A Flexible Distributed Design Assistance Tool for Early Design Phases

Yang Liu

December 2007

A Flexible Distributed Design Assistance Tool in Early Design Phases

Yang Liu

Dissertation presented for the degree of Doctor of Philosophy (Mechanical Engineering) at the University of Stellenbosch

Promoter: Prof. A.H. Basson

December 2007

Declaration

I, the undersigned, hereby declare that the work contained in this dissertation is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature: Yang Liu

Date: 29/05/2007

Acknowledgements

The author wish to express his gratitude towards the following person and organizations:

- My promoter Prof. Anton H. Basson for his enthusiasm, guidance and support through the study.
- My sisters QingPing Liu, Qi Liu and Cheng Liu, and my friends for their moral support and encouragements.
- Chen Ling who provided soul support.
- My fellow student for their unselfish support and advice.

I thank especially my mother HuiQi Jiang and father YuanYu Liu, who gave me the essential support during my study.

Abstract

The globalisation is increasing the complexity of product development in terms of product variants and the range of technologies implemented. It emphasises the requirement for developing various design information support systems for the world market. However, small and medium enterprises that employ a wide range of design procedures may not be able to afford customised information support systems, with the result that there is a need for flexible, i.e. easily adaptable, design support tools.

Four case studies were carried out to investigate the requirements for an information support system aimed at the design process and design documents. They indicated that a design information support system aimed at supporting design teams in the pre-detail mechanical design phases should be able to adapt various design methods and handle design information in a flexible way. Flexible here means being applicable over a wide range of contexts and extendable without affecting data already captured.

Ontology based approaches are widely applied where diverse information has to be handled. The development of the Internet today also makes a distributed design approach more and more popular for mechanical design. An internet-based design support system called DiDeas II (**Di**stributed **De**sign **as**sistant) was developed here with an ontology-based approach implemented to provide distributed and flexible assistance during concept generation in small companies. The DiDeas II has separate server side and client side programs, which communicate through a TCP/IP connection.

DiDeas II allows design teams to manage their design information according to various design methods, to decrease time-delays and to improve communication between team members. These benefits were confirmed in two case studies carried out to evaluate DiDeas II.

Keywords:

Distributed design; ontology; concept design, web-based system.

Opsomming

Globalisering verhoog die kompleksiteit van produkontwikkeling, in terme van produk variante en die bereik van tegnologieë wat geïmplementeer word. Dit beklemtoon die behoefte om verskeie ontwerp-inligting-ondersteuningstelsels vir die wêreldmark te ontwikkel. Klein en medium ondernemings wat 'n wye spektrum ontwerpsprosedures gebruik, kan egter nie doelgemaakte inligting-ondersteuningstelsels bekostig nie, met die gevolg dat daar 'n behoefte vir maklik-aanpasbare ontwerp ondersteuningstelsels is.

Vier gevallestudies is uitgevoer om die vereistes vir 'n inligting-ondersteuningstelsel gemik op die ontwerpproses en ontwerp dokumente, te ondersoek. Dit het aangetoon dat 'n ontwerp-inligting-ondersteuningstelsel, wat ontwerpspanne in die voor-detail meganiese ontwerp fases moet ondersteun, by verskeie ontwerpmetodes moet kan aanpas en ontwerpsinligting op 'n aanpasbare manier kan hanteer. Aanpasbaarheid in hierdie konteks beteken toepaslik oor 'n wye spektrum kontekste en uitbreibaar sonder om data wat alreeds ingevoer is, te beïnvloed.

Ontologie-gebaseerde benaderings word wyd toegepas waar diverse inligting hanteer moet word. Die ontwikkeling van die Internet maak 'n verspreide-ontwerpbenadering meer en meer gewild vir meganiese ontwerp. 'n Internet-gebaseerde ontwerp-ondersteuningstelstel genaamd DiDeas II (**Di**stributed **De**sign **as**sistant) is hier ontwikkel met 'n ontologie-gebaseerde benadering wat daarop gemik is om verspreide, aanpasbare hulp te verleen aan klein maatskappye gedurende konsep- ontwikkeling. Die DiDeas II stelsel het afsonderlike bediener en kliënt programme wat deur 'n TCP/IP verbinding kommunikeer.

DiDeas II laat ontwerpspanne toe om hulle ontwerp inligting volgens verskeie ontwerpmetodes te bestuur, tydvertragings te verminder en om kommunikasie tussen spanlede te verbeter. Hierdie voordele is bevestig in twee gevallestudies wat uitgevoer is om DiDeas II te evalueer.

Sleutelwoorde: Verspreide ontwerp; ontologie; konsepontwerp; web-gebaseerde stelsel.

CONTENTS

List of Figures

List of Tables

1. INTRODUCTION AND OVERVIEW

1.1. INTERNET-BASED DISTRIBUTED CONCEPTUAL DESIGN

At the present time, product design and manufacturing are more complex than ever before. It is beyond the scope of a single person, a single team or even a single department to comprehend fully all the aspects of the design effort. As a result, design and manufacturing collaboration do not only exist between the members of design teams, but also between designers from different companies and even between designers from different countries. This globalisation of design and manufacturing is assuming a leading position in industry.

The growing globalisation is based on the principal of making the most efficient use of the resources that a specific task would require. In product design, this principle means that the knowledge and expertise of all the groups involved in the design team, including marketing, design, suppliers, product management, etc., are exploited independently of how these groups are distributed geographically and organisationally. Distributed product development therefore offers a wide spectrum of new possibilities, such as enterprises sharing information via the Internet and the knowledge of experts from all over the world being exploited.

Anderl et al. [1999] have described a number of characteristics which are important for distributed product development processes. These characteristics are, amongst others:

- number of partners
- location
- time
- language
- intensity of collaboration
- distribution of components

- number of interfaces
- tool compatibility
- compatibility of methods

These characteristics provide the factors that will affect the development of a distributed system. In this system, confirmed links to each of the distributed group are necessary in order to accomplish the transformation of information.

The Internet, and in particular the web infrastructure being utilised as a new information highway, can link together distributed activity execution agents, persons or more basic computational processes. With the advances of computer technologies, many Internet-based systems have been developed for improving the efficiency and capability of distributed design. People can share knowledge, can negotiate, can coordinate and can manage activities via the Internet. Such collaboration activities can occur all over the world without any physical boundaries.

To develop an Internet-based system is becoming the focus of researchers in many different fields. For the manufacturing industry, in particular, these applications can allow designers to interact efficiently, but can also support knowledge capture, transformation and collaboration.

Conceptual design is an early phase of the product development process, and it is characterised by fuzzy problems and by tolerating a high degree of uncertainty. During the conceptual phase of design, designers not only need to determine the physical structure of the design, but also need to verify design functions and specifications. To make an effective conceptual design, a designer needs a clear understanding of the design requirements and sufficient design knowledge to be able to develop creative ideas. The successful management of the design requirements, as well as engineering knowledge, can bring large benefits for the development of a quality concept, which can lead to a competitive product.

1.2. FLEXIBLE DESIGN INFORMATION MANAGEMENT

Design consists of procedures involving the search, retrieval, transformation, transportation, representation and interpretation of information. In the early design phase, a designer spends most of his/her time searching and creating ideas, and communicating and negotiating with colleagues, customers, suppliers and manufacturers in order to come up with a good design.

During the process, the designer has to collect, organise and filter related information. Because of the different types of product manufacturing, he/she would need different kinds of information in order to accomplish his/her design work.

To manage the design information in an organised way is very important for getting an ideal solution. It also decreases the time that the designer has to spend on searching information, and this time can be used more productively to develop creative ideas for the design of the product. Different design methodologies are applied in different companies, and there is no single design model that can cover the range of all companies; the design process is mainly determined by the company resources such as the engineers' experience, the project type and the project scope. A flexible design information system can provide the opportunity for companies to alternate different design methodologies for different design resources and projects – it can allow the user to adapt different design methods. There are many support tools available, however they mostly use pre-defined design methods without flexibility.

