A DESIGN INTENT INFORMATION MODEL TO SUPPORT ENGINEERING DESIGN

By

Adrián Espinosa Bautista

Under supervision of

Dr R.I.M. Young

A Doctoral Thesis

Submitted in partial fulfilment of the requirements

for the award of

Doctor of Philosophy of Loughborough University

July 2004

DEDICATION

To my parents Agustin and Magdalena.

To Monica.

To Ernesto and Carolina.

ACKNOWLEDGMENTS

This thesis was done with the financial support of the Wolfson School of Mechanical and Manufacturing Engineering at Loughborough University, and the *Universidad Nacional Autonoma de Mexico*.

First I want to express my sincere gratitude and admiration to Dr. Bob Young. His help has gone beyond his duties as supervisor. His professional supervision, constant encouragement and admirable patient helped me to achieve all the progress in this research. Thanks to Sue Young for being Monica's Guardian Angel.

I want to thank Monica for her love, courage, strength and support. This achievement is ours. Thanks for always being aside of me. My love and admiration is for you.

Thanks to my father and mother, their example, love and unconditioned support for my family and me, it was essential for this achievement and helped us not to feel the distance. God bless you.

Thanks to Lilia, Martin, for their support and friendship.

Thanks to our friends from the Manufacturing and Design Centre, for their friendship and support.

Thanks to all the people in the lab.

Thanks to all the friends, we made during this research, and their families, in no particular order: Samir, Shilpa, Roberto, Marcia, George, Marine, Feng, Francisco, Jasmine, Alejandro, Isela, Shahin, Ayshin, David and Claudia

Thanks to God for giving my family and me the strength, health, and intelligence to start, finish this work and continue.

ABSTRACT

Redesign of products has grown in importance because of fast changes in the customer requirements, fierce competition between competing companies and globalisation of markets, among other reasons. Therefore, new tools and methods have been conceived for developing products in shorter times, with lower cost and with higher quality. Concurrent engineering aims at achieving these objectives by promoting the participation, during the development of a product, of experts from all aspects of the product life cycle. Thus, all the members of the team should share the information produced. One way to share information efficiently is through common information model structures. This thesis reports on the research into information models to support decision-making. It focuses on understanding the design history of the product, highlighting the capture and use of the Design Intent underlying the changes in a redesign.

The research has explored information structures which can capture the changes done to the previous versions of the product, the Design Intent underlying those changes, and the known consequences of the changes, such that, the engineering designer can be supported in developing the next version of the product by having a tool that helps him or her to better understand the design history of the product. A new product information structure to support Design Intent has been defined which comprised four interrelated classes; a Configuration class, a Version class, a Characteristics class and a Views class.

The information structure was tested with an experimental system developed using the ObjectStore 6.0 SP 8. This was used to implement the object-oriented structure while Visual C++ 6 was used to design the dialogs to interact with the database. The development history of a wire-straightening machine was used as the case study to explore the information structure. The result of this research is a novel product model structure that can work as a repository to capture and use information related to the Design Intent.

GLOSSARY OF TERMS

IDEF	-	Integration Definition for Function
CE	-	Concurrent Engineering
CAD	-	Computer Aided Design
ISO	-	International Organization for Standardization
MM	-	Manufacturing Model
OO	-	Object Oriented
PM	-	Product Model
PMS	-	Product Model Structure
PLM		Product Lifecycle Mangement
PDS	-	Product Design Specification
STEP	-	Standard for the Transfer and Exchange of Product Model Data
UML	-	Unified Modeling Language

TABLE OF CONTENTS

1 INTRODUCTION 1
1.1 RESEARCH CONTEXT 1
1.2 AIM AND OBJECTIVES 3
1.3 SCOPE 3
1.4 UML 4
1.5 EXPERIMENTAL ENVIRONMENT 4
1.6 THESIS ORGANISATION 6
2 LITERATURE REVIEW 8
2.1 INTRODUCTION 8
2.2 PRODUCT DESIGN AND REDESIGN 8
2.2.1 The General Product Design Process 8
2.2.2 Redesign as a Process in Product Development 9
2.2.3 Reuse of Information 9
2.2.4 Design History of a Product 10
2.2.5 Design Rationale 13
2.2.6 The Role of Design Intent 14
2.2.6.1 Definitions of Design Intent 15
2.2.6.2 Methods to capture Design Intent 15
2.2.6.3 Design Intent of a Product 16
2.2.6.4 Design Intent Underlying the Design Decisions 18
2.2.7 Summary 19
2.3 DECISION SUPPORT SYSTEMS FOR REDESIGN 19
2.3.1 Current Computer Aided Design Capabilities 19
2.3.2 Intelligent Support Systems Approaches 21
2.3.2.1 Introduction 21
2.3.2.2 Case-Based Reasoning 21
2.3.2.3 Agent Technologies 22
2.3.2.4 Fuzzy Logic 22
2.3.2.5 Expert systems 22
2.3.2.6 Object Oriented Approaches 23
2.3.3 Information Management Systems 24
2.3.3.1 Introduction 24
2.3.3.2 Product Data Management Systems 24

2.3.3.3 Product Lifecycle Management Systems 25
2.3.3.4 Information Sharing Through the Use of Information Models ... 26
2.3.4 Summary 27
2.4 INFORMATION MODELS AND DESIGN INTENT 27
2.4.1 Approaches to Product Information Reuse 27
2.4.1.1 Product Range Models 27
2.4.1.2 Product Models 28
2.4.2 Methods to Support Information Modelling 30
2.4.2.1 Introduction 30
2.4.2.2 RM-ODP 31
2.4.2.3 IDEF0 31
2.4.2.4 EXPRESS 32
2.4.2.5 UML 32
2.4.3 Summary 33
3 DESIGN INTENT IN THE REDESIGN PROCESS 34
3.1 INTRODUCTION 34
3.2 DESIGN SUPPORT SYSTEMS 34
3.2.1 General Issues 34
3.2.2 Computational Design Support Tools 35
3.3 INFORMATION MODELLING TO SUPPORT REDESIGN 36
3.3.1 Information Related to Design Intent 37
3.3.2 Characteristics of the Product Related to Design Intent 39
3.3.3 Classifying Design Intent Information 39
3.4 DESIGN INTENT IN A PRODUCT MODEL TO SUPPORT THE REDESIGN PROCESS 40
3.4.1 Initial Product Model Structure 40
3.5 SUMMARY 42
4 EXPLORATION AND DEFINITION OF THE PRODUCT MODEL STRUCTURE TO CAPTURE DESIGN INTENT 43
4.1 INTRODUCTION 43
4.2 CONFIGURATION OF THE PRODUCT 43
4.2.1 Function in the Product 45
4.2.1.1 Systems, subsystem, parts and components 45
4.2.1.2 Parts 46

4.2.1.3 Commercial parts and components 46
4.2.2 Structure for Configuration of the Product 46
4.3 DESIGN CHARACTERISTICS OF THE PRODUCT 48
4.3.1 Characteristics in the Redesign of a Product 48
4.3.1.1 Geometry of the product 48
4.3.1.2 Dimensions of the product 49
4.3.1.3 Tolerance of the product 49
4.3.1.4 Material of the product 49
4.3.1.5 Function of the product 50
4.3.1.6 Specifications of the product 51
4.3.2 The Structure for the Characteristics of the Product 52
4.3.2.1 Function Subclass 52
4.3.2.2 Requirements subclass 52
4.4 VERSIONS OF THE PRODUCT 54
4.4.1 Case Example 54
4.4.1.1. Third version of straightening die 56
4.4.2 Design Intent Underlying a Change in a Version 57
4.4.3 Changes in a Version 58
4.4.4 Consequences of Change 58
4.4.5 Structure for the versions of a product 59
4.4.5.1 Version Class 60
4.4.5.2 Changes Class 60
4.4.5.3 Design Intent Class 61
4.4.5.4 Consequences Class 61
4.5 VIEWS OF A PRODUCT 62
4.5.1 Product Design Specifications in the Design Intent 62
A) High-level criteria to evaluate products 63
B) General generators of design specifications 63
C) Core-views to classify design intent and consequences 64
4.5.2 Structure for the Views of a Product 66
4.6 RELATIONSHIPS BETWEEN STRUCTURES 67
4.6.1 Relationship Between Configuration Class and Characteristics Class 67

4.6.2 Relationship Between Version Structure, Configuration Structure and Characteristics Class 68
4.6.3 Relationships Between Views Class and Version Structure 69
4.7 EXTENDED INFORMATION STRUCTURE FOR THE PRODUCT MODEL 70
4.8 SUMMARY 71
5 DEVELOPMENT OF THE EXPERIMENTAL SYSTEM 72
5.1 INTRODUCTION 72
5.1 OVERVIEW OF THE DESIGN OF THE DESIGN INTENT SYSTEM . 72
5.1.1 The experimental system environment 72
5.1.2 A Use Case Model of the Design Intent System 73
5.2 IMPLEMENTATION OF THE EXPERIMENTAL SYSTEM 74
5.2.1 Implementation of the Product Model Structure 75
5.2.2 Capture New Target Specifications. 77
5.2.3 Changes in the redesign process 81
5.2.4 Capture of the Design Intent 85
5.2.4.1 Capture of design specifications 88
5.2.5 Evaluation of the new version 90
5.2.6 Check Design Intent 93
5.3 CASE STUDY 95
5.3.1 Background of the redesign problem 95
5.3.2 Models to support the redesign process 98
5.3.3 Models to support the use of the Design Intent information 104
5.3.3.1 Assessment of the new version 107
5.3.4 The Navigation Process. 109
5.4 SUMMARY 112
6 DISCUSSION, CONCLUSIONS AND FURTHER WORK 113
6.1 INTRODUCTION 113
6.2 DISCUSSION 113
6.2.1 Structure for the Product Model 114
6.2.1.1 Class Configuration of the Product 114
6.2.1.2 Class Characteristics that Describe the Product 115
6.3.2.3 Class Views of the product 115
6.2.1.3 Structure for the Version of a Product 116

6.3.2.4 Relationship between the structures of the PMS 117
6.2.2 Application of the Product Model Structure 117
6.2.2.1 Assessment of the PMS for the redesign process 118
6.2.2.2 The relationship between the operational data and the PMS 118
6.2.2.3 The role of the PMS in the context of redesign 118
6.2.2.4 The relationship between the PMS and the PLM systems . 119
6.3 CONCLUSIONS 120
6.4 RECOMMENDATIONS AND FURTHER RESEARCH 121
Papers 123
REFERENCES 124
APPENDICES 135
APPENDIX A. UNIFIED MODELLING LANGUAGE 135
APPENDIX B. MEXICAN STANDARD ON SURGICAL NEEDLES 138
APPENDIX C. INITIAL CAPTURE OF INFORMATION OF THE PRODUCT 142
APPENDIX D. CHANGES TO PRODUCT 150

1 INTRODUCTION

1.1 RESEARCH CONTEXT

The redesign of products has become increasingly important because of the globalisation of markets, faster change of customer requirements, shorter product life cycles, new standards, and an increasing demand for better products in shorter times. Some methods have been sought to ensure a transfer of the information, between the designers of different versions of a product, to achieve a good understanding of the design history of a product, thus enabling products to be produced with a higher quality, lower cost and in shorter times to be produced.

The design history of a product includes important pieces of technical information produced during the development of the product, such as drawings, calculations, sketches, diagrams, simulations, tests, etc. However, Design Intent is information that has not been fully explored (Ullman, 2002). Design Intent needs to be considered to achieve a good understanding of the design history of the product. This research considers Design Intent as the reason underlying the changes in the redesign of a product. The capture and use of the information related to Design Intent is needed to help designers, during the redesign process, to understand the changes and produce successful versions of a product.

Many commercial software applications are available in the market to support designers in the redesign process; some commercial applications such as Product Data Management Systems are beginning to include an information sharing capability. These systems provide the means to manage all product-related information as well as the processes used throughout the product's life cycle. Information managed by these systems includes product definition data, such as configuration information, as well as data or documents which are used to describe the product or how it is produced. Process management capabilities are provided, by these systems, to support the various workflows or procedures

used during the life cycle of any product, including definitions of the roles of people within the processes. However, more work is needed in information modelling to support all the concepts involved in the product life cycle. The research reported in this thesis provides a contribution to the understanding of Information Models related to Design Intent.

Information Models are computer representations of the information of a system. Therefore better and faster solutions can be found to many design problems if Information Models can be used to offer designers supporting information. Two important qualities of Information Models are: the ability to allow software applications to use them as repositories to capture, make queries, and retrieve useful information, and to have this information shareable among all the members of the design team. This makes information modelling a powerful tool to support the decision-making in developing products. Two information models have proved their ability to support different aspects of the re-use of information: the Product Range Model and the Product Model.

The Product Range Model contains information about a range of solutions for the different systems of an injection-moulding machine. The interactions between the solutions and the functions of the systems are used to provide a number of solutions to the designer providing support in the decision-making (Costa, 2000).

The Product Model contains detailed information related to the life cycle of a product, i.e. design information; how this product was manufactured and assembled; technological data; etc (Harding, 1996, McKay, 1993, 1994, Mckay and Bloor, 1991, Young, Dorador *et al*, 2001).

This research has extended the capability of the Product Model by including the information related to Design Intent. Figure 1.1 illustrates the two information models, including Design Intent, in the Product Model and computer-aided design applications interacting with them.

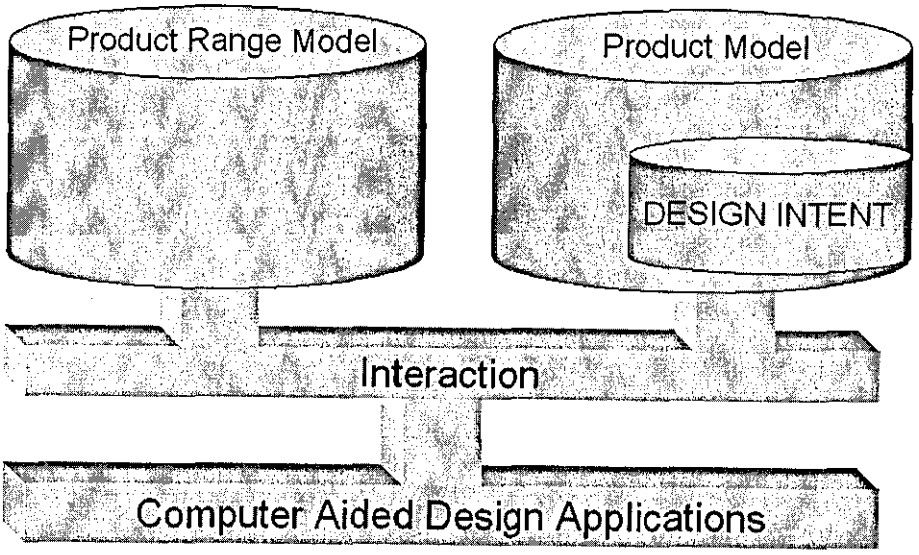

Figure 1.1 Information Models focused on the Reuse of Information

1.2 AIM AND OBJECTIVES

The aim of this research has been to explore and define information structures which can be used to support the designer in the decision making process in terms of intent. This leads to the following objectives:

1. To understand the current approaches related to Design Intent and identify missing gaps.
2. To define an information structure, which can capture the intent during redesign.
3. To design an experimental software system capable of capturing, storing and retrieving the information of the Product Model.
4. To populate the experimental system with data obtained from an actual redesign case study to perform experiments for testing and validating the proposed ideas.
5. To evaluate the results achieved by the experimental system.

1.3 SCOPE

In principle the ideas explored in this research can be applied to the development process of any electromechanical product. However this research

was developed by focusing on particular products from simple drinks containers through to washing machines and finally, in the case study experiment, to a machine developed in the Manufacturing and Design Centre of the National Autonomous University of Mexico (*Universidad Nacional Autonoma de México, UNAM*), co-supporter of this research.

1.4 UML

To support the development of the Product Model structure the Unified Modelling Language (UML) was used as an object oriented analysis tool. This tool is accepted as a standard modelling language and provides the basis of a *de facto* standard in the domain of object-oriented analysis and design founded on a wide base of user experience (Quatrani, 1999).

Two main UML diagrams were used in this research: Use Case Diagrams and Class Diagrams. The Use Case Diagram is a graphical view of the functions, named "use cases", the actors named "surroundings" and the relationships between them named "includes" and "extends". This diagram was used to document the behaviour of the experimental system.

A Class Diagram is used to model the real world employing 'Classes' and 'relationships' between these classes. As in the real world an object has properties (attributes), behaviour (operations), common relationships to other objects (associations and aggregations), and generalizations (inheritances). In this research the class diagram was applied to define the Product Model structure. The Use Case Diagrams with the Class Diagrams help in the analysis of the system. The description of the class diagram defined in this research can be found in Chapter 4. A more detailed explanation of Use Case Diagrams and Class Diagrams is provided in Appendix A

1.5 EXPERIMENTAL ENVIRONMENT

Three techniques were used to develop an experimental system to test the validity of the product information model, Quality Function Deployment,

ObjectStore and Visual C++ software. A more detailed explanation of how these tools were applied, is found in Chapter 5.

Two important methods were identified to get the target requirements from the customer requirements; these are Taguchi and QFD (Pugh, 1991). This research found that the QFD method was the most direct method to get the target requirements from the customer requirements.

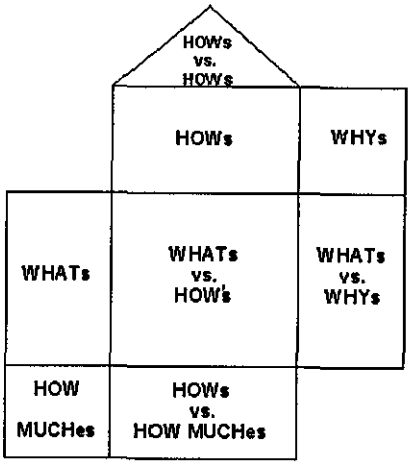

Figure 1.2 House of Quality taken from Ullman 1997

Most of the time customer requirements are vague and difficult to measure, QFD provides a way to translate the customer requirements into quantifiable requirements through the House of Quality. *Figure 1.2 shows the House of* Quality with the basic issues in the different sections of the house to get the target requirements.

ObjectStore© (OS) is an object-oriented database used to implement the information structure of the Product Model. Two tools namely ObjectStore Designer and ObjectStore Inspector support the OS database. The first one supports the design of the Product Model schema based in UML class diagrams as input to generate the Visual C++ code for each class and relationship in the database schema. The ObjectStore Designer helps to visualize the information in the database, as well as the objects and relationships between classes after

the database was populated. The accessibility and availability of these tools are the reasons for their use with Visual C++. A more detailed explanation is given in Chapter 5.

The visual interfaces were developed using Visual C++, which is an object-oriented language. Since OS generates the C++ code for the information model core of the experimental system, Visual C++ in combination with ObjectStore was the most suitable choice to design the experimental system to validate the Product Model.

1.6 THESIS ORGANISATION

The organisation of the chapters of this thesis is shown below.

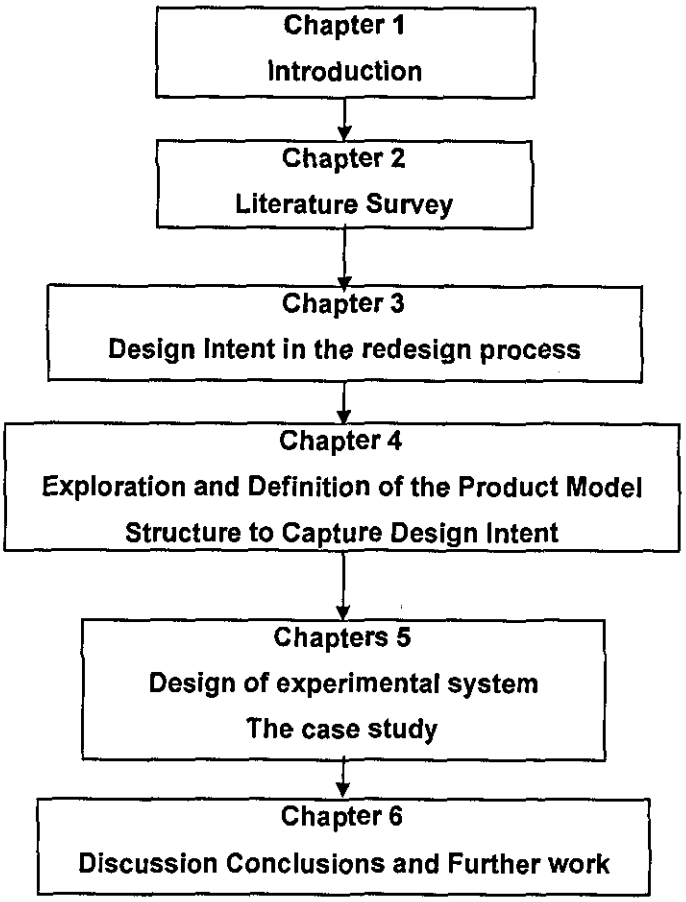

Figure 1.3 Structure of the thesis

2 LITERATURE REVIEW

2.1 INTRODUCTION

This chapter presents the literature review of three areas related to the topic of this research. Section 2.2 reviews the research done on product design and the redesign of products. Section 2.3 presents the main aspects and previous research into decision support systems. Section 2.4 describes the relationship between the previous sections and how important they are to support the redesign of products and understand the Design History of a product.

This literature review helps to understand the contributions made by other people in the related areas and to identify the gap between the proposed research and the previous works. This provides the justification for this research.

2.2 PRODUCT DESIGN AND REDESIGN

2.2.1 The General Product Design Process

A product is the result of any natural or artificial process (ISO10303-1, 1994). Many products resulting from an artificial process are the result of a design process. These products are designed to satisfy the needs of the people in different aspects of their lives. A sequence of activities or tasks is followed to accomplish these products. This sequence is named the *design process* and many authors have proposed many different descriptions in order to formalise this process. This section describes some of the most important descriptions of the design process, however, many other descriptions of it have been explored.

The four main tasks of the design process follow a logical sequence of activities, i.e. clarification of the task, conceptual design, embodiment design and detail design (Pahl and Beitz, 2001)

Following a similar sequence, another process based on different phases, such as the analysis of the problem, the conceptual design phase, the embodiment of

schemes and the detailing phase is proposed by (French, 1971). Both design processes follow a similar sequence of activities and have been followed by many engineering designers.

A different design process considers the systematic activity necessary from identification of the market to the successful selling of the product (Pugh, 1991). This design process is based on a central design core consisting of market or user needs, product design specifications, conceptual design, detail design, manufacture and sales.

All these design processes are focused to the development of products, from some initial customer requirements. However, the customer requirements change and many of the stages of these design process can be applied for the redesign of a product.

2.2.2 Redesign as a Process in Product Development

This research explores decision support systems for the redesign of products therefore some important aspects of redesign of products are reviewed. The first part reviews the idea of reuse of information as an introduction to the redesign of products. The next section concentrates closely on design history as a relevant area in the redesign of products. Finally, areas closely related to design history such as design rationale and design intent, are discussed.

2.2.3 Reuse of Information

Different approaches have been explored related to the reuse of information. Two information schemes have been identified: Process Information, which refers to the information about the reasons or history of the designer's thoughts and Product Information which refers to the attributes of the product (Taura and Kubota, 1999). Based on this classification, a database is built named Engineering Design Base. The engineer designer can retrieve from this database the explanations to enable the reuse of design information.

Variant Design is a technique that supports the reuse of information by retrieving an existing design specification for adaptation to a new design

specification for the design of a new, but similar artefact (Fowler, 1996). This is in line with the design reuse area that is aimed at the reuse of successful past designs partially or as whole in new designs (Sivaloganathan and Shahin, 1999).

On another aspect of the reuse of information, some problems were detected in the behaviour of the designers. In a contradictory behaviour, to find a new design the designer looks for what is already done (Busby, 1999). Therefore a way to improve the design reuse process is to provide better design databases, as well as a knowledge base of qualitative design behaviour, which enables the reuse of knowledge.

To reuse knowledge, the use of technologies, such as Web-based technologies, is important for their friendly characteristics and ability to share, capture and preserve information (Rezayat, 1999). Design reuse can take place by using standard parts from electronic catalogues found on CD-ROM or the Internet (Culley, 1998). This is an option for the designers to obtain well-proven parts as part of a design reuse process.

The computer can be an ally in the redesign process by acting as a colleague to the designer (Intelligent Design Assistant, IDA), providing guidance, learning from past design experiences, carrying out semi and fully automated tasks (Duffy, Brown and Maher, 1996). It can also help to explain his reasoning and in essence complement the designer's own natural skills through understanding the Design History of a product, leaving the final decision, control and responsibility to the designer.

2.2.4 Design History of a Product

Design History is well accepted as being the record of the design activity, the reasoning process behind the decisions made or not made for some choices, and the Design Intent of a product (Al-Salka, Cartmell and Hardy, 1998, Goodwin and Chung, 1997, Jeon, Urban and Shah, 1994, Lee and Lee, 2001, Mostow, 1985, Shah, Jeon *et al*, 1995, Taura and Kubota, 1999, Ullman, 1994). Recording the Design History of a product bring advantages in:

- Tracking the origin of the problem (Al-Salka, Cartmell and Hardy, 1998).
- Promoting a better understanding of the designed product and therefore enabling a faster and easier redesign process (Goodwin and Chung, 1997, Taura and Kubota, 1999, Ullman, 1994).
- Improving the quality of the decisions by having full information available avoiding guessing (Ullman, 1994).

Therefore it is important to record the Design History of a product to support the designer and achieve successful designs.

Design History is an area, where more work is needed to have a reliable system that allows the capture and querying of the information related to the evolution of the product (Ullman, 1994). To have a complete Design History, it is necessary to include more information than simply the drawings or the calculations. Information about the decisions must be considered. Many decisions at many levels are made in developing a product. Part of the work needed around the Design History is in identifying which level of detail in the decision made, known as the *granularity*, must be considered.

In a different approach, Design History is based in the idea that an entity can be associated to a function with three variables: entity, state and version (Barbosa, Feijo *et al*, 2003). The variable "state" refers to different stages of the product life cycle. The variable "entity" represents any element of the product. The variable "version" is associated to an entity at an instant in time. The association of these three variables provides information about the Design History of the product.

Some tools have been developed to capture and use the information related to the Design History of the products. One of these tools consists of three systems: the history capture and recording system, the history representation system and the history playback system. The history representation is the modelling and Design History knowledge base made up of decision, design operator, constraints and the design objects (Chen, McGinnis and Ullman,

1990). This tool acknowledges Design History as an important piece of information that needs to be captured and used.

Another tool related to Design History records all the micro decisions taken during the design process and good results have been achieved by using it (Yakemovic and Conklin, 1989). However, it is accepted that, for the designer, this requires a special effort to record the reasons behind the decisions in an indexed way.

A different approach to Design History makes an analogy with chromosomes (Malmqvist, 1995). In this analogy, the genetic information in a chromosome represents the information of the origin of the design characteristics and the organs are materialised through the components of a product. He considers that the *function-means tree* needs to be extended considering all the design characteristics such as: parameters, materials, tolerances and surface quality so it can include more Design History. This way, the function-means tree becomes a tool to represent the Design History of a product capable of making clear how the function and form of a product co-evolve in a redesign.

Another approach to recording the Design History derives from an Issue Based Information System (IBIS), that records the chain of design decisions (Nagy, Ullman and Dietterich, 1991). This approach uses four data elements: Issue, that is the problem identified; Proposal, that is the solution suggested to solve the issue; Argument, that is the designer's rationale behind the proposal and Decision, that is to accept or reject the different proposals based on the arguments. This system supports the redesign of products considering the different options for the solution of the problem identified, thereby capturing the design rationale.

Design History includes "how" as well as "why" the design evolves as it does (Ganeshan, Garret and Finger, 1994). Two important aspects are identified from them: Design intent is the reason behind the design decisions, the second important aspect is that the characteristics are the values that are measured and which make up the description of the product.

The Design History includes the design rationale and the Design Intent; including the process of the design, that is how the product comes into being. The design rationale includes the reasons behind the decisions and their justification, that is the options not chosen as well as the trade-offs evaluated. The Design Intent is the reason why an artefact exists (Andersson, 2001). Figure 2.2 illustrates the relationship between Design History, Design Rationale and Design Intent.

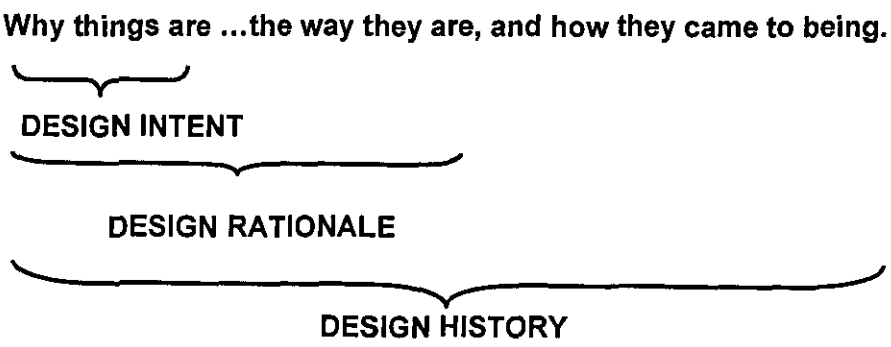

Figure 2.2 Design History including Design Intent and Design Rationale. Adapted from Andersson (2001).

Some research has followed the idea that learning is possible from past designs by capturing the knowledge in the form of design plans. Design plans usually do not capture the failed line of design decisions (Sim and Duffy, 1998). However, this author states that if a design plan is expanded to lower levels of abstraction, then the decisions made form of the Design History and can be captured. Thus the Design Intent, the rationale and the alternatives can be captured too.

2.2.5 Design Rationale

Design Rationale is generally accepted as an explanation that answers a question about why an artefact is designed as it is (Andersson, 2001). Most research on Design Rationale is concerned with developing effective methods and computer-supported representations for capturing, maintaining and reusing records of why designers have made their decisions (KMI, 2000, Shum, MacLean *et al*, 1993).

A design rationale is a representation of the reasoning behind the design of an artefact (Rational, 2001). The purpose of argument-based design rationale capturing methods is to track:

- The discussions and deliberations that occur during early requirement analysis.
- The reasons behind design decisions.
- The changes in the product over the course of its life, whether they are changes in requirements, design, or code i.e. any software artefact.
- The reasons for the changes and their impact on the product.

A Design Rationale is a representation of the reasoning, which has been invested in a design. A method based on Questions, Options a Criteria-Based Design Rationale as a representation to integrate key aspects of the design knowledge invested in a development project, and reactions from the design team is described by (Shum, MacLean *et al*, 1993). This work highlights issues of what kind of Design Rationale designers perceive as being most valuable to preserve, and how to represent and integrate rationales of varying degrees of stability, at different stages in the design process.

2.2.6 The Role of Design Intent

The capture and use of Design Intent is important to understand the Design History and achieve successful design products (Ault, 1999, Fisher, Nakakoji and Otswald, 1995, Gaughran, 2000, Henderson, 1993, Horváth, Rudas and Bitó, 1999, Nielsen, Dixon and Zinsmeister, 1991, Ullman, 2002)

The major issue in developing a Design Intent system is to determine when the information can be managed as direct history, as designer input rationale, as modelled parameters, as automatically parameterised by the systems or as needing inference. It is suspected that a successful Design Intent system will use all of these types of information (Ullman, 1994). The usefulness of a Design Intent system can be measured by its ability to supply users with the correct information when they may have guessed mistakenly without system support.

Therefore, Design Intent systems must be focused to provide the most complete information possible to support the designer.

2.2.6.1 Definitions of Design Intent

Three different approaches have been explored around Design Intent. Some researchers follow the idea that Design Intent is related to the reasons for a product to exist (Andersson, 2001, Henderson, 1993). In line with this definition some methods have been developed to capture Design Intent. Another approach follows a different definition of Design Intent namely: Design Intent is the reasons underlying the design decisions (Ganeshan, Garret and Finger, 1994, Horváth, Rudas and Bitó, 1999, Ullman, 1994). The following sections explain in more detail these approaches.

2.2.6.2 Methods to capture Design Intent

Different types of methods have been researched to capture the Design Intent of a product. These types of systems range from the use of simple notebooks to the use of parametric tools (Kasprzak, 1998, Silva and Chang, 2002). Among the efforts made to support the capture of Design Intent, some work is focused on capturing Design Intent through sketching in the early stages of design (Stevenson, Duffy and Lim, 1999).

Some other efforts have focused on developing unobtrusive systems to capture Design Intent (Jeon, Urban and Shah, 1994). Some of these attempts included the use of video equipment to record the design process followed by the designer (Carrol, Alpert *et al*, 94, Salomons, 1995, Ullman, 1994).

Following the idea of Design Intent as the purpose of an object, a difference with functionality, which describes what the design does, should be established (Andersson, 2001). A prototype system called my_acis was developed as an extension of ACIS (registered trade mark of Spatial Technology Inc) (Henderson, 1993). The purpose of this system is to capture the Design Intent and the functionality of physical objects and their features. This system is based in two realms, the physical realm and the metaphysical realm; the first one captures the physical information of the components, such as geometry,

materials, features, etc, information typically contained in modern CAD systems. The latter is the capture of information related to the relationships between the entities contained in the physical realm, such as Design Intent, functions, needs, constraints, physical principles etc. ACIS uses a number of commands to capture, make queries and retrieve the information related to Design Intent, such as intent, query, show, save etc. This author acknowledges 'function' as a key aspect for the description of a design.

A Design-History Tool must be capable of capturing, recording and playing back the information related to design history (Chen, McGinnis and Ullman, 1990). This method is based on four parts: decision, design operator, constraints and design object. The constraints are divided in three types: given, derived and introduced. The given constraints form part of the specifications of the product, the derived constraints are those generated during the design stage and the introduced constraints come from the outside and are part of the knowledge domain of the designer. The design object is the physical artefact; the features consist of information such as length, width, functional description, Design Intent and behaviour. The capturing is done through audiovisual methods. The playback is done through diagrams and text tables to enable the designer to understand the information of the Design History of the product. The information given relates to 'why' and the 'how' the designer decided on the original or previous object.

2.2.6.3 Design Intent of a Product

A different approach to Design Intent is related to the issue: Why is the product design as it is? To answer this question the author explores the advantages of cognitive modelling to represent the product, simulate how it works and allow refinement of the product. Cognitive ability is the mind's image of an artefact (Gaughran, 2000). The use of this ability with different types of modelling methods such as solid models, computer generated models, mathematical models, etc, is known as *cognitive modelling*. Therefore, to have a good cognitive modelling capacity provides a better description of the Design Intent of the product.

CAD systems have powerful visualisation capabilities (Rosenman and Gero, 1999, Rowe, 1999, Ullman, 2002). The use of shadows, colours, virtual three dimensions, movements, rotations and even simulation capabilities have improved to communicate the purpose or intent of the product between the designer and other users or the members of the design team.

To have accessible the experience of previous problems can help to reduce design time and increase the quality of the products (Werner and Ahmed, 1999). The implementation, of an on-line knowledge capturing system was done with an object-oriented-database. The model developed in this implementation contains functional information and describes the Design Intent of the objects designed.

A Design and Constraint model is proposed, based in the platform STEP, aimed to capture Design Intent based on constraints, parametrics, Design History and features (Shih, 1997). This model contains three modules: Design, Constraint and Parametric Modules. The Design Module contains defining shape information, i.e. vectors, points, lines, etc. The Constraint Module contains rules that apply to all the instances whilst the Parametric Module uses the entities of the Design module and the rules of the Constrain Module to give as a result a "product design". Therefore, a strong relationship is identified between Design Intent and the product described from the geometric point of view and its constraints.

The lack of constraints in a geometric model is a reason which may lead to undesirable changes being made by the downstream users; therefore the Design Intent may be lost (Ault, 1999). Therefore, it is important to use constraints in the geometric model of parts or assemblies to help the capture of Design Intent and avoid or reduce the possibility of inadequate changes by the downstream users.

2.2.6.4 Design Intent Underlying the Design Decisions

Following the idea that Design intent represents the reasons underlying the design decisions, other research has been done and this section describes these approaches.

Exploring the communication of Design Intent between designers, four problems were identified, these are: Communication of Design Intent between humans and computer systems. The second problem is how Design Intent information can be represented in a model. The third problem is how Design Intent can be communicated among concurrent engineering systems and the fourth problem is how Design Intent description can be communicated from the computer system to the engineer (Horvath and Rudas, 1999, Horváth and Rudas, 1998, Horváth, Rudas and Bitó, 1999). The structure proposed tries to solve these problems. However the solution proposed by the authors considers Design Intent as the description of the thinking process of the designer behind the decision. However they constrain to the shape and the associated dimensions for the parts of the product, named in this paper as *'constraints'*.

In the search for the ideal mechanical engineering design support system, some requirements related to Design Intent should be satisfied: Provide information about the problems; provide information about the arguments for or against the options and provide information about the decision reached. The Design Intent systems must record the information about how and why a decision was made (Ullman, 2002). Although CAD systems are beginning to manage this information, a great potential area for exploration around Design Intent is missing if the ideal mechanical engineering design support system is to be achieved.

Going through the rationale systems and the product model structures, more formal techniques need to be explored to manage the information relating to Design Intent, (Ganeshan, Garret and Finger, 1994, Horváth and Rudas, 1998, Jeon, Urban and Shah, 1994, Ullman, 1994).

Assembly features are part of the design features that can be used to create design data (Deneux, 1999). These design features play an important role in the capture of the Design Intent.

2.2.7 Summary

To capture a complete Design History requires consideration of all the design characteristics, materials, functions, surface quality and tolerances, further to the current characteristics considered such as geometry and dimensions.

Currently Design History is poorly considered. Understanding the Design History is necessary to produce successful products. Design History includes Design Intent and Design Rationale. Design Rationale is generally accepted as the reasons underlying the decisions, including the reasoning followed in order to chose certain options as solutions as well as the reasoning for others not being chosen.

Related to Design Intent, three main approaches are being researched:.

- One approach explores unobtrusive methods for the designer to capture Design Intent.
- A second approach explores Design Intent as the reason for a product or its entities to exist.
- Finally, the third approach, followed by this research, explores Design Intent as the reasons underlying the decisions and changes when a product is developed.

2.3 DECISION SUPPORT SYSTEMS FOR REDESIGN

2.3.1 Current Computer Aided Design Capabilities

Despite all the efforts made in the area of design support systems to support the designer in the decision-making process, the ideal design support system has not yet been reached yet (Ullman, 2002). Different philosophies have been explored trying to provide support for decision-making at in different stages of the product life cycle. One of these philosophies is Concurrent Engineering.

The main purpose of the Concurrent Engineering philosophy (CE) is to produce more efficient products in shorter times by integrating all the relevant knowledge at the design stage (Jo, Parsaei and Sullivan, 1993, Molina, Ellis *et al*, 1995, Prasad, 1996, Young, Dorador *et al*, 2001). To get the most out of this philosophy, some tools like computers are of help.

From the computational viewpoint, the Concurrent Engineering philosophy can be seen as an integration of the functional software tools, deployed to enable design team members to share knowledge and information and in the same way, to keep track of other's needs, constraints, decisions and assumptions (Harrington, 1996, Tichem and Storm, 1997). Computer systems, therefore, have become of great help in managing information. The increases in the power of computers, more complex software and new theories have combined to create better ways to support product design (Duffy and Duffy, 1996, Salomons, Slooten *et al*, 1993, Ullman, 2002). Two approaches are usually addressed for the CE implementation namely: team based and computational based (Jo, Parsaei and Sullivan, 1993).

From the *team-based approach* the computer is seen as an assistant providing support for the designer in the decision-making process, while from the *computer-based approach* the computer is seen as a substitute for the designer (Duffy and Duffy, 1996). Following the first approach, two more views are explored: one view sees the computer as an Intelligent Computer Aided-Design tool empowered to learn through the interaction with designers (Bañares-Alcantara and King, 1997, Duffy and Duffy, 1996); the other view explores the use of the computer as a repository of information shareable between the members of the design team (Rezayat, 1999, Ullman, 2002, Young, Canciglieri-Jnr *et al*, 2000). The research reported in this thesis follows this latter view.

Most CAD systems provide the possibility to capturing some of the Design History of a product (Ullman, 2002). However, this is constrained to the capture of the chronological changes made to the components designed, the time and date of change and, to some extent, comments for the reasons behind the

changes. Therefore more work is needed to make CAD systems capture the Design History in such way that it represents a valuable support to the designer by providing the capability to capture the reasoning underlying the decisions.

2.3.2 Intelligent Support Systems Approaches

2.3.2.1 Introduction

The increasing power of computers has lead to the development of tools and techniques to support the design activity, using the computer as an intelligent tool with the capability to make decisions; this section describes some of the most important approaches explored.

2.3.2.2 Case-Based Reasoning

Case-Based Reasoning (CBR) is a problem solving technique based on the adaptation of previous examples that are similar to the current problem (Maher and Garza, 1997).

The main explanation underlying CBR systems, is based on the fact that human problem solving does not always involve reasoning from first principle, but alternatively may be a matter of relating information about a problem to past experience of solving problems (Lees, Chen and Lee, 1997). This argument provides some advantages for CBR systems compared to Knowledge Based Systems, such as not needing an explicit domain model and identifying only important features that describe a case (Maher and Pu, 1997, Watson and Marir, 1994). Also, CBR systems use actual past experiences to learn and solve new problems, rather than generalised heuristic rules, as in knowledge-based systems/expert systems.

Since CBR uses previous cases to solve design problems, some key research is done to define similarity between cases and how to use temporal information (Encyclopedia, 2004).

This research is not focused on reusing similar cases but in reusing the Design History information of a product.

2.3.2.3 Agent Technologies

Agent technologies refer to autonomous or semi-autonomous pieces of computer software and contribute to the understanding of information integration of multiple viewpoints by communicating with other software agents to form a Multi-Agent System (Encyclopedia, 2004, Genesereth and Ketchpel, 1994, Wooldridge and Jennings, 1995).

Being autonomous or semi-autonomous implies that agents do not need to know about other agents, the communication is done through high-level languages that provide the capability to make statements about their internal state and the domain in which they interact.

Despite the advantages agent systems offer, there are problems with the communications or translation between different domains due to the diversity of the information involved in the multiple viewpoint environment.

2.3.2.4 Fuzzy Logic

Fussy Logic relates to the used of Boolean logic. Fuzzy Logic is based on the idea that everything can be expressed in binary terms. Fuzzy Logic replaces Boolean truth-values with degrees of truth, which are very similar to probabilities. This allows values between 0 and 1. Fuzzy Logic is polemical. Some critics argue that it cannot be a superset of ordinary set theory since membership functions are defined in terms of conventional sets. (Encyclopedia, 2004).

Fussy Logic has been applied in the control of household appliances such as washing machines i.e. selecting the most appropriate cycle according to the concentration of the detergent and the size of the load.

2.3.2.5 Expert systems

An Expert System is decision-making software based on the expertise of the users of the system that can reach a level of performance comparable to the performance of a human expert (Liao, 2001).

