the axes of each hole pair in the feature sets parallel?

When any of these constraints are not satisfied, a message stating the inconsistency and the corresponding parameter values is displayed. The user can then investigate the problem visually in the MERGE environment as presented in Section 5.2.4.

#### 5.2.4 Visual Analysis of Assembly Relationships

Once the interfaces have been specified, the user can visually analyze the entire assembly and various assembly relationships in the MERGE environment. An interface *lens* is defined for each pair of interacting features in an interface. Lenses display geometry of the participating features at a larger scale. Translucent spheres are drawn around the location of the feature pairs to present spatial context around the region of interest in the assembly. Translucent spheres are also drawn around the magnified features. Geometry surrounding the feature pairs in the assembly are drawn inside the magnified sphere to provide focus within the region of interest. Various assembly relationships specified in the interface are also visually presented within the magnified sphere.

In the FormulaSAE engine mount assembly example, participating interfaces are all hole pattern type interfaces. For each interface, the user can select a pair of matching hole features to be visualized using the lens. The hole features are displayed at a larger scale. The axes of the hole features are also displayed thus making it easier for the user to visually identify mismatches. Figure 5.12 shows a lens visualization of pair of holes, selected from the interface between the engine and the left side-plate. The same visualization is shown with the rest of the assembly in Figure 5.13. Recall from Section 5.2.3 that a message is displayed to the user whenever an interface constraint is not satisfied. It may not be straightforward to isolate the inconsistent features from the entire assembly model and decide a course of action from the message alone. The user can then investigate the problem using the lens visualization. Figure 5.14 shows the lens visualization of a pair of inconsistent hole features. The error in the hole axes orientation is immediately clear and the user can locate the error in the model. The magnified display of feature geometry along with additional information (such as visual presentation of

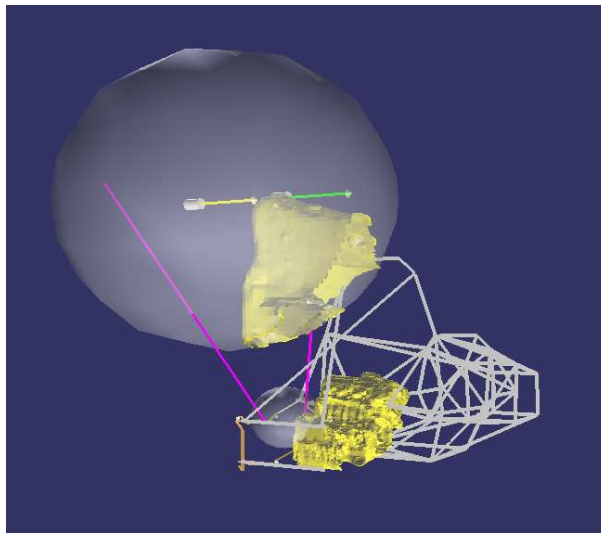


**Figure 5.12.** Enhanced lens visualization of hole feature pairs from an interface. The translucent spheres show the mapping of the focus of interest in the model. The relative size of the spheres give an estimate of the magnification scale.

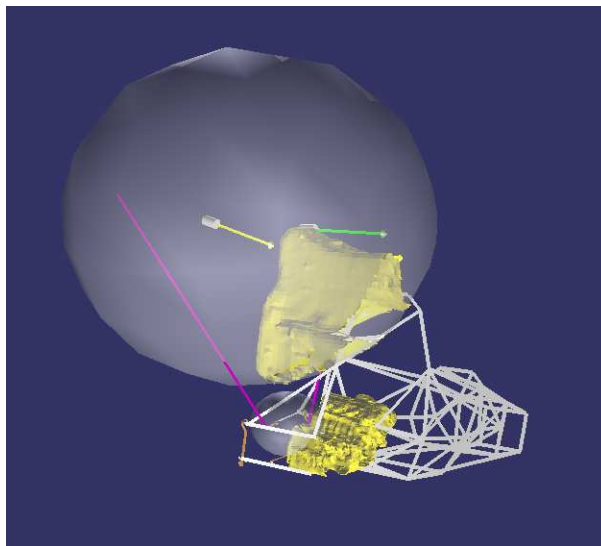
hole axes) help users in intuitively analyzing inconsistencies between different parts of an assembly.