Furthermore, due to today's competitive market where technology develops at a very fast pace, product design is getting more and more complicated and requires that the development time should be as short as possible without losing product quality. This requires that the design information system is expandable (another type of flexibility), in that new information can always be added without effecting the existing information.

Because of the high complexity of the process and the integration needed to accomplish a desired effect, product development requires a team of experts with different backgrounds. The development process also involves co-workers from a variety of specialised fields and who are located in different places. The design process is therefore an integration of knowledge from various perspectives at different locations, i.e. distributed collaborative design. It is essential that the developing system discussed here should support distributed collaborative design.

1.3. MOTIVATION

Design information systems are becoming more and more important due to the globalisation of product development, the increasing complexity of products (in terms of product variants and range of technologies employed) and the increasing emphasis on improving product development efficiency (shortening times, reducing cost, increasing performance). Although much research is being done to develop design information systems, the current systems are typically applicable in a limited range of contexts. To be flexible, i.e. to be applicable over a wide range of contexts, design information systems have to acknowledge the diversity of the design process. The diversity of the design process includes issues such as aspects of scope, level of abstraction, level of detail, conflicting use of terminology, wide-ranging procedures, multi-disciplinary character, and different information types. A flexible way for handling all the design information in the conceptual design phase of the product design and for creating smooth communication among designers is needed.

A design information system is proposed that is based on the use of ontologies and conceptual graphs. This approach provides for a great degree of flexibility with the result that main data structures and, to a lesser extent, user interfaces are not affected by the introduction of new terminology, procedures and tools.

Due to the explosive growth of the Internet and the associated information infrastructure, as well as the ubiquity of World Wide Web browsers, the use of the Internet and the World Wide Web as methods for communication and information transfer is increasing. This information revolution is impacting on how companies will operate in the future, because companies are always looking for those methods that can help them to improve their work efficiency. By applying Internet technology, the information management system can utilise more information resources from different locations to assist the designer in obtaining more competitive products.

1.4. THE OBJECTIVES

DiDeas (Distributed Design assistant, called DiDeas I in the following sections) is an Internet-based system which was developed by Andreas Schueller [2002]. This system provides facilities to support the early design phases.

The objective of this dissertation is to provide a distributed, flexible, early design assistant system, based on the DiDeas I (named as DiDeas II in the following sections), for concept generation in small companies that would normally have a low budget. The system's flexibility will allow improvement of a design team's work by providing the ability to adapt to the specific design procedure applied in the group, without requiring changes to the source code and the database structure. In this way, DiDeas II can adapt to different design methodologies so that the designer can use their preferred design methods and terminologies. Based on an ontology database, the DiDeas II system would be able to handle different kinds of information to fulfil different usages. The system will use the Internet as communication medium so that groups who are in different/separate geographical locations can work together.

1.5. THESIS OUTLINE

In the literature review in Chapter 2, research carried out in the various fields related to distributed conceptual design and the ontology background and its applications are presented. The steps of the early phases of the design procedure are also discussed in detail.

In Chapter 3, case studies about design information during concept design are presented.

In Chapter 4, the methodology of applying the ontology to the DiDeas II system to manage design information is described. The data base structure is also identified.

Chapter 5 discusses the details of the implementation of the DiDeas II system. This discussion includes the user interface development with aspects of the Internet programming.

In Chapter 6 several case studies in which the DiDeas II system was evaluated, are described. The results of these case studies are also presented.

In the final chapter, Chapter 7, the conclusions and a few suggestions for future work are made.

2. BACKGROUND AND LITERATURE REVIEW

2.1. INTRODUCTION

This chapter starts with a description of several design models and their design processes. This is followed by a detailed discussion of the conceptual design phase and a review of research done on conceptual design. The following section deals with research done on distributed design and Internet-based systems. The next section describes ontology definitions and applications related to engineering design such as the organisation of flexible information systems. Especially the use of ontology in the field of database management is investigated in detail.

2.2. DESIGN METHODOLOGY AND DESIGN PROCESS

The design process can be described as a map with instructions on how to get from the identification of a need for a specific object to the final product [Ullman, 2003]. It can therefore be said that this process transforms available information, knowledge and expertise in order to construct a means to get from an expressed need to a solution. The transformation has been viewed as an iterative evolution of design from the abstract to the concrete.

Different models of the design process have been developed since the early 1960s, and a few of these typical design models will be discussed below.

The new Guideline VDI 2221, which was developed in Germany, can be applied to a number of design disciplines, such as mechanical engineering, precision engineering, software engineering and process engineering. However, the focus seems to be on the mechanical design process [Eekels & Poelman, 1995]. The general approach according to the VDI 2221 is illustrated in Figure 2-1.

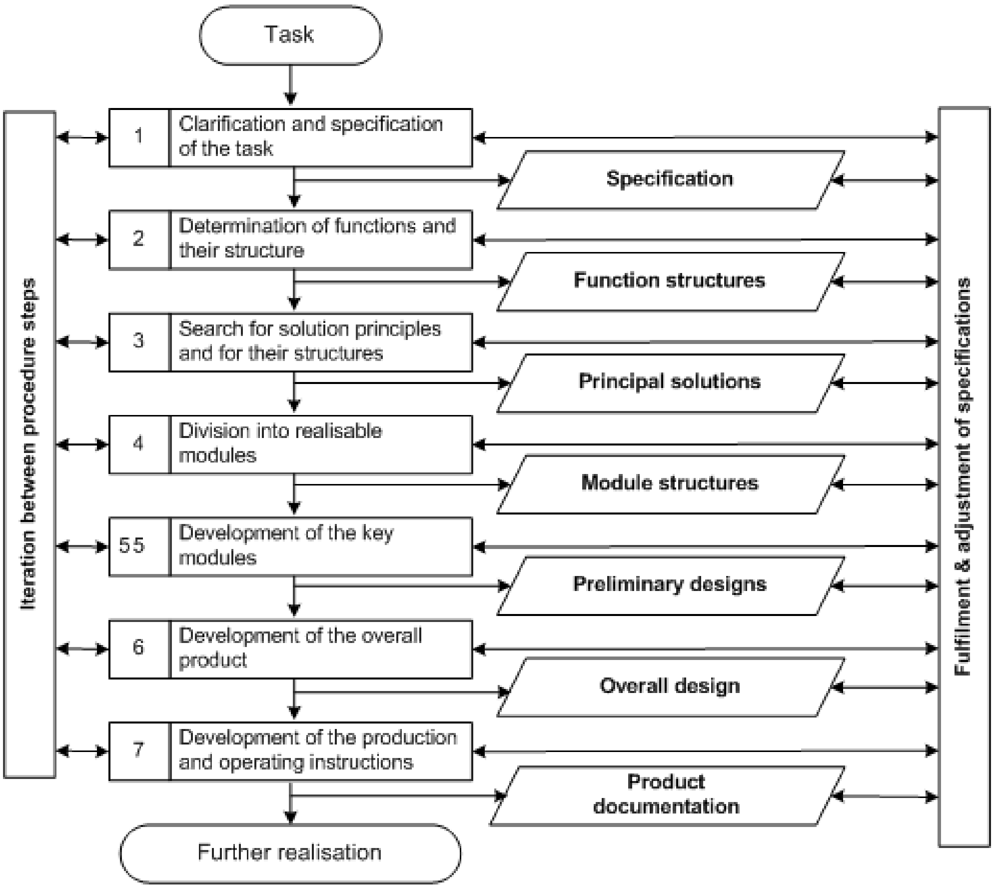

Figure 2-1 General Approach to Design [VDI 2221, 1993]

Pahl and Beitz [1997] point out that the structure of the VDI guideline must not be seen as a step-by-step process that a designer needs to adhere to strictly. Instead, the procedure has an iterative character, where some steps can be skipped and others are repeated. The focus is more on the required functions than on the processes to be used. Their design model is shown in Figure 2-2.