The main characteristics of the Expert Systems can be briefly described as: an increased output, productivity and quality; a capture of limited expertise and its diffusion, accessibility to knowledge, ability to work with incomplete information, provision of training, enhancement of problem solving capabilities and reduced decision making time. Despite these advantages some mistakes were detected in the expert systems due to lack of maintenance of the information used, the lack of standards that have not been forthcoming (Williams, 1990), and the lack of sophistication (Encyclopedia, 2004)

This research sees the designer as the ultimate decision maker, using the computer technology as a tool to make better decisions.

2.3.2.6 Object Oriented Approaches

Some work has done to integrate an object-oriented method to CAD systems to capture the Design History and Design Intent. A system proposed seeks integrating the store and use of the design information with the shape manipulation of CAD systems to achieve a system that can incorporate the Design History of a product (Lee and Lee, 2001). This author considers that designers see Design Intent as the goals and design parameters. This system provides an efficient use of the information for re-evaluating and redesigning products.

In line with the above research, some process-modelling extensions are presented to provide object-oriented databases with the capability to deal with versioning and history mechanisms. Object-oriented databases are considered useful for capturing the description of the product. (Jeon, Urban and Shah, 1994). The research reported in this thesis seeks to determine how object-oriented databases can capture and use information related to the Design History of a product.

2.3.3 Information Management Systems

2.3.3.1 Introduction

During the development of a product a huge amount of information is produced. The correct use of this information provides the support to make better and faster decisions (Young, 2003). Some of the most important information management systems explored are described in this section.

2.3.3.2 Product Data Management Systems

A different type of decision support systems is the Product Data Management (PDM) system. PDM systems are focused to manage the information produced making it exchangeable by following STEP. PDM systems need to be linked to CAD systems in order to reduce time in the management of information. PDM systems have the ability to manage versioning of the products (Oh, Han and Suh, 2001, Ullman, 2002).

Most commercial PDM systems are based on five basic management functions (Chen and Jan, 2000, Chen and Tsao, 1998, Liu and Xu, 2001):

1. Data vault and document management for storage and retrieval of product information.
2. Workflow and process management, focused to control the procedures for handling product data and support the driving of a business with information.
3. Product structure management, to handle the bills of material, the configuration of the product, versioning and variation in designs.
4. Parts management, aimed to provide information on standard components as well as re-use of designs.
5. Program management, which allows coordination between processes, scheduling of resources and project tracking.

Despite the extensive development of these systems, a number of disadvantages have been identified such as the lack of ability to support the early stages of the development of the product, as well as to represent the

product in a formal way. (Gao, Maropoulos and Cheung, 2003, Sharma and Gao, 2002, Szykman, Fenves *et al*, 2001). Additionally, present PDM systems are unable to respond sensitively to the frequent changes of the design parameters (Chung and Lee, 2002).

2.3.3.3 Product Lifecycle Management Systems

Product Lifecycle Management (PLM) is an integrated business approach that allows the design, analysis and management of the information of the company's products throughout the product lifecycle, i.e. from the initial conception to the retirement of the product (PLM03, 2003). Two main tools have been identified as being involved in PLM: tools to create the product content, such as MCAD/CAE; and tools to manage the process concerning the lifecycle of the product from conception to retirement of the product. Despite the many areas involved in the PLM, there is a lack of research in issues related to the early stages of the product design, especially in identifying and deciding the important requirements of a product.

In line with PLM systems, some attempts have been made to develop an information model pattern in Unified Modelling Language (Dhar and Rangani, 2003). This is aimed at the requirements of the initial stages and product characteristics information to support autonomous decision-making and the redesign across projects. This approach was proposed for application in the automotive industry.

Focused to capture issues related to the evolution of the product, a framework information model to represent products was described (Fenves, Sriram *et al*, 2003). This framework is intended to capture product information, design rationale, assembly and tolerance information from the initial design stages to the full lifecycle. This is done to facilitate the interoperability of next generation CAD/CAE/CAM systems. This information-model framework is aimed to provide access to the product's description and design rationale for the families of products.

Following the efforts to capture information on the reasoning behind a product, a well-structured Knowledge-Based Engineering system was proposed that allows the capture of the engineering intent and the reuse product development experience gained during the different phases of the product life cycle (Shyamasunder and Rao, 2003). This paper introduces a Product Model and a Design Process Model to capture the reusable true design intent making it available throughout the enterprise. Therefore sharing information is important in the PLM systems.

2.3.3.4 Information Sharing Through the Use of Information Models

Information modelling has been increasing its importance because it is a shareable source of information between the members of a design team, exchanges information between different systems following the STandard for the Exchange of Product model data (STEP) and has the flexibility to model complex problems (Dorador-Gonzalez and Young, 1999, McKay, 1993, 1994, Molina, Ellis *et al*, 1995, Young, Canciglieri-Jnr and Costa, 1998).

The modelling of complex engineering situations helps to provide an understanding of different types of problems in developing products and produce better solutions (Arnold and Kunz, 2000, Ault, 1999, Barbosa, Feijo *et al*, 2003, Belhe and Kusiak, 1996, Brazier, Pieter and Treur, 1997, Chen, McGinnis and Ullman, 1990, Molina, Ellis *et al*, 1995, Young, Canciglieri-Jnr *et al*, 2000). Information models are some of the main elements in major CAE system architectures to support design and manufacturing applications throughout the product life (Costa, 2000, Kimura, 1992, Krause, Kimura *et al*, 1993).

Another information model, the Manufacturing Model, describes available manufacturing processes such as injection moulding, machining processes, resources-machines, tools, fixtures and strategy -how these resources and processes are used and organised-, providing a consistent source of manufacturing information and knowledge for applications (Ellis, Molina *et al*, 1994, Guerra-Zubiaga, 2004, Harding and Popplewell, 1996, Liu, 2004). The Manufacturing Model has four levels based on a generally accepted standard,

namely, Factory, Shop, Cell and Station, which represent the functionality of the manufacturing facility of any firm. It has been modelled using information related to machining and injection moulding.

Therefore, information models have become an effective tool to manage information in a shareable environment used by the information support systems to provide support for the engineering designers. The research reported in this thesis focuses on the Product Model in the belief that it has the potential to capture the Design History and support the redesign of a product.

2.3.4 Summary

Decision support systems are increasing their importance in supporting the development of products. Development of these systems following the Concurrent Engineering philosophy has strengthened the use of information model structures due to their capacity to share information between the members of the design team, exchange information between different systems following STEP, and the flexibility to model complex problems.

Within the Concurrent Engineering philosophy the Product Model (PM) stores all the information of the product life cycle. A main advantage of this Product Model is the shareability of the information for all the members of the design team and the exchangeability of information among different applications.

2.4 INFORMATION MODELS AND DESIGN INTENT

2.4.1 Approaches to Product Information Reuse

2.4.1.1 Product Range Models

One approach followed to support product range definitions, is mainly focused on representing the most suitable product architecture defined in the product model (Baxter, Juster and de Pennington, 1994, Gorti, Gupta *et al*, 1998, McKay, Bloor and De Pennington, 1996). This approach can provide support for composing variants of existing designs based on sets of product specifications.

A further model has also been researched, called the Product Range Model (Costa, 2000, Young, Dorador *et al*, 2001). The purpose of this model is to capture information and knowledge, based on Design History, of the range of ways in which a product can be designed.

2.4.1.2 Product Models

The Product Model is central to the concept of information to support a concurrent engineering environment. The Product Model contains information about the product life cycle and is based, wherever possible, on the evolving STEP standard (Dorador-Gonzalez, 2001, Ellis, Molina *et al*, 1994, Harding and Popplewell, 1996, McKay, Bloor and De Pennington, 1996, Young, Dorador *et al*, 2001). The Product Model is a source and repository of information for many applications, and allows information to be shared between the many users and software components of CAE systems. Thus, all agents involved in the design process (humans or software) must actively take part in information-sharing by using the common product model database.

Based on a product model structure a new web based infrastructure is proposed to capture and reuse the Design History (Shah, Jeon *et al*, 1995). This infrastructure is based in the STEP models to capture the design steps and the design parameters and follows the design rationale based on IBIS. Figure 2.1 shows the elements needed for this infrastructure. He considers that the databases need more work to deal with versioning of the products.

This Design History Model is an important development because it is aimed to enhance the participation of experts of different areas through the Internet. This research identifies some important elements related to the Design History of the product.

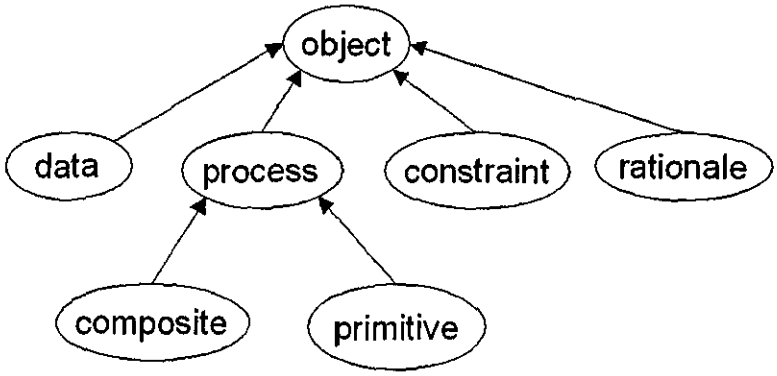

Figure 2.1 Elements in a Design History Model, taken from Shah (1996)

2.4.1.2.1 Product structure

From a BOM point of view, the structure of a product can be described in terms of assemblies, subassemblies, and parts (Dorador-Gonzalez, 2001). The structure of the product in this Product Model is most suitable for assembly purposes.

A different approach to describe the structure of a product is based in the function of the assemblies; these assemblies are named systems and subsystems (Ullman, 1997). For design purposes this description is more suitable because design seeks the accomplishment of the functionality of the product.

A different product model structure follows the description of the product in terms of product version and product entities. These entities are composed by shape definitions, features and relationships among the shape definitions (Shah, Jeon *et al*, 1995). In order to have a complete description of the product, three different contexts are applied: 'as specified', 'as designed' and 'as manufactured'.

This research explores the best formal representation of the structure of a product to support the decision making process from the design point of view.

2.4.1.2.2 Version Control

The concept of Version Control must be extended in the product model, to describe the specific relationships of versioned data to process objects.

Therefore a Design History data model should support the evolution of processes as well as of data objects. On the other hand, Version Control based in databases can be used to create an environment, which could automatically manage versions and alternatives of a design (Shah, Jeon *et al*, 1995). This author acknowledges that it is difficult to establish what level of change or changes defines a new version.

2.4.1.2.3 Multiple Views of Information

A product can be described from different points of views: i.e. the design, the manufacturing and the assembly point of view. Product models are beginning to explore these multiple views by which a product can be described (Dorador-Gonzalez, 2001).

The knowledge about the relationships between the viewpoints is necessary to transform the information between the different viewpoints (Canciglieri and Young, 2003). A new product model was developed seeking the integration knowledge to support the information integration between multiple viewpoints. This is important to transform the information between viewpoints and achieve a balanced product solution (Gunendran and Young, 2002).

The modelling of multiple viewpoints in the product model helps in the integration of the information from the multiple viewpoints to achieve integrated solutions.

2.4.2 Methods to Support Information Modelling

2.4.2.1 Introduction

As information involved in the development of a product has a great level of complexity, disciplined methodologies and languages are required to capture and represent this information and the way that it is changed (processes), providing a reference for discussions and software systems design and implementation (Court, Culley and McMahon, 1995, McKay, Bloor and De Pennington, 1997).

2.4.2.2 RM-ODP

International Standard Organization/Reference Model for Open Distributed Processing (ISO/RM-ODP) is a standard for open distributed processing, aimed to enable the interaction between the systems and the applications. This standard allows the development of system architectures which are themselves open, achieving interoperability and portability among their individual component systems (Blair, Coulson and Davies, 1996). This standard defines a general framework upon which the system must be developed. However, formal languages or methodologies to denote specific levels of such as frameworks are still required.

One of the foundations of the RM-ODP is a model of multiple viewpoints, which facilitates different participant's viewpoints to observe a system form different perspectives at a suitable level of abstraction (Encyclopedia, 2004).

The RM-ODP highlights more importantly the content of the important views of the system and does not influence how the information system should be designed and implemented, or which tools and techniques should be used to design and implement the requirements represented at each viewpoint. Therefore, to provide support and consistent, efficient guidance for to the design and implementation of the software system progressing through each level of the RM-ODP, different formal languages and techniques can be applied. One of these techniques is the IDEF0.

2.4.2.3 IDEF0

This is a generally accepted standard modelling technique used to some extent with success to model enterprise activities and information flows (Dorador-Gonzalez and Young, 2000).

IDEF0 is strongly recommended for projects that:

- Have the need for a modelling technique for the analysis, development, re-engineering, integration, or acquisition of information systems;
- Include a system or enterprise modelling technique within a business process analysis or software engineering methodology (**IDEF0**, 1993).

However, IDEF0 has a weakness because it cannot model process flows. Therefore it is necessary to complement this technique with other methods and techniques and languages, such as EXPRESS.

2.4.2.4 EXPRESS

Express is a data definition language developed and used by the STEP community (Ellis, 1993). EXPRESS language is an object-oriented language for information modelling. A graphical variant of EXPRESS is identified in The EXPRESS Language Reference Manual called EXPRESS-G.

EXPRESS was originally developed to provide a formal, and computerised means of defining the data necessary to describe any product through its lifecycle. The ISO 10303, commonly known as STEP (STandard for the Exchange of Product model data), uses EXPRESS as the formal specification of the required data and its relationships (Wilson, 1998).

2.4.2.5 UML

The research reported in this thesis uses the object-oriented technology to define information structures and development of software applications due to its versatility, which enables it to model any product. One of the latest and more important developments in object-oriented technology analysis, design and representation is the Unified Modelling Language (UML). It allows capturing the functionality until the final representation of the objects and their properties into an object-oriented system (Booch, Rumbaugh and Jacobson, 1999, Jacobson, Booch and Rumbaugh, 1999, Texel and Williams, 1997). UML follows the STEP standard to define the classes and their relationships, making UML a powerful tool for the sharing of information among the various CAD applications. How this notation was applied in this research is explained in Chapter 4.

Some approaches start to develop to capture the versioning of the artefacts using UML (Kovse, 2002). This research is in line with this approach by using UML as a standard language to capture the design history of a product.

2.4.3 Summary

There is an agreement within the research community on the fact that the capture and playback of the design history of a product is important to avoid loss of information, help in the tracking of possible problems and provide the means to understand the reasons underlying the different decisions made during the design or redesign of a product.

Most products have been the result of a redesign process; therefore it is important to understand the reasons underlying the decisions made in a redesign process, this is the Design Intent. Despite the importance of Design Intent, this is an area that has not been fully explored.

A number of different design support systems have been developed to support different or all the stages of the development of the product, e.g. the PLM's, as well as to manage the information and share it throughout the whole enterprise, e.g. the PMD's. However, despite the acknowledged importance of supporting the all the stages of the product lifecycle, little work has been done to develop tools for the support of the redesign process.

This research identifies two gaps: one in the exploration of Design Intent as a valuable piece of information in need of further research and the other in the understanding of the tools used to support redesign

3 DESIGN INTENT IN THE REDESIGN PROCESS

3.1 INTRODUCTION

This chapter identifies the contribution of this research, defines the key issues of this research and provides an outline of how the problems were addressed. Three main subsections identify issues related to: design support systems, information modelling to support redesign; and Design Intent in the product model to support the redesign process.

As well as presenting the problems addressed by this research, this chapter sets the basis for the resulting research described in the following chapters.

3.2 DESIGN SUPPORT SYSTEMS

3.2.1 General Issues

Concurrent Engineering (CE) is a well-accepted philosophy in the research and industry communities. This philosophy is based on the simultaneous participation of the experts in the different stages of the product life cycle, from the early stages of the development of a product (Prasad, 1996). The aim of CE is to develop higher quality products at lower costs in shorter times. To achieve these aims, sharing information is critical. Information model structures provide the means to share information and support the designer in the decision-making process (Young, Dorador *et al*, 2001). In a re-design process, re-use of this information is vital to produce better versions of the product (Dixon and Colton, 1998, Otto and Wood, 1998, Ullman, 1997).

As previously highlighted in the literature survey some issues that need to be addressed, related to the reuse of information, were identified and listed below.

I The information related to design history has been poorly considered in current design support systems

II In the information modelling area, little work has done to support the redesign process.

III Many researchers recognise how important it is to communicate Design Intent among the designers of a product. However, little work has been done related to Design Intent in the design support systems.

IV Other important characteristics, in addition to the geometry and dimensions generally identified as the main changing characteristics when a product is redesigned, need to be considered. These are: function, tolerances and material.

V Some research has been done to develop non-intrusive methods to capture information related to Design Intent.

To provide a complete redesign support system, the above issues should be solved. However this research focuses on issues I to IV because of their strong relationship with information model structures to support the concurrent engineering philosophy. These four issues are discussed in the following sections to establish the line of research reasoning of this work.

3.2.2 Computational Design Support Tools

Computational tools play an essential role in providing support for the designer, because of their speed and capability for handling huge amounts of information at fairly low costs. Despite this advantage, to identify which is the appropriate information that must be supplied to the designer is still an issue in providing support for effective decision-making. Decisions made in the early developing stages of the product influence the manufacturing cost by up to 50% (Ullman, 1997). Therefore research in design decision support systems is an important area with the potential to provide significant benefits to industry.

A variety of work around design support systems for engineering designers is targeted to provide support in the early stages of the design of new products and little work has been done to support the redesign task. Redesign and design reuse are important issues which have in common the continuous

search for the higher efficiency of a product (Dixon and Colton, 1998, Duffy, 1996, Hsu and Lin, 1998, Salomons, Slooten *et al*, 1993).

Design and redesign processes involve similar stages, however two major differences involve the starting point and result in both processes. The starting point in the design process is from scratch whereas for the redesign process the starting point are the design specifications of the previous version (Dixon and Colton, 1998, Dixon and Colton, 2000). The result of a design process is a new product while the final result of a re-design process is generally a new version of an already designed product (Dixon and Colton, 1998). Chapter 4 describes how the evolution of a product might have a bearing on the development of a new product. Despite these differences, stages between the starting point and result are similar (Ullman, 1997). This similarity makes possible, to some extent, the use of commercial design support systems for both processes. Product Data Management (PDM) is emerging, with some success, as a design support tool for the design and redesign processes. Despite the great advance of this tool, more work is needed to capture the design history of a product. Currently, some of these systems can manage variations in different versions; however, only limited support for the designer in the decision process is achieved. (Ullman, 2002, Ungerer and Buchanan, 2002)

This thesis contributes to the area of design support systems, by exploring the capture and storage of information produced during the redesign process of a product through information model structures.

3.3 INFORMATION MODELLING TO SUPPORT REDESIGN

Information model systems provide sources and repositories for information and knowledge related to the life cycle of the product (Young, Dorador *et al*, 2001).

Information modelling has been applied successfully in several areas related to redesign such as the reuse of information applied in the selection of a predefined range of solutions which satisfy a set of functions in Costa, (2000) and reverse engineering (Borja, 1997). This research is in line with the use of information models to support the redesign of products.

Some research has defined additional information models beyond the Product Model Structure (PMS) to provide support and information to the engineering designer. For example Costa, (2000) identifies the Product Range Model to capture the information and knowledge related to the ways in which a product can be configured. However, this research considers that the evolution of the design needs to be captured in a Product Model. Therefore, the issue is how to structure a product model in such a way that it can capture the design history of the evolving product.

3.3.1 Information Related to Design Intent

Design Intent, as part of design history, is an important piece of information and, as a topic, has not been deeply researched (Ullman, 2002). Recording design history brings many benefits for the evolution of the product (Goodwin and Chung, 1997, Malmqvist, 1997, Salomons, Slooten *et al*, 1993, Shah, Jeon *et al*, 1995, Taura and Kubota, 1999, Ullman, 2002). Design history consists of many pieces of information such as product data, e.g. calculations drawings, sketches, written notes; and process data, e.g. assembly, manufacturing; and the interactions between both types of data (Shah, Jeon *et al*, 1995). However, more research is needed to explore Design Intent as this will help to better

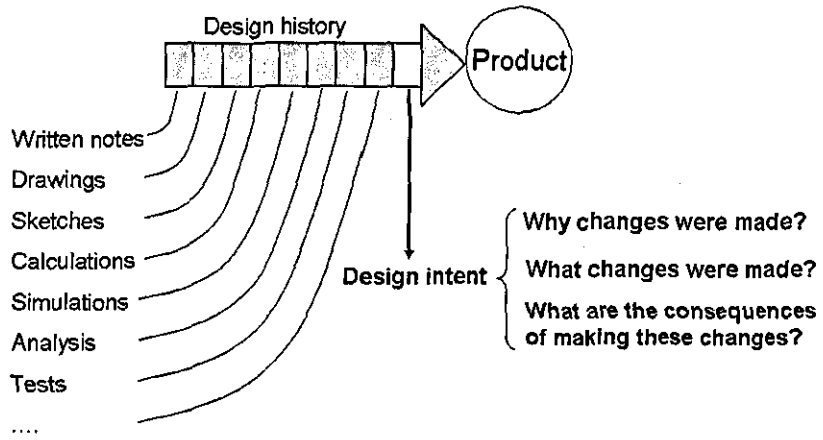

Figure 3.1. Design History

understand the design history of a product and influence the redesign process. Figure 3.1, shows the pieces of information that comprise the design history (Goodwin and Chung, 1997, Shah, Jeon *et al*, 1995, Ullman, 1994).

The research and industrial communities follow the next interpretations of Design Intent:

The first interpretation relates to the geometric relationships in the parametric Computer Aided Design (CAD) systems that help to record changes. However the order of the reasoning followed by the designer to design or redesign the product is not considered (Silva and Chang, 2002, Ullman, 2002).

The next interpretation relates to understanding the reason for the product to be designed through the improvement of the visualization capabilities of the software (Autodesk, 2002). Many commercial CAD developers follow this interpretation. Therefore much effort has gone into improving the abilities to display the product on the computer screen by simulating lights, shadows, rotation, animation and rendering. (Rosenman and Gero, 1999, Sim and Duffy, 1994).

The last description relates to the reasons behind changes and decisions made to a product when it is designed or redesigned. This concerns understanding capture, storage and querying information related to Design Intent (Espinosa and Young, 2002, Ganeshan, Garret and Finger, 1994, Henderson, 1993, Ullman, 2002).

Following this last interpretation, some researchers have focused their efforts on the methods for capturing Design Intent, which go from the use of notebooks to make simple notes to the use of multimedia devices (Ishino and Jin, 2002, Myers, Zumel and Garcia, 2000, Ullman, 2002).

This research provides a contribution in line with the third interpretation above, targeted to explore the reasons underlying the changes and decisions in the redesign of a product.

3.3.2 Characteristics of the Product Related to Design Intent

Redesign of a product shapes the new version of a product. Different characteristics of the elements, which comprise the product, change. Most research into Design Intent has focused on the capture of information related to changes in geometry, tolerances or dimensions (Ault, 1999, Horváth and Rudas, 1998, Horváth, Rudas and Bitó, 1999).

This work goes beyond considering, dimensions, tolerances and geometry and believes that materials, requirements and functions are equally important issues to consider in a redesign.

3.3.3 Classifying Design Intent Information

Any change has consequences (Henderson, 1993, Horváth and Rudas, 1998), but little has been done to make the designer aware of the consequences in a redesign (Ullman, 1994).

Although redesign seeks to benefit the product, some negative consequences may arise. Assessing the positive and negative consequences is important in evaluating the actual advantage of the new version.

Classifying these consequences in areas of influence around the product is important to provide a better support to the designer. (Pugh, 1991) identifies many elements that define the 'Product Design Specifications' (PDS), which sets the context of the design. The more elements considered when defining the PDS, the higher the chances of achieving a better product. Some of these elements relate to the product life cycle, e.g. disposal, maintenance, life in service, etc; other elements relate to the physical properties of the product, e.g. weight, size, material; some other elements relate to legal issues of the product, e.g. legality, standards, patents; other elements relate to behavioural aspects of the product, e.g. performance, efficiency, etc. This research explores which of these elements are considered to be a major influence on Design Intent and which areas can be grouped into other higher-level areas to simplify the capture of Design Intent.

Following the previous analysis, if an information model is to be able to support Design Intent, then it must be able to capture information to answer the following three questions, see Figure 3.1:

Why were these changes made?
What changes were made?
What are the consequences of making these changes?

To build an information model to capture the answers to these questions, this research explores four areas to classify information related to Design Intent:

1. *Configuration* of the product.
2. *Characteristics* of the product that influence the functionality.
3. *Views* to describe the version.
4. *Versions* of the product.

The next section introduces the limitations of existing product model structure to support this kind of information.

3.4 DESIGN INTENT IN A PRODUCT MODEL TO SUPPORT THE REDESIGN PROCESS

3.4.1 Initial Product Model Structure

It is accepted that the inclusion of Design Intent in product models and design support systems has great importance (Henderson, 1993, Horváth and Rudas, 1998, Ullman, 1994). PMS, the Product Model Structure, provides the capability to capture and store the design information of the product and provide support for the designer in the decision process (Young, Dorador *et al*, 2001). This research has as a starting point the PMS proposed by Dorador-Gonzalez, (2001) shown in figure 3.2. This PMS tends to focus on the Bill Of Material (BOM) style configuration of the product and is not well developed to deal with the history of a product. It captures information of the characteristics, the configuration and the views of the product but leaves out the versioning of a

product when it evolves. These classes seem to fit to some extent with the description of the product followed by this research. The areas: "Characteristics", "Views" and "Configuration" in the initial model seemed in general to fit with the ideas described in this chapter. However an improved PMS is needed to add the capability to capture the evolution of the product and its relation to design intent.

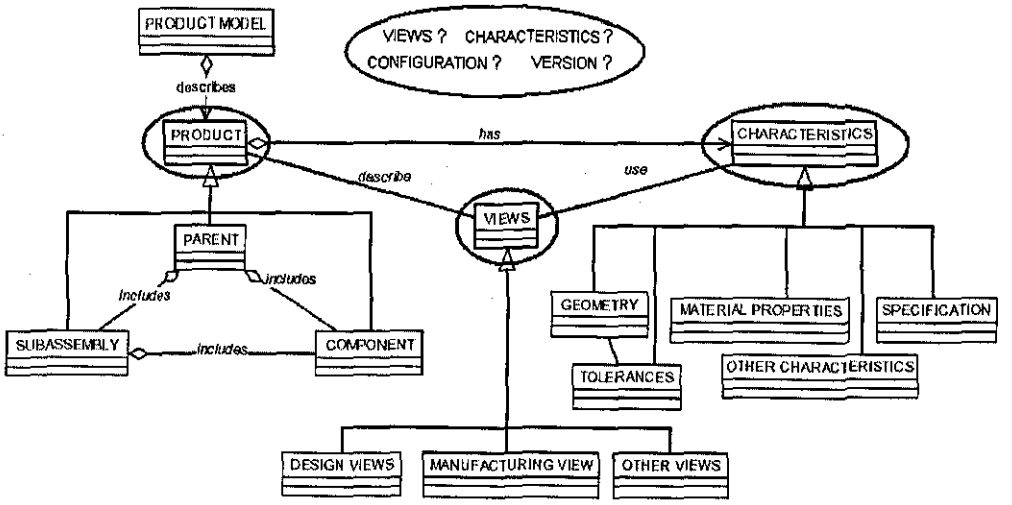

Figure 3.2 Product model structure used a start point for this research, adapted from Dorador, (2000).

Figure 3.3 illustrates the four areas around the product related to Design Intent explored in this research. Further to the three areas identified in the initial PMS the area "Version" is a new area focused on capturing information about the evolution of a product (ISO10303-203, 2000). Storing information of how the product evolved is an important part of the design history of the product to be captured in one database system. This helps to have access to all the

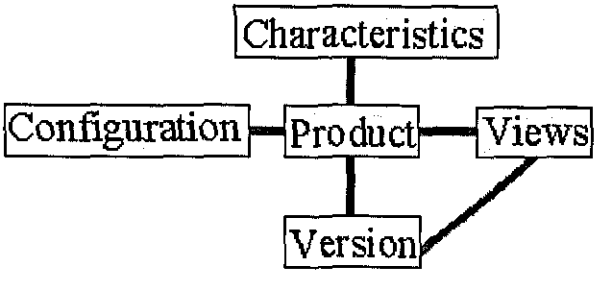

Figure 3.3 General structure of areas related to Design Intent

information related to the versions in a more direct and faster way (Shah, Jeon

et al, 1995). This research explores how these areas and their interactions can be structured to capture the information related to the changes in a version, the Design Intent underlying these changes and the consequences of the changes.

The major contribution of this research is to provide a product information structure which deals with the areas introduced above. The way this is explored, is explained in detail in Chapter 4.

3.5 SUMMARY

This chapter presented an outline of the basic contributions of this research, related to the issues identified in the previous chapter, in the context of the information modelling to support the redesign activities.

Three questions arose from the analysis done to the information related to Design Intent: What changes were made? Why were these changes made? And what are the consequences of making these changes?

To answer these questions, four areas named (Figure 3.3): Structure, characteristics, views and version, were explored. A new PMS based on these four areas will be discussed in the next chapter.

4 EXPLORATION AND DEFINITION OF THE PRODUCT MODEL STRUCTURE TO CAPTURE DESIGN INTENT

4.1 INTRODUCTION

This chapter discusses the four key areas of product information introduced in the previous chapter and illustrated in Figure 4.1. This chapter explores and defines a new Product Model Structure (PMS). In the following subsections five different product examples are used to explore the issues and support the definition of the structures. UML has been used to define the PMS structure. The last section explores and defines the internal relationships between the defined structures.

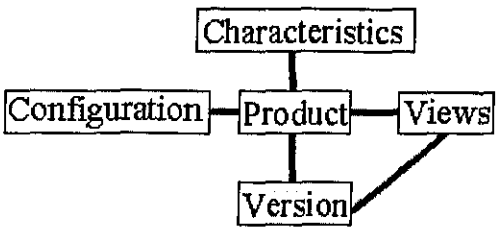

Figure 4.1 General Structure of Areas Related to Design Intent

4.2 CONFIGURATION OF THE PRODUCT

The configuration of a product can be described in different ways from different points of views. However, independently of the point of view, different types of entities comprise most products. Through a BOM point of view the configuration includes information about the name and quantity of components, material and origin of the part, bought outside or manufactured within the facilities of the company (BS5191, 1975, Dorador-Gonzalez, 2001, ISO16100-1, 2001). From this point of view, assemblies, sub-assemblies and parts comprise a product. Figure 4.2 illustrates this. This description although useful for the manufacturing and assembly process, is limited from the design point of view.

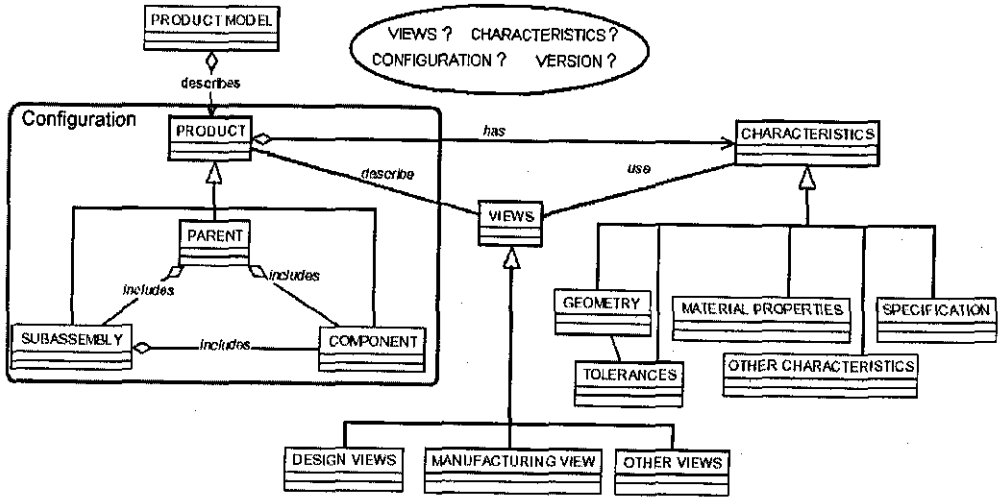

Figure 4.2 Configuration of a Product in the Founding Product Model Structure.

From the design point of view, it is important to illustrate two linked issues: the Design Intent underlying the changes and decisions and the functionality of the product (Espinosa and Young, 2002, Henderson, 1993 #66). This is because important design decisions are concerned with functionality (Malmqvist, 1995) and redesign pursuits accomplishing the function of the product through better quality, and cheaper, faster produced versions. Therefore, to catch Design Intent, a description of the configuration of the product needs to consider function.

A system is generally accepted as a collection of objects that performs a specific function based on a working principle (Hubka, 1980, Ullman, 1997). Following the argument set out by (Ullman, 1997), a mechatronics device, such as a photographic camera, is used to illustrate the importance of a systems approach. Following the description of systems stated above, the camera is itself a photographic system. Systems may be comprised of other systems at lower levels with more specific functions and working principles: e.g. a shutter system is part of the exposure system, which comprised the light meter part among many other entities with defined functions, see Figure 4.3.

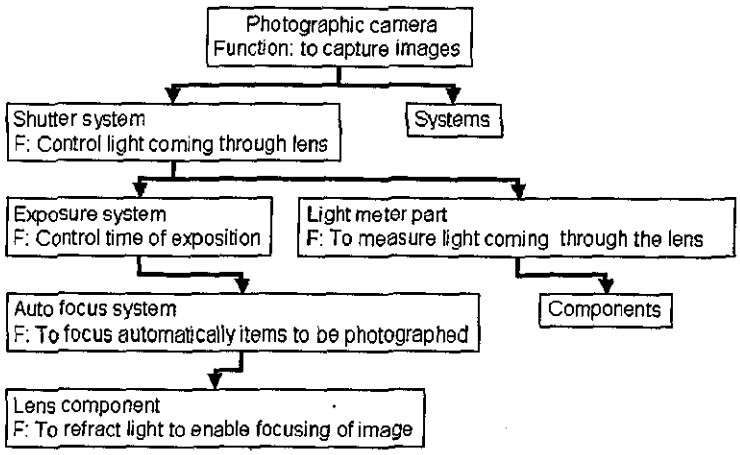

Figure 4.3. Example of Configuration of Product Adapted from (Ullman, 1997).

This leads to the exploration of a structure that describes a different configuration based on systems.

4.2.1 Function in the Product

Function is accepted as the desired task accomplished by an entity to solve a problem (Hubka, 1980, Rosenman and Gero, 1999, Ullman, 1993). Each entity within the product has a function, which harmonized with the other entities named systems, subsystems, parts and components, accomplishes the overall function of the product.

4.2.1.1 Systems, subsystem, parts and components

Following the description of system, a product can be described as a system comprising systems at lower levels, which can be divided into subsystems with more specific functions (Johanneson, 1993). To make a difference, the prefix "sub" was used to identify the subsystems that comprise the systems.

Parts and components comprise the systems and the subsystems (Johanneson, 1993, Ullman, 1997). Some authors use the terms component and part synonymously (Hubka, 1980, Ullman, 1997). However, a different meanings of 'part' and 'component' are used here. A component, in line with

(ISO10303-1, 1994) and (Ellis, 1993) is not subject to decomposition and is therefore made from the same material or composite material.

4.2.1.2 Parts

As stated before, a system is a collection of objects with a function, and components are non-decomposable objects. However, the part is an important entity that has not been clearly defined and sometimes used as a synonym for component (Engineering, 1998). In practice, when a product is developed, part is an entity that comprises the systems and subsystems. To describe a part this research follows the definition found in the system engineering handbook (INCOSE, 2000): *A Part is a separately cleanly identifiable group of components.* As an example, a car headlamp is an identifiable group of components, which can be separated from the electric system yet still hold as a unit (Hella, 2003). The definitions of part and component are reversed in the handbook for those used here. However, with regard to this research the definitions used are more in line with the ISO information modelling standards.

4.2.1.3 Commercial parts and components

Some parts and components are the final products of other companies like the tyres in a car or the bulbs in a headlamp. Some of these products are designed and manufactured within the facilities of the company. This creates a new division: Bought-out part or component and designed parts or components. Parts either bought-out or designed are comprised of components, either bought-out (e.g. bolts) or designed.

4.2.2 Structure for Configuration of the Product

The previous discussion on the configuration of a product from the design point of view sets the basis for the definition of a UML structure of the configuration of a product as shown in Figure 4.4. This research builds on the work done by (Ullman, 1997) around the design of a product, by defining a formal PMS, which can capture the configuration of a product. This PMS enables the capture of the function of the product, through the Function class, and the relationship with its elements.

Figure 4.4 shows a UML class diagram representing the configuration of a product as defined in this research. This representation highlights the internal relationships between the entities of a product themselves and their attributes. This Product class describes the configuration of the product. The internal relationships in this structure are largely hierarchical, i.e. System, subsystem, part and component are child classes of product, i.e. system ***is_a*** product, subsystem ***is_a*** product, part ***is_a*** product and component ***is_a*** product. However, a second relationship is necessary to be have a full description of the configuration of a product, i.e. a product ***has*** products. This structure provides an important means of capturing the product information related to what comprises a product and how it is comprised.

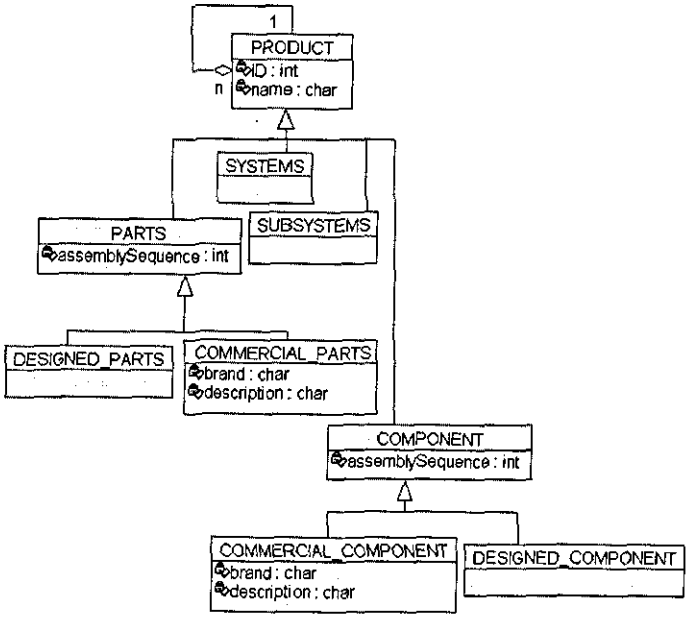

Figure 4.4 UML Product Model Structure showing the Configuration of a Product and the Function Class

Each of these subclasses has been defined with the attributes *name* and an *Id* for identification purposes. Systems and subsystems Subclasses are comprised by Commercial_parts and Commercial_components subclasses, each of these have two attributes: *commercialBrand* and a *description*. This attribute *description* is necessary since the commercial entities will not be redesigned, if

necessary, they will be replaced by a different commercial entity. Components and parts have an assemblySequence attribute to provide a sequence when assembled either to a part, in the case of the component or to the system, in the case of the part.

This structure provides an important means of capturing product information related to what comprises a product and how this is comprised, in line with (Hubka, 1980, Ullman, 1997).

4.3 DESIGN CHARACTERISTICS OF THE PRODUCT

The redesign of a product typically results in changes to components in terms of the geometry, the materials and the tolerances. This section has investigated the way in which information relates to the changes, which have been made to products, and defines a class structure to capture these changes.

It is important to understand the description of the product in engineering terms as this forms the basis for the transfer of information between engineering designers. The characteristics accomplish this requirement by being measurable traits that describe the product as a whole and in each of its systems, subsystems, components and parts. The classification of these characteristics provides the means not only to describe a product but also to identify the changes made when a product is redesigned.

4.3.1 Characteristics in the Redesign of a Product

This research takes as starting point an existing PMS, which classifies characteristics in terms of tolerances, geometry, function, dimensions, material and specifications (Dorador-Gonzalez, 2001). Although this structure was defined from an assembly point of view, these characteristics seem applicable from the design point of view.

4.3.1.1 Geometry of the product

The geometry describes the shape of the product. This characteristic is important not only from the appearance but also from the functionality point of

view (Henderson, 1993). Therefore, changes to geometry have important consequences beyond appearance related to functionality such as easing the assembly, accelerating manufacturability or improving performance.

The use of simple geometric shapes, such as dots, lines, arcs, etc, makes the description of any component difficult. Geometrical features are an accepted way to describe the shape of a component through already defined geometrical primitives i.e. fillet, box, chamfer, thread, cone, etc (Ault, 1999, Lenau and Mu, 1993, Nielsen, Dixon and Zinsmeister, 1991). Therefore, this research uses geometrical features to describe the shape of a component.

4.3.1.2 Dimensions of the product

Dimensions are associated geometrical features and their position (Ullman, 1997). Changes to dimensions have a great impact in aspects such as performance. Variation in dimensions could result in consequences such as lighter products, components that are more resistant and reduction of concentration of stress by increasing the radius of a fillet, etc.

4.3.1.3 Tolerance of the product

Tolerances are associated with dimensions and geometry. Because of the impossibility of manufacturing components to an exact dimension or geometry, tolerances should be provided. This makes tolerances an important characteristic that describes a component. Tighter tolerances imply the use of machinery or tools that are more precise and expensive. Therefore, changes to tolerances have a high impact on the cost of a product.

4.3.1.4 Material of the product

The Material is another characteristic that describes a component. It is important because of their different mechanical and chemical properties. The consequences of changing the material go from small costs to high costs, which imply a change to the whole manufacturing process. Therefore, material is considered as an important characteristic.

4.3.1.5 Function of the product

Subsection 4.2.1 described the configuration of a product in terms of its functionality. This subsection deals with the function as a characteristic that not only describes the task of a product, but also can change in a redesign process.

In line with the description given in subsection 4.2.1, from the characteristic point of view, function is the measurable characteristic that describes the desired task of the different entities that configure a product.

This research considers two different aspects that define the function: how it is achieved, and the nature of the output. The first one is the working principle; the latter is the function output. The changes to any of these aspects can produce a new product. E.g. Dyson vacuum cleaners, based on the cyclone principle, follow a different working principle to that of the traditional vacuum cleaner based on a bag to contain the dust. This change in the working principle whilst keeping the same function, leads to the development of a new product (Dyson, 2003).

As far as the fundamental essence of the function of the product remains unchanged from version to version, the product is the same. To ease a better understanding, the term function is associated with the whole product while subfunction is associated to the function of the entities of the product such as system, subsystem, part or component. Some additional subfunctions can be added keeping the function untouched. However, evolution of a product may reach a point in time when the function will not be the same as the original; in this case a new product is born, see Figure 4.5.

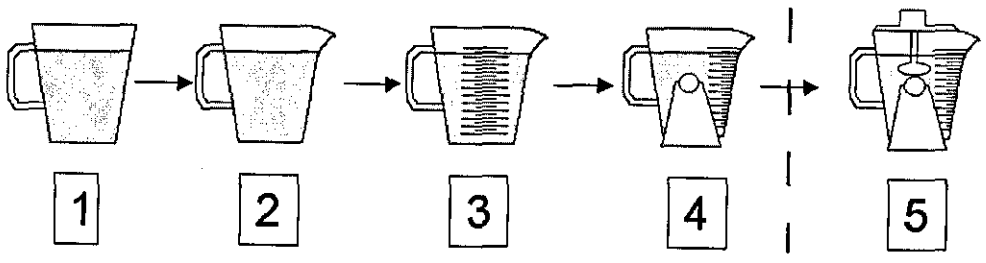

Figure 4.5 Evolution of a Jug until a New Product is Created.