### 5.3 Summary

Mechanical systems consist of assemblies rather than individual parts. Assembly interfaces are essential for ensuring compatibility of different parts within assemblies and provide a basis for interactive analysis of reconstructed assemblies. Moreover, interfaces enable flexibility of working simultaneously with heterogeneous resources such as CAD models and point cloud scans. Only the set of interacting features need to be derived from the scans. Derived features can be compounded with scans and linked with interfaces for analysis, thereby alleviating the problem of reconstructing entire models from incomplete and inaccurate scans. Thus assembly interfaces make the LSE process simpler and more effective.



**Figure 5.13.** Lens visualization of hole feature pairs with entire assembly context. Surrounding engine surface geometry is also magnified for presenting context.



**Figure 5.14.** Lens visualization of a pair of inconsistent hole features. From the magnified visualization, it is possible to isolate the inconsistency to this particular pair of hole features and their locations in the assembly. By visually presenting the hole feature axes, analysis of the inconsistency is further simplified.

## CHAPTER 6

### CONCLUSIONS AND FUTURE WORK

Reconstruction of models from different resources has so far been achieved by separate standalone solutions. This thesis presents a new approach, that, we believe, enhances the effectiveness of LSE by integrating multiple engineering resources into a unified interactive environment. An easy-to-use virtual environment enabling intuitive interaction and visualization allows a user to readily compare different resources, understand the ongoing process and manipulate derived information to obtain more accurate CAD models of physical parts and assemblies that are compatible with the target mechanical system. Section 6.1 presents further insights gained from the research presented in this thesis. Section 6.2 presents possible directions for future work to improve the MERGE system in particular, and LSE in general.

#### 6.1 Conclusions

The MERGE system allows augmentation of drawings, scans, and CAD models. Each of these resources are visualized in a 3D graphical environment. The resources are transformed from static entities to interactive and dynamic objects, which intuitively (visually) provide the user with necessary information for evaluation of the model reconstruction process, and allow the user to intuitively transfer information for creating more accurate models.

This thesis presents research embedded in the current capabilities of the MERGE system using the rocker mount part and the FormulaSAE engine mount assembly examples. Key abilities include projection of the model structure beside the CAD model, registration of multiple data sources such as drawings and scans with the model, folding of drawing views on the CAD model, display and modification of

feature parameters in the regions of interest in the CAD model, explicit modeling of assembly relationships and enhanced visual analysis of assembly relationships via lenses. Thus the user is allowed to focus on high level engineering tasks, thereby making the LSE process simpler, faster and more effective.

The components of the MERGE system have been developed specifically to enhance the LSE process. Although not all of the individual components of the system are significant technical advances in themselves, when utilized together in an interactive environment, they synergistically simplify, speed up and improve model synthesis in legacy engineering scenarios.

## 6.2 Future Work

Avenues for future work includes providing more functionality in the MERGE system and incorporating other interaction techniques. Although model creation is a critical step in the LSE process, it is by no means the final step. Synthesized models must be evaluated using engineering analysis techniques such as finite element simulations, etc. Manufacturing techniques, heat treatment processes, and other such information has to be inferred. Work needs to be done to integrate such analysis and other CAD tools with the MERGE system to evaluate replacements and enable re-engineering and re-design of replacements. It might be an interesting option for commercial CAD vendors to integrate LSE capabilities within existing CAD systems.

Mechanical systems can get damaged due to a variety of reasons. Recreating scenarios for part failure might provide insights toward the design of more effective replacements. It would be interesting to incorporate simulations for analysis of failure-case scenarios and provide flexibility to explore various *what-if* scenarios in the virtual environment.

Further, work needs to be done on improving the human-computer interface to the MERGE system. Interaction techniques such as gestures and speech can make the interface more intuitive. Head-mounted display devices may be employed to increase the degree of immersion in the virtual environment. Further work needs to

be done on interface management for handling large systems (e.g., entire aircrafts). Realistic or physically-based interaction (haptic or otherwise) and simulation would definitely benefit user performance in the virtual environment.

This thesis presents insights on enhancing LSE process by integrating multiple engineering resources in an interactive environment. Model creation is used to validate this claim. However, this thesis does not solve the entire LSE problem but instead, opens further opportunities for improving LSE in particular, and design engineering in general. Future work may involve evaluating the system via user studies to understand the extent to which this approach helps an engineer in performing LSE and determining areas for improvement.