According to Ullman [2003], the mechanical design process consists of five main steps: project definition and planning, specification definition, conceptual design, product development and product support. These phases are performed in an iterative way, and the refinement always occurs before the final requirements are satisfied. This model tries to identify all phases that a technical system has to pass through during its origination and operation in order to find all the factors that will affect the technical system. These design phases are listed in Figure 2-3.

Ulrich and Eppinger [1995] classify the product developement process according to five phases:

1. Concept development
2. System-level design
3. Detail design
4. Testing and refinement
5. Production ramp-up

These phases provide the general design phases that are not specific to engineering design, and can be seen as a combination of the marketing, the designing and the manufacturing issues.

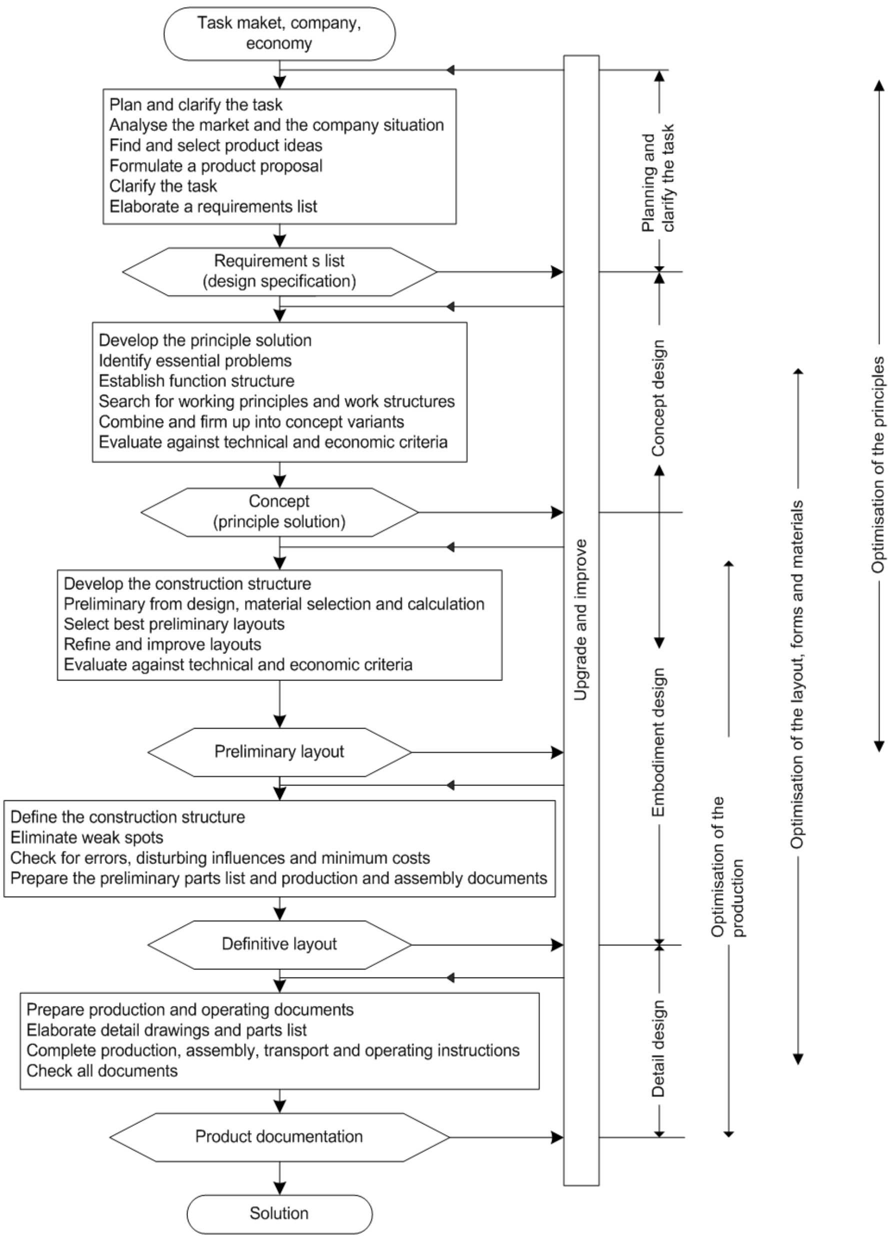

Figure 2-2 Pahl & Beitz, Design Model [1997]

From the point of view of system design, Blanchard and Fabrycky [1998] describe design as an iterative process that consists of three steps:

1. Synthesis: putting together of the parts or elements to produce new effects and to demonstrate that these effects create an overall order

2. Analysis: the resolution of anything complex into its elements and the study of these elements and their interrelationships

3. Evaluation: a prediction of how good the design alternative might be if it is chosen for implementation

These steps describe a more general design process that happens in different types of design and at different levels of the system design process.

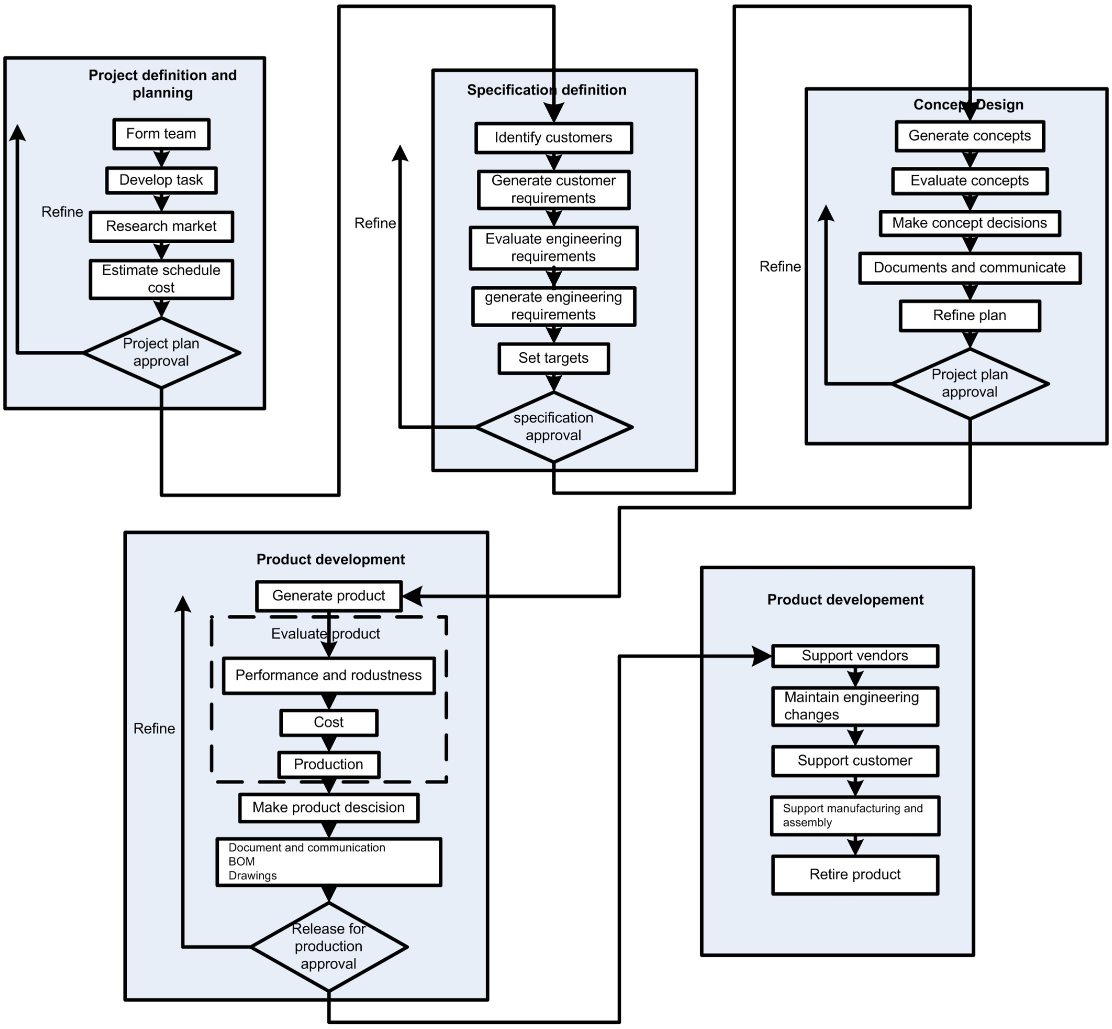

Figure 2-3 Ullman's [2003] Design Method Steps

As these design models use different technical terms to illustrate the various design processes, designers who use different design methodologies have problems understanding each other when they have to communicate with each other in their design work. In the manufacturing industry, it happens that different companies apply different design methodologies and form their own design style, i.e. they develop technical terms to build their own technical standards.