The inclusion of new technologies, materials or ideas from different products helps to create better versions always pursuing the aim of having cheaper, better quality, faster produced products. Figure 4.5 illustrates with a simple example how a new product emerges from a previous one. Subfunctions are added from jug 1 to 4, i.e. a guide to pour the liquid, a measuring scale and two pins to aid the user to pour the liquid into another container. Although new subfunctions are added in developing the jug, the main function is not changed. Once a mixing device is added to the fifth jug the main function has not been changed. Therefore, the main function changes from "to contain liquid" to "to mix substances" and a new product starts. The design history of the new jug will include the versions of the previous jug since this is the result of the evolution. Hence, when a new product is created, it usually does not start from scratch, but in its design history, plus ideas, which may be taken from other products (Dixon and Colton, 2000).

Therefore, the PMS needs to have the capacity to capture the history of a product including the previous products and its version that evolved into the last product.

4.3.1.6 Specifications of the product

The specifications of the products have been well researched (Hubka, 1980, Pugh, 1991, Schachinger and Johannesson, 2000, Ullman, 1997). The customer specifications transform into target specifications as a formalisation to make them more easily understood by the engineer. The final specifications help to measure the function of a product. The success of the evolution of the product can be verified through the product design specifications (Schachinger and Johannesson, 2000). This makes target requirements and final specifications important in order to find out the true advantages of a new version.

4.3.2 The Structure for the Characteristics of the Product

The discussion in the previous subsection provides the basis for the definition of a UML structure of the characteristics that describe the product and identify the changes made to a product. Six characteristics classes were identified: geometry, tolerance, dimensions, material, function and requirements. These classes have an 'is_a' relationship with the Characteristics class. As discussed in previous subsections, two subclasses of characteristics have a particularly important influence in this research: Function and Specifications. Therefore, discussion in this section focuses on these characteristics and their relationships.

4.3.2.1 Function Subclass

The Function subclass describes the configuration of a product from a design point of view.

This Function subclass has two main attributes considered important from the view of this research, i.e. *functionOutput* and *working_principle.* The changes to the function or the working principle can lead to development of new products, rather than new versions. Therefore, it is important to have identified these two attributes.

The requirements help to assess the function; this relationship is highlighted in Figure 4.6. One function is related to one or more requirements. Therefore, the Function Subclass and the Requirements Subclass need to be related, i.e. one Function *has* one or more Requirements.

4.3.2.2 Requirements subclass

The requirements subclass considers requirements from the customer point of view and the technical point of view, i.e. *Customer_Requirements* and *Specifications* of the product. Therefore, a relationship *'is_a'* is established between these classes and the Requirements class, as shown in Figure 4.6.

4.3.2.2.1 Specifications subclass

The Specifications subclass contains technical information that describes the design criteria of the product. This is done from two different periods, before and after the design process, i.e. Target_Requirements and Final_Technical_Specifications. Therefore, an inheritance relationship with the Specifications class is established as shown in Figure 4.6.

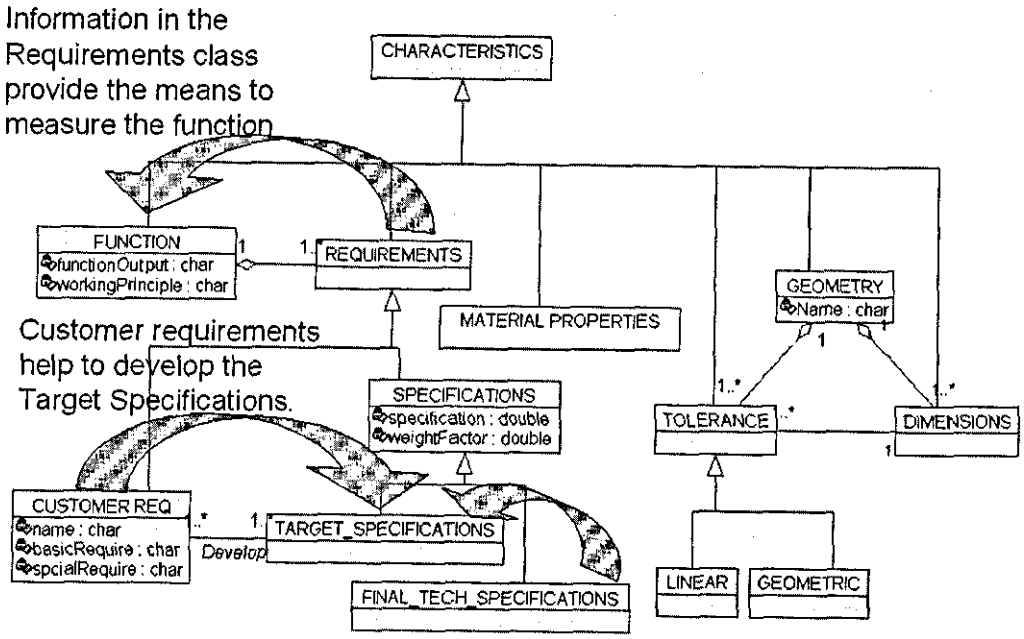

Figure 4.6 UML Product Model Structure showing the Characteristics Class

To provide the basis to assess the product, the Specifications class needs a description of the specification and the importance of that specification, i.e. attributes *specification* and the corresponding *weightFactor*.

4.3.2.2.2 Customer Requirements subclass

The Customer_requirements represent the wishes a customer has for a product. Target_requirements are the engineering interpretation of the customer_requirements. One customer requirement can lead to one or several

target requirements, in the same way as several customer requirements can be related to one target requirement. Therefore, one or more Customer_requirements *associates* with one or more Target_requirements.

The wishes of the customer as the Customer_requirements subclass are represented as two important attributes, i.e. *basicRequirements* and *specialRequirements*, in line with the PMS defined by (Dorador-Gonzalez, 2001).

4.4 VERSIONS OF THE PRODUCT

Whenever a product is redesigned, the changes done to the product create a new version. Issues related to the changes and the Design Intent underlying these changes are discussed using the case of a wire-straightening machine.

4.4.1 Case Example

In the Manufacturing and Design Centre of the National University of Mexico, a company in the medical area sponsored a project to develop the process to manufacture surgical needles. As part of the process, the wire from which the needles are manufactured needs to be straightened from its coil. A commercial straightening machine to work with 0.7 mm diameter stainless steel wire was sought. However, because of constraints in time and cost from the sponsoring company, a commercial straightening wire machine to straighten 6 mm diameter bars was adapted. The working principle of the straightening system is based on forcing the wire into a rotating system of seven alternated dies that describe a sine wave, see Figure 4.7.

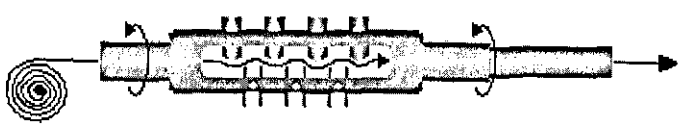

Figure 4.7 System to Straighten Wire

Two versions of the die are shown in Figure 4.8. Die "A" is one single component with a slot, whose purpose is to guide the wire, see encircled area in

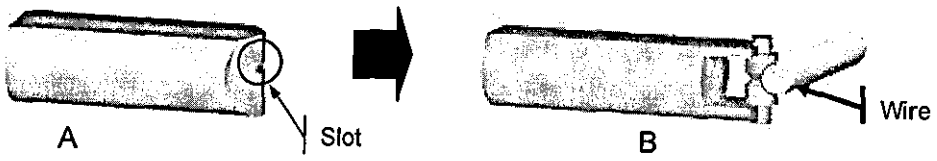

Figure 4.8. Die "A" one component; die "B" with two components.

Figure 4.8. However, spiral marks with a depth around 0.1 mm, are scraped along the wire because of the friction produced by the direct contact between die and wire. For the 6 mm diameter bar, friction is not a problem since once straightened; this bar is used in the building industry. However, for the suture needles, these marks were unacceptable. The die "B" was developed to overcome the problem of the marks by adding a roller. This roller is held in place in the die by the wire, as seen in Figure 4.8 "B".

The change made to die "A" to overcome the problem of the marks improved some aspects of the machine and prejudiced some others. Mexican Official Standards require that no marks should be present on the needle surface (NOM, 1993). This is shown in *Appendix B*.

By overcoming the problem of the marks on the wire, the legal requirements on the wire have been achieved. This is the primary driver for this change in the product design. In addition, there are both positive and negative consequences, which are listed and explained below.

The positive consequences are

1) Aesthetics: By removing surface marks from the wire the product also looks good as well as meeting the standard requirements.
   Performance: By removing marks from the surface of the wire, the needle performs more effectively.

The negative consequences are:

1) Manufacture: The die is more complex and therefore takes more time and is more expensive to manufacture. Adding the roller implies an increase in the cost and time to produce the components.

2) Safety: The design of the new die has a problem, leaving the roller shown in Figure 4.8, with potential to fly out from the machine.
3) Maintenance: Every time a wire break occurs, the roller needs to be reinserted into the machine.

Each change produces variations in the production time and cost. The sum of this variation in terms of time and money sets the basis to make an initial evaluation of the final production time and cost of the current version designed.

4.4.1.1. Third version of straightening die

A version "C" shown in Figure 4.9 was proposed to improve version "B" in Figure 4.8. For this version, the information produced when version "B" was generated, was provided to the designer. With this information the designer becomes aware of the changes and the Design Intent underlying the changes done from version "A" to "B" and targeted the redesign effort towards improving a different aspect: safety.

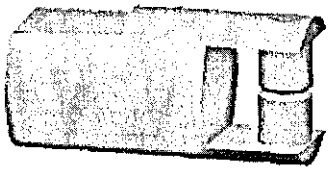

Figure 4.9. Third Version of Die.

Figure 4.9 shows die "C", the third version of the die. This version changes how the roller is held in the die. An axle provides the means to constrain the roller and make a safer machine which is the Design Intent underlying this change. The positive consequence is a low-maintenance machine, therefore the benefit reflects on the aspect of maintenance. Despite this advantage, two negative consequences are produced: the assembly and manufacturing become more complex and expensive.

This example raised three issues related to a version.

- Why a change is done?

- How the change is done?
- What consequences of the change are produced?

These issues are discussed below.

4.4.2 Design Intent Underlying a Change in a Version

The example described used three versions of the die. However, it would be possible in actual designing, to go from version "A" of the design to version "C" of design without actually manufacturing version "B" by anticipating problems in different aspects, i.e. manufacturing, assembly, performance and safety. Foreseeing future problems of the initial change aids the resolution of potential secondary problems. In the above example this would have led to the production of a different version of the die with one Design Intent Driver underlying the changes, i.e. accomplishment of Mexican Standards on surgical needles; and two secondary Design Intents, i.e. increase of safety in the use of the machine and reduced maintenance to the machine. Therefore Design Intent Driver is defined as the main reason that promotes change to the characteristics of the entities of the product. And the secondary Design Intents are defined as those less significant reasons related to the entity changed that reinforces the justification of the change.

The process of identifying the Design Intent Driver requires experience and expertise from the designer to jump several steps forward in developing the next version by anticipating problems in different areas leading to there being many reasons related to the changes.

This identification and capture of the Design Intent Driver leads the designer to understand the thinking process followed by the previous designer, in the same way as he or she will transfer this information to the next designer. By ensuring this transfer of information, designers will be capable of making better and faster decisions.

The above discussion highlighted the importance of identifying the Design Intent Driver, which underlies a change in a new version of a product. The next subsection discusses the changes. This structure provides the basis to produce a PMS.

4.4.3 Changes in a Version

The changes to a product produce new versions, and this is part of the design history of the product. To understand this information it is vital to provide the designer with useful information of how the product has evolved from its creation.

Strong competition, problems with the product, new materials, improved manufacturing technologies, changes in the customer requirements, among other reasons, make it necessary to produce improved versions of a product. Therefore, a variety of different changes can be made to a product at different elements in its configuration e.g. systems, subsystems, parts and components. The aim of these changes is to obtain higher quality products produced with lower production costs and in a shorter time.

Differentiation of the status of the change provides the ability to understand how the product has evolved. Changes may modify, add or eliminate elements of the product configuration. By identifying the status of the elements of the configuration of the product, the designer is able to track how the product has evolved.

It is unlikely to have changes without having consequences. Most of the time a change produces consequences, both positive and negative. This relationship needs to be considered in the PMS and is discussed in next section.

4.4.4 Consequences of Change

Changes in a version are aimed at improving the product. However, each change may bring both positive consequences and negative consequences.

Identification of these consequences, when the changes are made, requires experience and knowledge from the designer.

Identification of the consequences is important because negative consequences represent potential future aims to improve the product, while the positive consequences represent improvements, which are potentially desirable to keep. Some of these consequences can be evaluated in terms of production time and cost.

A positive consequence implies, in most cases, a reduction of the production time or the production cost, while a negative consequence implies the opposite. An evaluation of the consequences in terms of the production time and the production cost provides the means to make a further global evaluation once the design of the version has been accomplished.

In the example used in 4.4.1, adding the roller to die "A" produced one negative consequence, i.e. the manufacture process became more complex. Complexity in the manufacturing aspect would increase the time and cost to produce the product. However, the positive consequence was the improvement in the performance of the machine since a better product was achieved. Despite the negative consequences in the manufacturing process, a better quality product was achieved.

By capturing the Design Intent Driver underlying the changes, the consequences of the changes and the status of the change, a complete range of information is captured. Therefore, these issues can be answered:

- Why is a change done?
- How is the change done?
- What consequences of the change are produced?

4.4.5 Structure for the versions of a product

The above discussion sets the basis for the definition of the UML structure of the versions of a product. This subsection describes the classes that comprise

this structure and their internal relationships. Four classes comprise this structure: version, changes, consequences and intent. Figure 4.10 illustrates this structure.

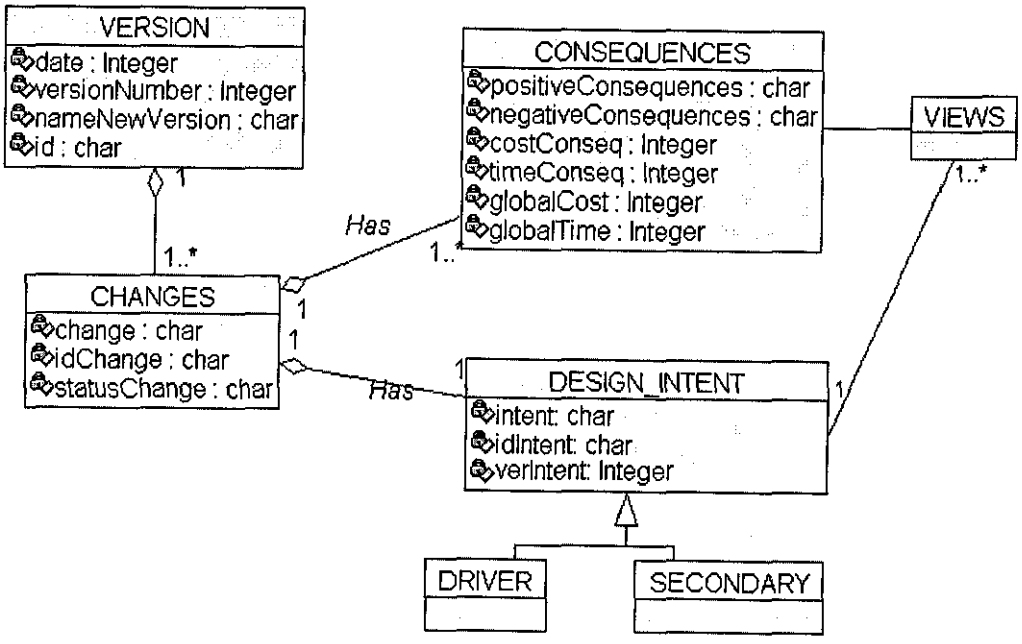

Figure 4.10 UML Product Model Structure showing the Version Class

4.4.5.1 Version Class

A version needs to be positioned in time and sequential order; therefore, the attributes *date* and *versionNumber* are included in the Version class. The *nameNewVersion* and *id* attributes are necessary for identification purposes from other versions of the product.

4.4.5.2 Changes Class

The Changes class captures the information of what and how changes are done in a redesign. This is captured through the attributes *change* to capture the description of the change itself and *statusChange* to capture if the change was an addition, elimination or modification, as described in subsection 4.4.3. Because many changes can be made to a product, the changes need to be identified, therefore the idChange needs to be included in the Change Class.

4.4.5.3 Design Intent Class

The key issue of this research is the capture of the Design Intent underlying the changes. This information is captured through the attribute *intent* in the Design Intent Class.

As described in section 4.4.2 each change has associated one Design intent Driver and several secondary Design Intents. Following this description, two child classes are defined named Driver class and the Secondary class, these classes inherit the attribute *intent* as well as the attributes defined to associate one change in one version of the product to the Design Intent. These are termed *idIntent* and *verIntent*.

4.4.5.4 Consequences Class

The negative and the positive consequences described in section 4.4.5.4 are captured in the Consequences class. The attributes used to capture the consequences are *negativeConsequences* and *positiveConsequences*. The variations in the production time and cost are captured through the attributes *timeConseq* and *costConseq*. Finally, the attributes *globalCost* and *globalTime* capture the sum of the variations in the production time and cost of the version. This last attribute helps to assess the accomplishment of two of the three criteria to evaluate a new version.

The internal relationships between the classes Consequences and Changes follow the description given in section 4.4.4. A version is featured by changes; therefore, the relationship between Version class and Changes class was defined as: one Version *has* one or many Changes. Each change *has* one Design Intent Driver and secondary Design Intents underlying it. Therefore Driver Class and Secondary Class represent these two types of Design Intents, an *inherit* relationship is established between these classes. As a change produces consequences, Changes Class *has* Consequences Class. Figure 4.10 highlights the relationships described above between Changes Class and Intent Class.

The structure illustrated in Figure 4.10 and its relationship to the rest of the PMS is very important for the ability to reuse design information. The definition of a formal structure, which can enable the capture and reuse of Design Intent, is the main contribution of this research.

4.5 VIEWS OF A PRODUCT

This section is concerned with identifying views to classify the information related to Design Intent, in terms of the reasons behind the changes and the resultant consequences of these changes. Classifying this information is important because it has the potential to provide answers to the questions around the evolution of the product to support the engineering designer (Ullman, 2002).

The example described in section 4.4.1 highlighted the importance of organising the information around the evolution of the product to provide support to the designer of the next version of a product. The next subsection identifies the core-views where Design Intent and the consequences can be classified.

From the example above, the underlying intent was to produce a product that met the Mexican standard requirements. However, the consequence of this was to produce a product, which had a number of positive consequences but also a number of negative consequences. Little work has been done to consider the consequences of the changes (Goodwin and Chung, 1997). The need then is for an information system capable of capturing the relationship between the intent and the consequences of a change.

4.5.1 Product Design Specifications in the Design Intent

The Product Design Specifications (PDS) set the specifications of a product, which can be used to evaluate the resulting product. This PDS is the starting point to classify the Design Intent and the consequences of the changes. The next section discusses how it is used.

Section 4.3.1.6 explored the specifications as a characteristic that describes the product. The PDS explored in this section is the starting point for the definition of the Views of a product.

From the many authors who discuss and explain PDS and their importance for the design process, perhaps the most comprehensive lead is from (Pugh, 1991). He identifies thirty-two elements that help to make a complete definition of the specifications a product should accomplish when developed. These are: Patents, environment, testing, safety, legality, documentation, quantity, product life span, materials, ergonomics, standard specifications, aesthetics, installation, life in service, performance, product cost, timescale, customer, processes, size, shipping, company constraints, disposal, manufacturing facility, politics, market constraint, weight, maintenance, competition, packing, quality reliability and short life storage. These elements cover a wide range of aspects, all of them necessary to define the specifications of a product to design.

In the process of evaluating these thirty-two elements, three categories have been identified: high-level criteria to evaluate products, general generators of design specifications and core-views to classify Design Intent and consequences.

A) High-level criteria to evaluate products

In category one, three elements of the PDS relate to very high-level issues, which are generally applicable to every product. These are: timescale, product cost and quality reliability, which are typically related to high-level criteria to evaluate the design of a product.

B) General generators of design specifications

In category two, some of the thirty-two elements of the PDS are identified with the generation of the design specifications and how these elements relate to other elements of the PDS. For example, the element of the PDS: customer, is actually related to meeting the customer requirements, which then have implications for other issues related to other elements of the PDS such as performance or aesthetics. The element: manufacturing facility, relates to a

general issue for a company, which again has implications for other elements of the PDS such as: processes, which include the manufacturing and assembly processes. Therefore, each of the elements of the PDS is not fully independent.

In a full evaluation of the elements of PDS, standards specification; politics; competition; market constraints; manufacturing facility and company constraints are all elements of Pugh's description, which are general generators of design specifications that describe the product as explored in section 4.3.1.6.

C) Core-views to classify design intent and consequences

For the third category, ten elements of the Pugh's PDS were grouped into the above categories. The remaining elements seem to provide the basis to classify the Design Intent and the consequences of the changes into twelve core-views, as illustrated in Figure 4.11. The provision of high-level categories against which the negativeness or positiveness of the consequences of the changes can be assessed, is the main justification for the useful association of the twenty-two remaining elements.

Figure 4.11 c) shows the twelve core-views identified. These core-views group the remaining elements of the PDS following a process based on concepts that can be associated. For example the core-view Physical, is associated to the elements: quantity, materials, weight and size, all of them physical attributes of a product.

Standards specifications is the only element of the PDS, with a double role, as a general generator of specifications and as a core-view to classify the Design Intent. As a part of the core-view Legal, Standard specifications are guidelines to be followed to produce accepted products by both domestic and international communities. However, their association with legal issues is strong since in many countries they are the only accepted way to carry out some process or develop a product. Patents are legal sources of information as well as the means to protect the intellectual work that represents the design, and therefore related to the core-view Legal. Documentation of the product includes the instructions for the user, the maintainer, and other people around the product,

this includes legal warnings for use and safety therefore it is associated to the core-view Legal.

The element of the PDS "Processes" was divided in two core-views not considered in the original thirty-two elements: manufacturing and assembly. Following the association of concepts, the other associated views are: "Sustainability" associated with product life span, disposal and life in service;

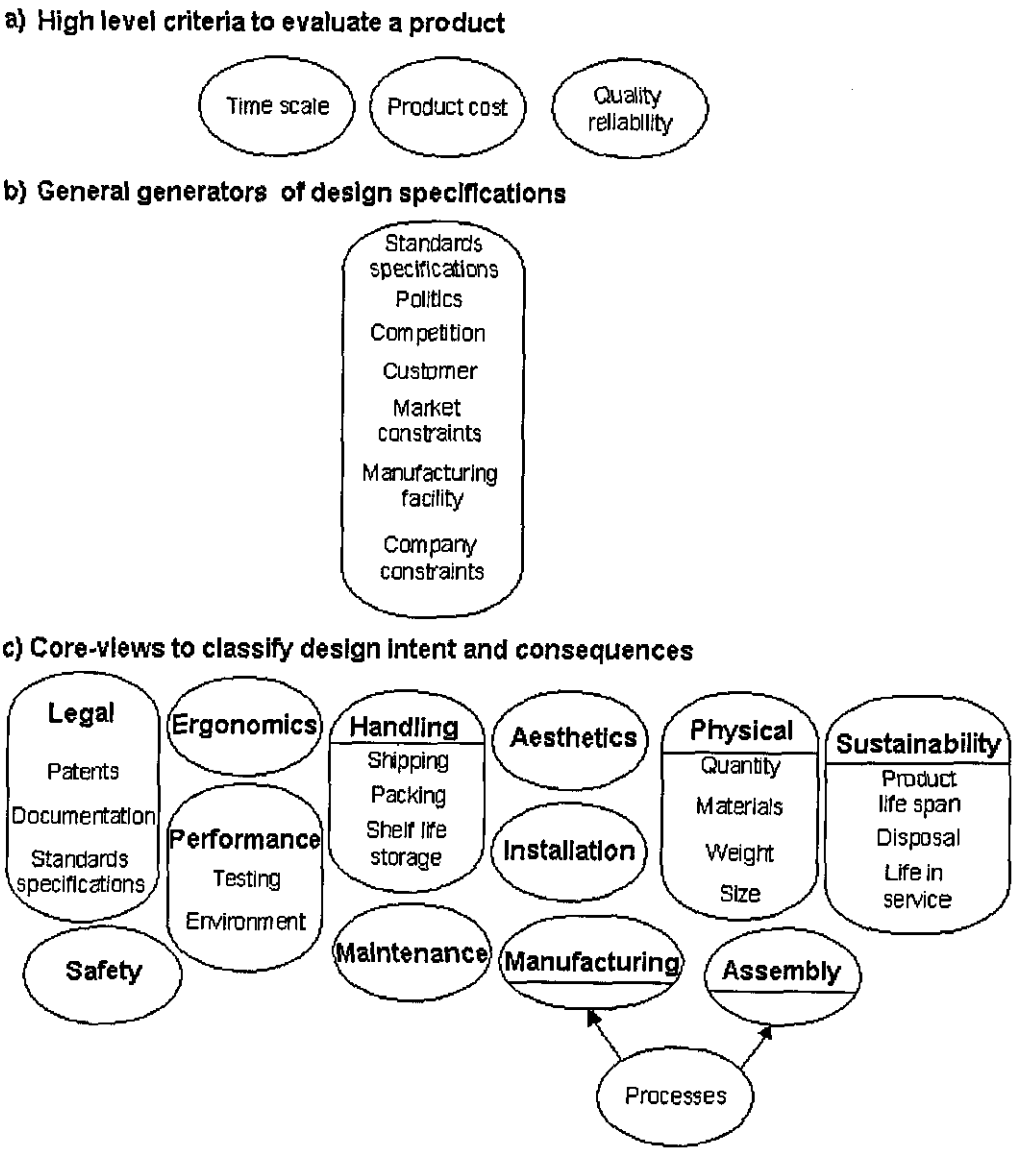

Figure 4.11. Regrouping of Pugh's PDS elements

"Functionality" associated with performance, testing and environment; "Handling" associated with shipping, packing and shelf-life storage.

4.5.2 Structure for the Views of a Product

As described in section 4.5.4 it is important to classify Design Intent and the consequences of the changes in the main core-views identified. Twelve core-views were identified as part of the design views class: aesthetic, physical, assembly, sustainability, installation, safety, handling, maintenance, production, ergonomic and performance.

Because the twelve-core-views have the same classifying nature, an 'is_a' relationship is needed between Design view class and the core-views subclasses. The Views class is shown in Figure 4.12.

As described in section 4.4.2, Design Intent is aimed at improving the product in some aspect, namely core-view. This improvement can be captured as a positive variation of that core-view product. The capture of this variation

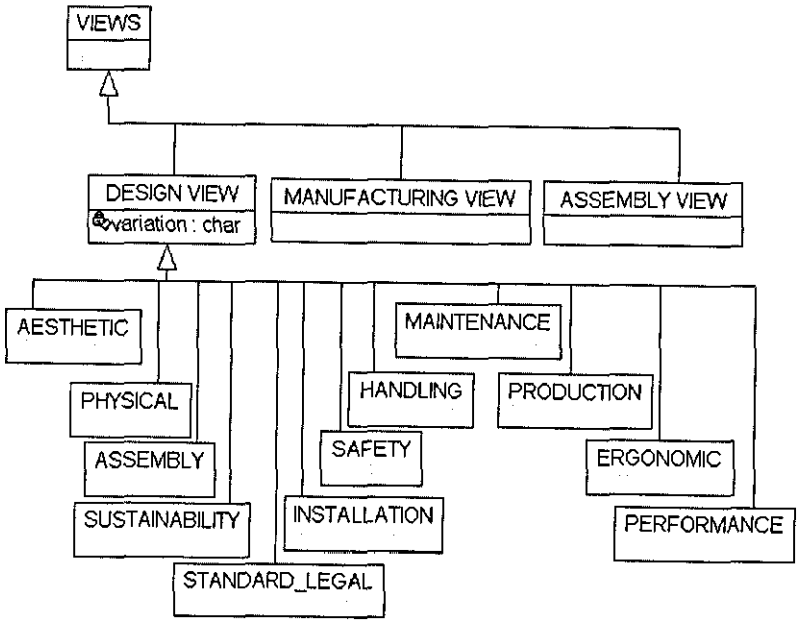

Figure 4.12 UML Product Model Structure showing the Views Class

provides supplementary information to the company for the next version on the degree of improvement achieved with the change.

4.6 RELATIONSHIPS BETWEEN STRUCTURES

This section describes the relationships between the structures defined in previous sections. Subsection 4.6.1 describes the relationship between the Configuration structure and the Characteristics class as the first important relationship that describes the product. The next subsection, 4.6.2 describes the most important set of relationships which help to capture the Design Intent information. The last subsection describes the relationship between the Intent class and the Views class.

4.6.1 Relationship Between Configuration Class and Characteristics Class

A product has a range of characteristics that describe it, this leads to the establishment of a '*has*' relationship between the Configuration Class and the Characteristics Class. This gives the Configuration class access to all the attributes captured by the instances of the Characteristics Class. Thus, all the entities of the Configuration Class have access to the information captured in

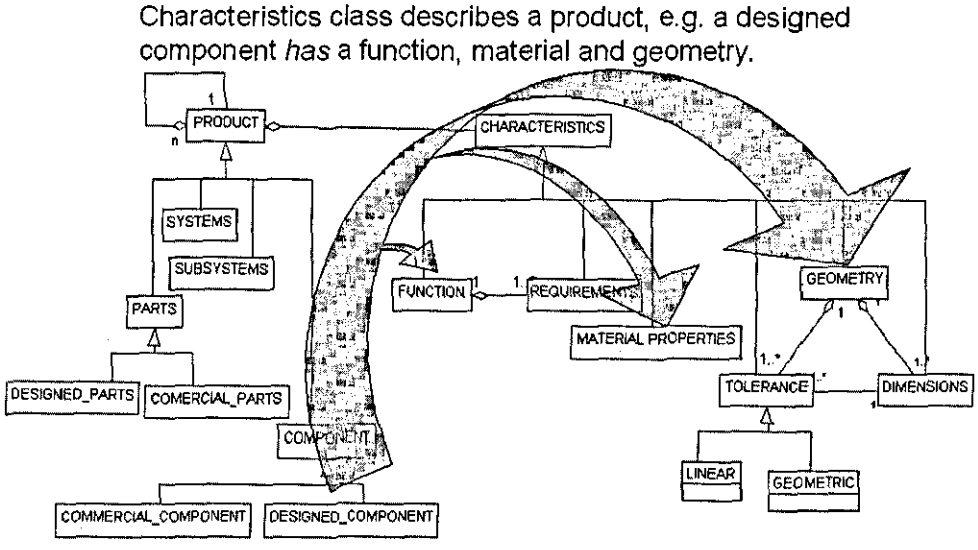

Figure 4.13 Relationship between Product Class and Characteristics Class

any instance of the Characteristic Class. Figure 4.13 highlights this relationship using the Geometry subclass to illustrate how the Characteristics Class has the ability to describe a Designed component subclass.

4.6.2 Relationship Between Version Structure, Configuration Structure and Characteristics Class

The Characteristics class as described in the previous section has the ability to capture the information of the characteristics of a product. However, this research sees a second ability, which is to identify what changes are made to the characteristics of a product when it is redesigned. This section defines this new ability through an associative relationship between Characteristics Class and Version Class.

The Configuration structure links to the Version structure through a *has* relationship following the description made in section 4.4: Product *has* versions. The first two relationships, Product *has* Versions and Versions of Characteristics, provide the way to capture initially, the information of a new version taking the target requirements as a start point, Figure 4.14 highlights this relationship.

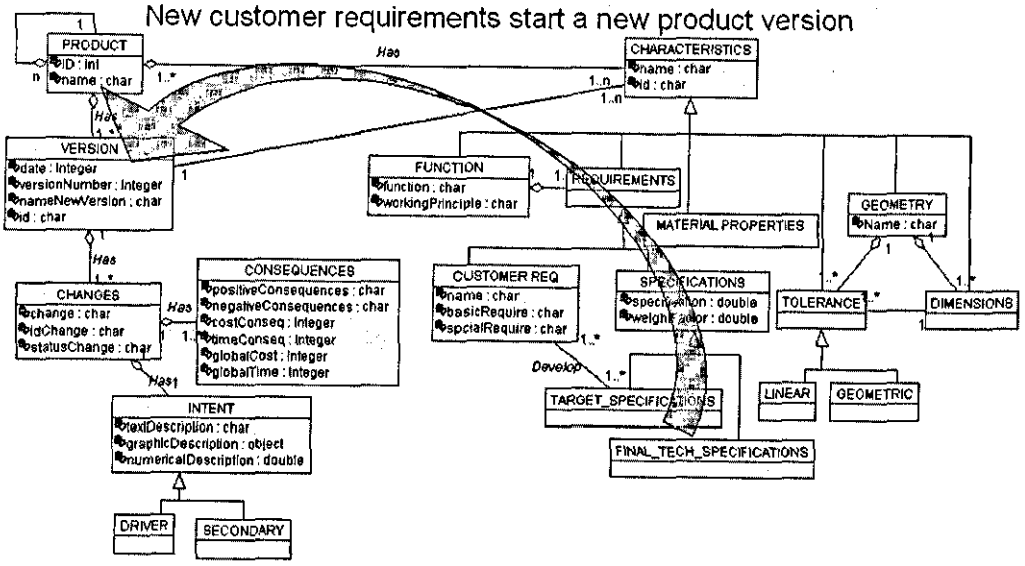

Figure 4.14. Target Specifications Give Way to a New Version

A second use for this relationship is to capture the information of the changes done to the characteristics of a chosen entity of the product. Figure 4.15 highlights this relationship.

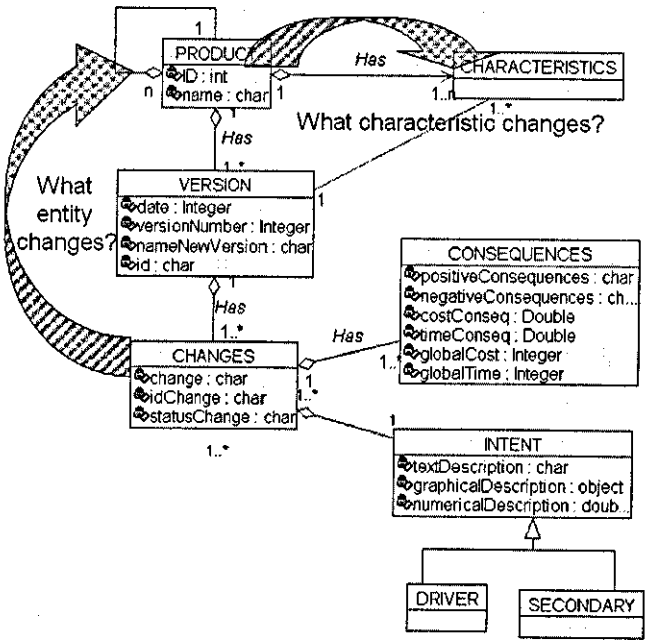

Figure 4.15. Relationship between Configuration Structure, Characteristics Class and Version Structure.

4.6.3 Relationships Between Views Class and Version Structure

Views Class was conceived to *describe* the product from different points of view. To do this the Views Class *uses* the Characteristics Class information, this is shown in Figure 4.16.

The Design View class has been defined to classify the Design Intent and the consequences of the changes. Therefore, the Consequences class and the Intent class are associated to the Views class extends. This association extends the scope of the Views class by providing the means to classify the many consequences resulting from the change with one or many design views and the Design Intent Driver with one design view.

However, this research sees a new benefit for the association between the Views class and the Intent class, by providing the potential capability to capture the manufacture intent and assembly intent, which are the reasons behind the way the product is manufactured or assembled.

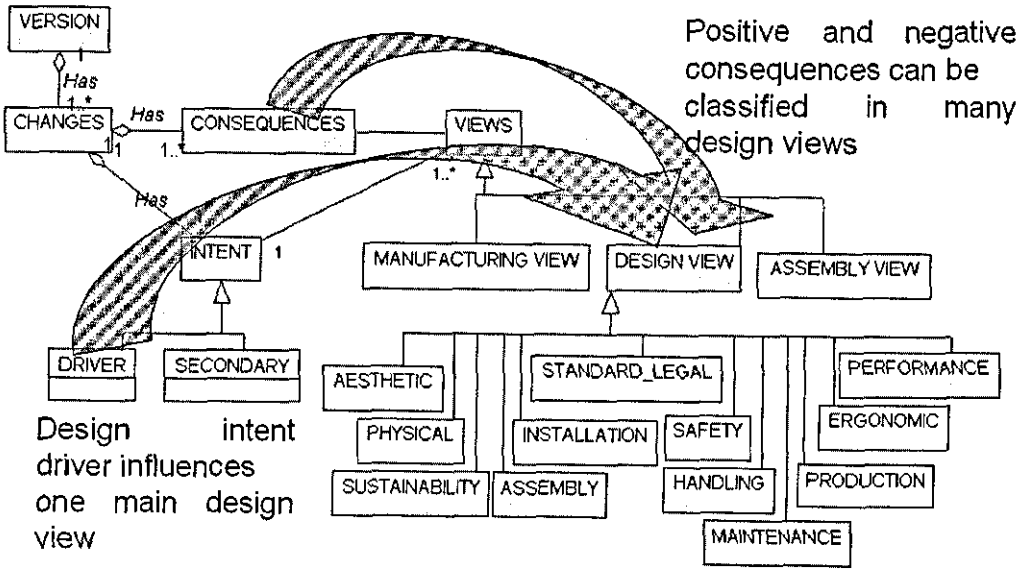

Figure 4.16. Interactions between Views Class and Version Structure

The previous sections defined the structures that comprise the extended UML structure to capture the Design Intent information, shown in Figure 4.17 at the end of this chapter.

4.7 EXTENDED INFORMATION STRUCTURE FOR THE PRODUCT MODEL

The previous sections have discussed the four structures, and their relationships, to comprise the extended PMS.

This PMS contains detailed information about the product, changes and views so it can support software applications that interact following the concurrent engineering philosophy.

This extended PMS provides the capability to capture and store the information related to Design Intent, to support the decision making process in the redesign process.

To validate this PMS, an experimental software environment was developed. The next chapter describes this experimental system and its exploration in relation to a wire-straightening machine.

4.8 SUMMARY

Four key areas and their relationships were explored in this section to define a Product Model Structure related to Design Intent. The area "Configuration" describes a product from the design point of view in terms of entities with function such as systems, subsystems, parts and components. The area "Characteristics" describes the different measurable characteristics that describe a product and represents what changes when a product is redesigned. The area "Version" discusses the types of changes and the importance of identifying the Design Intent Driver behind the changes of a version. Finally, the area "Views" describes the different core-views in which Design Intent and consequences can be classified based on their influence. Understanding the information and its relationship sets the basis to develop the main contribution of this research: a PMS capable of capturing and storing Design Intent.

This chapter presented the resulting structure for the Product Model. This structure was designed to provide information support to the decision-making process in the redesign process.

The configuration of the product was modelled following a design point of view, i.e. considering the function of each of the entities of a product. This allows a better understanding of how a product is comprised and the relationships between the entities at different levels of assembly. A new structure, named Version, was introduced in the Product Model, which captures the changes made to a product, the Design Intent Driver underlying the changes and the consequences of the changes. This structure captures the information about the

changes in new versions providing answers to a number of issues around the changes. These answers go from the origin of the change, to the consequences of the change, going through to how the change is made and the change itself. This provides a better understanding of the design history of a product and enhances the transfer of information between designers.

The Views class identifies which aspect of the product is influenced by the Design Intent Driver as well as identifying those aspects affected by the consequences of the change. Therefore, information about the changes can be more efficiently captured and used. The link between the Design Intent Class and the Views Class provides the possibility to consider manufacture and assembly intent, interpreted as the reasons underlying how a product is manufactured and assembled.

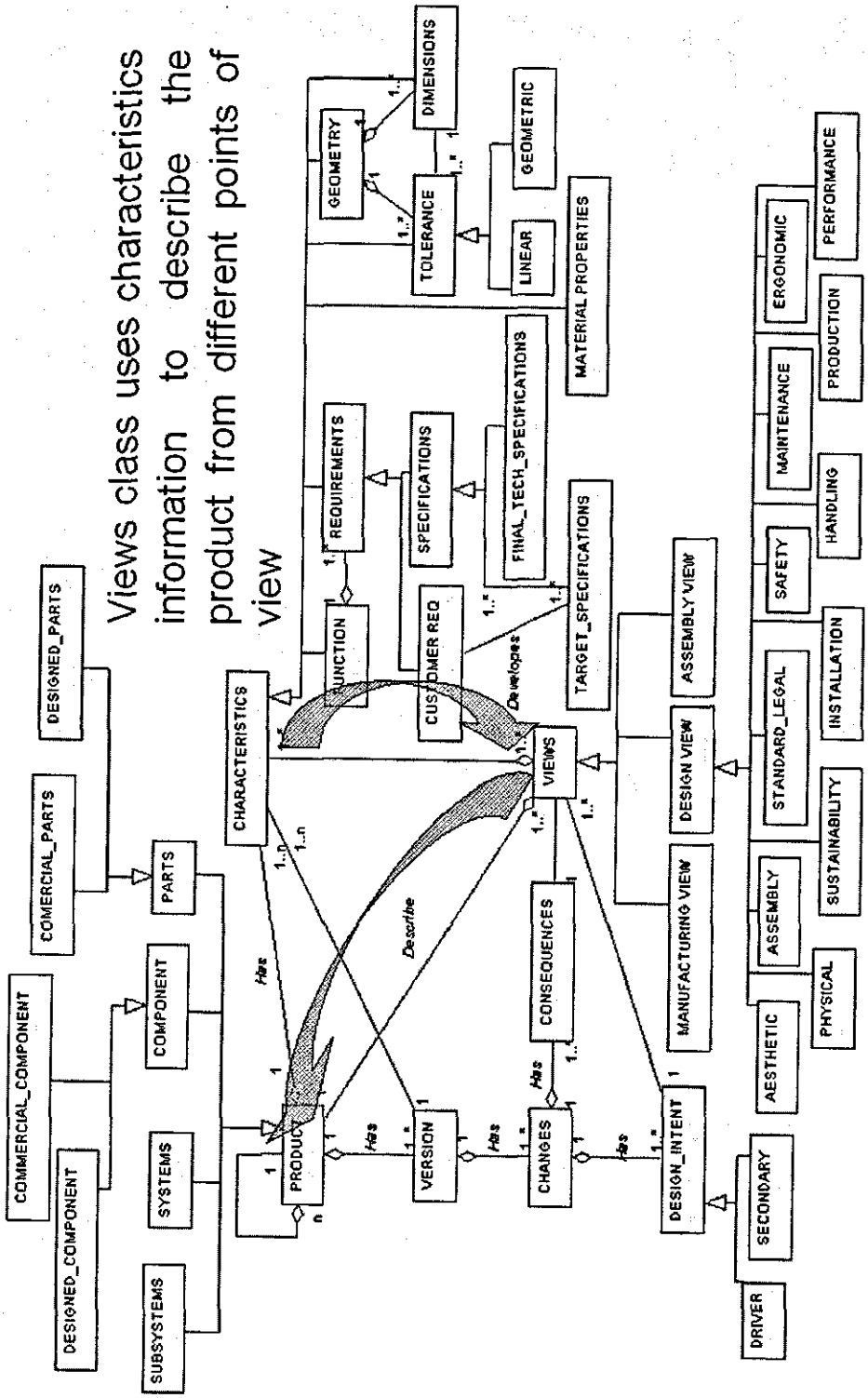

Figure 4.17. Extended General Product Model Structure

5 DEVELOPMENT OF THE EXPERIMENTAL SYSTEM

5.1 INTRODUCTION

This chapter describes the experiments that show how the information structures defined for the PMS can capture, and use the information related to Design Intent to support the redesign activity.