## REFERENCES

- [1] Army Research Office, Virtual Parts Engineering Research Initiative, Final Report: Workshop on Legacy Systems Engineering - A Holistic View. Tech. rep., Arizona State University, Tempe, Arizona, 2003.
- [2] AGRAWALA, M., AND STOLTE, C. Rendering effective route maps: improving usability through generalization. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM Press, pp. 241–249.
- [3] ARANGARASAN, R., AND GADH, R. Geometric modeling and collaborative design in a multi-modal multi-sensory virtual environment. In *DETC 2000* (2000).
- [4] BELL, B., FEINER, S., AND HÖLLERER, T. View management for virtual and augmented reality. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology* (2001), ACM Press, pp. 101–110.
- [5] BIER, E. A., STONE, M. C., PIER, K., BUXTON, W., AND DEROSE, T. D. Toolglass and magic lenses: the see-through interface. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1993), ACM Press, pp. 73–80.
- [6] BRONSVOORT, W. F., BIDARRA, R., AND NOORT, A. Feature model visualization. *Computer Graphics Forum* 29, 4 (2002), 661–673.
- [7] CHEN, M., KANADE, T., POMERLEAU, D., AND SCHNEIDER, J. 3d deformable registration of medical images using a statistical atlas. In *Second International Conference on Medical Image Computing and Computer-Assisted Intervention* (Cambridge, UK, September 1999), Springer, pp. 621–630.
- [8] DAILY, M., HOWARD, M., JERALD, J., LEE, C., MARTIN, K., MCINNES, D., TINKER, P., AND SMITH, R. Distributed design review in virtual environments. In *Proceedings of ACM Conference on Collaborative Virtual Environments*, ACM Press (Sep 2000), pp. 57–63.
- [9] DE ST. GERMAIN, H., STARK, S., THOMPSON, W., AND HENDERSON, T. Constraint optimization and feature-based model construction for reverse engineering. In *Proceedings of ARPA Image Understanding Workshop* (Feb 1996).

- [10] DE ST. GERMAIN, H. J., JOHNSON, D. E., AND COHEN, E. Integrating freeform and feature-based fitting methods. In *ASME Design Engineering Technical Conference* (2004). DETC2004-57298.
- [11] DE ST. GERMAIN, H. J. *Reverse engineering utilizing domain specific knowledge*. PhD thesis, University of Utah, Salt Lake City, Utah, December 2002.
- [12] DORI, D. Dimensioning analysis toward automatic understanding of engineering drawings. *Communications of the ACM* 35, 10 (October 1992), 92–103.
- [13] DRISKILL, E., AND COHEN, E. Interactive design, analysis, and illustration of assemblies. In *Proc. Symposium on Interactive 3D Graphics* (April 1995), ACM, pp. 27–32.
- [14] EASTMAN, C., ASSAL, H., AND JENG, T. Structure of a product database supporting model evolution. In *Proceedings of CIB workshop on computers and information in construction: Modeling of buildings through their life-cycle* (Stanford, California, 1995).
- [15] EASTMAN, C. M. A data model for design knowledge. *Automation in Construction* 3 (1994), 135–147.
- [16] EDMONDSON, S., CHRISTENSEN, J., MARKS, J., AND SHIEBER, S. A general cartographic labeling algorithm. *Cartographica* 33, 4 (1997), 13–23.
- [17] FEKETE, J.-D., AND PLAISANT, C. Excentric labeling: dynamic neighborhood labeling for data visualization. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems* (1999), ACM Press, pp. 512–519.
- [18] FENG SU, JIQIANG SONG, CHIEW-LAN TAI AND SHIJIE CAI. Dimension recognition and geometry reconstruction in vectorization of engineering drawings. In *Computer Vision and Pattern Recognition, 2001 (CVPR 2001)* (2001), vol. 1, pp. 682–688.
- [19] GUI, J.-K., AND MÄNTYLÄ, M. Functional understanding of assembly modeling. *Computer Aided Design* 26, 6 (June 1994), 435–451.
- [20] HENDERSON, T. C. Explicit and persistent knowledge in engineering drawing analysis. Research Report UUCS-03-018, School of Computing, University of Utah, Salt Lake City, Utah, 2003.
- [21] HOPPE, H., DEROSE, T., DUCHAMP, T., McDONALD, J., AND STUETZLE, W. Surface reconstruction from unorganized points. *Computer Graphics* 26, 2 (1992), 71–78.
- [22] HORN, B. K. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America* 4 (1987), 629–642.