Normally when a designer joins a design department, he/she needs to be trained in the specific design knowledge that that company uses so that there can be effective communication between designers in the same technical term environment. To build an interface in a certain design style will then not only improve the communication between the members of a design team, but by reviewing a previous design project that was organised by such a design style, a new designer can become familiar with the design environment in a specific company.

2.3. METHODICAL SPECIFICATION AND CONCEPT DEVELOPMENT

A concept is a description of the form, function and features of a product, and usually flows from a set of specifications, an analysis of competitive products, and an economic justification of the project [Ulrich & Eppinger, 1995]. Conceptual design is the formulation of the concept upon which the design will be based and the approximate determination of major dimensions, and selection of major components. At the conceptual design phase, components are represented by certain symbols and connected to each other in a configuration that performs a set of functions. Conceptual design is a crucial phase in engineering product development cycle, since it has been shown that the design phase determines the quality of products and 70 to 80% of the final production cost [O'Grady *et al*., 1988].

Conceptual design commences with the high-level descriptions of requirements, and proceeds with a high level description of a solution [McNiel *et al*., 1998]. Conceptual design is that phase in the product design cycle when the basic solution path is laid down through the elaboration of a solution principle. It involves formulation of abstract ideas with approximate concrete representations. The early or conceptual phase of the design process is dominated by the generation of ideas, which are subsequently evaluated against the general requirements' criteria. It is followed by a process whereby additional data are incorporated, allowing decisions to be made between competing alternatives as more tangible evidence of function is derived. Concept generation here can be defined as the procedure stretching from the customer requirements to the solution that can bring the physical actions.

Conceptual design is perhaps the most crucial phase in the development cycle of an engineering product. According to Wang *et al*. [1994], conceptual design is of great importance in computer-aided design (CAD), but it is very difficult to accomplish. Therefore, although computers have been used extensively in areas such as simulations, analysis and optimisation, there have been relatively few applications at the conceptual design phase. This is because knowledge of the design requirements and constraints during this early phase of a product's life cycle is usually imprecise and incomplete, making it difficult to utilise computer-based systems or prototypes [Basson *et a.l*, 2003].

Pahl & Beitz [1997] give a detailed description of the steps involved in the first two phases of the design process, as illustrated in Figure 2-4.

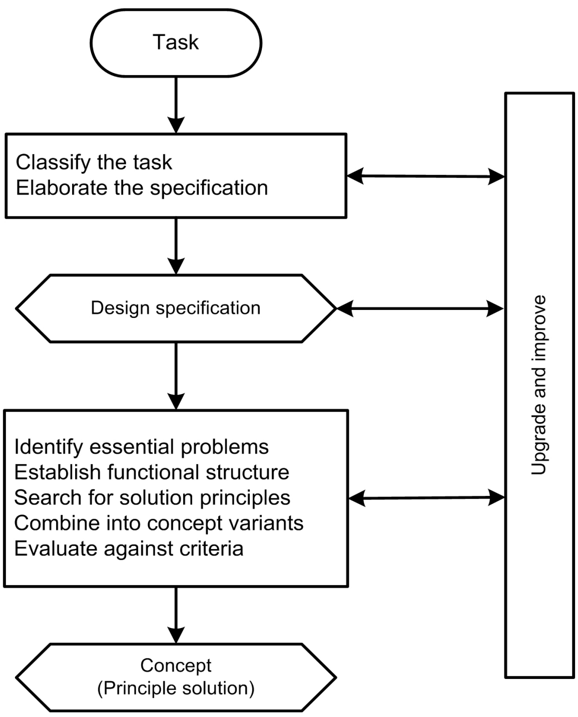

Figure 2-4 Early Steps of the Design Process

2.3.1. Identifying Design Problems

The specific goal of identifying the design problems is to get an understanding of these problems and to clarify them to the design team. The output of this design step is a set of constructed problem statements, organised in a hierarchical list.

Based on the Quality Function Deployment (QFD) technique, Ullman [2003] sees the design problems as being represented by the customer requirements. He suggests the following eight steps for the specification development:

1. Identify the customers
2. Determine the customers' requirements
3. Determine the relative importance of these requirements
4. Identify and evaluate the competition
5. Generate engineering specifications
6. Relate customer requirements to engineering specifications.
7. Identify relationship between engineering requirements
8. Set engineering targets

Ulrich and Eppinger [1995] also prefer "customer requirements" instead of the "design problem". They use the following six steps to identify these requirements:

1. Define the scope of the effort
2. Gather raw data from customers
3. Interpret the raw data in terms of customer needs
4. Organise the needs into a hierarchy of primary, secondary and (if necessary) tertiary needs
5. Establish the relative importance of the needs
6. Reflect on the results and the process

These steps ensure that the design product is focused on the customer's needs and that no critical customer need is missed.

Pahl and Beitz [1997] list four main steps for the specification development:

1. Compile the requirements, including information on quality and quantity, and clearly specify demands and wishes. The wishes can be classified as wishes of major, medium and minor importance.

2. Arrange the requirements in logical order, e.g. by defining the main objective, and the main characteristics, identifiable sub-systems or categories.
3. Distribute the requirements on forms to all project participants.
4. Update the list of requirements with the feedback.

Both Ullman [2003] and Pahl and Beitz [1997] recommend the use of guidelines and checklists to ensure the completeness of the customer and engineering requirements.

QFD (Quality Function Deployment) is a popular technique to help the designer to understand the design problem in an organised way (see Figure 2-5). It sets targets to be achieved for the engineering characteristics of a product, such that they satisfy the customer requirements. In this way the voice of the customer is fully recognised, and the customer requirements are not subject to reinterpretation by the design team.

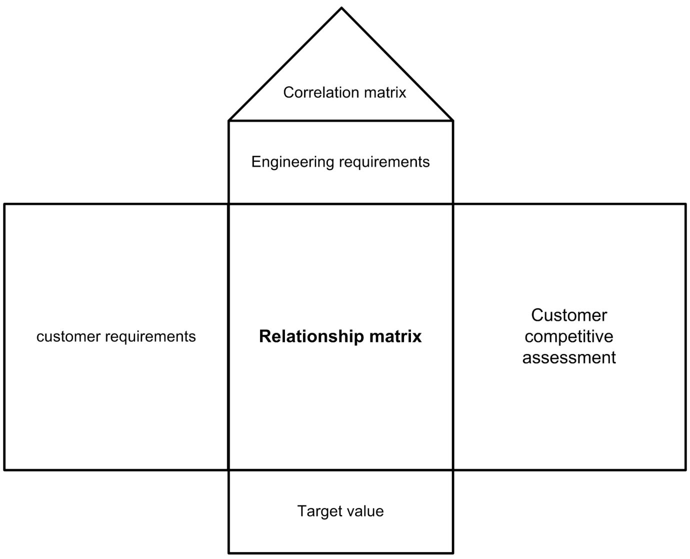

Figure 2-5 QFD Matrix [Ullman, 2003]

2.3.2. Functional Analysis

Functions are closely related to requirements. Generally the overall function of a product is equal to the main requirement, and this function can often be divided directly into identifiable sub-functions corresponding with sub-tasks. The development of a function structure for a product enables the designer to deal with a number of simple design problems instead of a single, more complex problem.