Section 5.2 outlines the objectives and scope of the experimental software. Section 5.3 describes the implementation of the experiment. Section 5.4 describes in detail how the experimental system captures and uses the Design Intent information, the main contribution of this research.

5.2 OVERVIEW OF THE DESIGN OF THE DESIGN INTENT SYSTEM

5.2.1 The experimental system environment

The development of the Design Intent experimental software included the use of four elements: the ObjectStore Designer, the Component Wizard, the Visual C++ 6.0 programming language and the ObjectStore Inspector.

The ObjectStore designer was used to build the object oriented database structure. This object-oriented modeller allows the modelling of complex problems through persistent objects using classes, attributes, methods and relationships.

The component wizard creates the code from the database structure to implement the application in Visual C++ 6.0 and develop the experimental software.

Visual C++ 6.0 was used to design the visual interfaces to create the database, capture, visualize, make queries and edit the information stored in the database.

Another way to visualize the information is through the ObjectStore Inspector. This tool allows the visualization of the structures, the querying and editing of the information and the verification of the representation of the relationships between the persistent objects.

5.2.2 A Use Case Model of the Design Intent System

To create the implementation it is important to identify the functionalities of the system. A UML use case model is a tool that helps the identification of the primary elements and processes, and the functionality of the system. It describes what the system will do at a high-level, based on *actors* who interact with the system (Booch, Rumbaugh and Jacobson, 1999).

An oval is used to depict the use cases while a verb is used to describe the functionality. The two relationships used in modelling the diagram are: <<includes>> and <<extends>>. The first one represents the use of the functionality of another use case; the latter represents the extension of the functionality through optional functionalities.

The use case diagram helps the understanding of the interaction between the system and the actors. This description can be done by such phrases as "the actor does" and "the system does". Therefore the use case diagram models the dialog between the users and the system. For example, in this diagram the actor checks the design history and the system finds the changes; the consequences of the changes; the Design Intent underlying the changes; the status of the change and the entities of the product related to the entity the designer is checking.

The designer communicates with four more use cases of first level, these are: Make changes; Update design history, Evaluate new version and Capture target specifications. These use cases further use or extend second level use cases to provide the required functions of the system, as illustrated in Figure 5.1.

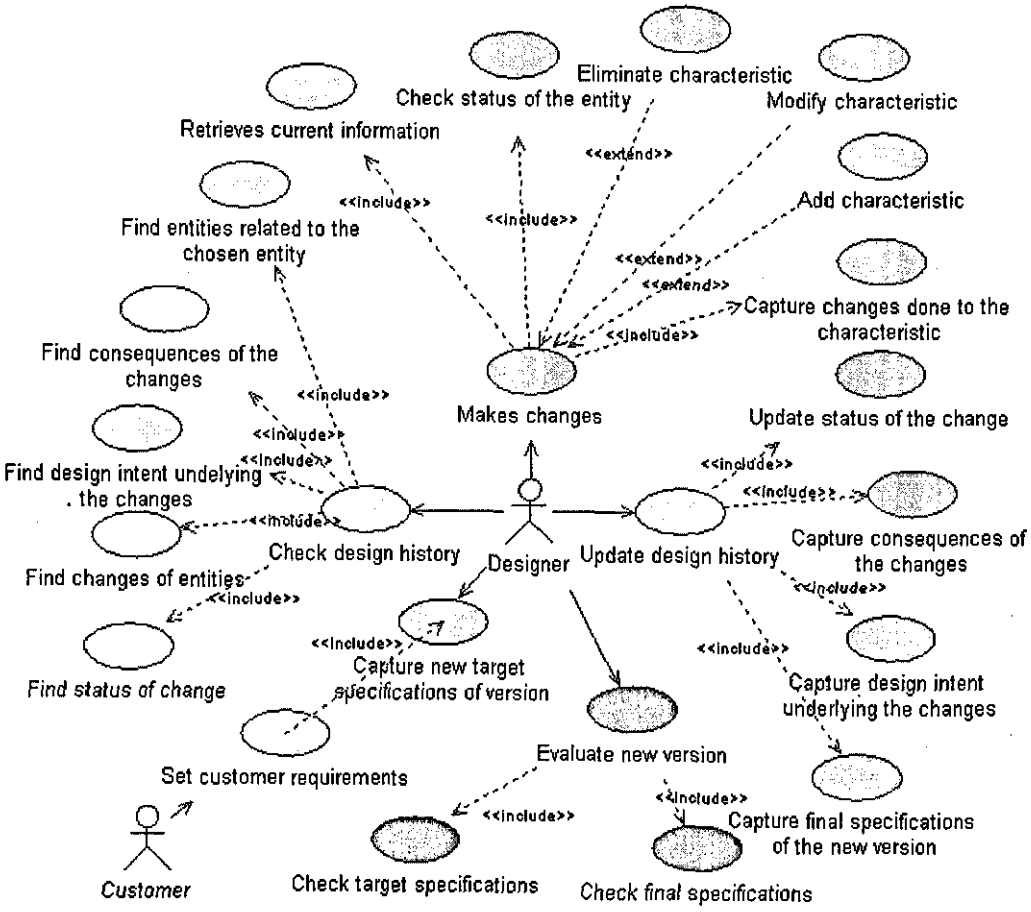

Figure 5.1 Use case model of Design Intent.

The capture of the initial information about a product, is not included in the use case model because this was done from the perspective of the redesign of a product. Therefore the information of the configuration and the characteristics of the product are assumed to be already stored in the database; hence a redesign can be done. The images of the visual interfaces and the structures applied that capture the initial information of the product and its characteristics are illustrated in Appendix C.

5.3 IMPLEMENTATION OF THE EXPERIMENTAL SYSTEM

This section discusses the implementation of the experimental system that tests the capability of the information structures of the product, version, characteristics and views. The information used to populate the experimental

system was taken from a sponsored project from the Manufacturing and Design Centre of the National Autonomous University of Mexico.

5.3.1 Implementation of the Product Model Structure

The PMS defined in chapter 4 was redesigned using ObjectStore Designer to generate C++ programming code. This redesigned PMS is illustrated in Figure 5.2.

This redesigned PMS keeps the main four structures as defined in chapter 5. However this structure has four main differences from the PMS defined in chapter 5, and illustrated in Figure 4.17. These differences are.

1) This research focused on the identification and use of a Design Intent Driver; therefore secondary Design Intent is not included in the implementation.

2) The implementation captures the consequences of a change, through attributes in the Views class. Although this structure captured the information adequately, it is not the best representation and as a result, the Consequence Class in chapter 4 was defined. Although the implementation would need some adjustments to follow the amended PMS, the representation succeeded in the capture, storage and reuse of information of the consequences of the change.

3) Three subclasses were not considered in the implementation as part of the Views class, i.e. Handling, installation and security subclasses. However the other nine subclasses of the views class are considered to be enough to test the PMS.

4) To simplify the software code, the dimensions of the geometric features were considered as attributes of the geometry class in the implementation structure. However a Dimension class was associated to the Geometry class and to the Tolerances class in the PMS defined in the previous chapter.

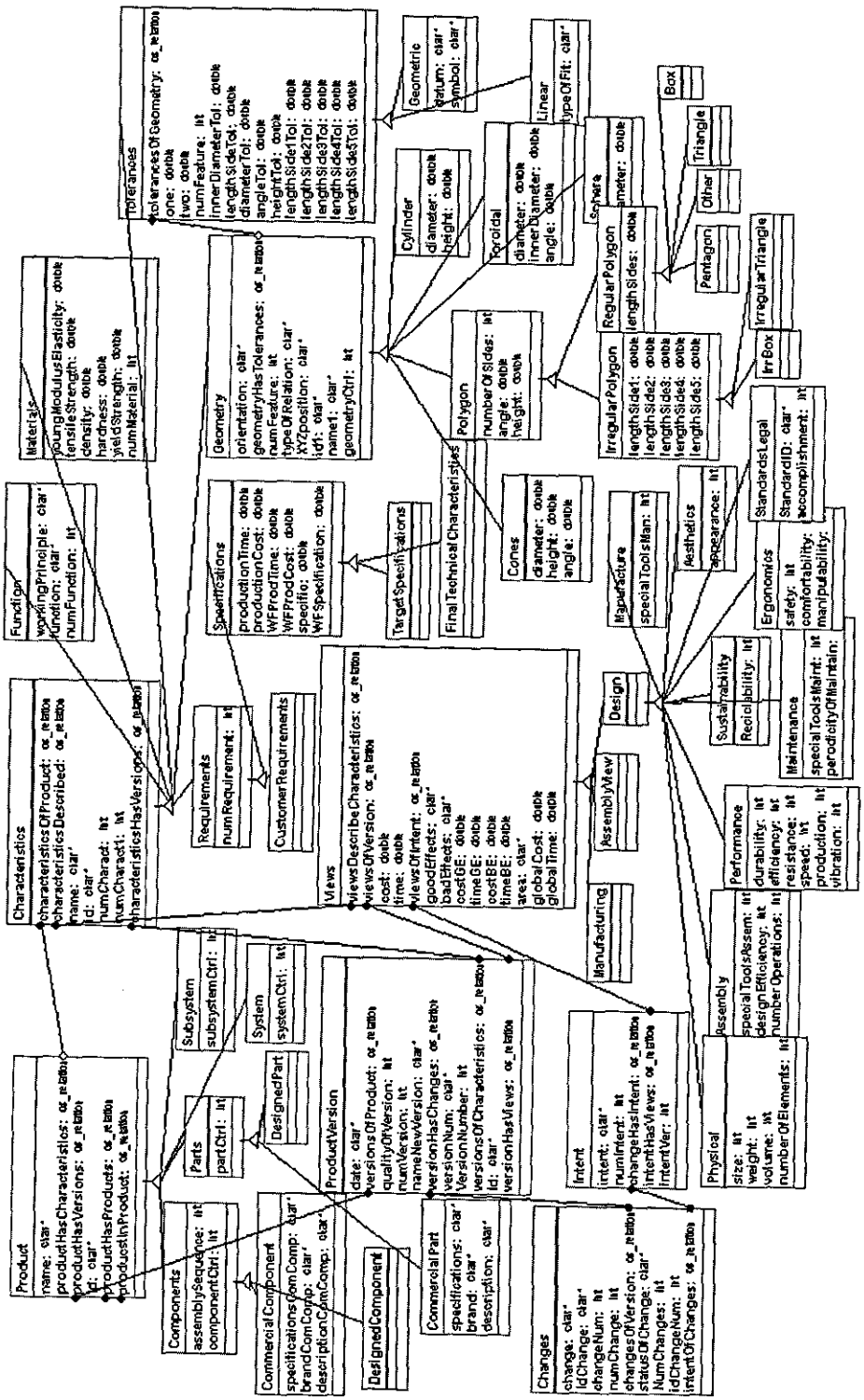

Figure 5.2. PMS implemented in the experimental system

5.3.2 Capture New Target Specifications.

The customer requirements are one of the factors that boost the redesign of a product. These requirements are expressed in terms of wishes for a product and need to be translated into a technical language: Target Specifications. One generally accepted technique that helps to translate the customer requirements into target specifications is the House of Quality (Dieter, 2000, Pugh, 1991, Ullman, 1997). The Visual C++ implementation of the experimental system does not include the translation of the customer requirements into target specifications. Therefore the target specifications need to be obtained separately.

Some target specifications represent an improvement for the product if the value is higher while some other target specifications represent an improvement if the value is lower, e.g. higher efficiency is usually sought in a redesign while weight is usually sought to be lower. In evaluating versions this requires knowledge of the improvement direction. An alternative approach, used in this experimental implementation, is to evaluate the specification elements in terms of the convenience of the solution (Dieter, 2000). This evaluation scheme has values going from to 0 to 10 according to the convenience of the solution that assigns them to the target specifications. The evaluation scheme values are illustrated in Table 5.1.

Point scale	Description
0	Useless solution
1	Very inadequate solution
2	Weak solution
3	Poor solution
4	Tolerable solution
5	Satisfactory solution
6	Good solution with a few disadvantages
7	Good solution
8	Very good solution
9	Excellent solution
10	Ideal solution

Table 5.1 Evaluation Scheme Values for the Target Specifications

The decision matrix method has been a generally accepted method to evaluate different options of solution (Dieter, 2000). This research applies this method as an initial way of evaluating the quality of the versions of the product. The decision matrix method uses weight factors since not all the specifications are equally important. The weight factors were calculated following a normalisation process, which compares the relative importance between the specifications. Table 5.2 shows the values used to compare pairs of relative importance and get the weight factors.

Intensity of importance	Definition
1	Equal importance
3	Moderate importance
5	Strong importance
7	Very strong or demonstrated importance
9	Extreme importance
2,4,6,8	These rating are used to compromise between the above values

Table 5.2 Scale for pair-wise comparison, adapted from (Dieter, 2000)

An example of this process is illustrated in the Table 5.3. Using the values shown in Table 5.2 the importance of the specification labelled as Straightening Quality is compared to all the specifications, e.g. the value 2 means that straightening quality is slightly more important than the surface quality. The same process carried out for the specifications: Production of wires and Length precision. It is clear that comparing straightening quality with itself must be 1. The bottom row shows the sum of the relative importance values in each column. For the specification labelled 'surface quality' the sum is 3.64. The partial weight factor, between the straightening quality and surface quality is 0.55. The average of all the partial weight factors results in the total weight factor of 0.39 for the straightening quality. (Dieter, 2000).

	Specifications	Surface quality	Straightening quality	Production of wires	Length precision	Rating (Average)
1	Surface quality	1.00	0.50	7.00	2.00	
		0.27	0.21	0.47	0.38	**0.33**
2	Straightening quality	2.00	1.00	3.00	2.00	
		0.55	0.43	0.20	0.38	**0.39**
3	Production of wires	0.14	0.33	1.00	0.25	
		0.04	0.14	0.07	0.05	**0.07**
4	Length precision	0.50	0.50	4.00	1.00	
		0.14	0.21	0.27	0.19	**0.20**
	Total	3.64	2.33	15.00	5.25	

Table 5.3 Weight Factors for Version 1

For the capturing of the information described above, three types of attributes were used in the Specification class: *EditOne* attribute of character type to capture the description of the specification; *EspecifOne* attribute of integer type to capture the value for the evaluation scheme; and *WFEspecifOne* attribute of double type to capture the weight factor. Capturing this information sets the start of the redesign of the product. Because of the huge number of attributes, this class was not included in the extended PMS in Figure 5.2. However, in Figure 5.3 the Specifications class is illustrated with the total of attributes to capture nine *descriptions of specifications*, their *Evaluation Scheme Number* and the corresponding *weight factor.*

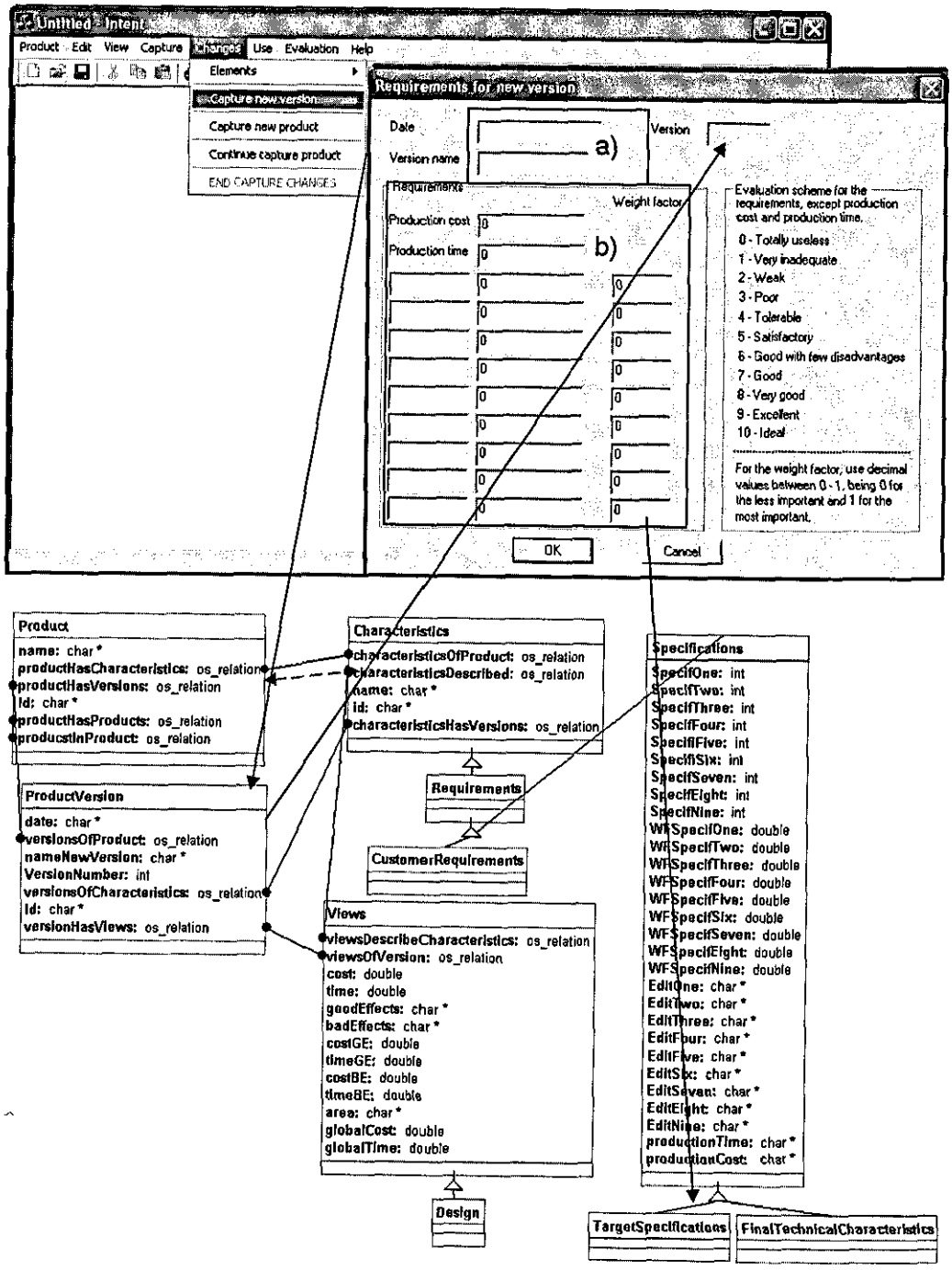

Figure 5.3 Capture of Specifications of New Version.

In box a) of Figure 5.3 production cost and production time are captured in the targetSpecifications class in the experimental system through attributes *productionTime* and *productionCost.* These are two of the three criteria used to evaluate the new version.

The experimental system assigns a new value for the VersionNumber every time a new version is started. This is shown in most visual interfaces to point out to the user the version on which he or she is working.

To identify the version and link the new version to a specific time, the date and version name are captured in the box a) illustrated in Figure 5.3. This information is stored in the ProductVersion class through the attributes *date* and *nameNewVersion*, both of type character. Once this information is captured, changes to the product can be made.

5.3.3 Changes in the redesign process

The implementation of the "Changes" in a redesign can be done through three types of changes: Addition of new entities, modification of current entities or elimination of existing ones. These changes can be influential at each level in the product configuration, i.e. systems, subsystems, parts or components

Figure 5.4 shows a sample of the Visual C++ interface to change systems. It shows which classes of the PMS store the names of the systems, and the classes that store the information about the version number, function, working principle and the status of the system selected. To do this, two relationships of the PMS are used, *productHasCharacteristics* and *productHasVersions.*

The author of this research is in the belief that the PMS defined has the elements to capture the design history of a product including the evolution of previous products underlying a final product, through the Version structure. However, this implementation was constrained to capture the versions of a product without reaching the stage of developing a new product by evolving the overall function of the product. Therefore two buttons, 'Add' and 'Delete', are used to make the changes in the systems and subsystems. In Figure 5.4 the button 'Delete' changes the status to 'Deleted', however no system is actually deleted from the database, the entities that comprise the system change their status to 'Deleted'. The system automatically prevents any change being made to any entity with this status. Because no information is actually deleted, the

design history of the product is stored. The button 'Add' opens the visual interface to capture a new system. This process is the same for the other entities.

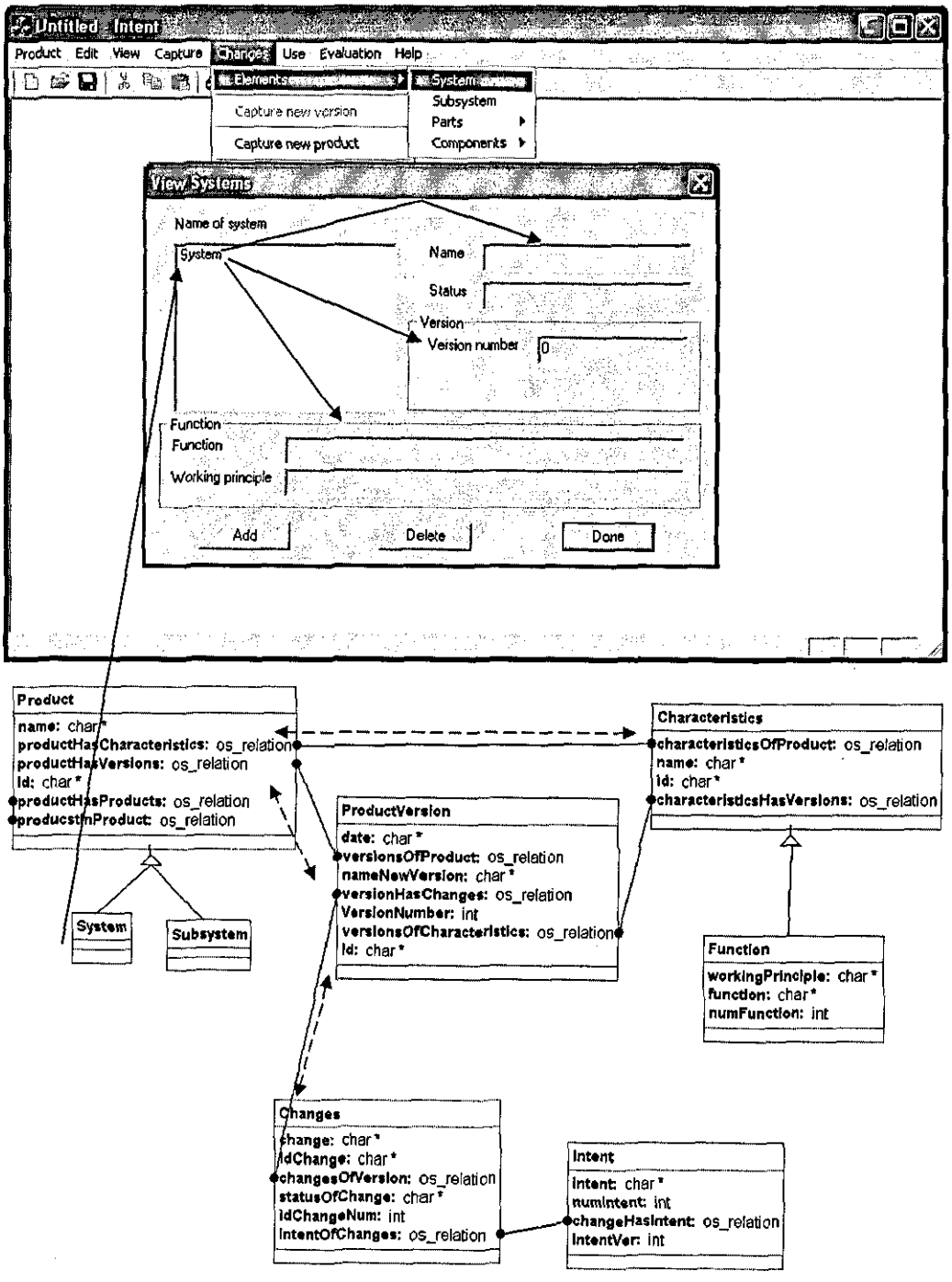

Figure 5.4 Selection of a System to be Changed

Figure 5.5 shows the visual interface used to make changes to the designed components. This dialog has four main areas where information about the components is retrieved from the database and displayed to be changed. The first area 5.5 a), shows the components of the product. When one of the components is chosen, the system retrieves the information of the component such as the function; the assembly sequence number, the last version captured, the status of the component, i.e. original, modified and deleted; the features associated to the component and the material from which the component is made. When a geometric feature is chosen the information related to the dimensions, tolerances and spatial location is retrieved and displayed in area 5.5 c). Finally, the area 5.5 d) contains two groups of three buttons each to make changes to the component or the geometric features.

With the button 'Add' for the component, the dialog to capture a new component appears. When the button 'Deleted' is pressed the status of the component is change to 'Deleted', and the geometric features also change their status as well to 'Deleted'. The systems, subsystems and parts associated to the component that have been modified in any way have their status changed to 'Modified'.

One important aspect of this visual interface rises when changes are made to the geometrical features. When a new version of an entity is created, e.g. part, component, system, the relationship between Version Class and Configuration Class helps to capture this information. However, when new versions of geometrical features are produced the relationship between the Version Class and the Characteristics class helps to capture this information. The component is the only entity that can add or eliminate a characteristic such as a geometrical feature.

A whole set of visual interfaces used for changing subsystems, parts, and components are illustrated in Appendix D.

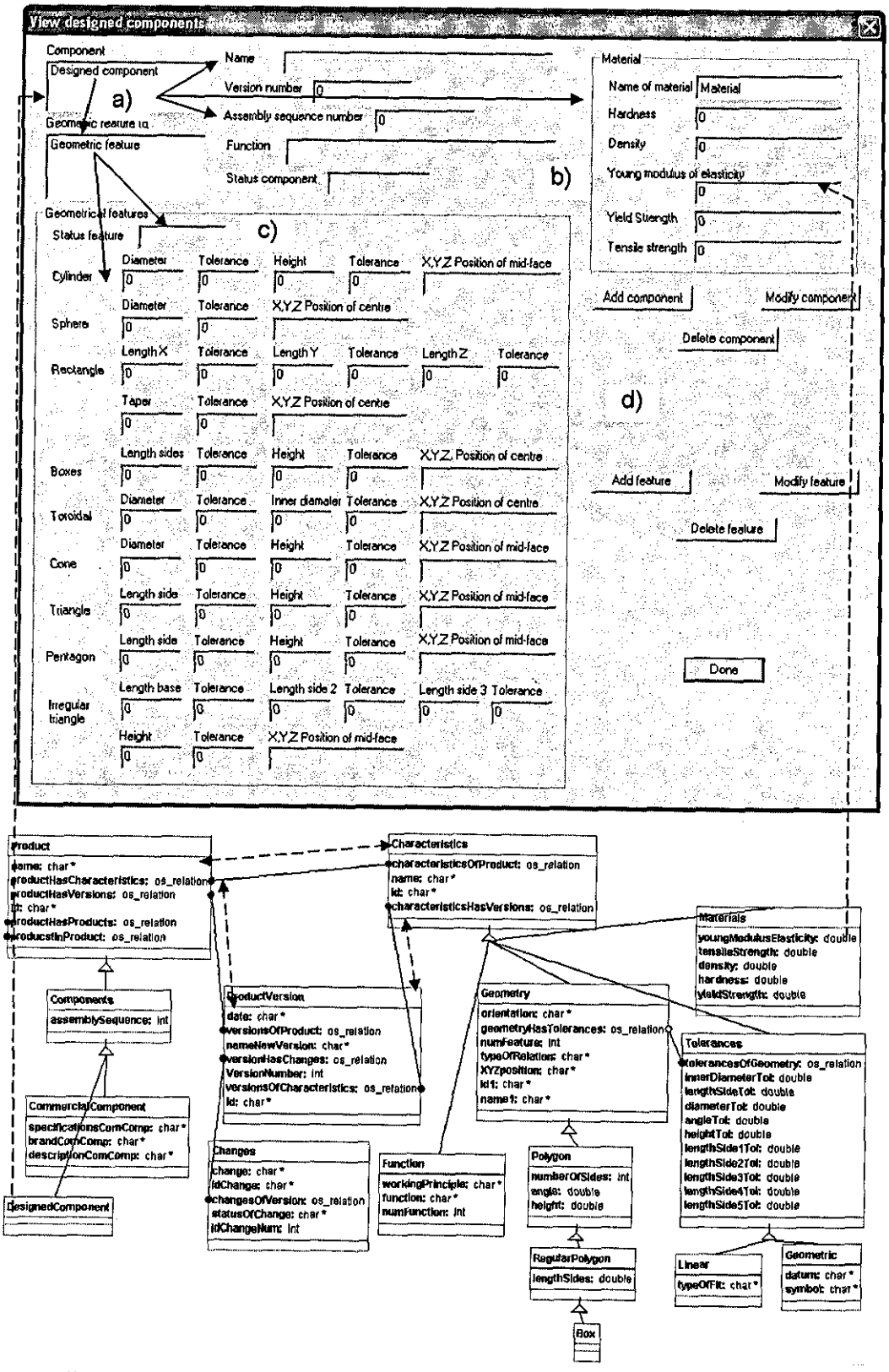

Figure 5.5 Selection of Designed Component and Geometrical Features

An important feature of this structure is that it allows the capture of the changes associated with the versions, keeping the unchanged entities as part of the previous and the new version. The implementation used this feature to avoid copying of the whole information of the product from version to version, keeping databases in a reduced size to promote a faster searching of information. This is important for industrial applications where the amount of information of a product is huge and computational resources need to be optimised.

5.3.4 Capture of the Design Intent

To capture the Design intent, a visual interface was implemented which opens immediately after any change is done. This visual interface implemented, illustrated in Figure 5.6, has five main areas focused on capturing the changes, to capture the Design Intent, to classify the Design Intent, to capture the variations in production cost and time due to the change made, to capture the positive and negative consequences including the variations associated with these consequences.

The area 5.6 a) captures the description of the change made. The system takes information from the dialogs used to make the changes, such as the entity changed, the status of the change and the version where the change is made, e.g. system deleted in version 2. However, the system allows the designer to add or modify this description, e.g. the radius of the fillet was modified from 3 mm to 5 mm

The box in the area 5.6 b) captures the Design Intent Driver underlying the change. This box is inactive when the visual interface comes out. The area 5.6 c) shows nine of the twelve core views where the Design Intent Driver is classified. The system allows selecting only one core view for each Design Intent Driver. Once selected, any core view of the box in the area 5.6 b) becomes active. Furthermore, some aspects are included in area 5.6 c) to capture the positive variation associated to the core view, e.g. for core view Performance, an increase of 5% in the efficiency can be captured in area 5.6 c)

The area d) captures variations in time and cost of the Design Intent. In this implementation, advantageous variation in production time and cost is represented by negative values meaning lower production cost or lower production time.

The area 5.6 e) captures the consequences of the changes in nine different core views. Each area has two boxes to capture the positive and/or negative consequences. The author conveys the belief that one area could have negative and positive consequences due to one change, for example: Changes to the components of a part, i.e. more components with simpler geometries. This change has consequences in the manufacture area. The positive consequence is that each component takes less time to manufacture, but the negative consequence is that more time is needed to manufacture all the components of that part. Therefore when one area becomes active these two boxes become active too. This allows the capture of positive and/or negative consequences in one area.

Four more boxes are associated to each area, labelled *Cost £* and *Time hrs*, to capture the variation in the production cost and time for the positive or negative consequences. The sum of the variations to the initial production time and cost gives an initial idea of the increase or reduction of the cost and time to produce the new version of the product. Production time and cost are two of the three criteria used to evaluate a product.

To assign the Production time and cost associated to the different aspects a deep analysis, which include the participation of the experts in the different areas of the development of the product, is needed in terms of how the consequence in that area affect the production time and cost.

Quality is evaluated using the target specifications, the design specifications and the decision matrix as described in the next section.

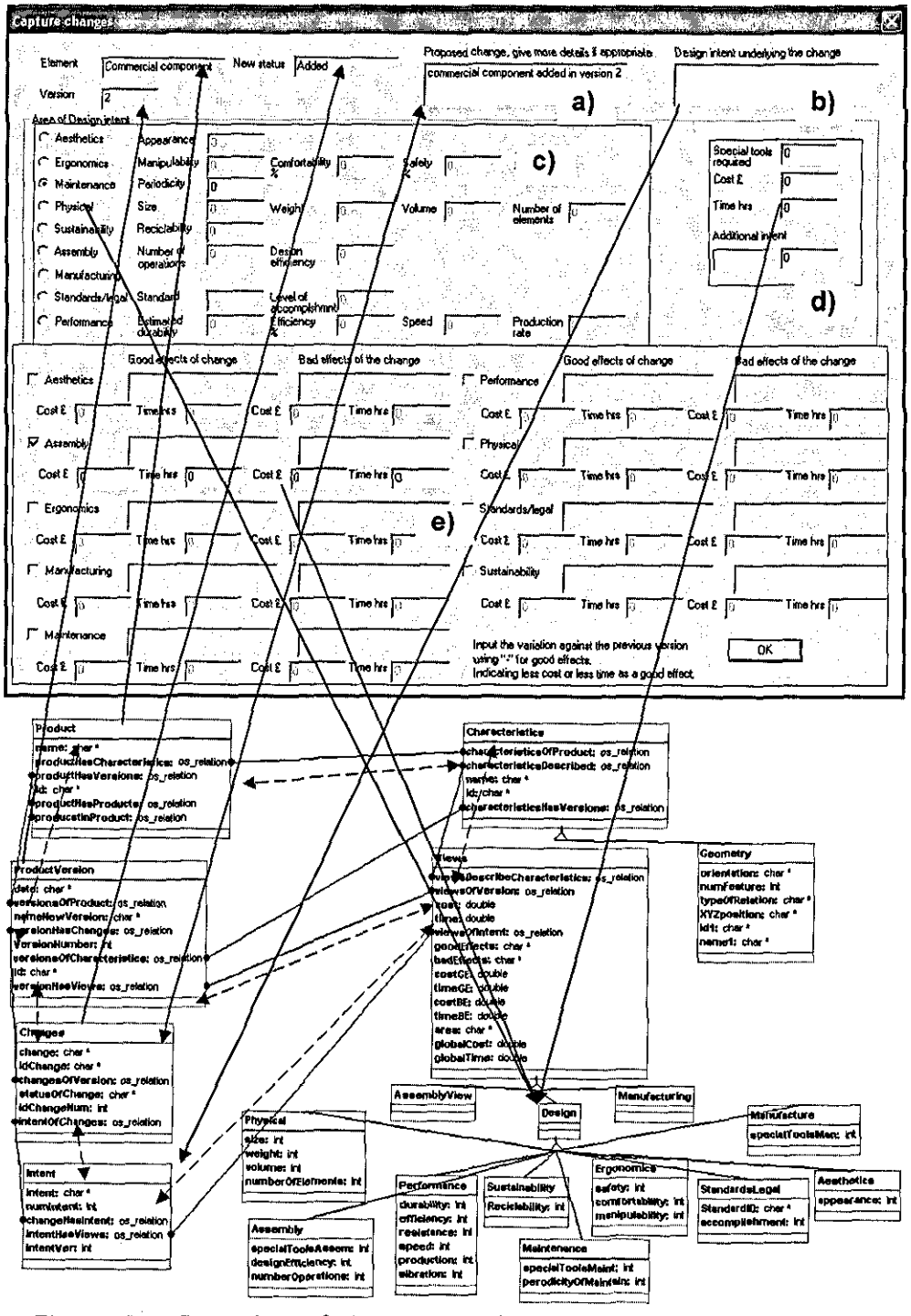

Figure 5.6 Capturing of the Design Intent and Consequences of the Changes

5.3.4.1 Capture of design specifications

The last part of the redesign process is the capture of the final design specifications. The Section 5.3.2 described how the evaluation schema values were used to capture the target specifications.

These final evaluation scheme values are captured through the visual interface illustrated in Figure 5.7. Only the evaluation scheme values are captured because the experimental system uses the same labels for the specifications and the weight factors captured with the initial target specifications. For the implementation it is assumed that the specification names and the weight factors do not change at the end of the redesign. The final evaluation schema values are the result of an internal evaluation process for the new version carried out by the company.

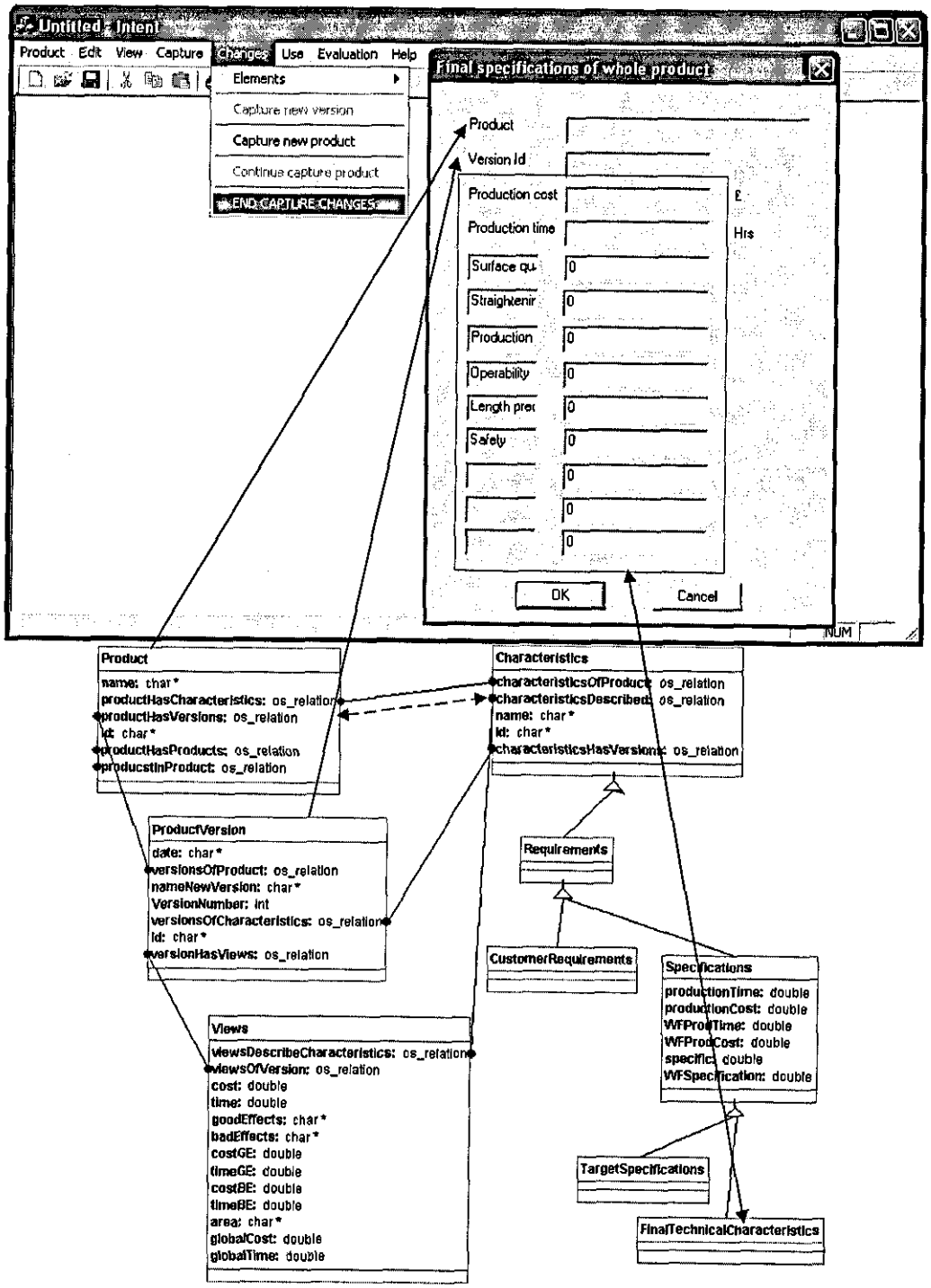

Figure 5.7 Capture of Final Specifications of Redesign

5.3.5 Evaluation of the new version

This section describes how the process to evaluate the new version was implemented following the three criteria used to evaluate a product; these are production cost, production time and quality.

The production cost and time in Figure 5.8 a), are the sum of the numerical variations of production cost and production time for the positive consequences, negative consequences and the Design Intent Driver. Therefore a positive value, which means more production time or higher production cost, or negative value, which means less production time or lower production cost, can be obtained.

As described in section 5.3.2 the decision matrix method was adapted to assess the quality of the new version. This method is also used to assess to what extent the target specifications were accomplished.

Table 5.4 shows the decision matrix in the implementation. The weight factors (WF) captured when the target specifications were introduced are multiplied by the corresponding value of the evaluation scheme (ES) of each specification. The sum of the results of the multiplication gives a quality rate for each version.

	Specification 1 e.g. Weight WF1	Specification 2 e.g. Efficiency WF2	Specification 3 e.g. Speed WF3	Specification n WFn	Quality Rate
ES's of Previous Version 1	ES11 / WF1*ES11	ES12 / WF2*ES12	ES13 / WF3*ES13	ES1n / WFn*ES1n	QF1
ES's of New Version 2	ES21 / WF1*ES21	ES22 / WF2*ES22	ES23 / WF3*ES23	ES2n / WFn*ES2n	QF2

Table 5.4 Decision Matrix Applied to get the Quality Factor of Two Versions

This quality rate is captured in the *qualityOfVersion* attribute in ProductVersion class. This provides an initial way to recall the quality of the version.

Figure 5.8 b) shows three quality rates. The first two quality rates shown belong to the same version. First quality rate is calculated using the final design specifications of the new version. The second value is calculated using the target specifications of the same new version. With these two values an assessment can be made of the level of accomplishment of the new version against the target requirements. The third value, is calculated using the design specifications of the previous version, therefore an assessment of the quality of the new version against the previous version can be carried out.

Finally Figure 5.8 c) shows the conclusions for the three criteria used to evaluate a new version: time, cost and quality based on the results illustrated in Figure 5.8 b).

This section described an initial evaluation aimed to show the potential benefits of the changes made to a product. However, deeper analysis is necessary to actually have a better assessment of the new versions of a product.