- [23] JACOBS, T., AND COHEN, E. Aggregation and controlled interaction: Automated mechanisms for managing design complexity. In *Proceedings of ASME Design Engineering Technical Conference in Design Theory and Methodology* (September 1998).
- [24] JAYARAM, S., JAYARAM, U., WANG, Y., TIRUMALI, H., LYONS, K., AND HART, P. Vade: A virtual assembly design environment. *IEEE Computer Graphics and Applications* 19, 6 (1999), 44–50.
- [25] JAYARAM, S., VANCE, J., GADH, R., JAYARAM, U., AND SRINIVASAN, H. Assessment of VR technology and its applications to engineering problems. *Journal of Computing and Information Science in Engineering* 1, 1 (2001), 72–83.
- [26] JOHNSON, T. A general virtual environment for part assembly method evaluation. MS Thesis, Department of Mechanical Engineering, Iowa State University, Ames, Iowa, 2000.
- [27] LAMAR, E., HAMANN, B., AND JOY, K. A magnification lens for interactive volume visualization. In *Ninth Pacific Conference on Computer Graphics and Applications* (2001), IEEE Computer Society Press, pp. 223–232.
- [28] LI, Z., GRAHAM, M., AND ROBINSON, D. Point cloud to surface best-fitting. Technical Report 2002GRC175, GE Global Research, Shanghai, 2002.
- [29] MARTIN, J. *High Tech Illustration*. North Light Books, Cincinnati, OH, 1989.
- [30] NELDER, J., AND MEAD, R. A simplex method for function minimization. *Computer Journal* 7 (1965), 308–313.
- [31] OTTO, K., AND WOOD, K. A reverse engineering design methodology. In *ASME Design Theory and Methodology Conference* (1996).
- [32] RITTER, F., SONNET, H., HARTMANN, K., AND STROTHOTTE, T. Illustrative shadows: integrating 3d and 2d information displays. In *IUI '03: Proceedings of the 8th international conference on Intelligent user interfaces* (2003), ACM Press, pp. 166–173.
- [33] ROSE, E., BREEN, D., AHLERS, K. H., CRAMPTON, C., TUCERYAN, M., WHITAKER, R., AND GREER, D. Annotating real-world objects using augmented reality. *Computer graphics: developments in virtual environments* (1995), 357–370.
- [34] SHAH, J., AND MÄNTYLÄ, M. *Parametric and Feature-based CAD/CAM - Concepts, Techniques and Applications*. John Wiley and Sons, New York, 1995.
- [35] SHAH, J. J., AND TADEPALLI, R. Feature based assembly modeling. In *Proceedings of the 1992 International Computers in Engineering Conference and Exhibition* (1992), vol. 1, ASME, pp. 253–260.

- [36] SONNET, H., CARPENDALE, S., AND STROTHOTTE, T. Integrating expanding annotations with a 3d explosion probe. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces* (2004), ACM Press, pp. 63–70.
- [37] SWAMINATHAN, L. Agent-based engineering drawing analysis. Master's thesis, University of Utah, Salt Lake City, Utah, May 2003.
- [38] TASDIZEN, T., WHITAKER, R., BURCHARD, P., AND OSHER, S. Geometric surface smoothing via anisotropic diffusion of normals. In *VIS '02: Proceedings of the conference on Visualization '02* (Washington, DC, USA, 2002), IEEE Computer Society, pp. 125–132.
- [39] TOMBRE, K., AND DORE, D. Interpretation of engineering drawings. In *Handbook of Character Recognition and Document Image Analysis* (1997), pp. 457–484.
- [40] UNIVERSITY OF UTAH, DEPARTMENT OF COMPUTER SCIENCE. *Alpha1 User's Manual Version 99.01*, 1999. Online HTML Document. URL: <http://www.cs.utah.edu/gdc/projects/alpha1help/man/a1.html>.
- [41] VAN HOLLAND, W., BRONSVOORT, W. F., AND JANSEN, F. W. Feature modelling for assembly. Technical Report 93-103, Delft University of Technology, Delft, The Netherlands, 1993.
- [42] WHITAKER, R., AND GREGOR, J. A maximum-likelihood surface estimator for dense range data. *PAMI* 24, 10 (2002).
- [43] WHITAKER, R. T. A level-set approach to 3d reconstruction from range data. *International Journal of Computer Vision* 29, 3 (October 1998).
- [44] WIKIPEDIA. Boeing 707. URL [http://en.wikipedia.org/wiki/Boeing\\_707](http://en.wikipedia.org/wiki/Boeing_707).
- [45] YEH, T., AND VANCE, J. Applying virtual reality techniques to sensitivity-based structural shape design. *ASME Journal of Mechanical Design* 120, 4 (Dec 1998), 612–619.
- [46] Y.YU, A.SAMAL, AND S.C.SETH. A system for recognizing a large class of engineering drawings. *IEEE Transactions on PAMI* 19, 8 (1997), 868–890.
- [47] ZEID, I. *CAD/CAM - Theory and Practice*. McGraw-Hill, New York, 1991.