The procedure starts with the identification of the overall function, which gives a solution-neutral, abstract description of what the product must accomplish. This overall function is then divided into sub-functions to create a more specific description of what the elements of the product need to do in order to implement the overall function. Related functions, or functions that might be satisfied with similar solutions, can then be grouped into logical units, a procedure known as functional packaging [Blanchard & Fabrycky, 1997].

The functions are often represented as a black box where the input of the box is transformed to the required output. Three types of graphical illustration of the function structure are commonly used [Schueller, 2002]:

- the block diagram
- the logical flow diagram
- the hierarchical function tree.

2.3.2.1. Block Diagram

In a block diagram, a single function can be represented by a black box, in which the input in the form of energy, material and/or information is processed and transformed into the output, as illustrated in Figure 2-6.

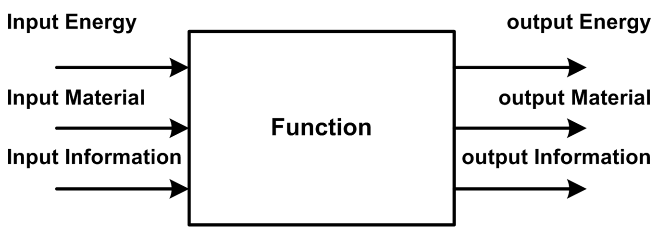

Figure 2-6 Inputs and Outputs of a Function [Pahl & Beitz, 1997]

The complete function structure consists of the overall function and various levels of sub-functions, as shown in Figure 2-7. The block diagram is the most complete representation of a function structure.

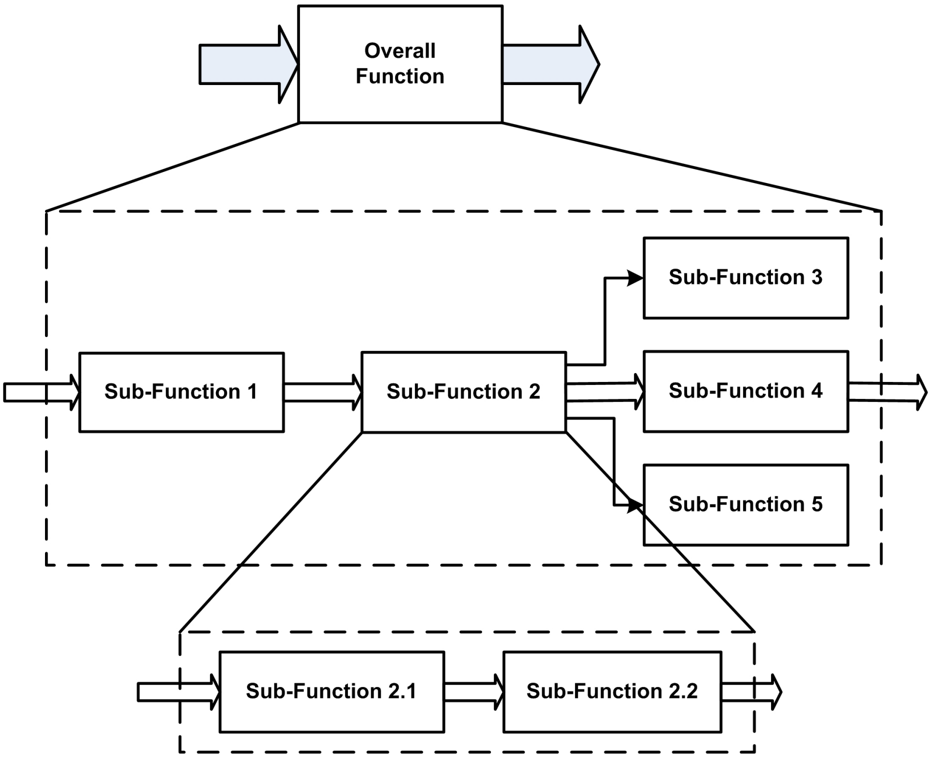

Figure 2-7 Function Structure with Overall Function and Sub-Functions [Pahl & Beitz, 1997]

2.3.2.2. Logical Flow Diagram

The logical flow diagram is widely used in the fields of systems engineering and computer science, but can also be applied to the mechanical design process. Instead of boxes it uses symbols to represent logical functions, such as AND and OR.

2.3.2.3. Function Tree

The third illustration type is the function tree structure. Ullman [2003] states that a tree structure is often more effective for the representation of a function structure than a block diagram is. An example is shown in Figure 2-8.

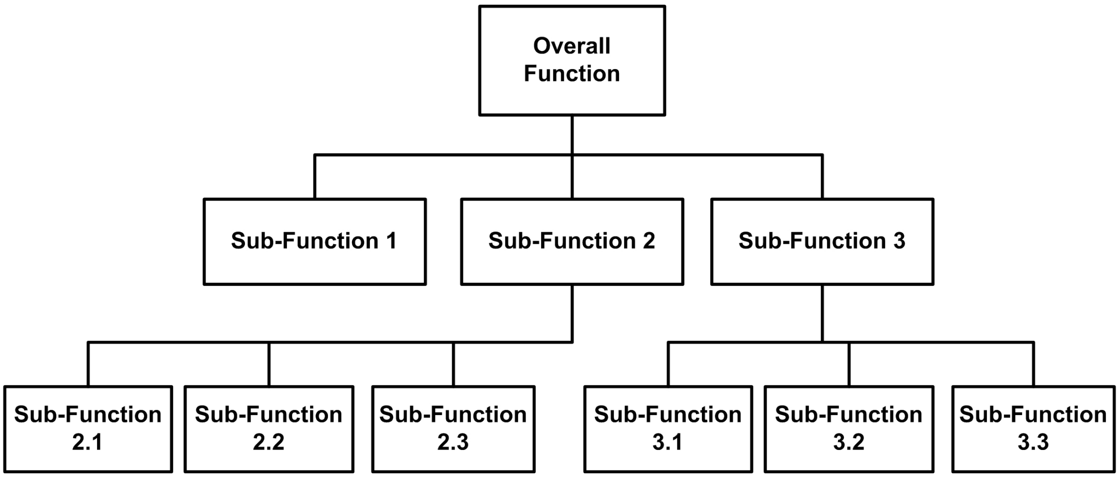

Figure 2-8 Hierarchical Function Structure

The hierarchical tree structure does not fully describe the function structure, but it allows for a better illustration of the different levels of decomposition than the block diagram does.

These function representations can give a designer a clear function decomposition in order to understand the product design needs. These functions always need a link to the solution. As can be seen in Figure 2-9, Begelinger *et al*. [1999] provide a function-means tree with an element component and (production) process. In this tree, a distinction is made between principle functions and connection functions. The connection functions represent the flows of energy, material and information between principle functions. It makes the use of the tree as a conceptual design modelling method more feasible.