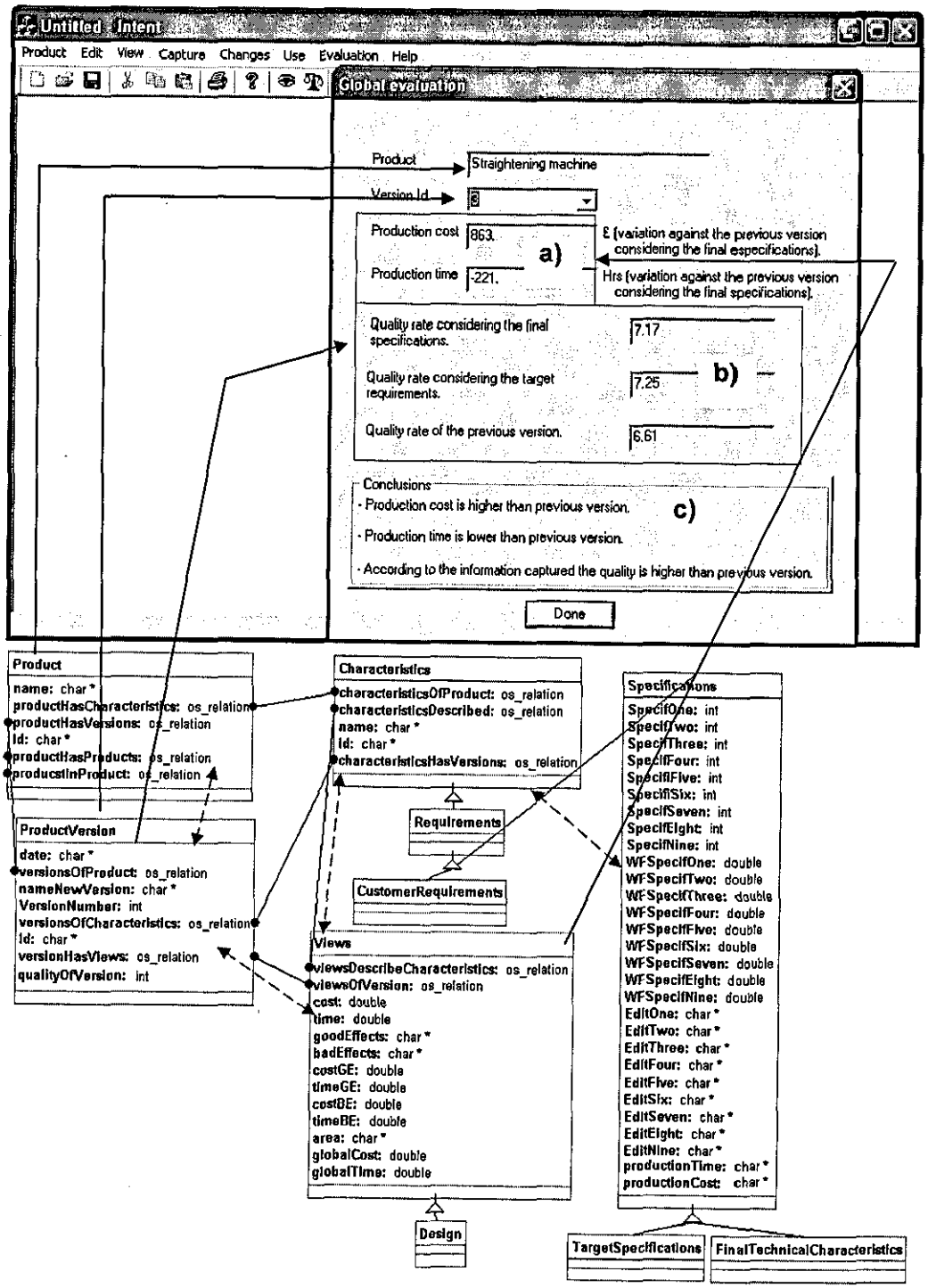

Figure 5.8 Evaluation of the New Versions

5.3.6 Check Design Intent

Redesign is a cycle; therefore this step might be the first for the redesign, rather than the last. However, for the implementation it was important first to capture the information then to show how to use it.

The main aim in this section is to make the next designer aware of the information related to Design Intent. Figure 5.9 shows a general visual interface where most of the information captured can be visualised. The first step was to choose the name of the product, and the version number to be reviewed. Following the sequence of the arrows illustrated in Figure 5.9 a), the systems comprising the product are illustrated. When a system is chosen the subsystems, parts and components within that system appear in their corresponding boxes. In the Figure 5.9 b) the entity chosen is illustrated, its status and the changes associated to the entity selected.

By selecting a change, the Design Intent underlying that change is shown in the box placed below the Changes box. In the boxes on the right, two boxes show the areas with the positive, the negative consequences and their associated variation in production cost and time.

The lower part of Figure 5.9 shows the classes used to retrieve the information and the relationships used. This visual interface provides the means of interacting with the product model and therefore how to access the captured design history of a product.

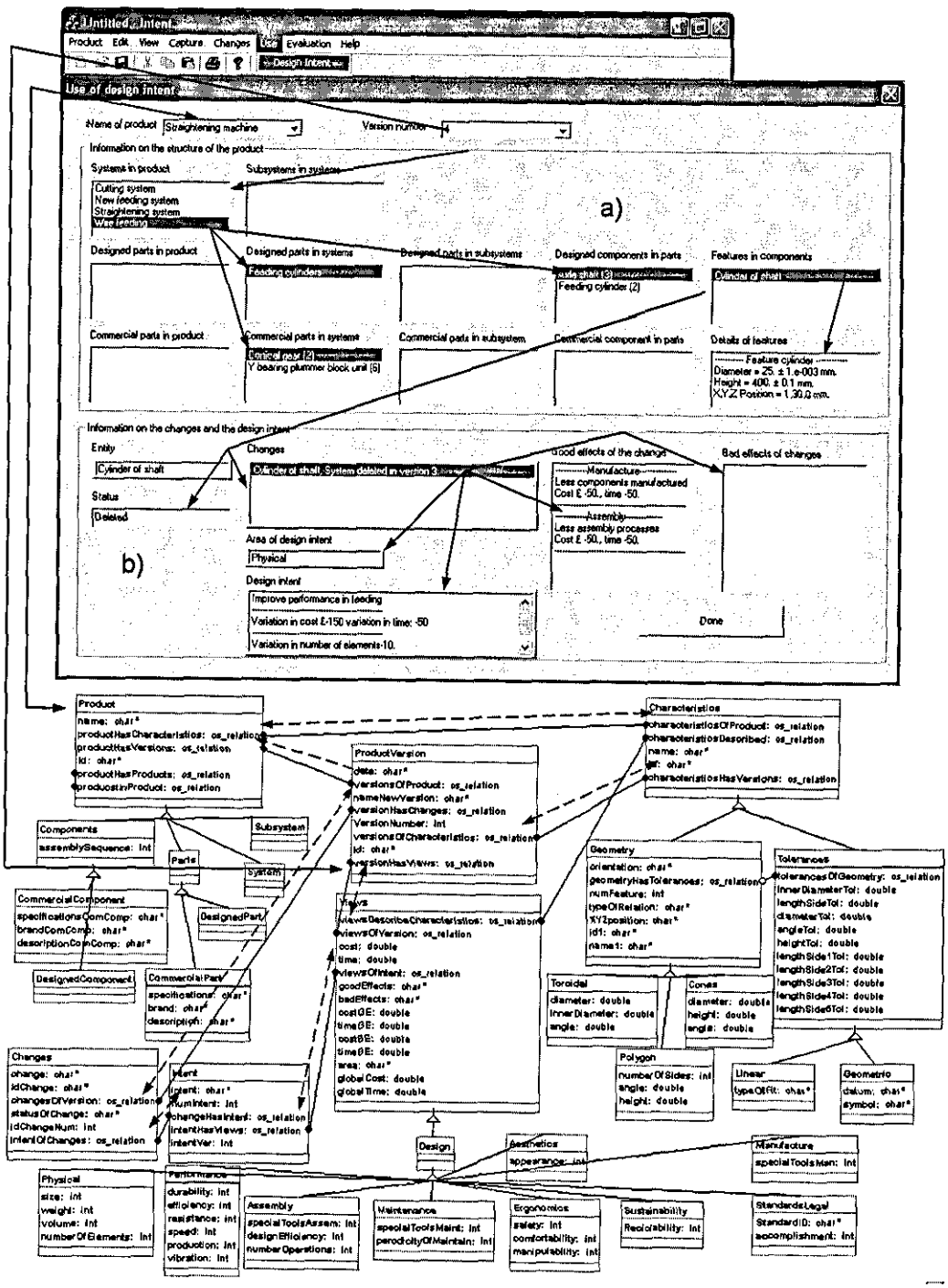

Figure 5.9 Implementation of the Review of the Design History of the Product

5.4 CASE STUDY

5.4.1 Background of the redesign problem

The Manufacturing and Design Centre of the National University of Mexico supports Mexican industry by solving design problems in many areas. One of these problems was the design and manufacturing of a group of industrial machines to manufacture surgical needles from wire in coils.

This case study focuses on the various redesigns done to the straightening machine, which is the first machine in the manufacturing process. This machine has the function to straighten the wire from the coil and cut it into segments with

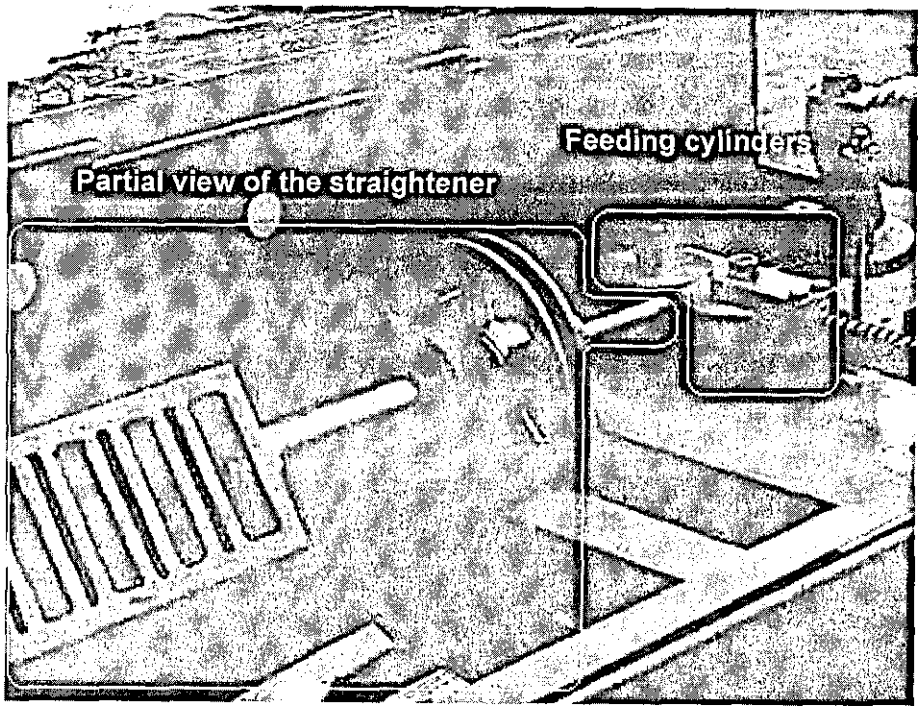

Figure 5.10 Photograph of the straightener and a pair of feeding cylinders at the end of the straightener

a predefined length segments. Section 4.4 introduced and explained this project to some extent. Figure 5.10 shows a photograph of a section of the machine. The straightener in the photograph is comprised of components welded into one straightener, which needs further balancing to work properly. The working

principle of the machine is to force the wire into a rotating sine wave; this process is done in the straightener illustrated partially in Figure 5.10. The straightener has 10 adapted studdings in a sine array, as illustrated in Figure 4.7. The straightener rotates making the wire follow a sine wave in an infinite number of planes. Figure 5.11 shows a different commercial straightener with two fixed planes. Since the cross-section of the wire is circular, the rotation of the straightener gives a better straightening quality.

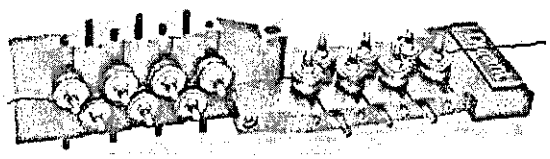

Figure 5.11 Commercial Straightener with Two Planes.

A commercial machine was bought to straighten the wire. However some problems arose because this machine was originally designed to handle 6 mm diameter bars for the building industry. The wire for surgical needles has a diameter of 0.7 mm. The straightening quality required for the bars is not the same as for the wire, this is in a length of 37 mm and the actual position of the tip of the wire must not be farther than 0.1 mm against an ideal straight line. Figure 5.12 illustrates how this error is measured. The rough design and manufacture of the straightener could not provide the straightening quality needed. A redesign was needed to sort out this initial problem.

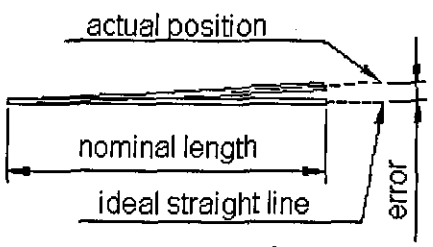

Figure 5.12 Measuring of the Straightening of a Wire.

A subsequent problem was detected after the first problem was resolved in the first redesign: spiral marks were detected on the surface of the wire. As

described in section 4.4.1 these marks were unacceptable. Therefore, a second redesign was necessary to correct this major problem.

Figure 5.13 shows the different main changes made to the different versions of the machine. Version 1 shows the original straightener, which uses studding as a die with the tip modified and two sets of cylinders that feed the wire into the straightener. As described in section 5.4.1, the straightener in version one was comprised of 30 welded components. In the second version, a new singled-component straightener which uses a new die, replaced the multicomponent straightener. In the third version a commercial wire feeder replaces the feeding

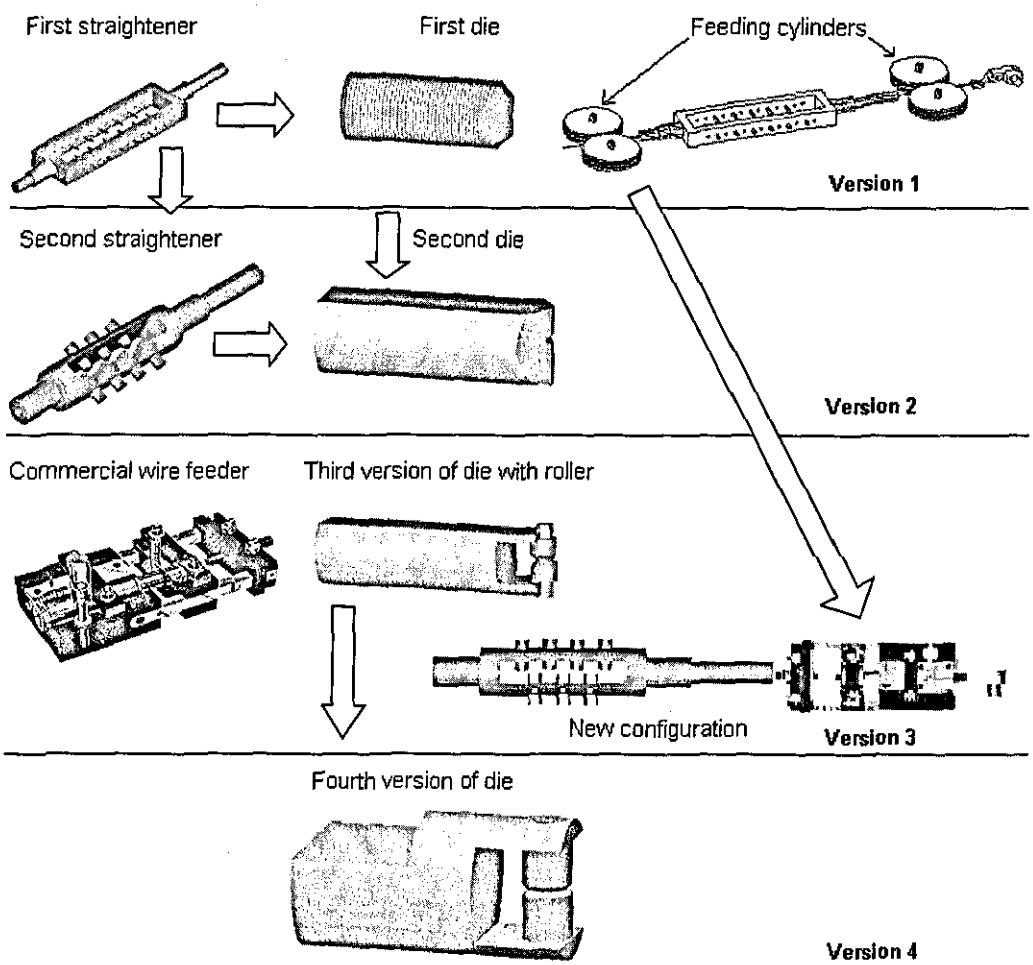

Figure 5.13 Main Changes in Different Versions

cylinders and a roller is added to the die. Figure 5.13 shows the new

configuration of straightener and feeding system in the third version. In version 4 a new version of the die is used in the straightener.

The redesign carried out to this straightening machine was used by this research as a case study. The next two sections describe the experiments carried out.

5.4.2 Models to support the redesign process

This experiment is to test if the product model can support the capture of the information produced when a redesign is done. This includes three different tasks: 1) The capture of the target specifications, 2) the capture of the changes made in each version, and 3) the capture of the Design Intent underlying these changes.

For the wire straightening example introduced in section 4.4.1.1, the target specifications defined are: Quality of surface, which is associated with the depth of the marks on the surface of the wire; Straightening quality; Length precision and Production associated to the number of straightened wires produced by the machine. The target production cost is £ 1758.36 and the production time is 288 hrs.

The evaluation scheme values assigned to the target specifications are shown in Table 5.1. A value of 9 is assigned to the Surface q*uality* representing an "excellent solution"; the same value as for the *Straightening quality*. For the *Production* a value of 8 is assigned, representing a "very good solution". For the *Length precision*, the value assigned is 5 representing a "satisfactory solution".

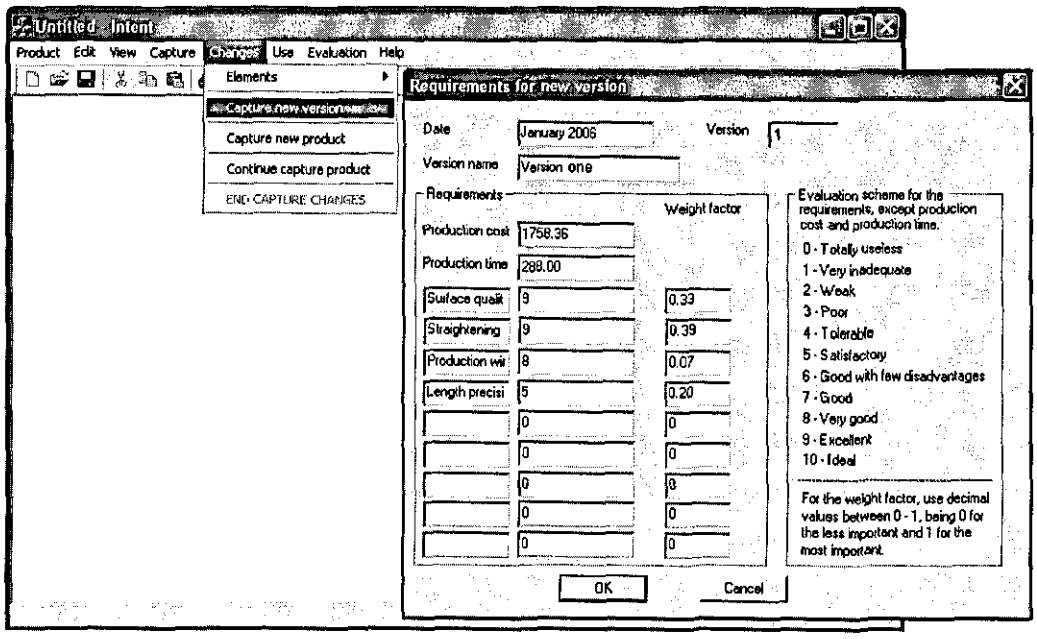

Figure 5.14 Capture of the Target Specifications

Three systems comprise the straightening machine; these are shown in the dialog to modify the systems: Cutting, straightening and wire feeding systems.

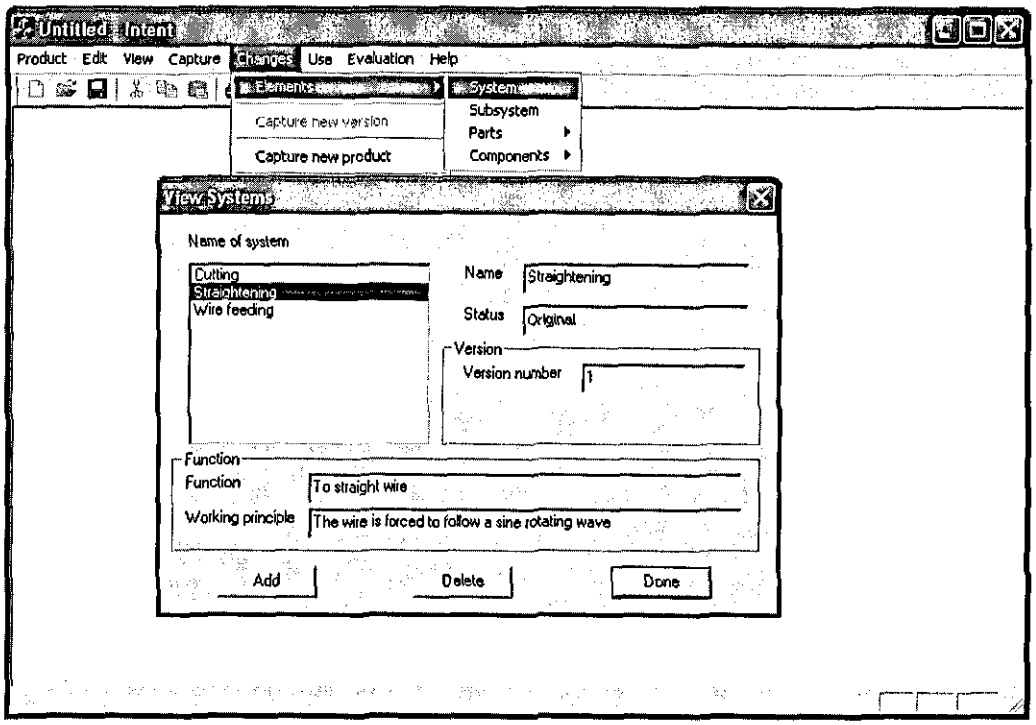

Figure 5.15 Selection of a System to be Changed

The capture of the changes is done through different visual interfaces for different entities of the configuration of the product. All the visual interfaces used to make the changes to all entities of the configuration of the product are illustrated in *Appendix D*.

Figure 5.16 shows the visual interface to change the designed components. A list of all the components are shown in box 5.16 a). Some of these components were eliminated such as Front axle of bolster in version 2. Therefore, in the box labelled Status component, appears the legend Deleted, consequently the geometrical feature associated to that component, named Front cylinder feature, appears deleted.

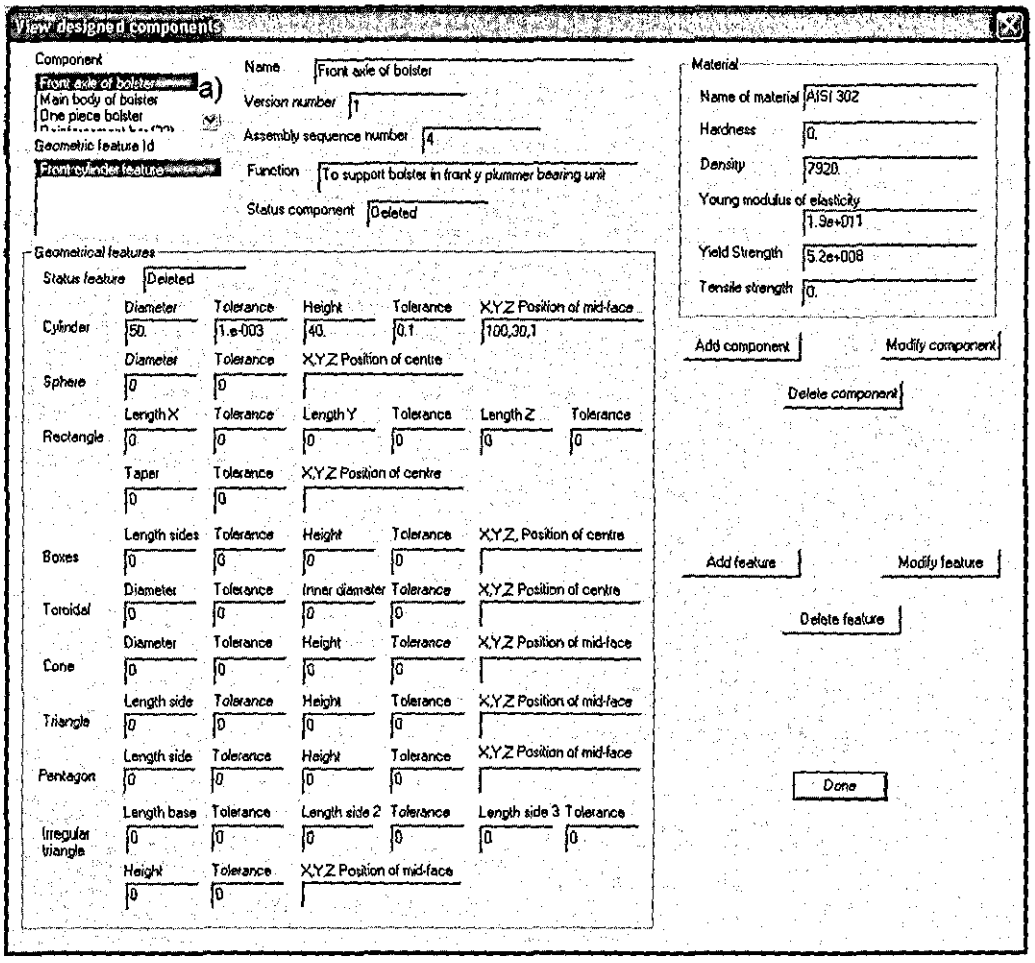

Figure 5.16 Dialog to modify the designed components showing a deleted component

Figure 5.17 shows a different case. Four geometrical features comprise the designed component 'New die': *Body of main die*; *Flat surface*; *Guide in new die* and *Tip of new die*. The material for this component is illustrated in Figure 5.17 d). The designed component 'New Die' is part of version 2 and its status appears as 'Modified'. One of the modifications shown was the added geometrical feature. When the geometrical feature "Flat surface in die" is chosen, the experimental system retrieves and shows it in area 5.17 c) the information of this feature such as the status, length on axes X, Y, and Z with its corresponding tolerances and the spatial position X, Y, Z.

The status of this geometrical feature is 'Added', this implies that this

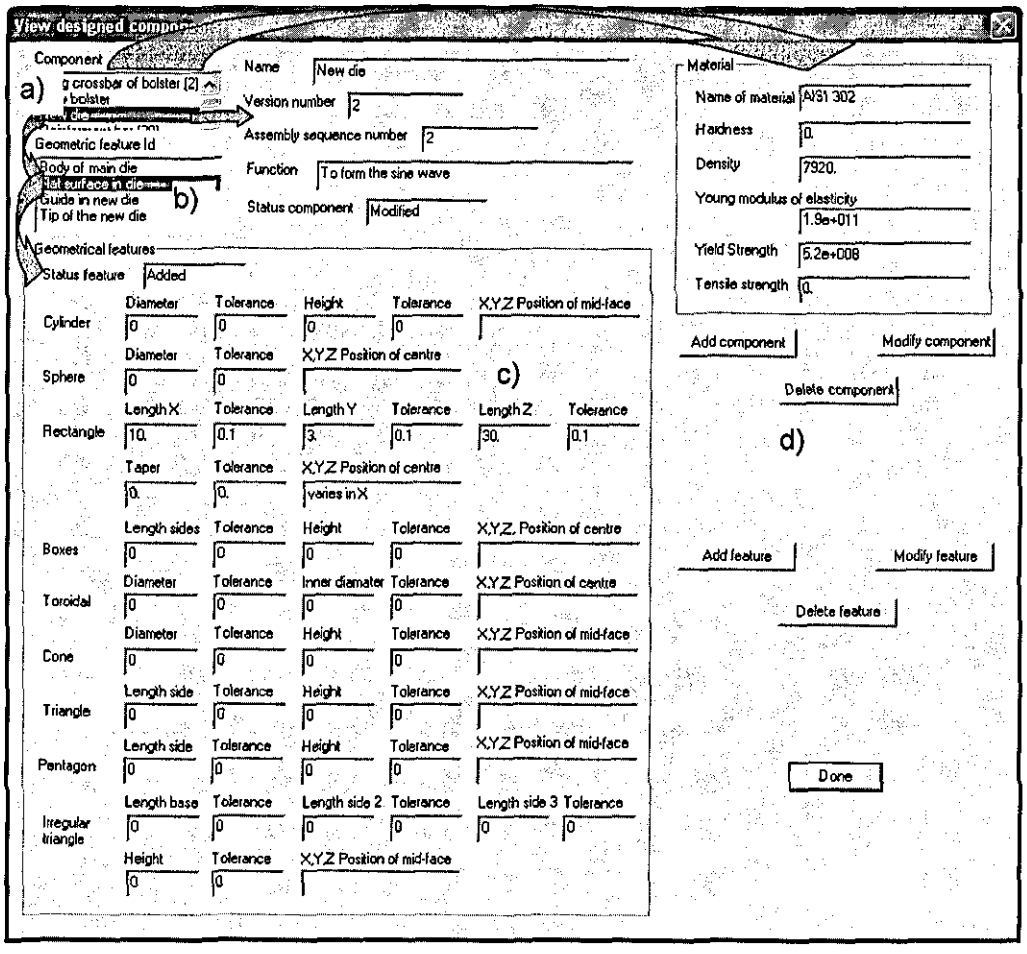

Figure 5.17 Visual Interface for the Changes to the Components

component had a change hence its status is 'Modified'. The experimental system identifies when an entity in the lower levels of the configuration was changed and adjusts the status of higher levels of the configuration. Therefore

the systems or subsystems to which this component belongs change their status automatically. On the other hand if an entity such as a system is deleted, all the entities below, change their status to "Deleted", the system identifies this, and no further changes can be made to these entities.

After a change is finished the system displays the dialog to capture the Design Intent Driver. Figure 5.18 illustrates the dialog used to capture the Design Intent Driver that underlies the addition of a roller in the die, see Figure 5.13 third version. The Design Intent Driver is to eliminate marks on the surface of the wire. The area with more influence is the Standards/Legal. With this change the expected accomplishment of the standard is 95%. The fact of accomplishing the standard does not imply a variation in the production time or cost. However the consequence of adding the role to accomplish the standards has consequences in the manufacturing, assembly and performance areas. For the areas of Manufacturing and Assembly the negative consequence, reflected in an increase in the production time and cost, is identified. For the area of Performance the positive consequence, improvement in the performance of the machine and the needle, did not reflect a direct variation in the production time and cost. However an important improvement in the quality was achieved.

One important aspect detected during this experiment was that many changes are associated to one Design Intent Driver e.g. Changing the geometry of the die in version 2 to version 3, shown in Figure 5.13, implied the elimination of some existing geometrical features to later add other geometrical features. All these are detailed changes, which actually represent the steps that a designer would follow when using commercial CAD software, and can be finally described for practical purposes as one change and therefore all the detailed changes are associated to one Design Intent Driver. Therefore the implementation needs some changes to allow the designer to associate the detailed changes to one important change, with one Design Intent Driver underlying it.

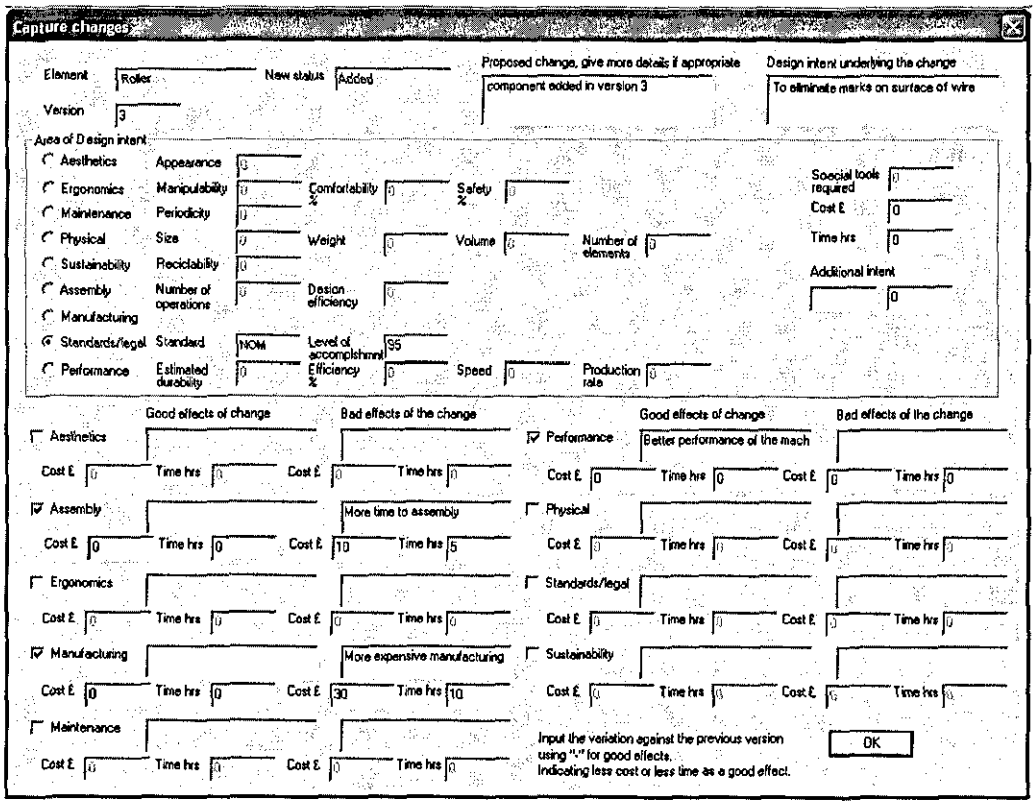

Figure 5.18 Capture of Design Intent

Once this information is captured for all the changes made, the last step in the redesign process was to capture the evaluation scheme values according to the convenience of the solution achieved, as described in section 5.3.2.

Following the values of Table 5.1, the values shown in Figure 5.19 reflect that the problem of the marks on the surface of the wire was not faced in this version; therefore the convenience of the solution was poor. However, the straightening quality was improved until the level of an 'excellent solution' level was attained. The production of wires did not change. Finally the length precision improved to reach the 'good solution' level.

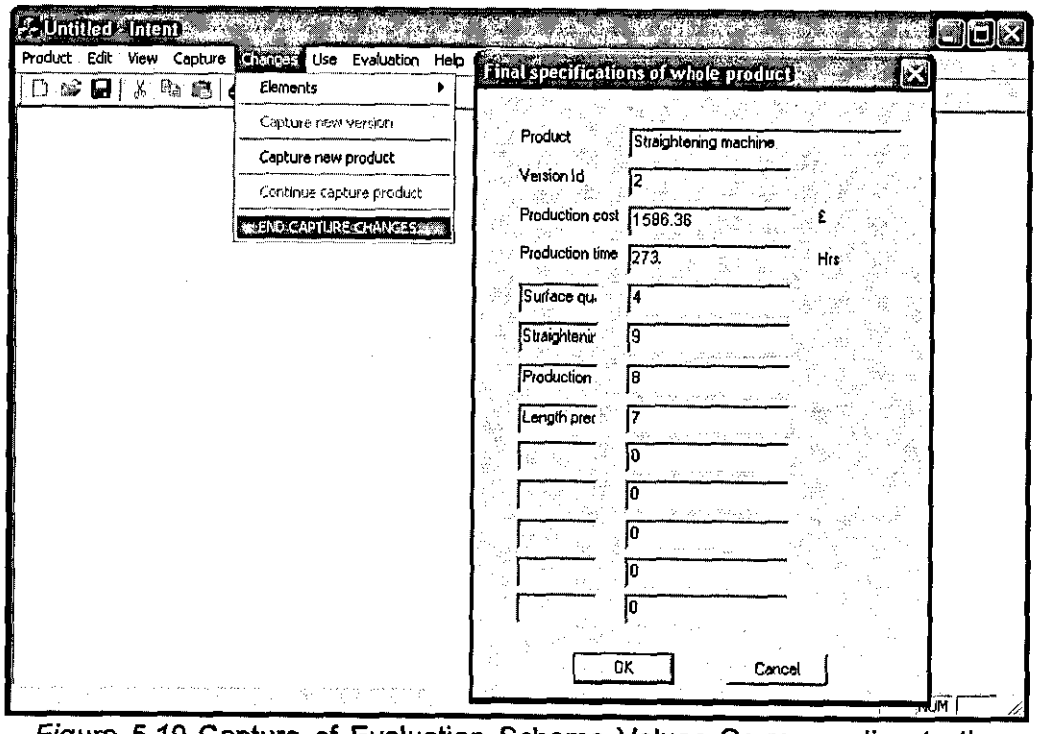

Figure 5.19 Capture of Evaluation Scheme Values Corresponding to the Design Specifications

This information is used by the system to calculate the quality rate for the target specifications and design specifications of the new version using the method described in section 5.3.5.

5.4.3 Models to support the use of the Design Intent information

This experiment is to test the ability of the Product Model Structure to be the repository of the information related to the Design Intent Driver and how the experimental system retrieves this information to support the redesign process. This support is achieved through one visual interface. Figure 5.20 shows the dialog used to retrieve and show the Design History information.

The process to review the Design Intent starts by choosing the product and the version of the product to be reviewed. The experimental system shows the three systems that comprise the product associated with the selected version. For this experiment the systems are: Straightening, Feeding and Cutting.

The upper part of Figure 5.20 shows part of the information of the configuration of the product and some of the changes made for version 2 of the product. Following the sequence of the arrows, Figure 5.20 b) shows the Design Intent of the change made to the Straightening system. A designed component named 'New bolster' was added in version 2. The Design Intent Driver of this addition is shown in Figure 5.20 b) and was to simplify the manufacturing process of the straightening element.

The lower part of Figure 5.20 shows the dialog for version 4. The box labelled 'Systems in product' shows four systems. The Wire Feeding System belongs to the first version of the machine; in version 3 an air feeding system replaces this system. In order to keep the Design history of the product, the old version of the system is shown, however, its status is deleted. This dialog provides support to make better decisions by offering a complete picture of the changes, their consequences and the Design Intent underlying them.

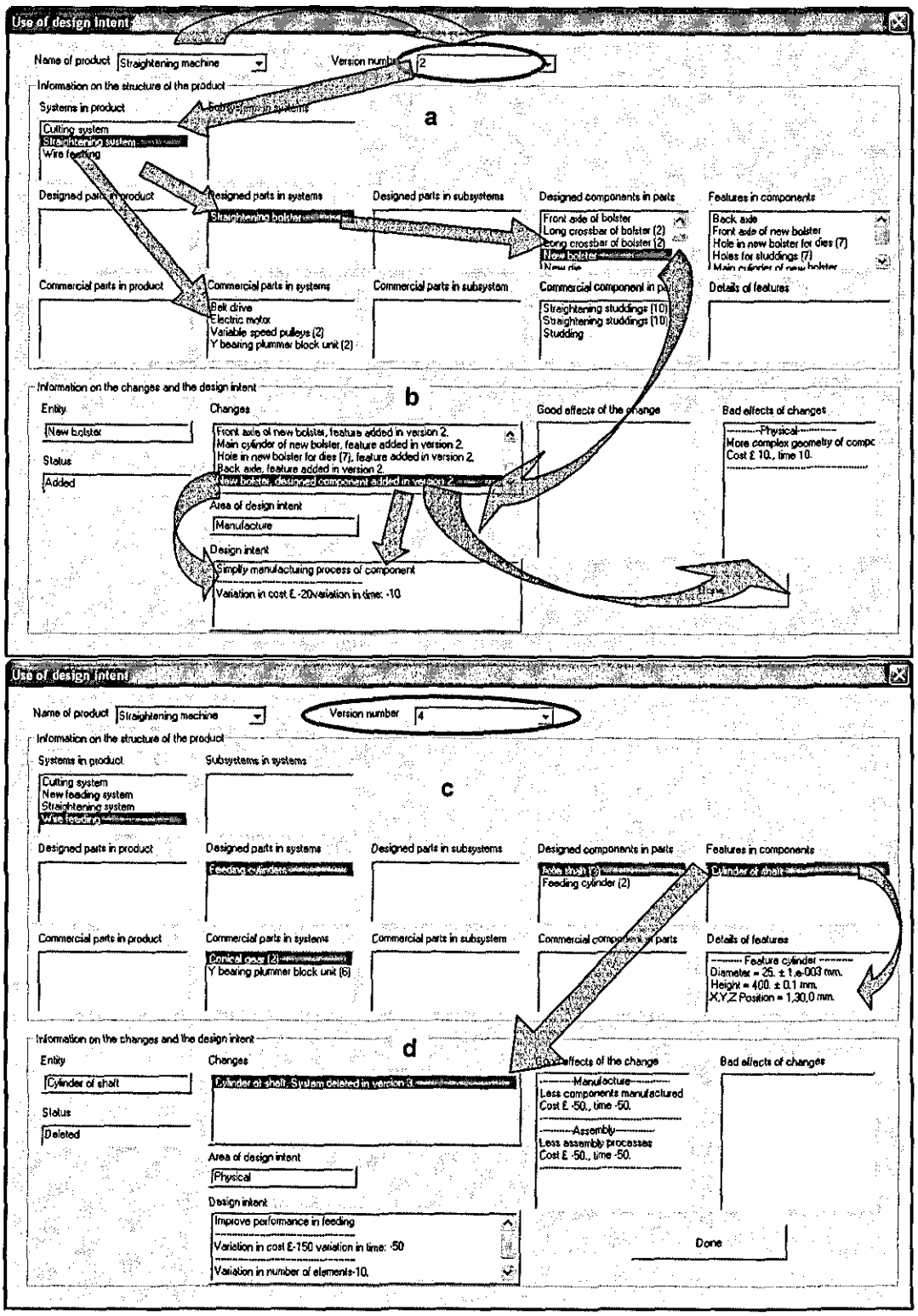

Figure 5.20 Use of Design History Information

5.4.3.1 Assessment of the new version

Lower production cost, shorter production time and higher quality are the three criteria generally accepted to make an assessment of a product.

The Figure 5.21 shows the evaluation of Version 2. The Production cost box shows a reduction in the production cost. For the Production time the changes reflect no variation. This result reflects that the sum of the production times of all the changes is zero.

The Quality rate calculated using the target specifications is 8.08, very high compared to the quality rate of the first version. For the second version there is an improvement in the quality reflected in the value of 6.61 against the 6.13 of the first version. However the quality rate for the target specifications is not reached in this version. The lower part of Figure 5.21 shows the conclusions based on the values shown.

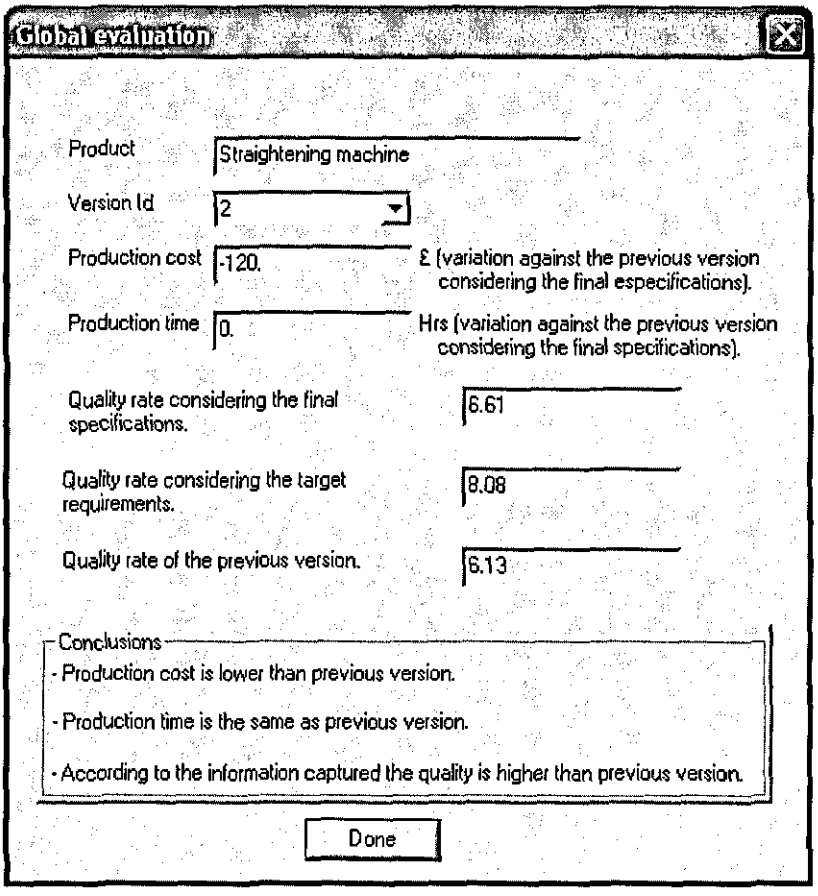

Figure 5.21 Evaluation of Version 2

The Figure 5.22 shows the results for version 4, which is the best version of the machine due to the changes in previous versions. Comparing the values of production cost and time, the negative values represent the reduction in production time and production cost against the previous version. For the quality, the previous version has a quality rate of 6.86 against 8.08 of the new version.

In Figure 5.22, the quality rate of the target specifications is 7.78, lower that the quality rate of the final specifications of the version, therefore the quality achieved surpassed the quality sought.

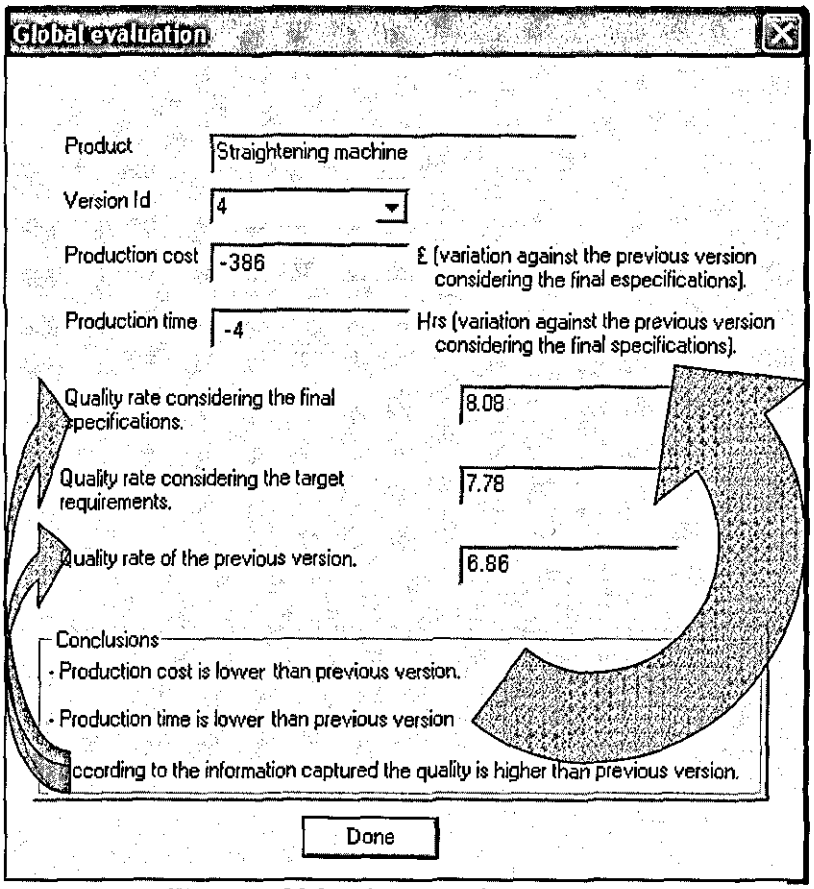

Figure 5.22 Evaluation of Versions

This information shows the designer how the new versions have improved the product.

5.4.4 The Navigation Process.

Some of the information was visualised through the dialogs developed in the implementation. A different way to visualise the information and verify that the relationships between the related objects was done properly in agreement with the PMS is by using the navigation process of the ObjectStore Inspector.

Figure 5.23 shows part of the navigation process between some descriptive instances of the classes of the structures related to the changes using the ObjectStore Inspector instance pane. It clearly shows that after the experiment, the PMS captured successfully the six classes of information: Configuration information identified with the car icon; Characteristics information, identified with the pencil; Views information identified with the glasses; Version information identified by the version logo; Changes information identified by the arrows icon and Intent information identified by the with Intent icon.

Figure 5.23 a) shows the instance Product as the start of the navigation process. This figure shows the systems Straightening, Feeding and Cutting, which comprise the product in a higher level. This figure shows also two sample characteristics of the Product, as a whole, such as Target specifications and Function.

Following the navigation process for the Straightening system located in the upper part of figure 5.23 a), two new instances are illustrated in Figure 5.23 b) *ProductVersion* and *DesignedPart*. The upper part of this figure shows the navigation process for the new version of the Straightening system, which was redesigned. Following this navigation the changes made to the Straightening system are illustrated.

In the lower part of Figure 5.23 b) the *DesignedPart* instance is illustrated as part of the Straightening system. The designed part straightener shown in this

figure is comprises the *DesignedComponents* instance. This figure shows only one designed component.

Following the navigation process from Figure 5.23 b) the Changes class continues in Figure 5.23 c) where the Design Intent class is shown at the top of this figure. Two instances of the Views class are shown: Performance and Manufacture. These two instances represent, *Performance*, as the main area where the Design Intent is classified and *Manufacturing,* the area where the consequences are presented. As stated in section 5.3.1 point 2, despite the structure implemented, is not the same as the defined PMS, it succeeds in the objective to capture the consequences and successfully classify it in the core views.

In the lower part of this figure the Characteristics instances of the design component are shown including the Function, Feature, Tolerance and Material instances.

The realisation of the information navigation process illustrated in Figure 5.21 shows that the PMS has been implemented successfully and is capable of capturing and storing the information related to Design Intent.

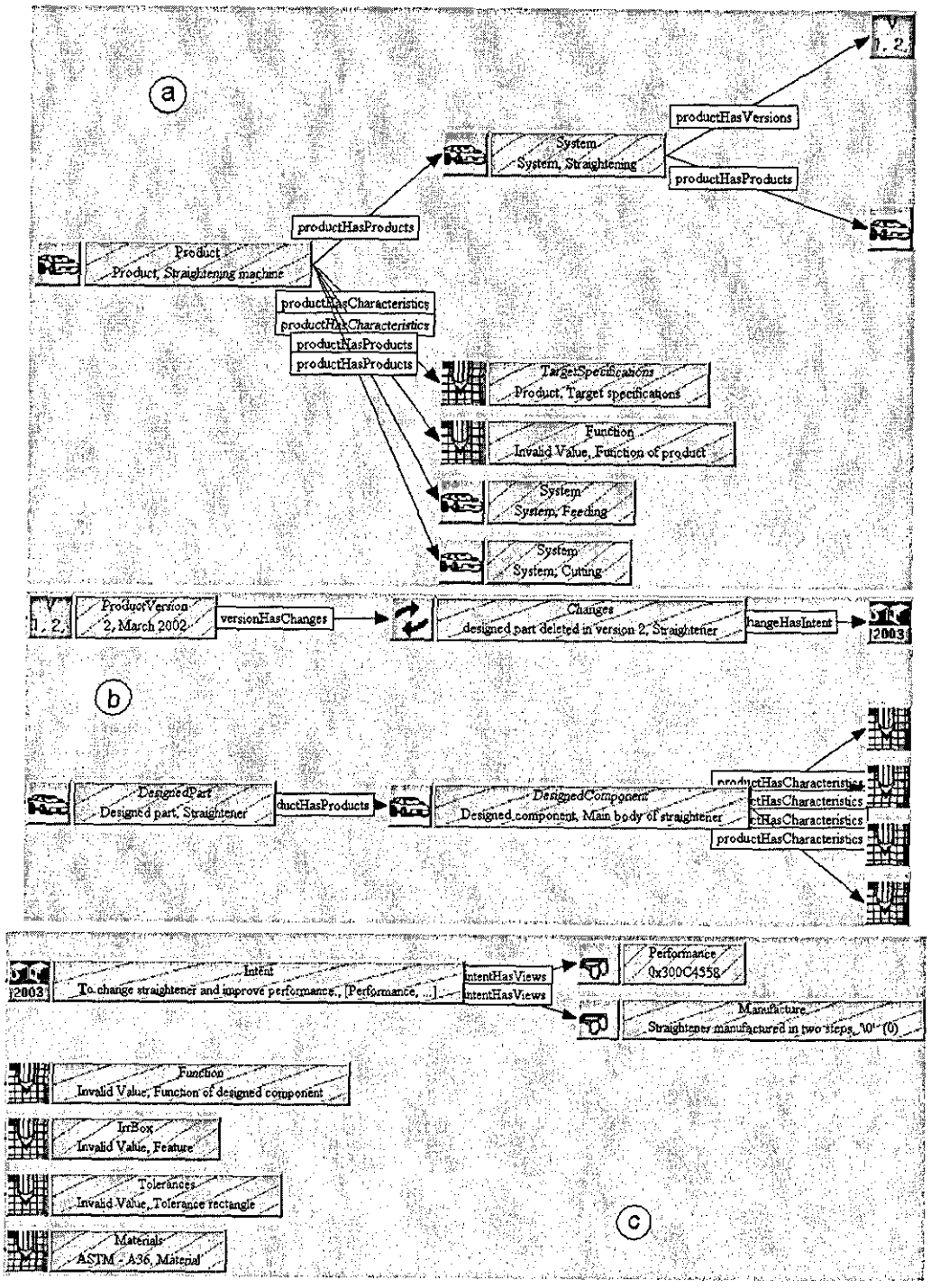

Figure 5.23 Information Navigation Trees for PMS

5.5 SUMMARY

This chapter described how the experimental system was implemented and demonstrated the results of the experiments.

The results achieved show that the PMS explored in this work can capture successfully the required classes and relationships between the classes. The PMS has been shown, in the context of an industrial problem, to adapt a commercial machine to produce straight wire for the manufacture of surgical needles. However this PMS can be applied to most electromechanical products, therefore it has a more extended applicability.

6 DISCUSSION, CONCLUSIONS AND FURTHER WORK

6.1 INTRODUCTION

The research reported in this thesis has explored how information modelling can support the redesign of products.

The main contribution of this research is a Product Model Structure composed of Configuration, Characteristics, Views and Version classes. It was designed to support the capture, storage and use of the information related to Design Intent.

An experimental system and a case study examined how the PMS works as a source and repository for the information needed to support the redesign of products.

Section 6.2 presents a discussion of the main issues arising from the research. Section 6.3 presents the conclusions reached and section 6.4 presents recommendations for further work.

6.2 DISCUSSION

This discussion consists of three sections. The first section discusses the method followed; the second section discusses the structure for the Product Model and the last section discusses the application of the Product Model Structure.

This research used some product design examples, information from published material and the background knowledge of the author to explore and model the information related to the Design Intent.

This research used examples ranging from vacuum cleaners to straightening machines, to explore the different ideas behind modelling the structures that comprise the Product Model Structure. The vacuum cleaner provided a good

example of how the change of working principle to accomplish the common function of all vacuum cleaners creates a new product. The photographic camera provided a good way to describe part of the configuration of the product. However, no difference is established between part and component. The headlamp of a car provided the possibility of illustrating the description of a part. The straightening machine used as the case study in the experiment provided a product with the complexity necessary to test the PMS and the simplicity to be understood and captured by the experimental system developed. However, unlike other mass production products, the customer requirements for this product were the technical specifications, which did not require the use of the House of Quality technique. Despite this, the House of Quality provides the means to translate the wishes of the customers into more technical specifications.

This research used UML and Use Case Diagrams for the design and representation of the experimental system developed. UML provided effective support to show in detail the classes and relationships necessary to give the PMS the capacity to act as both a source and repository for the information related to Design Intent. Despite the subjectivity of the design task, this language provides the means to create a formal representation of the information needed in the redesign process from the perspective of this research.

6.2.1 Structure for the Product Model

Four classes and the relationships among them were defined as part of a Product Model Structure aimed to capture the information related to Design Intent, these are: Configuration, Characteristics and Views of a product, which are related to the description of the product. To provide the ability to capture the Design Intent the structure Version was defined.

6.2.1.1 Class Configuration of the Product

From the design point of view, the description of the configuration of the product is based on the Function. This is not a new way to describe a product, however, this research formalises this description in the PMS.

The PMS mainly focuses on electromechanical products mainly. Electronic products are beyond the scope of this research. Despite that some electronic components might be part of a product; the PMS does not consider details of their design characteristics such as: input signals, output signals and frequency.

As products grow in complexity, systems become increasingly interrelated therefore difficulties are expected in identifying the systems, subsystems, parts and even components. This research does not deal with this type of interrelation.

6.2.1.2 Class Characteristics that Describe the Product

The redesign of a product is reflected in changes in five main characteristics: Function, Dimension, Geometry, Tolerances and Material. Changes to any of these characteristics may impact on the production time, production cost and quality of the new version.

Changes to the design characteristics of a product, result in new versions of the product. However, changes to the main function create a new product. In the PMS defined in this research, the class Function has two attributes: *functionOutput* and *workingPrinciple.* Changes to any of these attributes result in a new product. This is important because finally a product is the evolution of other products that become part of the Design History of any product.

6.2.1.3 Class Views of the product

Different points of view can describe a product, for example design, manufacture and assembly. In this research, this class extends its ability by providing the possibility to classify the Design Intent and the consequences of the changes into twelve core views. Pugh (1991) proposes thirty-two elements to define the specifications of a product. Chapter 4 discusses how three main areas reclassify these thirty-two elements: high-level criteria to evaluate products, general generators of design specifications and core views.

In an effort to simplify the classification of the Design Intent Driver and the consequences, the twelve core views defined for this class are aimed to cover the whole context around the product. Aspects such as product life cycle aspects, behavioural aspects, legal aspects and physical properties were considered in this definition. However, some difficulties may arise due to the interrelations among these areas. For example: if the geometry of the components of a large and heavy product is changed with the Design Intent of making the product easier and safer to assemble, it is not obvious which of the core views exerts more influence and therefore where the Design Intent Driver must be classified. The most logical is assembly, however ergonomics could also be the core view or perhaps safety. More core views could well result in an overload of information.

Not all the core views have a representation in terms of production cost and time. In fact, two of them are more strongly related to the production time and cost: Manufacture and Assembly. Aesthetic, for example, cannot have a direct representation in terms of production cost and time. The manufacture and assembly of a product with a better appearance has relationships with production time and cost, but not the aesthetic itself.

6.2.1.3 Structure for the Version of a Product

The structure Version in the defined PMS focuses on capturing the historical information about the product. This structure follows the idea that a redesigned product has versions. The versions have changes. Two types of Design Intent underlie the changes, i.e. the Design Intent Driver and the secondary Design Intents. Class Intent captures the Design Intent Driver. However, the experimental system was focused to implement the Design Intent Driver.

This research follows the idea that changes feature in the redesign stage, while decisions feature in the design stage. Changes and decisions have the same nature as both imply a choice of different options. However, just to establish a difference between the two stages, decisions are associated to the design stage and changes are associated to the redesign stage. Based on this, the PMS has the ability to capture the Design Intent underlying the changes made in the

redesign as well as the Design Intent underlying the decisions made in the design stage. However, the latter has not been implemented.

6.2.1.4 Relationship between the structures of the PMS

In order to ensure the capability of the PMS to capture, retrieve and be the repository of the information related to Design Intent, the classes and version structures need to work together.

The class Views has three subclasses: design, manufacture and assembly. The relation between class Intent, in the Version structure, and class Views opens the possibility to capture Manufacture Intent and Assembly Intent understood as the reasons underlying how the product was manufactured and assembled as it was.

6.2.2 Application of the Product Model Structure

To test the capabilities of the defined Product Model Structure defined; an experimental system was developed to prove the application of the PMS. The experimental system demonstrated how the PMS could capture and use the information related to the Design Intent. The use of the use cases and the class diagrams developed using the UML notation provided effective support for the development of the experimental system.

In UML, the classes have attributes and operations that represent the behaviour of the object. These operations were not used to develop the experimental system. This was not a problem to test the validity of the PMS, however, a higher capability would have been achieved by including these operations.

The experimental system developed proved to be effective for exploring the ideas discussed in this thesis. It proved how the PMS could capture the historical information of the changes, the Design Intent underlying them and the consequences of the changes. A more detailed discussion is made in the next sections around the PMS and the redesign process

6.2.2.1 Assessment of the PMS for the redesign process

This author considers that, as defined, the PMS has the potential to support the redesign process, because it covers a complete range of key information concerning of the product and its evolution. Despite the database was populated with actual information of the redesign made to an industrial machine, it is important to test the PMS in the real-time conditions of a redesign process, where the information is transferred between one designer and the next to test the actual level of support provided to the decision making process.

6.2.2.2 The relationship between the operational data and the PMS

The customer requirements are some of the most important aspects that boost the redesign of a product; therefore, once the product is in an operational stage it is important to 'listen' to the voice of the customer. However, this PMS does not consider a way to capture the criticism of the product by the customers as a feedback. It is constrained to capture the customer requirements once these have been were translated into the target requirements.

On the other hand, the evaluation of the product was carried out in a simple way, through a decision matrix technique. However this is an area that needs deeper exploration on how this evaluation can be modelled to provide support in the evaluation of the quality of a product including the customer feedback as well as the feedback from experts in different areas involved in the development of the product.

6.2.2.3 The role of the PMS in the context of redesign

It has been acknowledged that there is a growing research into tools to support the redesign of a product. The PMS, as defined here, contributes to these efforts by defining a PMS capable of capturing the Design Intent underlying the initial decisions made during the design of a product as well as the decisions related to the changes made during the redesign.

In order to have a more functional PMS, two changes are needed. It is important to make refinements to the structure in order to capture the configuration of more complex products, to add more detail to the

characteristics, i.e. materials, geometrical features, tolerances and requirements. It is also important to make refinements to the Version Class to consider the secondary Design Intent and to include a more formal representation to capture the quality of a product.

The second change needed is in the experimental system to link it to a CAD system as a new option after a change is made. This link would capture not only the information about the configuration of the product and the characteristics of the entities of the product in an automatic way, but as a new option in the menu. The information related to the Design Intent underlying the changes could thus be captured.

6.2.2.4 The relationship between the PMS and the PLM systems

In order to manage the information produced during the development of a product, different approaches have been adopted to capture the rationale and Design Intent in the development of a product as described in section 2.3.3.3. The PMS defined in this research is in line with the PLM systems providing support to manage the information produced in the initial stages and during the development of the product related to the design history of the product.

Although the experimental system was not linked to a commercial CAD system, it was developed such that it considered the changes at a detailed level following the practice of commercial CAD software. In a hypothetical situation where the experimental system would have been implemented in one of these commercial applications, once the change is finished then the designer could capture the Design Intent Driver. This experimental system needs to give the designer the chance to capture the Design Intent Driver when the changes are made.

6.3 CONCLUSIONS

(1) A new information structure for a product model has been defined to support the redesign of a product. Three classes were adapted and one structure was created. These are Product configuration class, Characteristics class, Views class and the Version structure.

(2) The Version Class provides the capacity to capture the Design History of the product. Specifically to capture and be a repository for the information related to the Design Intent.

(3) The Characteristics Class captures the information on five measurable, prominent attributes that describe the product. These characteristics are divided into physical attributes such as material, dimensions, tolerances and geometry and non-physical attributes such as the function. These characteristics reflect the changes made to the entities of the configuration of a product.

(4) The Configuration Class formalises, from a design point of view, the description of the configuration of a product based on the function. This class supports the redesign of a product because an improvement in the functionality is sought in a redesign.

(5) The Views Class provides the capacity to capture the consequences of change. This has been structured against 12 core views that can be used to capture the positive and negative consequences of the changes. These core views are related to different aspects of the product life cycle, the behaviour of the product, the legal and the physical aspects.

(6) An experimental system was developed using UML, the object-oriented database ObjectStore and the Visual C++ programming language. It has been used to successfully demonstrate the capacity of the Product Model

Structure to capture and be a repository for the information related to Design Intent.

6.4 RECOMMENDATIONS AND FURTHER RESEARCH

I. This research has defined a Product Model and has identified how the information structure can support the redesign of products. Implementing a Product Model Structure has also been demonstrated. However, there is a need to explore the applicability of the PMS against a broader range of products, especially those with higher levels of complexity. The products with high levels of complexity can cause problems in the identification of the entities of the product, since systems become increasingly interrelated. Therefore further research is needed to explore how to deal with the interrelation between systems in more complex products.

II. This author considers that the PMS defined here can deal with products that are the result of the evolution of previously designed products. Therefore, in the Design History of a product includes the previously designed products. However, the experimental system was constrained to deal with the Design History of a product with many versions but no changes in the function or working principle that result in the development of new product. Therefore, it is necessary to implement how a product can be the result of previously designed products.

III. This research focused on the use of a Design Intent Driver underlying the changes. However, secondary Design Intents are likely to be significant but were not implemented in the experimental system. Therefore, the application of the secondary Design Intent needs further exploration.

IV. This research defined a configuration class focused on electromechanical products. However, products tend to include more electronic devices in their configurations. Therefore there is the necessity to explore the configuration of a product for these types of products.

V. This research has explored the information structures making emphasis on the information related to Design Intent. The relationship between classes Intent and Views, with subclasses Design, Assembly and Manufacturing, opens the possibility of exploring the Manufacturing Intent and Assembly Intent areas.

VI. The product is a system with target specifications and design specifications; this is the same for the systems and subsystems. A change in the target specifications of a system needs necessarily to be reflected in changes to the characteristics of the entities that comprise the systems or subsystems. In the experimental system developed, the changes to the target requirements of the systems or subsystems were implemented. Therefore, the links between changes to these target specifications and the changes to the entities that comprise the system were not established. Therefore, the changes to the target specifications of the systems and subsystems need to be implemented and linked to the changes to lower entities in the experiment.

PAPERS

PAPERS TO WHICH THIS THESIS HAS CONTRIBUTED

1) **Espinosa, A.** and **B. Young**. 2002, *The Use and Representation of Design* Intent to Support Engineering Designers. in Engineering Design Conference 2002. London: Professional Engineering Publishing.

2) **R.I. Young, A. Espinosa, G. Gunendran and S. Liu**, 2003, Information and knowledge sharing in design decision support, Concurrent Engineering: *Advanced Design, Production and Management* Systems, 147 – 153.

REFERENCES

Al-Salka, M.A., M.P. Cartmell, and S.J. Hardy, 1998, *A Framework for Generalized Computer-based Support Environment for Conceptual Engineering Design.* Journal of Engineering Design 9(1): p. 57-88.

Andersson, F. 2001, *Functional Representation: creating a framework for design support.* in *7th International Conference On Concurrent Engineering.* Lyon, France.

Arnold, A.J. and J.C. Kunz, 2000, *Integrating Product Models with Engineering Analysis Applications: Two Case Studies.* Artificial Intelligence for Engineering Design 14: p. 137 - 147.

Ault, H.K., 1999, *Using Geometric Constraints to Capture Design Intent.* Journal for Geometry and Graphics 3(1): p. 39 - 45.

Autodesk, 2002, *2D Design Productivity*, http://www.autodesk.co.uk/adsk/item/0,,716810-452932,00.html

Bañares-Alcantara, R. and J.M.P. King, 1997, *Design Support Systems For Process Engineering - III. Design Rationale as a Requirement for Effective Support.* Computers Chemical Engineering 21(3): p. 263 - 267.

Barbosa, C.A.M., B. Feijo, M. Dreux, *et al.*, 2003, *Distributed Object Model for Collaborative CAD Environments Based on Design History.* Advances in Engineering Software (34): p. 621 - 631.

Baxter, J.E., N.P. Juster, and A. de Pennington, 1994, *A functional data model for assemblies used to verify product design specifications.* Proc. Instn. Mech. Engrs, Part B, Journal of Engineering Manufacture 208: p. 235-244.

Belhe, J. and A. Kusiak, 1996, *Modelling Relationships Among Design Activities.* Journal of Mechanical Design 118: p. 454 - 460.

Blair, G., G. Coulson, and N. Davies, 1996, *Standards and Platforms for Open Dsitributed Processing.* Electronics and Communication Engineering : p. 123 -- 133.

Booch, G., J. Rumbaugh, and I. Jacobson, *The Unified Modelling Language User Guide.* First ed. Object Technology Series, ed. Addison-Wesley. 1999, Boston: Pearson Education Corporates. 482.

Borja, V., 1997, *Redesign Supported by Data Models with Particular Reference to Reverse Engineering*, in *Department of Manufacturing Engineering*.PhD 1997, Loughborough University: Loughborough

Brazier, F.M.T., H.G.v.L. Pieter, and J. Treur, 1997, *A Compositional Approach to Modelling Design Rationale.* Artificial Intelligence for Engineering Design. 11: p. 125-139.

BS5191, 1975 *Production planning and control terms*, BS 5191,

Busby, J.S., 1999, *The Problem with Design Reuse: An Investigation into Outcomes and Antecedents.* Journal of Engineering Design 10(3): p. 277-296.

Canciglieri, O.J. and R.I.M. Young, 2003, *Information Sharing in Multi Viewpoint Injection Moulding Design and Manufacturing.* International Journal of Production Research 41(7): p. 1565 - 1586.

Carrol, J.M., S.R. Alpert, J. Karat, *et al.*, 94, *Raison d'Etre: Capturing Design History and Rationale in Multimedia Narratives.* Human Factors in Computing Systems 4: p. 192 - 197.

Chen, A., B. McGinnis, and D.G. Ullman, 1990, *Design history knowledge representation and its basic computer implementation.* ASME, Design Engineering Division 27: p. 175 - 184.

Chen, Y.M. and Y.D. Jan, 2000, *Enabling Allied Concurrent Engineering Through Distributed Engineering Information Management.* Robotics and Computer Integrated Manufacturing 16: p. 9 - 27.

Chen, Y.M. and T.H. Tsao, 1998, *A Structure methodology for Implementing Engineering Data Management.* Robotics and Computer Integrated Manufacturing 14: p. 275 - 296.

Chung, J. and K. Lee, 2002, *A framework of Collaborative Design Environment for Injection Moulding.* Computers in Industry 47: p. 319 - 337.

Costa, C.A., 2000, *Product Range Models in Injection Mould Tool Design*, in *Manufacturing Engineering*.PhD 2000, Loughborough University: Loughborough. p. 266

Court, A.W., S.J. Culley, and C.A. McMahon. 1995, *Modeling the Information Access Methods of Engineering Designer.* in *Design Engineering Technical Conferences-DE, ASME.*

Culley, S.J. 1998, *Classification Approaches for Standard Parts to Aid Design Reuse*: IMechE.

Deneux, D., 1999, *Introduction to Assembly Features: an Illustrated Synthesis Methodology.* Journal of Intelligent Manufacturing 10: p. 29-39.

Dhar, S. and R. Rangani. 2003, *Product Specification Patterns for Early Stage Development in the Automotive Industry.* in *International Symposium on Product Lifecycle Management.* Bangalore, Karnataka, India.

Dieter, G.E., *Engineering Design.* 3rd edition ed. Mechanical Engineering, ed. M.H.I. Editions. 2000: McGraw Hill. 798.

Dixon, L.A. and J.S. Colton, 1998, *An Anchoring Adjustment Process Model for Redesign.* Journal of Engineering Design 9(4): p. 297-314.

Dixon, L.A. and J.S. Colton, 2000, *A Process Management Strategy for Redesign: An Anchoring Adjustment Approach.* Journal in Engineering Design 11(2): p. 159-173.

Dorador-Gonzalez, J.M., 2001, *Product and Process Information Interactions in Assembly Decision Support System*, in *Manufacturing Engineering.*PhD Thesis 2001, Loughborough University: Loughborough

Dorador-Gonzalez, J.M. and I.M. Young. 1999, *Information Models to Support the Interaction Between Design for Assembly and Assembly Process Planning.* in *International Symposium on Assembly and Task Planning.* Porto, Portugal.

Dorador-Gonzalez, J.M. and R.I.M. Young, 2000, *Application of IDEF0, IDEF and UML Methodologies in the Creation of Information Models.* Computer Integrated Manufacturing 13(5): p. 430-445.

Duffy, A.H.B., 1996, *Learning for Design Reuse.* Artificial Intelligence for Engineering Design, Analysis and Manufacture 10: p. 139-142.

Duffy, A.H.B., D.C. Brown, and M.L. Maher, 1996, *Special Issue: Machine learning in design.* Artificial Intelligence for Engineering Design 10: p. 81-82.

Duffy, S.M. and A.H.B. Duffy, 1996, *Sharing the Learning Activity using Intelligent CAD.* Artificial Intelligence for Engineering Design 10: p. 83-100.

Dyson, 2003, *Dyson homepage*, http://www.dyson.co.uk

Ellis, T.I.A., 1993*Glossary of Terms MOSES Project*, . Loughborough University, University of Leeds: Leeds.

Ellis, T.I.A., A. Molina, R.I.M. Young, *et al.* 1994, *The Development of an Information Sharing Platform for Concurrent Engineering.* in *International Manufacturing Systems Engineering Workshop.* Grenoble.

Encyclopedia, 2004, *TheFreeDictionary.com*, http://encyclopedia.thefreedictionary.com/Engineering

Engineering, I.C.o.S., 1998 *INCOSE SE Terms Glossary*,

Espinosa, A. and B. Young. 2002, *The Use and Representation of Design Intent to Support Engineering Designers.* in *Engineering Design Conference 2002*. London: Professional Engineering Publishing.

Fenves, S.J., R.D. Sriram, R. Sudarsan, *et al.* 2003, *A product Information Modelling Framework for Product Lifecycle Management.* in *International Symposium on Product Lifecycle Management.* Bangalore Karnataka, India.

Fisher, G., K. Nakakoji, and J. Otswald. 1995, *Supporting the Evolution of Design Artifacts With Representations of Context And Intent.*

Fowler, J.E., 1996, *Variant Design for Mechanical Artifacts: A State-of-the-Art Survey.* Engineering with Computers 12: p. 1-15.

French, M.J., *Conceptual Design for Engineers.* First ed. 1971, London: Springer Verlag.

Ganeshan, R., J. Garret, and S. Finger, 1994, *A Framework for Representing Design Intent.* Design Studies 15(1): p. 59 - 84.

Gao, J.X., P.G. Maropoulos, and W.M. Cheung, 2003, *Application of Product Data Mangaement Technologies for Enterprise Integration.* International Journal of Computer Integrated Manufacturing 16(7 - 8): p. 491 - 500.

Gaughran, W.F. 2000, *Modelling and Design Intent.* in *Engineering and Product Design Education; Integrating Design Education Beyond 2000.* Brighton: Bury St. Edmunds; Professional Engineering Publishing Ltd.;2000.

Genesereth, M.R. and S.P. Ketchpel, 1994, *Software Agents.* Communication of the ACM 37(7): p. 48 - 53.

Goodwin, R. and P.W.H. Chung, 1997, *An Integrated Framework for Representing Design History.* Applied Intelligence 7: p. 167-181.

Gorti, S.R., A. Gupta, G.J. Kim, *et al.*, 1998, *An Object-oriented Representation for Product and Design Process.* Computer-Aided Design 30(7): p. 489-501.

Guerra-Zubiaga, D.A., 2004, *A Manufacturing Model to Enable Knowledge Maintenance in Decision Support Systems*, in *Wofson School of Engineering*.PhD 2004, Loughborough University: Loughborough. p. 228

Gunendran, A.G. and R.I.M. Young. 2002, *A Framework for Interoperability in Team Based Software*. in *Proceedings of the 9th International Conference on Concurrent Engineering*. Cranfield, U.K.

Harding, J.A., 1996, *A Knowledge Representation Model to Support Concurrent Engineering Team Working*, in *Department of Manufacturing Engineering*.Ph.D. Thesis 1996, Loughborough University: Loughborough. p. 160

Harding, J.A. and K. Popplewell, 1996, *Driving Concurrency in a Distributed Concurrent Engineering Project Team: a Specification for an Engineering Moderator.* International Journal of Production Research 34(3): p. 841-861.

Harrington, J.V., Soltan Forskitt, M., 1996, *Framework for knowledge based support in a concurrent engineering environment.* Knowledge based Systems 9: p. 207-215.

Hella, 2003*Technical information, Lighting headlamps*, . Hella.

Henderson, M.R. 1993, *Representing Functionality and Design intent in Product Models*. in *Proceedings of the 2nd Symposium on Solid Modeling and Applications*. Montreal, Quebec, Canada: ACM.

Horvath, L. and I.J. Rudas, 1999, *Human Intent Description as a Tool for Communication Between Engineers.* IEEE : p. 348 - 353.

Horváth, L. and I.J. Rudas, 1998, *Description of Design Intent in Product Models.* IEEE : p. 1312 - 1316.

Horváth, L., I.J. Rudas, and J.F. Bitó, 1999, *Modelling Design Intent in Concurrent Engineering.* IECON - PROCEEDINGS 2: p. 968 - 972.

Hsu, H.-Y. and G.C.I. Lin, 1998, *A Design-for-assembly-based Product Redesign Approach.* Journal of Engineering Design 9(2): p. 171-195.

Hubka, V., *Principles of Engineering Design*. First ed. 1980: Butterworth & Co Ltd. 118.

IDEF0, 1993 ***INTEGRATION DEFINITION FOR FUNCTION MODELING (IDEF0)***,

INCOSE, I.E.H., 2000 *Systems Engineering Handbook*,

Ishino, Y. and Y. Jin, 2002, *Estimate design intent: a multiple genetic programming and multivariate analysis based approach.* Advanced Engineering Informatics 16: p. 107 - 125.

ISO10303-1, 1994 *Industrial automation systems and integration -Product Data Representation and Exchange-*, BS ISO 10303-1,

ISO10303-203, 2000 *International Standard ISO 10303-203 AMENDMENT 1*, ISO 10303-203: 2000 Amd.1,

ISO16100-1, 2001 ***Industrial automation systems and integration — Manufacturing software capability profiling —***, ISO 16100 - 1,

Jacobson, I., G. Booch, and J. Rumbaugh, *The Unified Software Development Process*. 1999, Massachussets: Addison Wesley Longman, Inc.

Jeon, D.K., S.D. Urban, and J.J. Shah, 1994*Process Modeling Aspects of a Design History Data Model*, in *Software Systems Engineering*, ASME, Editor. p. 209 - 218.

Jo, H.H., H.R. Parsaei, and W.G. Sullivan, *Principles of concurrent engineering*, in *Concurrent Engineering: Contemporary issues and modern design tools*, H.R. Parsaei and W.G. Sullivan, Editors. 1993, Chapman & Hall: Cambridge. p. 3-23.

Johanneson, H.L., 1993, *Computer Aided Product Structure Modeling Using a DBMS*. Advances in Design Automation 1: p. 511 - 518.

Kasprzak, K., 1998, *Parametrics that don't lose design intent (engineering design)*. Machine Design .

Kimura, F., 1992, *Product and Process Modelling as a Kernel for Virtual Manufacturing Environment*. Annals of the CIRP 42(1): p. 147-150.

KMI, 2000, *Knowledge Media Institute*, http://kmi.open.ac.uk/home-f.cfm

Kovse, J. 2002, *Specifying Customized Versioning Facilities of Software Engineering Repositories by Using UML-based Design Templates*. in *GI - Workshop Grundlagen von Datenbanken*. Fischland/Darß.

Krause, F.L., F. Kimura, T. Kjelberg, *et al.*, 1993, *Product Modelling*. Annals of the CIRP 42(2): p. 695-705.

Lee, K.-S. and K. Lee, 2001, *Framework of an Evolutionary Design System Incorporating Design Information and History*. Computers in Industry 44: p. 205-227.

Lees, R.-S., Y.-M. Chen, and C.-Z. Lee. 1997, *Engineering Design Support Through Case-Based Reasoning*. in *IEE Coloquium*. London.

Lenau, T. and L. Mu, 1993, *Features in integrated modelling of products and their production*. International Journal of Computer Integrated Manufacturing 6(1 & 2): p. 65 - 73.

Liao, D.-Y., 2001*Intelligent Support Systems*, . Department of Information Management, National Chi-Nan University: Puli, Taiwan.

Liu, D.T. and W. Xu, 2001, *A Review of Web-Based Product Data Mangement Systems.* Computers in Industry 44: p. 251 - 262.

Liu, S., 2004, *Manufacturing Information and Knowledge Models to Support Global Manufacturing Co-ordination*, in *Wolfson School of Engineering*.PhD 2004, Loughborough University: Loughborough. p. 227

Maher, M.L. and A.G.d.S. Garza, 1997, *Case-Based Reasoning in Design.* IEEE Expert : p. 34 - 41.

Maher, M.L. and P. Pu, *Introduction to Issues and Applications of Case-Based Reasoning in Design.* Issues and Applications of Case-Based Reasoning in Design. 1997, New Jersey: Lawrence Erlbaum Associates.

Malmqvist, J., 1995, *A Computer-based Approach Towards Including Design History Information in Product Models and Function-Means Trees.* Proceedings of DIM-95 : p. 593 - 602.

Malmqvist, J., 1997, *Improved Function-means Trees by Inclusion of Design History Information.* Journal of Engineering Design 8(2): p. 107-117.

McKay, A., 1993, 1994*Data Model for the Description of Product Specifications*, . University of Leeds: Leeds. p. 2 - 8.

Mckay, A. and M.S. Bloor. 1991, *The role of product models in effective CADCAM.* in *European Conference.* Coventry.

McKay, A., M.S. Bloor, and A. De Pennington, 1996, *A Framework for Product Data.* IEEE Transactions on Knowledge and Data Engineering 8(5): p. 825-837.

McKay, A., M.S. Bloor, and A. De Pennington. 1997, *Realising the Potential of Product Data Engineering.* in *5th International Conference on FACTORY 2000.* Cambridge, UK: IEE.

Molina, A., T.I.A. Ellis, R.I.M. Young, *et al.*, 1995, *Modelling Manufacturing Capability to Support Concurrent Engineering.* Concurrent Engineering: Research and Applications 3(1): p. 29-42.

Mostow, 1985*Toward Better Models of the Design Process*, in *Artificial Intelligence.* p. 44-57.

Myers, K.L., N.B. Zumel, and P. Garcia, 2000, *Acquiring Design Rationale Automatically.* Artificial Intelligence for Engineering Design 14: p. 115-135.

Nagy, R.L., D.G. Ullman, and T.G. Dietterich, 1991, *A Knowledge Base Data Representation for Collaborative Mechanical Design.* Design Theory and Methodology 31: p. 69 - 76.

Nielsen, E.H., J.R. Dixon, and G.E. Zinsmeister, 1991, *Capturing and using Designer intent in a design-with-features system.* Design Theory and Methodology 31: p. 95 - 102.

NOM, S.d.S., 1993 ***ESPECIFICACIONES SANITARIAS DE LAS SUTURAS QUIRURGICAS.***, NOM-067-SSA1-1993,

Oh, Y., S.-h. Han, and H. Suh, 2001, *Mapping Product Structures Between CAD and PDM System Using UML.* Computer Aided Design 33: p. 521-529.

Otto, K.N. and K.L. Wood, 1998, *Product Evolution: A Reverse Engineering and Redesign Methodology.* Research in Engineering Design 10: p. 226-243.

Pahl, G. and W. Beitz, *Engineering Design: A systematic approach.* Second edition ed. 2001, London: Springer - Verlag. 544.

PLM03, 2003, *International Symposium on Product Lifecycle Mangement*, http://web.mecheng.iisc.ernet.in/~plm03/

Prasad, B., 1996, *Toward a Functional Design of a Concurrent Information Modelling System.* Electronic Data Systems .

Pugh, S., *Total design: Integrated Methods for Succesful Product Engineering.* 1991: Prentice Hall.

Quatrani, T., *Visual Modeling with Rational Rose 2000 and UML.* 2 ed, ed. G. Booch, I. Jacobson, and J. Rumbaugh. 1999, Massachusetts: Addison-Wesley. 256.

Rational, 2001, *Carnegie Mellon Software Engineering Institute,*, http://www.sei.cmu.edu/str/argument.html

Rezayat, M., 1999, *Knowledge-Based product development using XML and KCs.* Computer Aided Design 32(2000): p. 299-309.

Rosenman, M.A. and J.S. Gero, 1999, *Purpose and Function in a Collaborative CAD Environment.* Reliability Enginnering and Safety System 64: p. 167 - 179.

Rowe, J., 1999, *Knowledge-Driven Cad.* Computer Graphics World .

Salomons, O.W., 1995, *Design History in FROOM.* International Journal of CADCAM and Computer Graphics 10(1 - 2): p. 95 - 111.

Salomons, O.W., F.v. Slooten, F.J.A.M.v. Houten, *et al.* 1993, *A Computer Support Tool for Re-design.* in *International Conference on Engineering Design.* The Hague.

Schachinger, P. and H.L. Johannesson, 2000, *Computer Modelling of Design Specifications.* Journal of Engineering Design 11(4): p. 317-329.

Shah, J.J., D.K. Jeon, S.D. Urban, *et al.*, 1995, *Database Infrastructure for Supporting Engineering Design Histories.* Computer-Aided Design 28(5): p. 347 - 360.

Sharma, R. and J.X. Gao, 2002, *A Progressive DEsign and Manufacturing Evaluation System Incorporating STEP AP224.* Computers in Industry 47: p. 155 - 167.

Shih, C.-H., 1997, *A Design/Constrain Model to Capture Design Intent.* Solid Modeling : p. 255 - 263.

Shum, S., A. MacLean, J. Forder, *et al.* 1993, *Summarising the Evolution of Design Concepts Within a Design Rationale Framework.* in *Conference on Human Factors in Computing Systems.* Amsterdam.

Shyamasunder, P. and P.S. Rao. 2003, *Knowledge Based Engineering - an elegant approach for successful PLM implementation.* in *International Symposium on Product Lifecycle Mangement.* Bangalore, Karmataka, India.

Silva, J. and K.-H. Chang, 2002, *Design Parametrization for Concurrent Design and Manufacturing of Mechanical Systems.* CONCURRENT ENGINEERING: Research and Applications 10(1Q): p. 3 - 14.