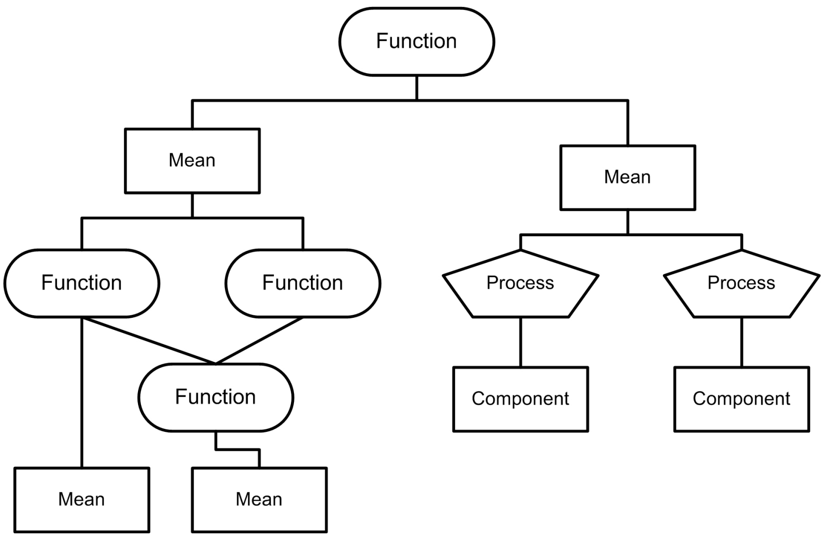

Figure 2-9 Function-Means Function Tree [Begelinger *et al*., 1999]

2.3.3. Search for Solution Principles

After the function tree is built, the overall function is split into sub-functions, and their structures on a number of hierarchical levels. These partial functions, with their inputs and outputs, are used to break the design problem down into manageable parts. For each sub-function, solutions are sought, which are evaluated, selected and finally combined to generate solution principles that again are evaluated and selected. The tools that can be used to generate different solution principles to fulfil a function include literature searches, analysis of natural and technical systems, intuitive solutions (such as brainstorm, 6-3-5 method, etc. [Ullman, 2003]) and design catalogues (handbooks).

Further, the morphological matrix, a classification scheme of Zwicky [1966], has been proposed for the synthesis of overall concepts. It is set up by constructing a table with graphical representations of solution principles for all functions. The synthesis takes place by selecting, per function, the best solution principle that is compatible with the selected solution principle of the other functions.

2.3.4. Development of Concepts

A product concept can be defined as an approximate description of the technology, the working principles, and the form of the product [Ulrich & Eppinger, 1995]. A concept is usually expressed in the form of sketches or a rough three-dimensional model that often includes a textual description. This step is quite dependent on the solution principles search stage; the development methods are also similar. According to Ullman [2003] there are two logical methods that evolved since the 1990s: TRIZ (Theory of Inventive Machines) and axiomatic design. TRIZ is a complex collection of methods that requires extensive study. Axiomatic design was developed by MIT (Massachusettes Institute of Technology) to describe how an academic theory could be used to develop a product.

2.3.5. Concept Evaluation

The degree to which a product can comply with customer requirements and can be developed successfully depends to a large extent on the quality of the underlying concept. A good concept can not always be implemented in subsequent development phases, but a poor concept can rarely be manipulated to a successful product.

The evaluation process consists of the following steps:

1. Rate the concepts
2. Rank the concepts
3. Combine and improve the concepts
4. Select the concepts
5. Evaluate the concepts

The decision matrix is a simple and effective method to identify the best concepts. It scores each of the concepts relative to the other according to certain criteria that are based on customer requirements. According to Ullman [2003], it is the most effective if each design team member does this scoring independently and the individual scores are then compared. The basic structure of the decision matrix is shown in Table 2-1.

		Alternatives		
Criteria (requirements or specifications)	**Importance**	Alternative 1 (A1)	Alternative 2 (A2)	Alternative 3 (A3)
Criteria 1 (C1)	xx	Evaluate A1 using C1	Evaluate A2 using C1	Evaluate A3 using C1
Criteria 2 (C2)	yy	Evaluate A1 using C2	Evaluate A2 using C2	Evaluate A3 using C2
…	…			
…	…			
	Satisfaction	A1 score	A2 score	A3 score

Table 2-1 The Basic Structure of the Decision Matrix [Ullman, 2003]

2.3.6. Documentation in Early Design

Design documentation would typically consist of a physical file or a set of physical files. This file would contain design notes, design reviews, analysis reports, source files and drawings or sketches that contain descriptive information to be handled by integrating further technical examination. Blanchard and Fabrycky [1997] advise that a formal design review be held after the conceptual design. This can provide a common baseline for all project personnel, and should record design decisions and the reasons for selecting the baseline.

Moreover, during the design process, it is useful to consolidate different of the product documentation that correspond to relevant phases of the design project [Giannini *et al*., 2001].

2.4. CONCEPTUAL DESIGN SUPPORT AND DISTRIBUTED DESIGN SYSTEM

Wiegers *et al*. [1999] outline the requirements for conceptual design support as follows:

1. The need to support the creative activity of designers
2. Natural interaction methods
3. The use of examples and real objects
4. Rapid feedback and evaluation
5. Knowledge about the process and the artefact
6. Information storage and retrieval

Because of the development of computer technology, designers in the manufacturing industry are today using computers instead of drawing-boards to do their design work. However, at present most of the commercial computer-supported tools address specific engineering issues in areas such as simulations, analysis, optimisation and geometric modelling and relatively few applications exist for use at the conceptual design phase.

The reason for this is that knowledge of the design requirements and constraints during this early phase of a product's life cycle is usually imprecise and incomplete, making it difficult to utilise computer-based systems that normally process the well-defined problems. A design concept is often difficult to capture, visualise or communicate electronically among the members of a multidisciplinary design team, especially when the team is geographically dispersed. Conceptual design issues at stake are highly interdisciplinary and often involve collaboration between customers, designers and engineers. Not only is conceptual design becoming more and more central in meeting the increasingly specialised demands of customers, it can also have a powerful impact on manufacturing productivity and product quality, as many manufacturing processes (e.g. moulding, casting, or machining) are indirectly determined at this phase.

Another factor in the conceptual design phase, as experienced by many industries today, is that not only the resources and equipment are geographically distributed, but also the knowledge and expertise [Wang *et al*., 2002].

The popularity of the Internet is largely due to the influence of the World Wide Web, which since its inception in 1989 has made the Internet accessible and available to the mass population [Yi, 2003] powered by the ever-improving information technologies, such as Java, search engines, and e-mail. Through HTML (Hyper Text Markup Language), XML (eXtensible Markup Language), and RMI (Remote Method Invocation), the Web provides another familiar interface and gives a common "look and feel" to information exchange. As the use of the Internet and the Web spreads, and because of globalisation, the paradigm of the design activity is changing drastically. Specifically, there is an ever-increasing need for the continuous collaboration among geographically distributed design teams. The collaborative conceptual design process is physically enabled by the Internet and Web technologies, and functionally supported by the technologies in the domain of artificial intelligence, such as agent technology, knowledge management, and knowledge-based systems.

In general, using the Web in engineering design can bring advantages such as the following:

1. Web client programs (browsers) are available for all popular computing platforms and operating systems, providing access to information in a platform independent manner [Bentley *et al*., 1997]
2. Information involved in the design can be kept constantly up-to-date, such as the latest version of product documents
3. Components from several manufacturers can be found from a single source
4. Designers can download selected components directly into their CAD models

5. Less time spent on travelling, and less time in face-to-face meetings can reduce the product development time
6. The Internet can be regarded as a source that can provide a user-friendly collection of online training facilities, tips and information likely to be of benefit to users [Curry & Stancich, 2000]

By utilising these advantages of the Internet, conceptual design in the manufacturing industry is increasingly becoming an Internet-based distributed process.

The Department of Mechanical and Mechatronic Engineering at Stellenbosch University developed an Internet-based system called DiDeas (*Di*stributed *De*sign *as*sistant) [Schueller & Basson, 2001] that can provide a collaborative environment to enhance the design process in the early design phases for medium and small enterprises. However, it is focused more on supporting the generation of original design ideas and decision making of product development solution than on the know-how of designers in the early design phase. At the conceptual design phase, designers need more information, such as identifying the most up-to-date technologies and an estimate of product cost, and they also need to communicate with the client.

In the conceptual design phase itself, more tools are available for the support of the later part of conceptual design than for its earlier part. The later part represents the boundary between conceptual design and the detail design. In the mechanical domain, the component shape is decided at this later part. Existing commercial tools that support conceptual design belong to this part, and most of the research tools being developed in universities and/or research also support this part.

Wang *et al*. [2002] identified the following directions in which distributed conceptual design research are making progress:

1. System architecture for web-based collaborative conceptual design
2. Collaborative conceptual design modelling and data sharing
3. Product-centric design methodology
4. Conceptual design selection
5. Knowledge management in collaborative environment
6. Intelligent web-based users interface
7. Distributed design project management

The most common techniques used in conceptual design include problem solving strategies, general algorithms, case-based reasoning and agent technology.

The research in distributed conceptual design systems can be classified in the following categories:

Drawing tools: These tools can provide support creating 2D or 3D drawings or in virtual reality. Most of these tools are currently available commercially.

Design data repositories: These repositories can assist designers with design knowledge from existing designs, such as the DARE (design-analysis-evaluation-redesign) model for conceptual design on numerical calculation with symbolic reasoning [Wang *et al*., 1994]; Concept Database [Varma *et al*., 2003], providing a navigation tool though a hypermedia database of linking design concepts, and CODAS (COncept Design Assistant) [Vinney *et al*., 1999], utilising knowledge of past successful solutions to help designers to generate new ideas.

Knowledge-based evaluation: These are domain dependent tools, such as Scheme Builder [Bracewell & Sharpe, 1996] for storing knowledge about the past solutions to enable effective design reuse; and WebCADET [Rodgers *et al.*, 1999], providing designers with feedback about alternative solutions by searching through design knowledge.

Design environment: These tools or systems provide a workspace for supporting the communication between designers as a Virtual Enterprise (VE). SHARE [Toye *et al*., 1993] is an open, heterogeneous, network-oriented environment for concurrent product development, enabling engineers to participate in a distributed team using their own tools and databases. It helps a design team to achieve a shared understanding of their designs and the design process using agent-based computational tools and services, and CoDISS (Collaborative Data and Information Sharing System) [Pahng *et al*., 1998] providing dynamic shared objects that can be built quickly and connected to expert data. As a first distributed collaborative product design system, CFW (CollabFraemWork) [Babu *et al*., 2003] focuses on providing 3D modules for thin clients (client side has limited hardware and software resources).

The researchers in these different categories provide support to designers in a number of ways so that they can work together in different locations. However, designers from different backgrounds need a support tool that they can build up in such a way that they would be able to organise the terminology so that they can improve the communication between them.

2.5. ONTOLOGY

In this section, the definition of ontology and its applications are discussed in general, with the engineering application thereof being highlighted.

2.5.1. The Definition of Ontology

In general, ontology is the branch of philosophy dealing with modelling reality and modelling information system concepts. The definition of ontology differs in different domains:

Firstly, the word "ontology" can refer to two things (http://www.huminf.aau.dk):

1. A study of the subject of the categories of things that exist or may exist in some domain. Thus ontology is here the study of categories.
2. The product of such a study can also be called an ontology.

Several definitions of ontology related to information processing have been formulated in recent years, corresponding to the different fields of application:

Neches *et al*. [1991] originally defined ontology as the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions of the vocabulary. Bernaras *et al*. [1996] extended this definition of ontology to be more specific: ontology provides the means for describing explicitly the conceptualisation behind the knowledge represented in a knowledge base.

Van Elst and Abecher [2002]: Ontology is a collection of key concepts and their inter-relationships collectively providing an abstract view of an application domain. With the support of ontology, users can communicate with each other by the shared and common understanding of the domain.

Guarino *et al.* [1995]: A logical theory, which gives an explicit, partial account of a conceptualisation.

Corazzon [2000]: Ontology is the theory of objects and their ties. The unfolding of ontology provides criteria for distinguishing various types of objects (concrete and abstract, existent and non-existent, real and ideal, independent and dependent) and their ties (relations, dependences and predication).

From these definitions, it can be stated that ontology focuses on using shared conceptualizations, including conceptual frameworks for modelling domain knowledge, while the definitions below list what ontology can do:

Gruber [1993]: An Ontology provides an explicit specification of a conceptualisation. This definition is the most quoted in literature and by the ontology community. Conceptualisation [Studer *et al.* [1998]] refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of the phenomenon. Explicit means that the type of concept used and the constraints on their use are explicitly defined.

Swartout *et al*. [1997]: An ontology is a hierarchically structured set of terms for describing a domain that can be used as a skeletal (wasted) foundation for a knowledge base.

Toye *et al*., [1993]: An ontology is a shared vocabulary of terms and definitions.

In general, ontology can provide a means to use elements, relations and attributes to build information structures for different purposes. Ontology is mainly used for abstracting, understanding and processing information over different domains: a knowledge field can be abstracted into one conceptual framework and then implemented into a concept framework. It contains each of the concept definition, the relations between different concepts and key statements to link relations and concepts. If this framework can be accepted and shared between the experts in this field, and can be implemented in order to be processed by computer, this concept framework can be regarded as ontology.

It is worth distinguishing between ontology and taxonomy: Ontology is more abstract than taxonomy since it focuses on the way to form taxonomy and expand it into more abstract level in order to apply in different fields.

A conceptual graph can be combined with an ontology-based approach and applied in database programming as in this research. Sowa [1992] gave the definition of a conceptual graph as an abstract representation for logic with nodes called concepts and conceptual relations, linked together with arcs. The arc in this definition means an ordered pair <r, c>, linking a conceptual relation r to a concept c. A conceptual graph is a way by which an ontology can be utilised in certain applications such as database programming and will be discussed in more detailed in 4.4.3.

2.5.2. Ontology Applications in Conceptual Design

Ontology is widely used in information representation and management applications, and it can be classified as common ontology, domain ontology, language ontology and formal ontology. It is considered to be essential in two of the currently most relevant research areas: knowledge management and the semantic web [García-Sánchez *et al*., 2005].

In knowledge management, applying ontology in a domain provides an opportunity to analyse domain knowledge, to make domain assumptions explicit, to separate domain knowledge from operational knowledge, and to provide common understanding of the information structure. Most of these applications are attempts to capture and represent the data items that are important to the application domain for which the database is being developed. The problem of database design is, however, that it relies on the user for all the information of that domain [Storey *et al*., 1998].

Sugumaran and Storey [2002] present a heuristics-based ontology creation methodology that can be applied to any application domain. It can guide the user through the process of specifying entities, relationships and attributes, and it can then apply its expertise to detect missing or conflicting data. After that it can express the requirements as a well-formed conceptual model, and transform this into a form suitable for implementation.

Kitamura and Mizoguchi [2003] provided an automatic identification method of functional structure of artefacts from given behavioural models of components and their connection information. In their later work, Kitamura and Mizoguchi [2003] and Yoshioka [2004] have developed a device technology model in which the functional concept ontology specifies the space of functions within the generic functions defined in the ontology. It enables designers to map functional concepts with behaviour automatically and identify a plausible (reasonable) structure from a given behavioural model. This ontology application provided two types of functional models, two types of organisation of generic knowledge and two ontologies of functionality. Kitamura and Mizoguchi [2003] emphasise that the functional ontologies can provide common concepts for its consistent and generic description by applying the device ontology technology.

Several ontology frameworks have been applied in information processing: CYC, KIF, Loom, Ontolingua (China XML, 2007), RDF etc. Since so much research is based on RDF, it is worth considering RDF here. Resource Description Framework (RDF) is a foundation for processing metadata since it provides interoperability between applications that exchange machine-understandable information on the Web. RDF emphasizes facilities to enable automated processing of Web resources. RDF can be used in a variety of application areas (Brickley, D and R.V. Guha, 2007). RDF and XML is the most two popular frameworks in web browsers (China XML, 2007). RDF is presented as an abstract information statement that has no specific ties to any language. It uses resources, properties and statements to form the description of the web resources and is mainly applied in the semantic web.