Sim, S.K. and A. Duffy. 1994, *A New Perspective to Design Intent and Design Rationale.* in *Artificial Intelligence in Design Workshop.*

Sim, S.K. and A.H.B. Duffy, 1998, *A Foundation for Machine Learning in Design.* Artificial for Engineering Design, Analysis and Manufacturing 12: p. 193-209.

Sivaloganathan, S. and T.M.M. Shahin, 1999, *Design reuse: on overview.* Proc. Instn. Mech. Engrs, Part B, Journal of Engineering Manufacture 213: p. 641-654.

Stevenson, D.A., A.H.B. Duffy, and S. Lim. 1999, *Supporting Design Intent in Sketching Activities.* in *International Conference on Engineering Design.* Munich, Gernany: ICED 99.

Szykman, S., S.J. Fenves, W. Keirouz, *et al.*, 2001, *A Foundation for Interoperability in Next-generarion Product Development Systems.* Computer Aided Design 33: p. 545 - 559.

Taura, T. and A. Kubota, 1999, *A Study on Engineering History Base.* Research in Engineering Design 11: p. 45-54.

Texel, P.P. and C.B. Williams, *Use Cases Combined with BOOCH / OMT / UML: Process and products.* 1997: Prentice Hall, Inc.

Tichem, M. and T. Storm, 1997, *Designer support for product structuring-development of a DFX tool within the design co-ordination framework.* Computer in Industry 33: p. 155-163.

Ullman, D., 1994, *Issues Critical to the Development of Design History, Design Rationale and Design Intent Systems.* Design Theory and Methodology - DTM ASME 68.

Ullman, D., *The Mechanical Design Process.* 2nd ed. 1997, London: The McGraw Hill Companies, Inc. 336.

Ullman, D., 2002, *Toward the Ideal Mechanical Engineering Design Support System.* Research in Engineering Design 13: p. 55 - 64.

Ullman, D.G. 1993, *The Evolution of Function and Behaviour During mechanical Design.* in *Design Theory and Methodology*: ASME.

UML, 2004, *Introduction to UML - OMG*, http://www.omg.org/

Ungerer, M. and K. Buchanan, 2002*Usage Guide for the STEP PDM Scheme V1.2*, : Darmstadt, Germany.

Watson, I. and F. Marir, 1994, *Case-based reasoning: A review.* The Knowledge Engineering Review 9(4): p. 327 - 354.

Werner, H. and S. Ahmed. 1999, *Design Capturing with a Model System Using Event Triggered Procedures.* in *Internationall Conference on Engineering Design* 99. Munich, Germany: WDK.

Williams, J., 1990, *When Expert Systems are Wrong.* ACM : p. 661 - 669.

Wilson, P.R. 1998, *STEP and EXPRESS.* in *Distributed Information, Computation and Process Management for Scientific and Engineering Environments.* Herndon, Virginia.

Wooldridge, M. and N. Jennings, 1995, *Intelligent Agents: Theory and Practice.* nowledge Engineering Review 10: p. 115 - 152.

Yakemovic, K. and J. Conklin, 1989*The capture of Design Rationale on an Industrial Development Project*, .

Young, R.I.M., 2003, *Informing Decision-Makers in Product Design and Manufacture.* International Journal of Computer Integrated Manufacturing 16(6): p. 428 - 438.

Young, R.I.M., O. Canciglieri-Jnr, and C.A. Costa, *Information Interactions in Data Model Driven Design for Manufacture*, in *Globalization of Manufacturing in the Digital Communications Era of the 21st Century: Innovation, Agility,and the*

Virtual Enterprise, G. Jacucci, G.J. Olling, K. Preiss, *et al.*, Editors. 1998, Kluwer Academic Publishers: Trento. p. 313-324.

Young, R.I.M., O. Canciglieri-Jnr, C.A. Costa, *et al.* 2000, *Information Support in an Integrated Product Development System.* in *3rd International Conference on Integrated Design and Manufacturing in Mechanical Engineering - IDMME'2000.* Montreal: Presses Internationales Polytechnique.

Young, R.I.M., J.M. Dorador, J. Zhao, *et al.* 2001, *A Shared Information/ Knowledge Environment to Support Concurrent Engineering.* in *International Conference on Concurrent Engineering Research and Applications, "Advances in Concurrent Engineering".* California, USA.

APPENDICES

APPENDIX A. UNIFIED MODELLING LANGUAGE

The Unified Modelling Language (UML) is the result of the merging of three modelling methods: Booch, Object Modelling Technique (OMT) and Objectory. UML is used to model the design of software applications. This modelling is important to develop software as important are the blueprints for a building. The UML is used for specifying, visualizing and documenting the artefacts of software systems (UML, 2004).

UML can be used for database professionals. UML provides the design databases to share a common language and establish communication between the database team.

For different levels of abstraction different diagrams are used, the diagrams used by UML are: Class diagram. Object diagram, Use case diagram, Sequence diagram, Collaboration diagram, Statechart diagram, activity diagram, Component diagram and Deployment diagram. However for this research two diagrams were applied these are the Use Case Diagrams and the Class Diagrams.

Use Case Diagrams

The use case diagram is comprised by a set of actors, use cases and the

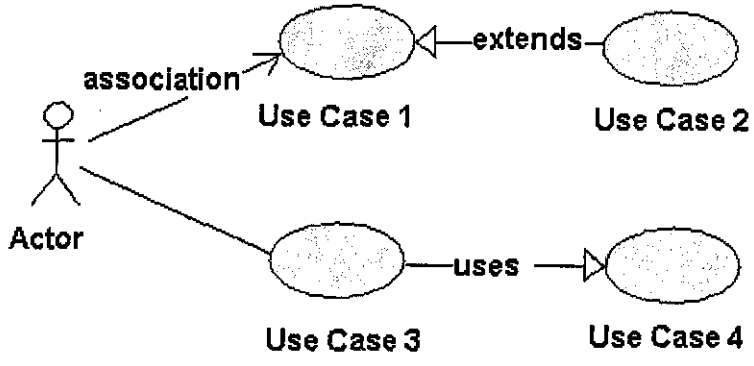

Figure A1 Basic elements of a Use Case Diagram

relationships between them. This diagram helps to organise and model the

behaviour of the system. The use cases represent a special kind of class and represent what the system does. The actors are the users of the system. Therefore a dialog can be modelled between the users and the system. Figure A1 shows the basic elements of a use case diagram

Class Diagrams

A class diagram is a group of classes, interfaces and collaborations and their relationships (Booch, Rumbaugh and Jacobson, 1999). This is the most common type of diagrams used in object-oriented systems. As described by the author a class describes a group of objects with common properties, common relationships and common semantics.

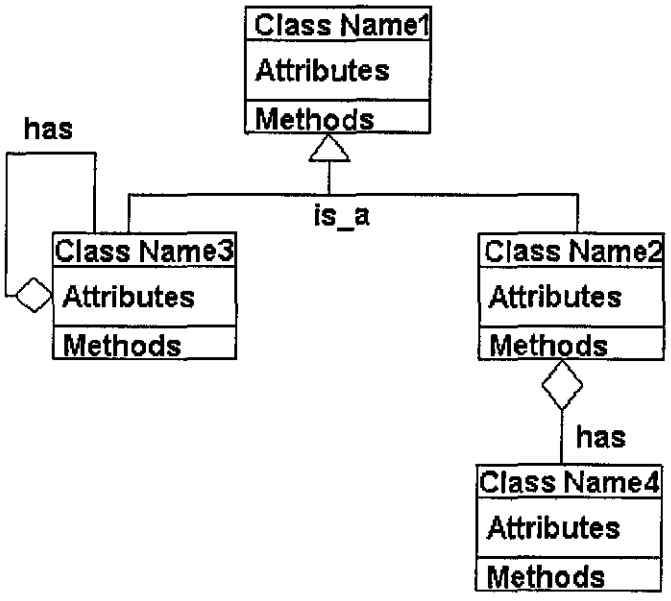

Figure A2 Basic Class Diagram

Figure A2 Shows a basic diagram with four classes with its relationships. Any object can be described in terms of its attributes, such as colour, name age, address. In the same way any object has methods that describe the behaviour of the object. Therefore a class has attributes and methods, as any real object.

The classes have relationships among them. Some of these are shown in Figure A1. A *is_a* relationship relates classes that have the same attributes. and methods. Therefore a parent class inherits to his children class through this relationship. For example a car is a way of transportation.

A different type of relationship is the *has* relationship. With this relation the diagram represents the idea that a class *has* another class. For example a car has accessories. In the Figure A1 Class Name3 shows a *has relationship to itself* representing a self-containing possibility. This provided this research with the possibility to represent the fact that a product has products.

UML is an important tool that provides the possibility to model complex situations, from business to engineering situations. This helps to model software systems before coding (UML, 2004).

APPENDIX B. MEXICAN STANDARD ON SURGICAL NEEDLES

NORMA OFICIAL MEXICANA NOM-067-SSA1-1993, QUE ESTABLECE LAS ESPECIFICACIONES SANITARIAS DE LAS SUTURAS QUIRURGICAS.

Al margen un sello con el Escudo Nacional, que dice: Estados Unidos Mexicanos.- Secretaría de Salud.

CARLOS R. PACHECO, Director General de Control de Insumos para la Salud, por acuerdo del Comité Consultivo Nacional de Normalización de Regulación y Fomento Sanitario, con fundamento en los artículos 39 de la Ley Orgánica de la Administración Pública Federal; 38 fracción II, 45, 46 fracción II, 47 de la Ley Federal sobre Metrología y Normalización; 8o. fracción IV y 12 fracción II del Reglamento Interior de la Secretaría de Salud.

INDICE

PREFACIO

1 OBJETIVO
2 CAMPO DE APLICACIÓN
3 REFERENCIAS
4 DEFINICIONES
5 NOMENCLATURA DE MATERIALES METALICOS
6 CLASIFICACION
7 ESPECIFICACIONES
8 ANALISIS DE LABORATORIO
9 ALMACENAMIENTO
10 CONCORDANCIA CON NORMAS INTERNACIONALES
11 BIBLIOGRAFIA
12 OBSERVANCIA DE ESTA NORMA
13 VIGENCIA

PREFACIO

Participaron en la elaboración de esta Norma: Dirección General de Control de Insumos para la Salud (SSA), Coordinación General de Hospitales en el D.F. (SSA), Subdirección General Médica y Subdirección Técnica (ISSSTE), Instituto Mexicano del Seguro Social, Cámara Nacional de la Industria Farmacéutica (CANIFARMA): Sección de Productos Auxiliares para la Salud, Asociación del Acero Inoxidable, A.C. (ADAI), y las siguientes empresas: Johnson & Johnson de México, S.A. de C.V., Mactur, S.A. de C.V., Internacional Farmacéutica, S.A. de C.V. y Serral, S.A. de C.V.

1. Objetivo

Esta Norma Oficial Mexicana especifica las características que deben llenar los materiales de curación conocidos bajo el nombre de suturas quirúrgicas (incluye proceso) y es de observancia obligatoria en el territorio nacional.

2. Campo de aplicación

Se utilizan en las áreas de cirugía y en las áreas generales de las Unidades Médicas.

3. Referencias

3.1 Farmacopea de los Estados Unidos Mexicanos 5ta. Edición Suplemento No. 1, Secretaría de Salud. México, D.F.

7. Especificaciones

7.1 Del producto.

De la aguja

Acabado	Libre de rebabas, puntas romas o deformes, fisuras fracturas, marcas de esmerilado, rayaduras, áreas rugosas, muescas, corrosión a simple vista, poros y deformaciones, debe tener un pulido final a espejo.	8.2.1
Longitud	Debe cumplir especificación.	8.2.2
Características (forma y dimensiones).	La forma debe cumplir con lo indicado para cada clave. Las dimensiones deben coincidir con el dibujo que viene en el empaque primario o secundario.	8.2.14
Dureza *	45 Rockwell-C a 55 Rockwell-C	8.2.15
Resistencia a la corrosión para aceros* :		
Austeníticos *	A simple vista no deben tener indicios de corrosión.	8.2.16
Martensíticos *	No deben tener depósitos de Cobre.	8.2.17
Composición química*.	Deben cumplir especificación.	8.2.18

8.2.1 Acabado de la sutura y la aguja.

Procedimiento.

inspeccionar visualmente la sutura y la aguja.

Interpretación.

Sutura.

Libre de nódulos, roturas, material extraño, piezas desensambladas, colores diferentes, porciones planas y deshilachamiento (separación de capas). Debe tener color homogéneo. Si la sutura es envasada en líquido, realizar las pruebas, después de 2 minutos de haber sacado la sutura del líquido.

Aguja.

Libre de rebabas, puntas romas o deformes, fisuras, fracturas, marcas de esmerilado, rayaduras, áreas rugosas, muescas, corrosión a simple vista, poros y deformaciones. Debe tener un pulido final a espejo.

8.2.2 Longitud de la sutura y aguja.

Procedimiento.

Sutura.

Medir la longitud de la sutura utilizando una regla metálica de 50 cm o 100 cm.

Aguja.

10. Concordancia con normas internacionales

Esta Norma no concuerda con ninguna Norma Internacional.

11. Bibliografía

11.1 Farmacopea de los Estados Unidos Mexicanos, 5a. edición, Secretaría de Salud, México, D.F.

11.2 The United States Pharmacopea XXII Ed. National Formulary 17th Ed. Mack Publishing Co. Easton, Pennsylvania, 1990 pp 1308, 1614, 1615.

11.3 Metals handbook, properties and selection: stainless steels, tool materials and special-purpose metals, ninth edition, American Society for Metals, vol. 3, USA, 1980, p. 10.

11.4 Ley General de Salud, título décimo segundo, capítulo I.

11.5 Reglamento de la Ley General de Salud en Materia de Control Sanitario de Actividades, Establecimientos, Productos y Servicios, título vigésimo cuarto. Envasado de los Productos.

11.6 Suplemento No.1 de la Farmacopea de los Estados Unidos Mexicanos, 5a. edición, Secretaría de Salud, México, D.F.

13. Vigencia

La presente Norma entrará en vigor con su carácter de obligatorio a partir del día siguiente de su publicación en el Diario Oficial de la Federación.

Sufragio Efectivo. No Reelección.

México, D.F., a 5 de septiembre de 1994.- El Director General de Control de Insumos para la Salud, Carlos R. Pacheco.- Rúbrica.

Fecha de publicación: 25 de mayo de 1995

APPENDIX C. INITIAL CAPTURE OF INFORMATION OF THE PRODUCT

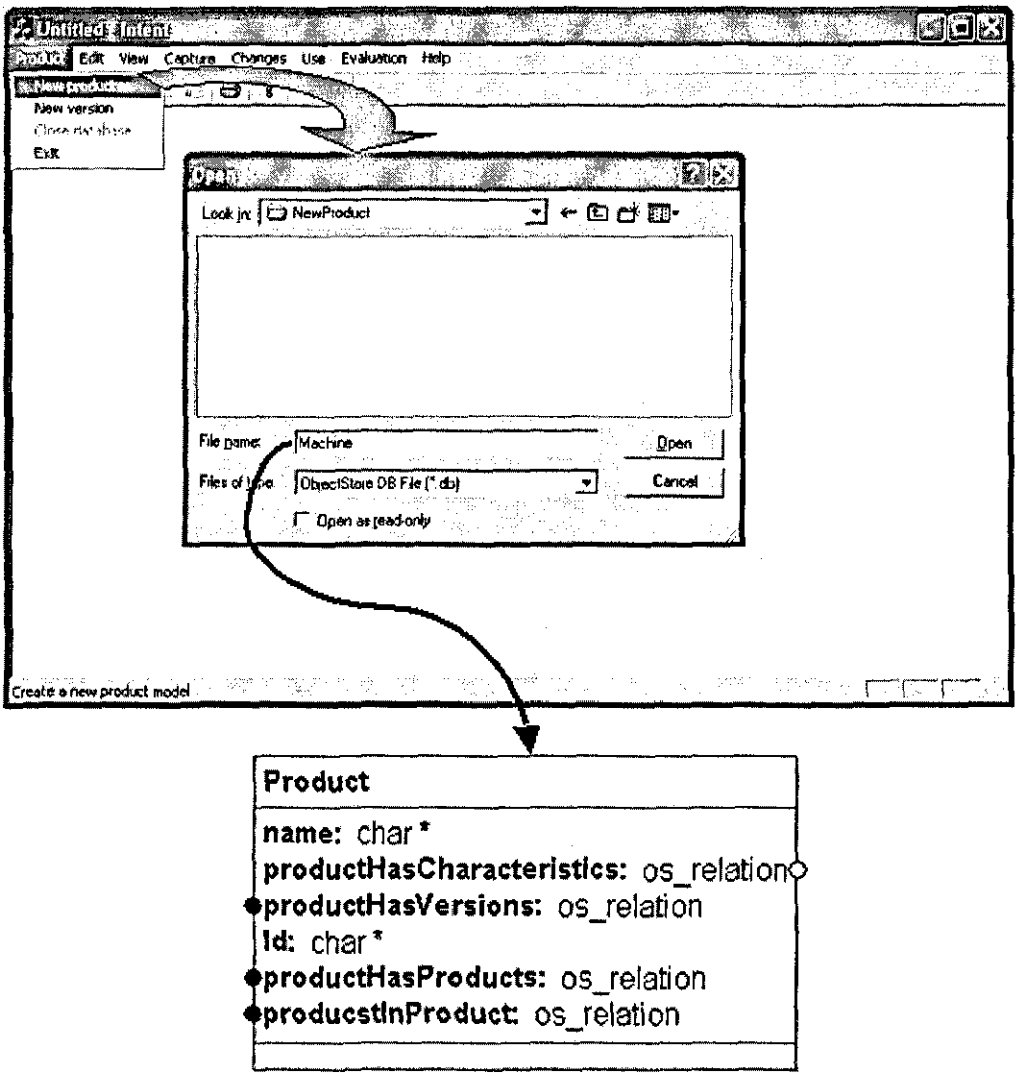

Figure C.1 Creation of the database

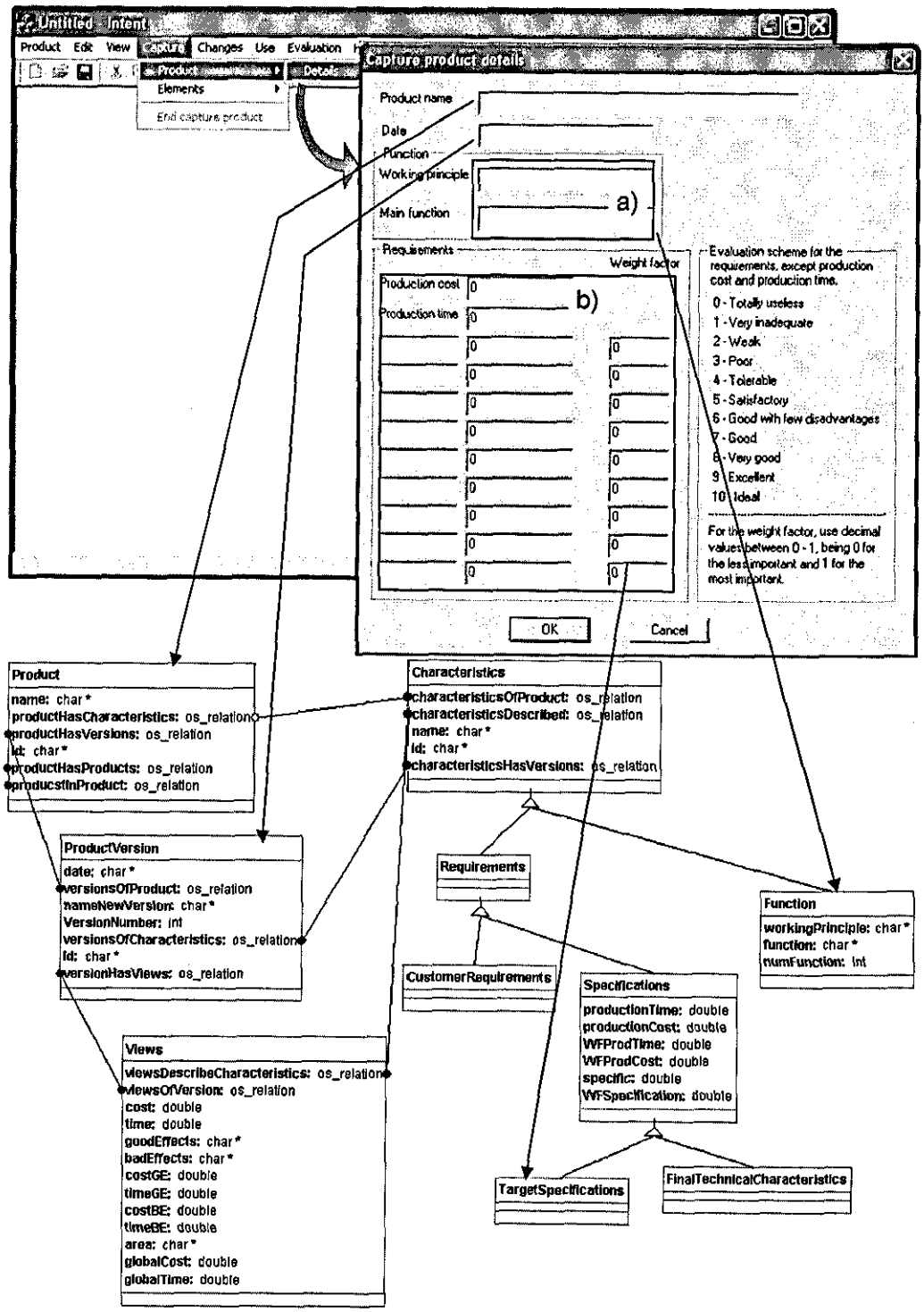

Figure C.2 Capture of the target specifications of the product

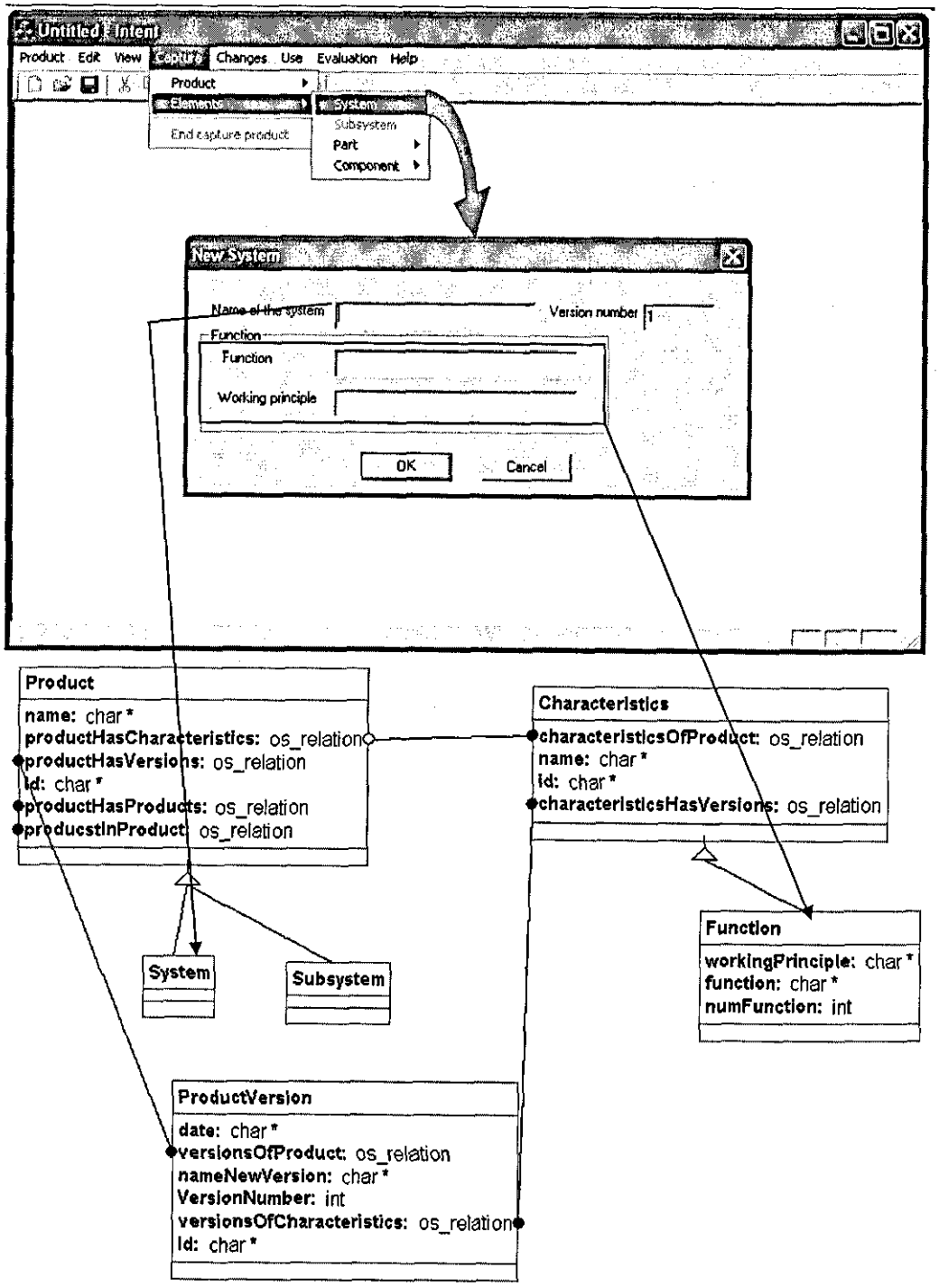

Figure C.3 Capture of the information of the system

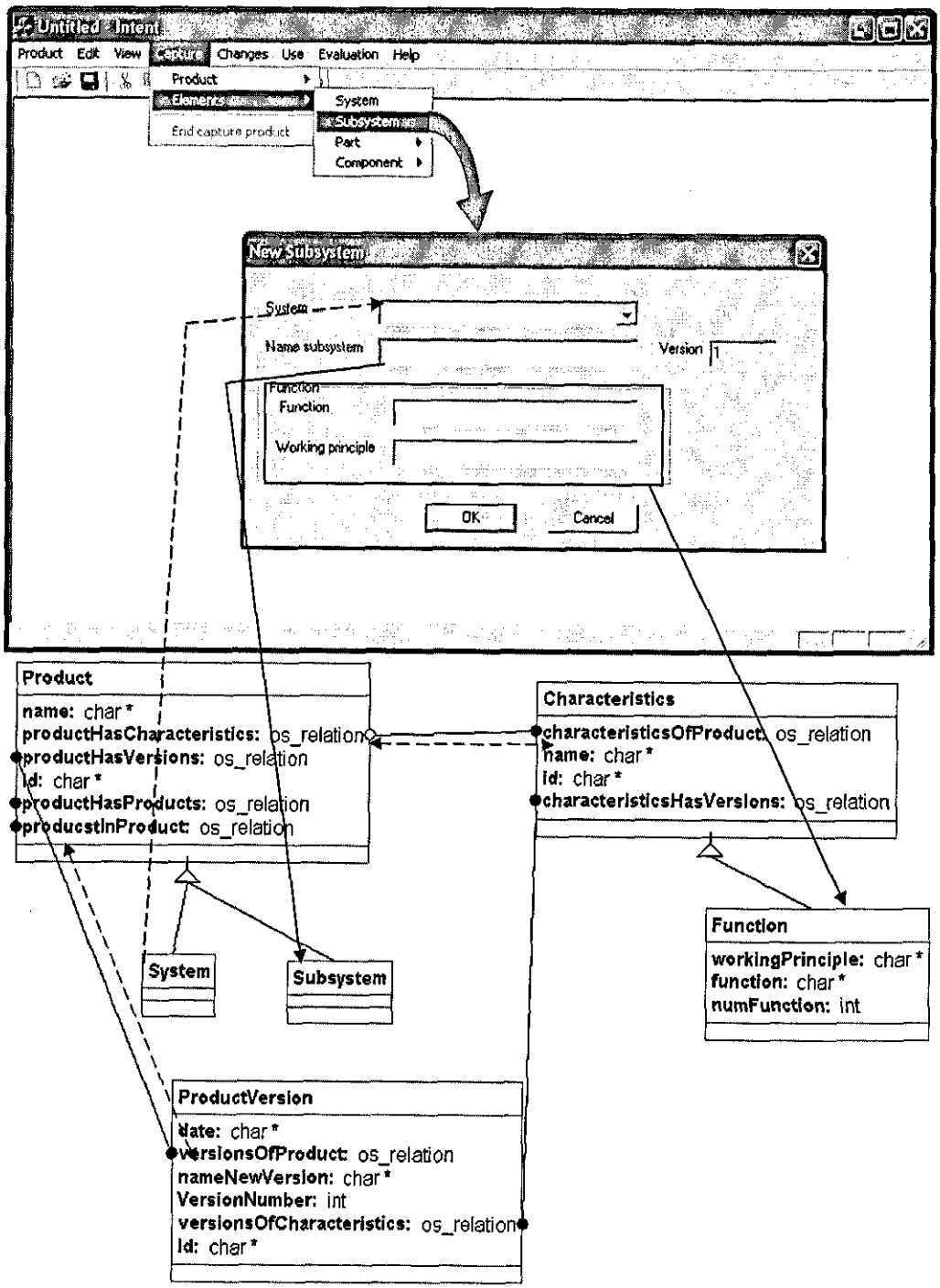

Figure C.4 Capture of the information of the subsystem

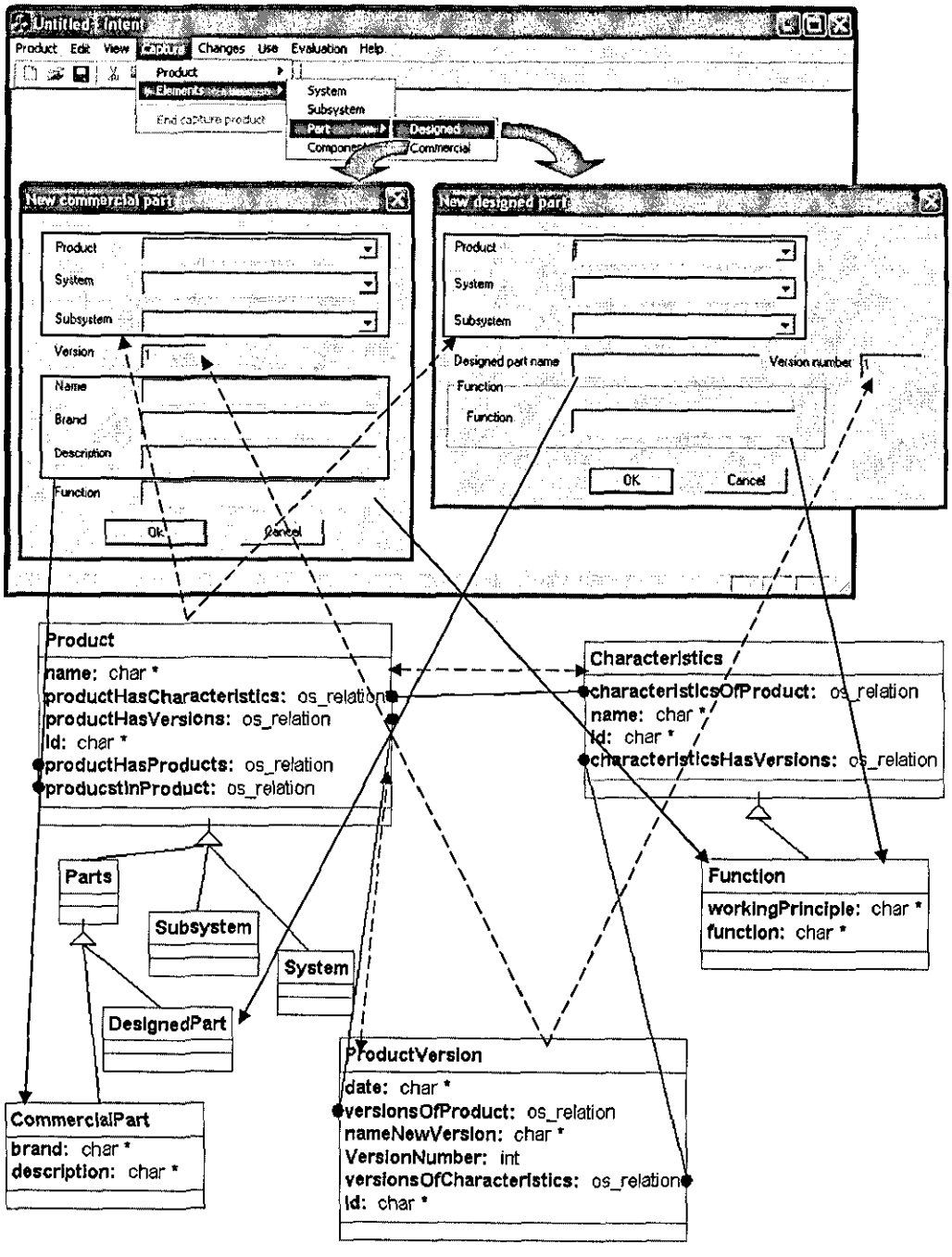

Figure C.5 Capture of the information of the parts

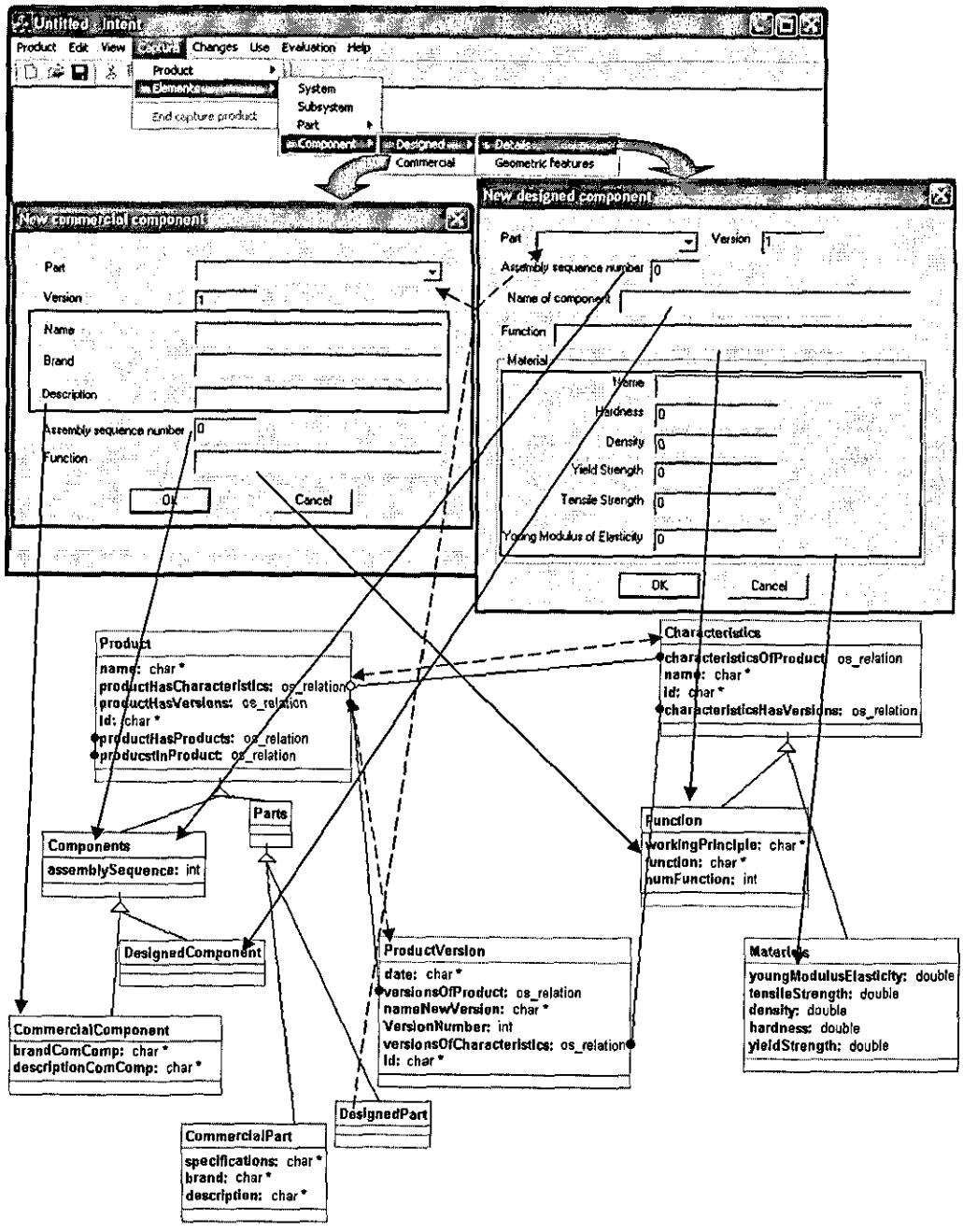

Figure C.6 Capture of the information of the components

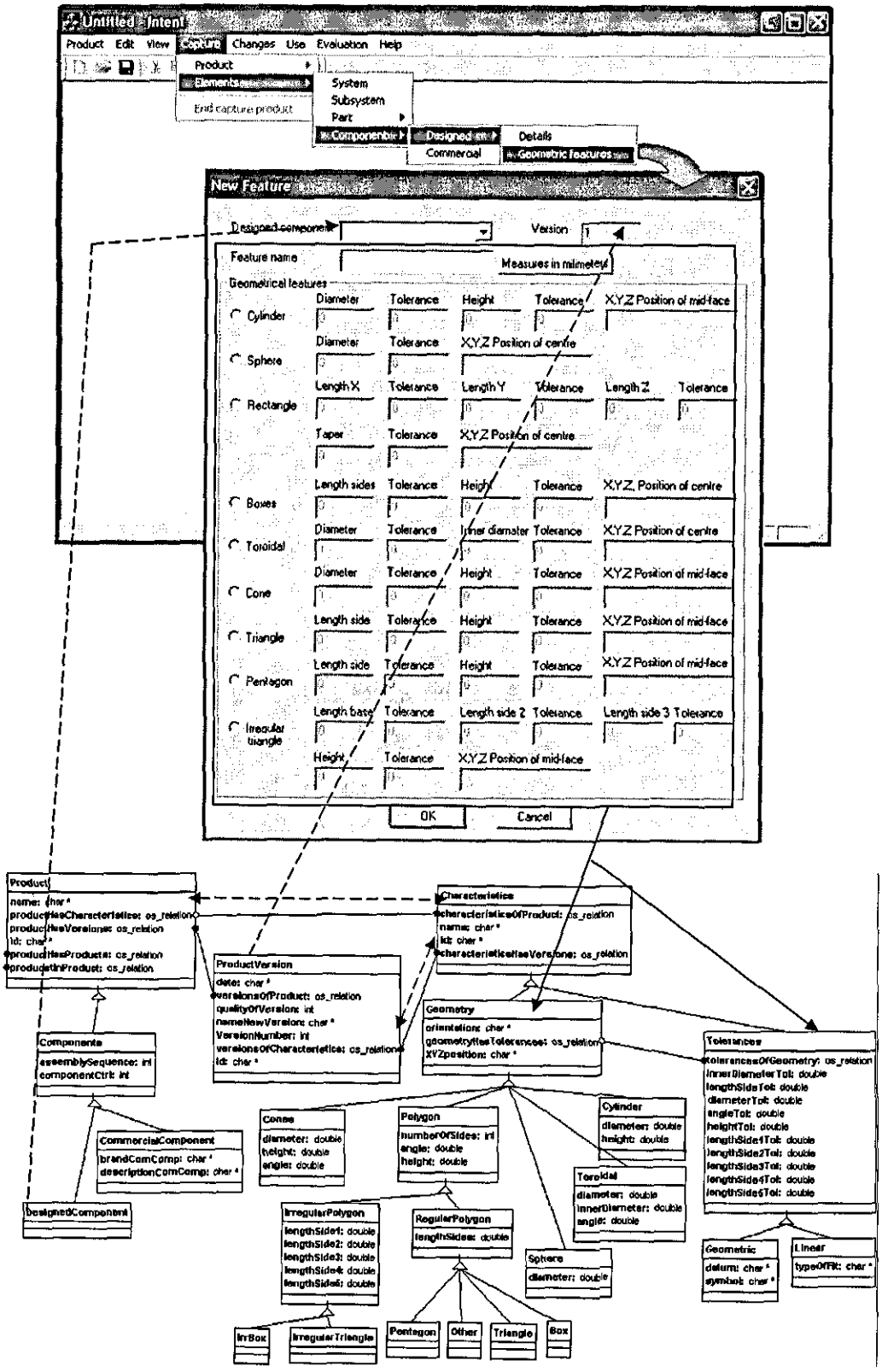

Figure C.7 Capture of details of geometric features

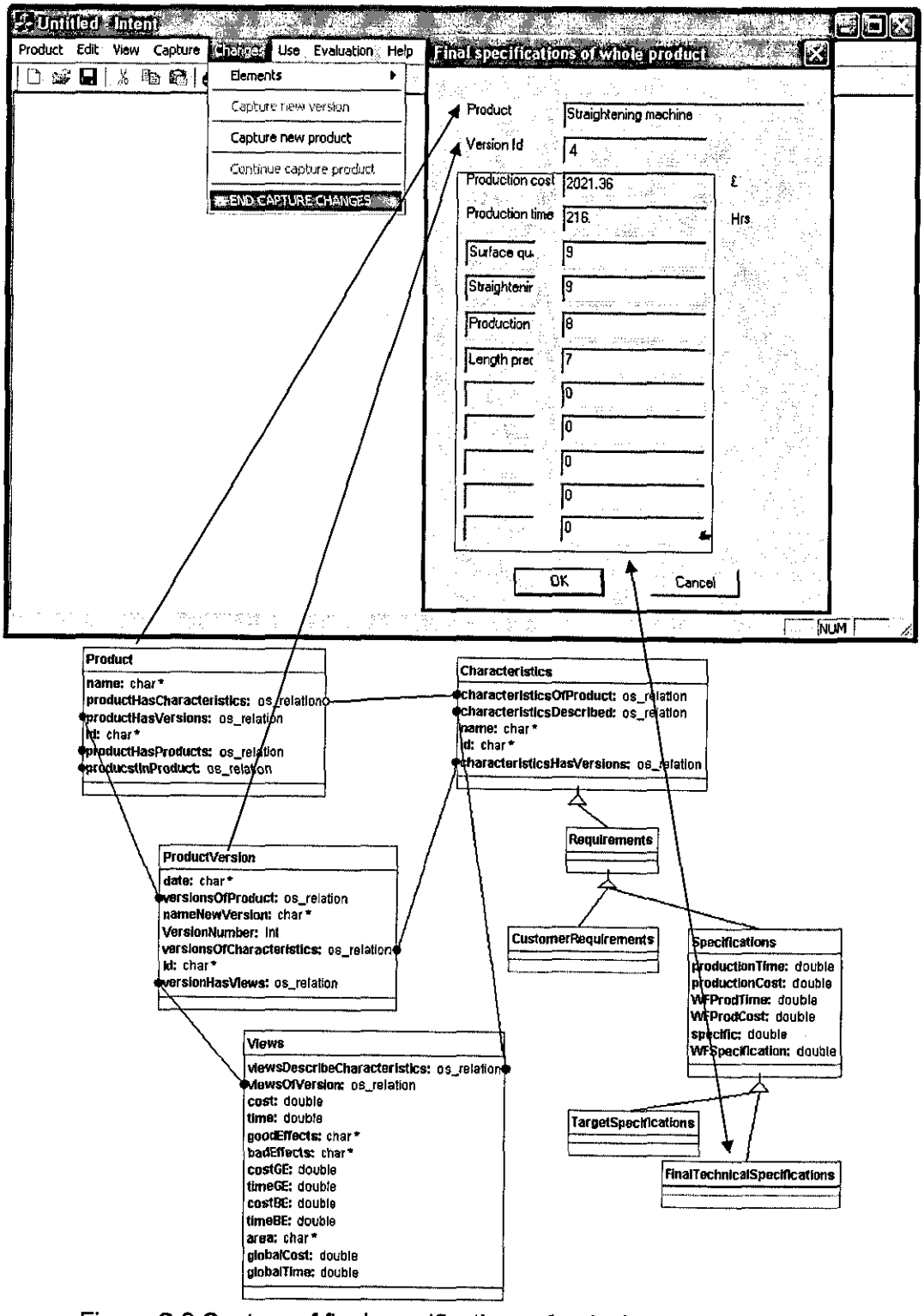

Figure C.8 Capture of final specifications of redesign

APPENDIX D. CHANGES TO PRODUCT

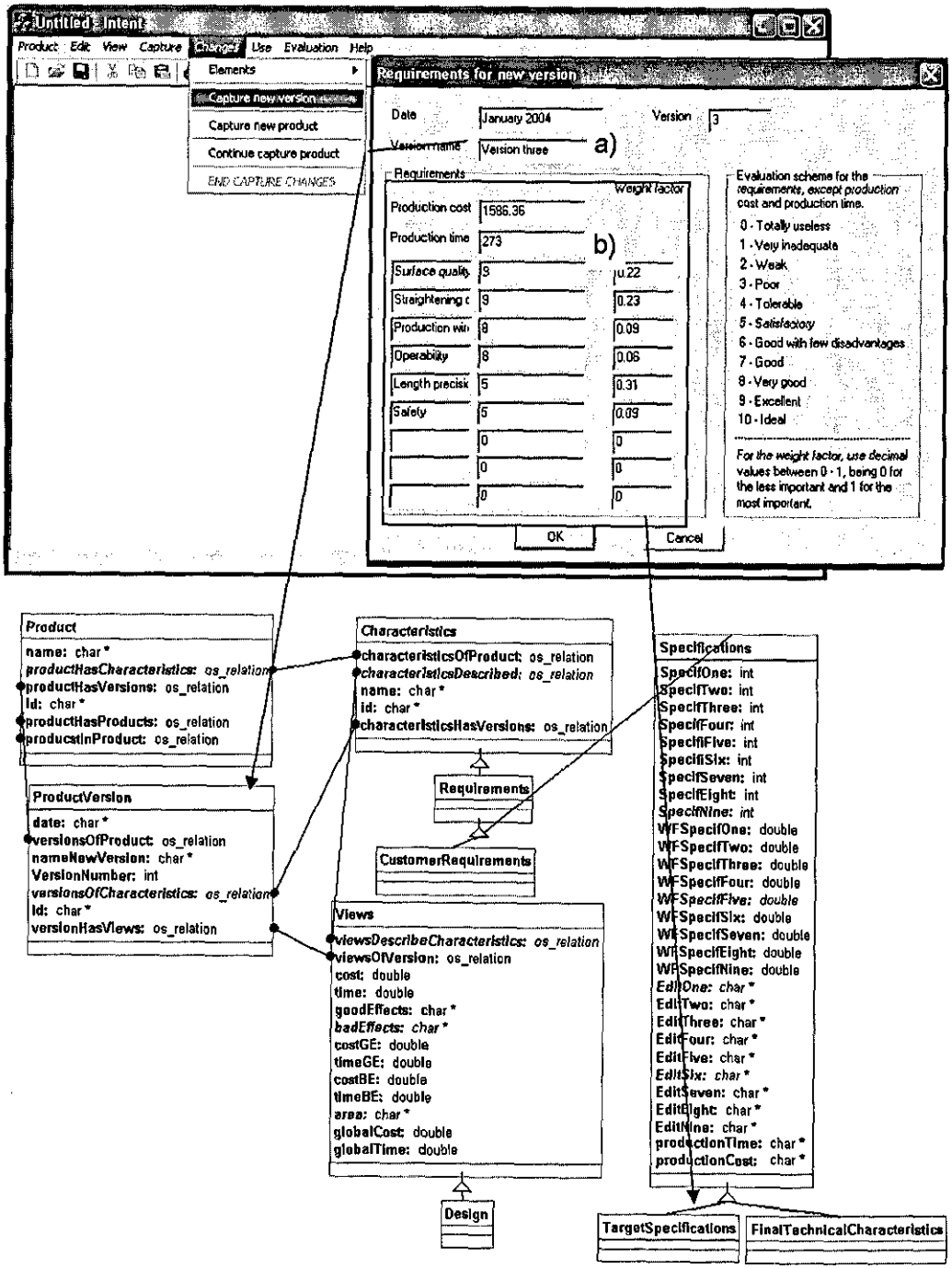

Figure D.1 Capture of specifications of new version.

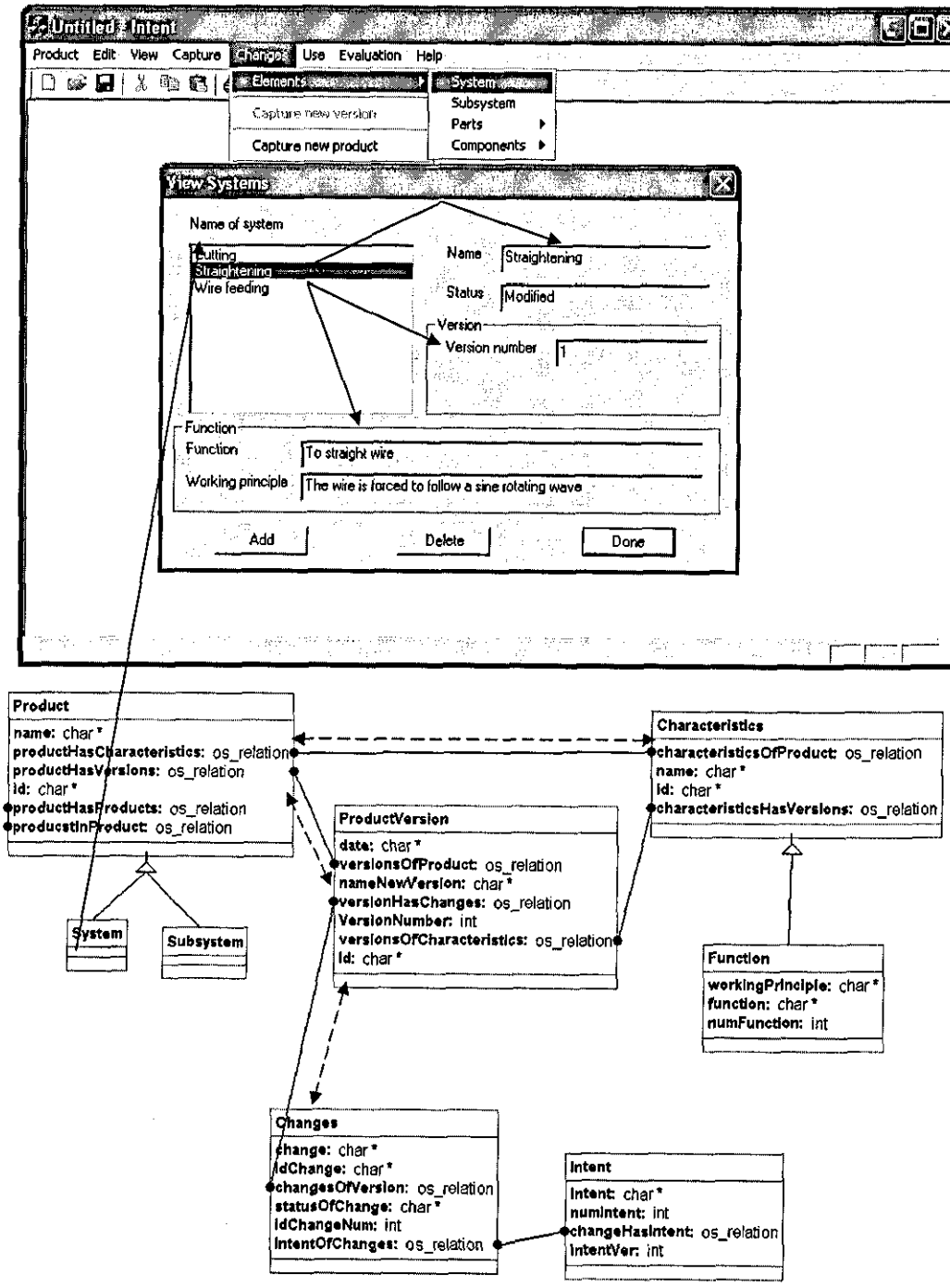

Figure D.2 Selection of a system to be changed

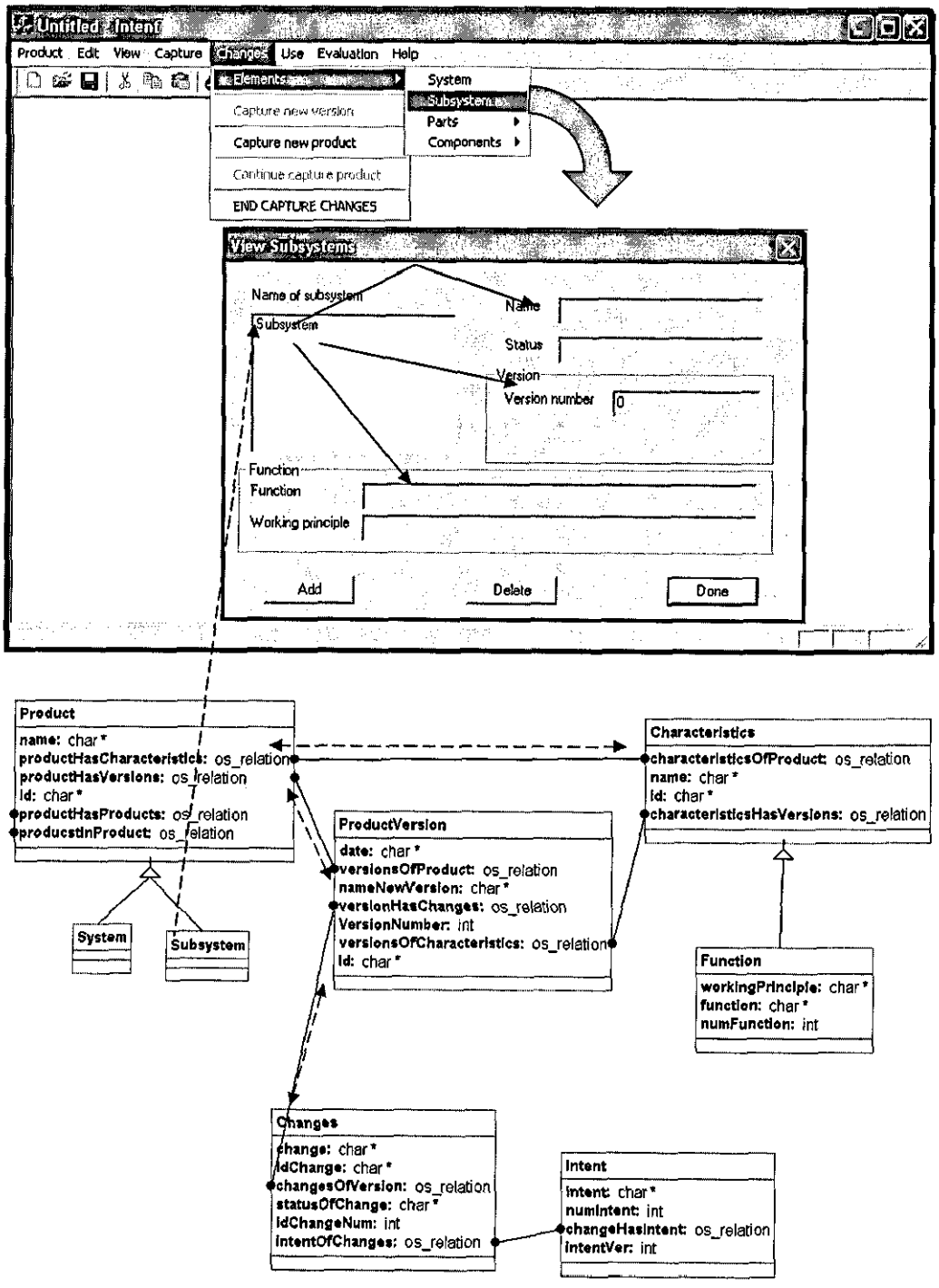

Figure D.3 Selection of a subsystem to be changed

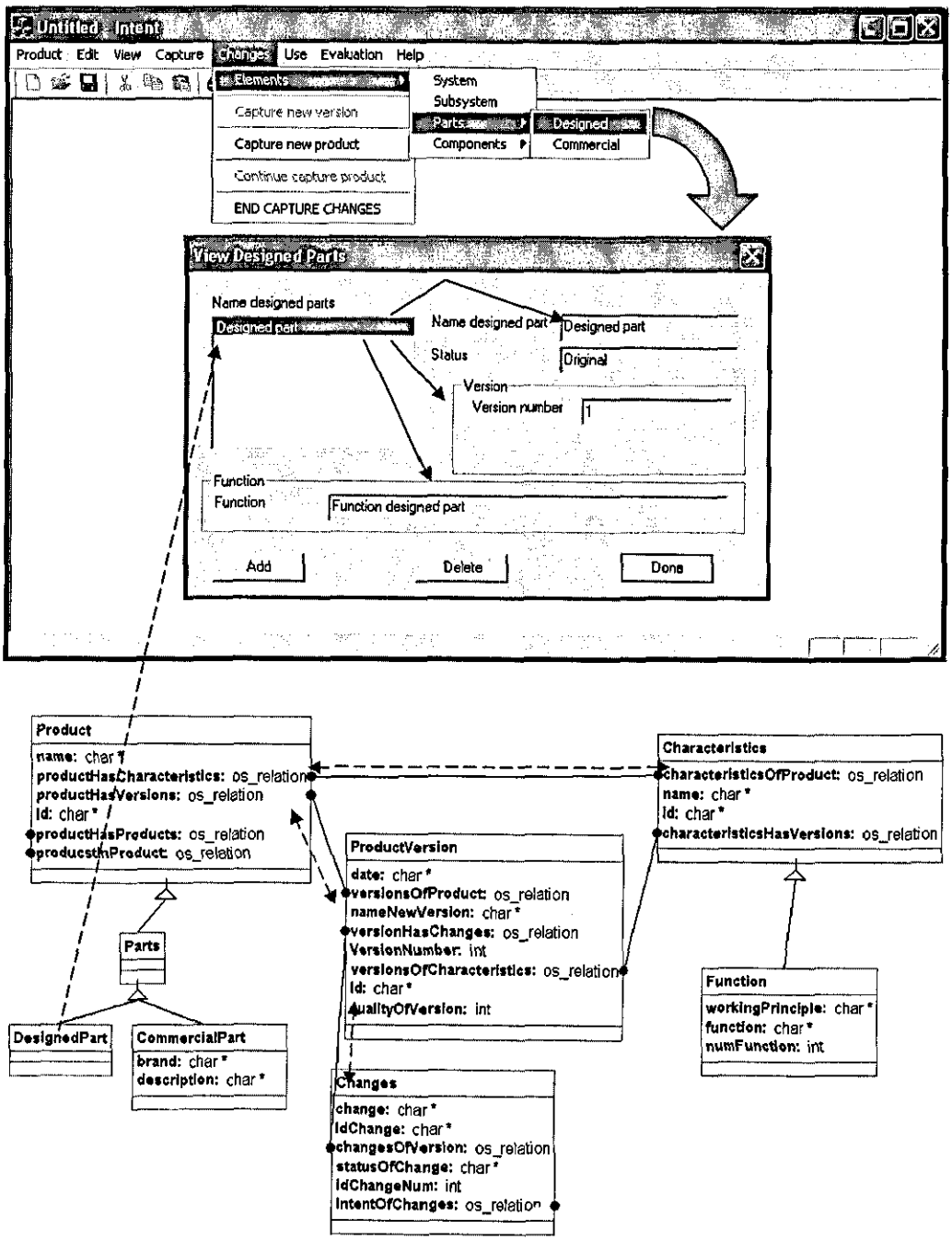

Figure D.4 Selection of Designed part

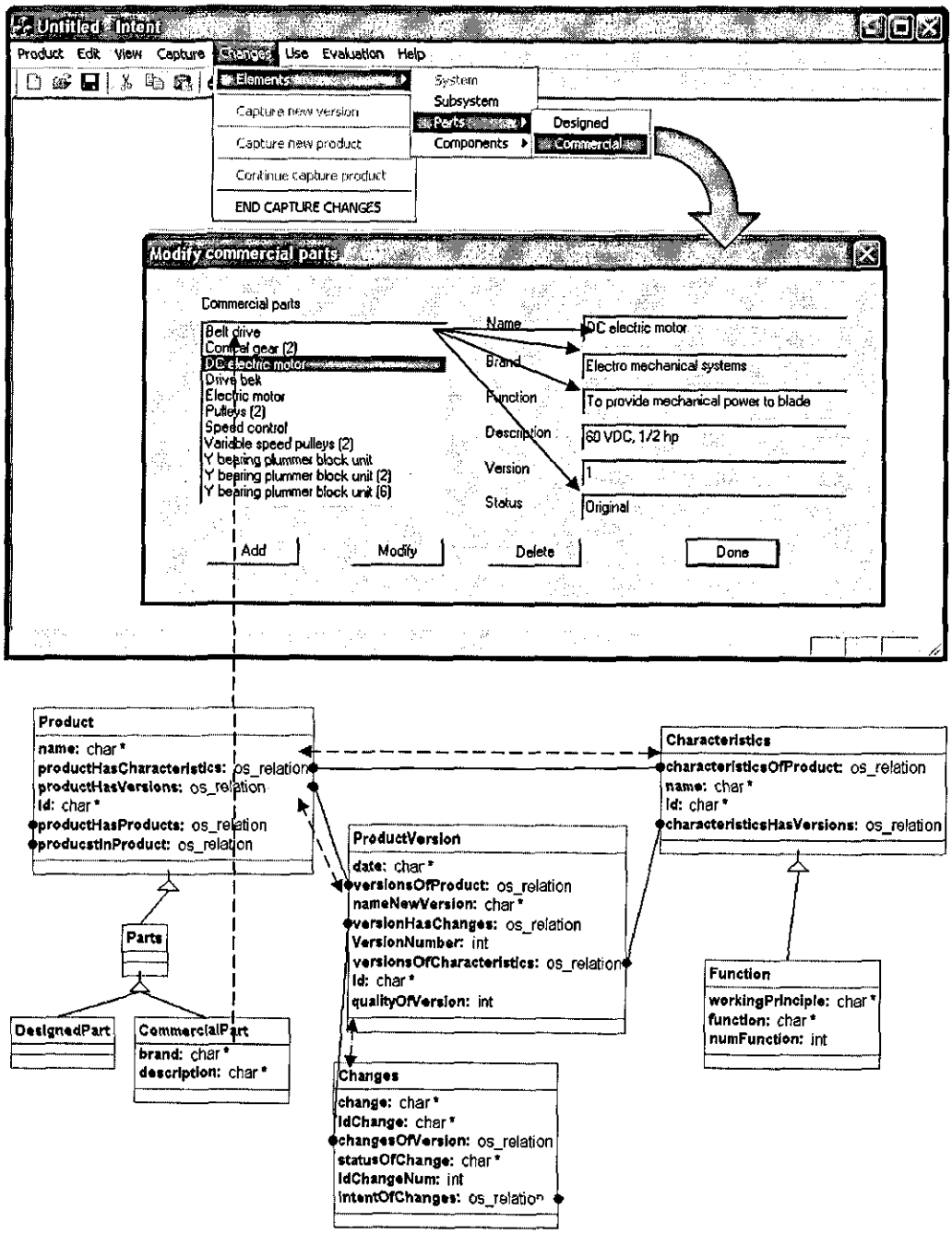

Figure D.5 Selection of Designed part

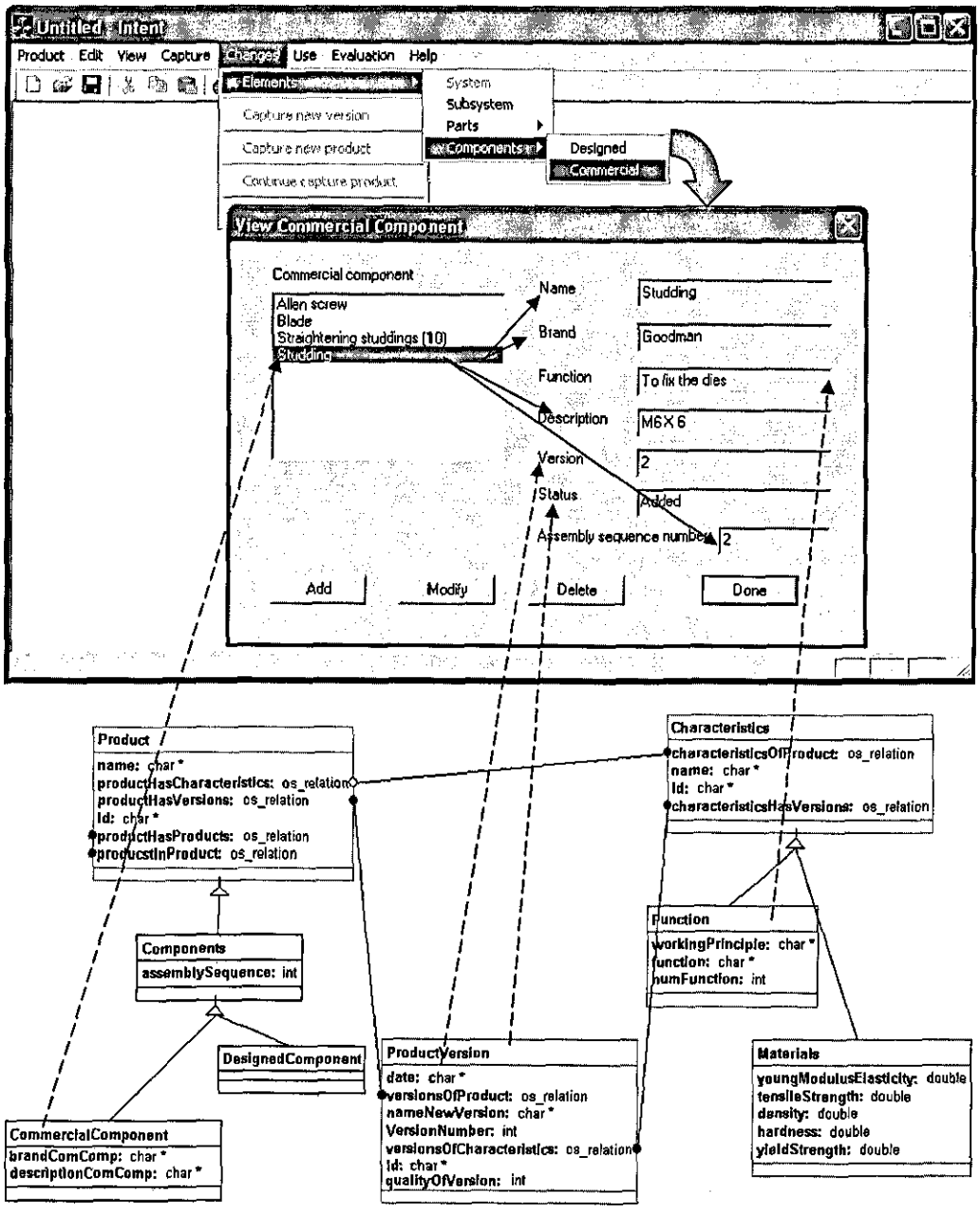

Figure D.6 Selection of Commercial component

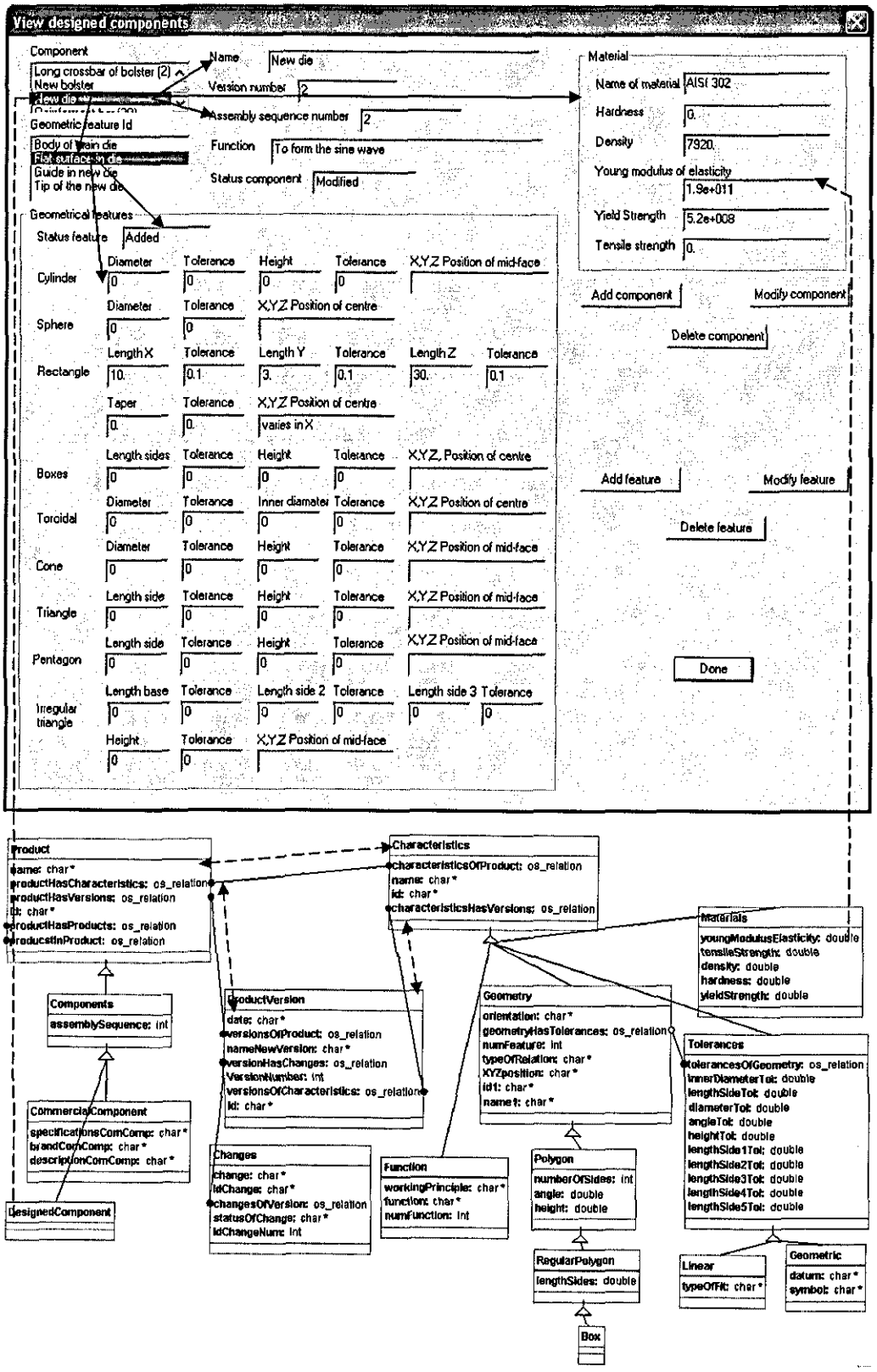

Figure D.7 Selection of designed component and geometrical features

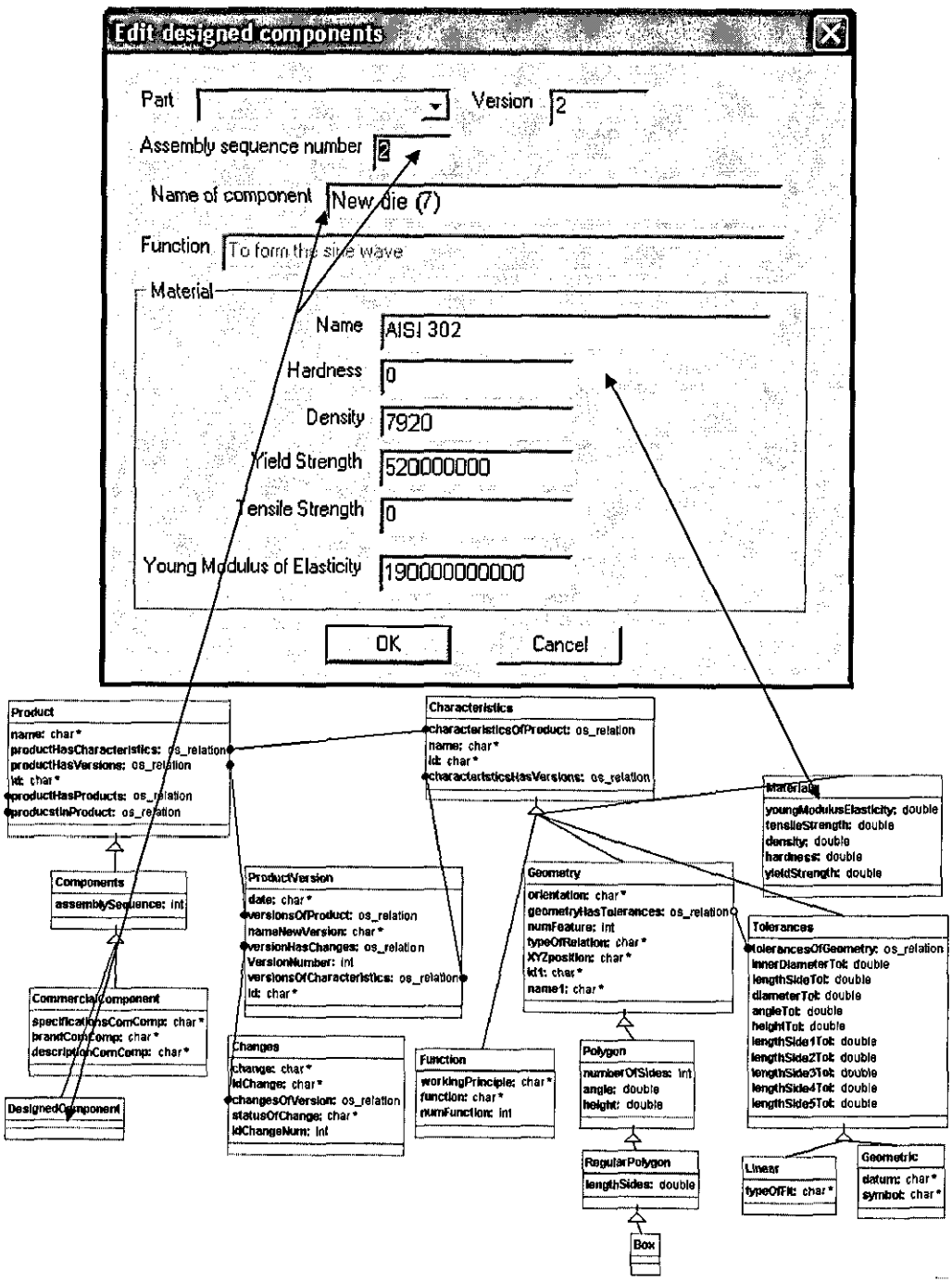

Figure D.8 Interface to modify the component

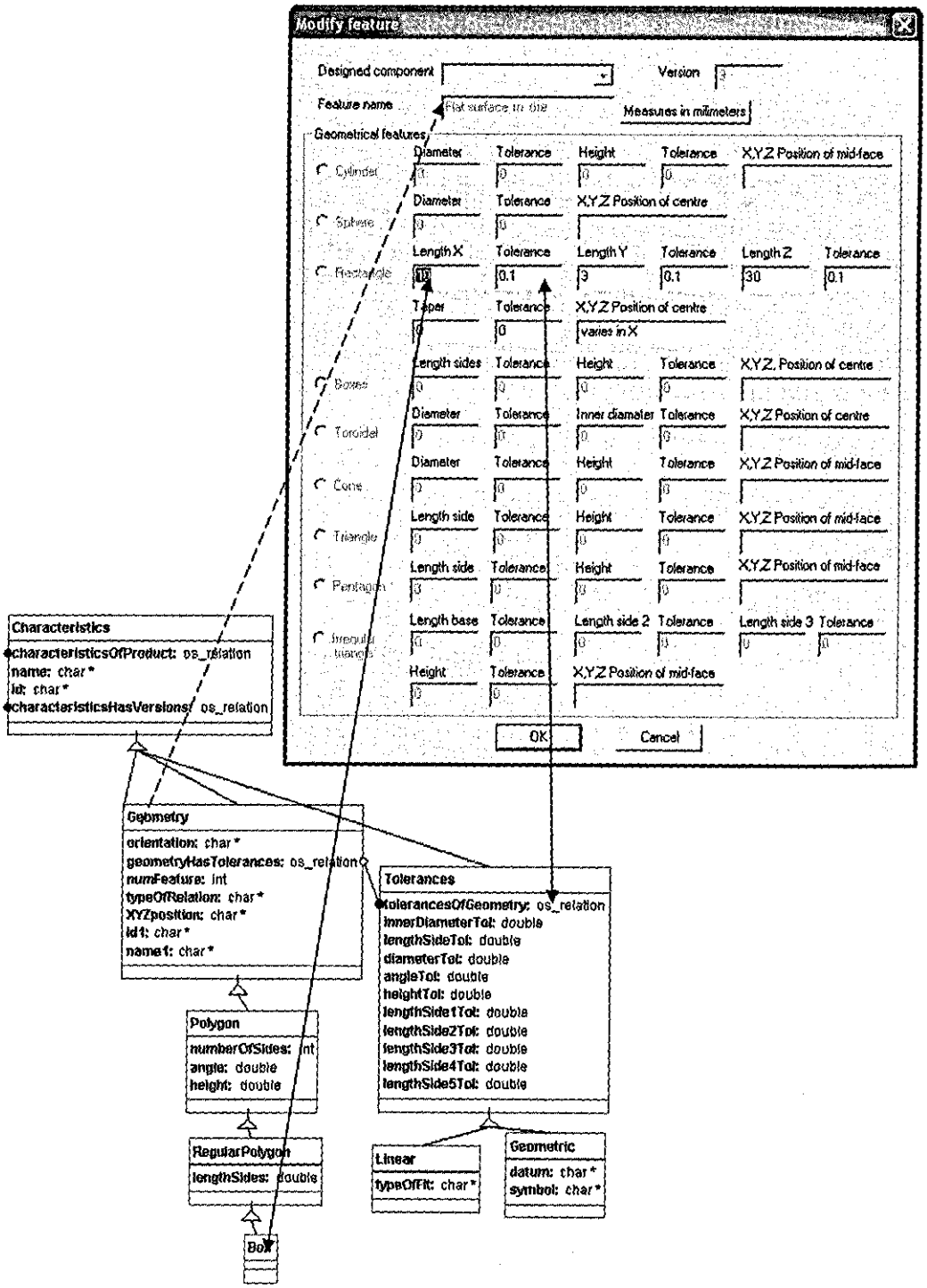

Figure D.9 Interface to modify the geometrical features

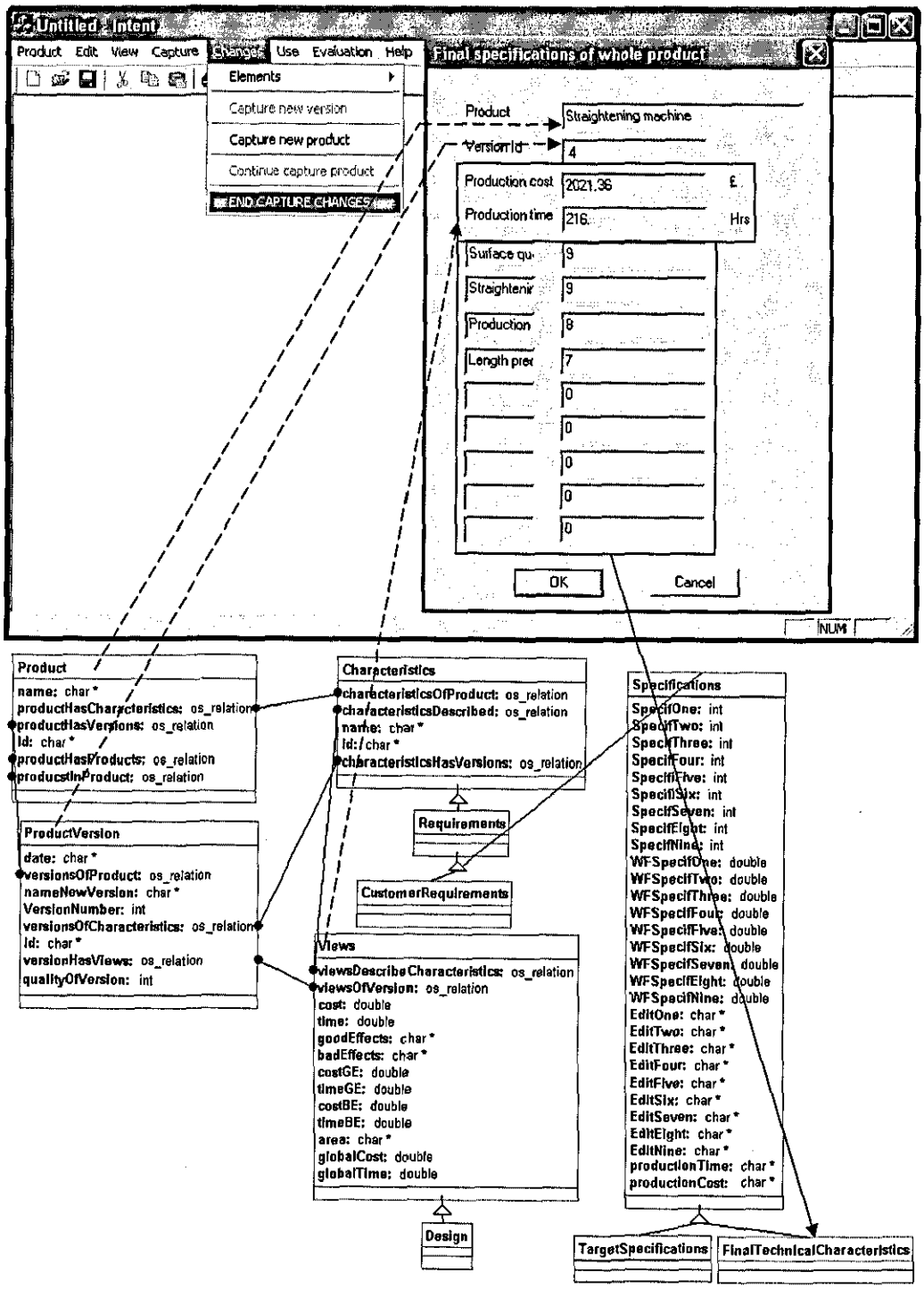

Figure D.10 Capture of final specifications of the version

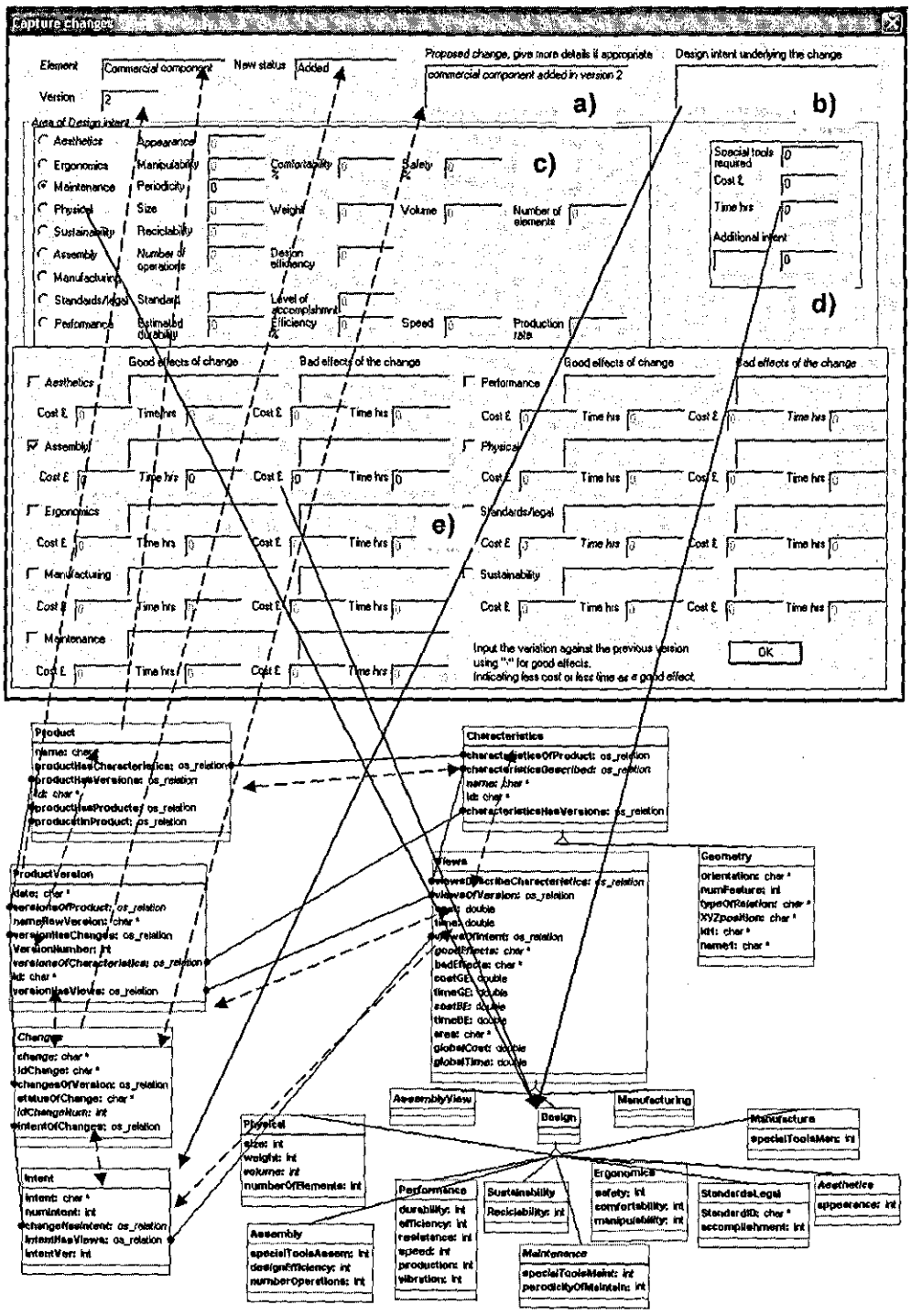

Figure D.11 *Capturing of the Design Intent and consequences of the changes*