A semantic web is an extended web of machine-readable information and of automated service [Lee *et al*,. 2001]. The key for designing such a semantic web is on-line ontological support for data, information and knowledge exchange. Given the exponential growth of the information available online, automatic processing is necessary for information managing and maintaining. Used to describe the structure and semantics of information exchange, ontologies play a key role in knowledge management.

Several applications were developed, such as Internet search engines [García-Sánchez *et al.*, 2005], an E-catalogue system, named KOCIS (Korea Ontology Catalog

Information Service) [Lee *et al.*, 2005] for the Public Procurement Services of Korea, category theory that uses ontology to refine the terminology and criteria in semantic web [Robert & Dampney 2005] etc. Ding *et al.* [2002] give a review of the research in the semantic web research field.

2.6. DIVERSITY OF DESIGN METHODOLOGIES

Engineering design can be regarded as the synthesis of new information for product realisation, establishing quality through defining functionality, materialisation and appearance of artifacts, and influencing the technology in the economical and marketing aspects of production. Most of the activities of design and production engineering are the handling/processing of the information. Steven and George [1999] point out that almost 75% of an engineer's design work consists of seeking, organising, modifying and translating information, often unrelated to his/her own personal discipline. Only 25% of his/her time remains for specific engineering efforts. Poor information management causes the problem of concurrent engineering processes and the minimal re-use of information. The information required and produced by a wide variety of design tools should be handled by the information system, even though this information will be in a wide variety of formats. In multi-disciplinary teams, this requirement becomes even more important.

Yoo and Kim [2002] point out that to provide a knowledge management system for seamless sharing of product data in a virtual enterprise, three types of design knowledge are needed: metadata, ontology and mapping relationships. They have developed a web-based knowledge management system to help designers to get a map of product data and to locate the proper information.

Engineering design is a complex process due to a strong coupling and an interdependence between the many processes. It involves expertise and technologies in a wide variety of different fields. The complexity is reflected in the mixture of design procedures and methods that have been proposed or are being used. Each procedure can be related to a particular design model. Cross [1994] gives a useful summary of several methods. The diversity of methods, procedures and models and the variations on each of these will not be considered here in detail.

Basson *et al*. [2003] pointed out two important aspects of the variety of procedures in engineering design that flexible design information systems have to consider. One distinction between different design processes is the extent to which a top-down approach (e.g. "function before structure") is promoted, as opposed to a "cut-and-try" approach (choosing a concept or building a prototype early and then gradually improving it). This aspect is related to Cross's [1994] distinction between prescriptive and descriptive design models. Prescriptive models try to persuade designers to follow a more-or-less algorithmic, systematic procedure as a better means of working. The descriptive models, on the other hand, simply describe the typical sequence of activities, generally reflecting a solution-focused approach where an initial solution is formulated early in the design process and this solution is then gradually improved.

An aspect closely related to the previous one and which is common between all design procedures, is the role of feedback and inevitable revisions. Prescriptive procedures attempt to minimise the revisions (to save time and cost) through structuring the process, but no process claims to eliminate revisions. But there are suggestions for minimising their cost, for instance by putting more effort in the early phases of design and thoroughly analysing the possible concepts (e.g. Ullman [2003]; Bonnema & Van Houten [2003]).

Another aspect of significant diversity is in the overall structure assigned to the design process. To illustrate this, a few well-known references can be compared. Ullman [2003] identifies the following main steps in the design process: Identify need, plan for the design process, develop engineering specifications, develop concepts, and develop product. These steps employ various methods such as functional decomposition. Suh [1990] has a much simpler, but more abstract structure. He considers the design to be a mapping between the "functional requirements" in the functional domain and the "design parameters" of the physical domain, which occurs through the proper selection of design parameters that satisfy functional requirements. Blanchard and Fabrycky [1998] consider the design process for large engineering systems to comprise conceptual design, preliminary design, detail design and development. Each of these parts is divided into five to eight sub-elements.

The differences between the design process structures of the various approaches are in some respects related to differences in scope (level of abstraction and size of the design project), but in many respects just show that there is no single structure that is being used in design processes in general. The commonalities are usually on a more abstract level, and are therefore not obvious to the design practitioners.

The confusion in the terminology used in design process literature is another indication of the diversity in the design process, with the same term used for different aspects in different contexts. To illustrate this, the use of the term "function" can be considered:

Ullman [2003] defines a "function" to be the logical flow of energy (including static forces), material, or information between objects or the change of state of an object caused by one or more of the flows, and "functional modelling" as a decomposition of the design problem in terms of the flow of energy, material and information. Suh [1990] makes extensive use of the concept of "functional requirements", since it is a core feature of his design process. In his approach, the functional requirements form a statement of the design's objectives in terms of specific requirements. This is in contrast to Ullman's [2003] approach where a distinction between requirements and functions is maintained. In Suh's approach "constraints" are distinct from "requirements", but there is no such distinction in Ullman's approach. Szykman *et al*. [2000] even make a distinction between "function" and "behaviour".

Two disparate contexts both using the notion of functions are systems engineering and industrial design. Blanchard and Fabrycky [1998] define "functional analysis", in the context of systems engineering, as the process of translating system requirements into detailed design criteria, along with the identification of specific resource requirements at the subsystem level and below. As with Suh's approach, functions and requirements are not clearly distinguishable in Blanchard and Fabrycky's approach, although they define a function as a "specific or discrete action that is necessary to achieve a given objective". Akiyama [1991] divides functions into "external" and "internal" functions, in which the latter more or less coincides with Suh's and Ullman's use of functions, but the former is closer to Blanchard and Frabrycky's use of the term. Eekels and Poelman [1995], working in industrial design, even use the concept of emotional functions and semantic functions, as shown in Figure 2-10.

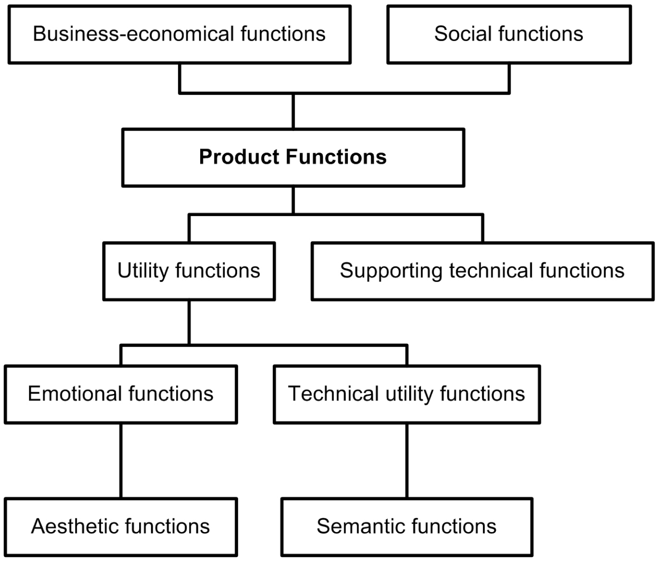

Figure 2-10 A View of Functions in Industrial Design [Eekels & Poelman, 1995]

Another fact is the diversity of product models. Product models are needed to aid the designer in imagining and validating the new design, also to communicate the design details to the designers of the production process [Pels, 1996]. The design models will be validated in different styles due to the complexity and variance of the project.

Lutters [2001] classified five types of product models:

1. Structure-oriented product models: the creation is dependent on the breaking down of the product based on e.g. bill of materials, classification or variant structures

2. Geometry-oriented product models: usually based on computer internal models with the primary purpose of representing the shape of one specific product

3. Feature-oriented product models: this extension of geometry-based models provides the ability to represent shape patterns as coherent geometric objects

4. Knowledge-based product models: based on artificial intelligence techniques, these models allow for the building of abstract taxonomies of products and/or processes as objects

5. Integrated product models: integrated models cover the abilities of the other models emphasising semantic integration

These design models can be found in many design products in different domains. They assist the designer to understand the structure of the design product, and even more than one of such a model are required for one product. This also raises the question of which product model should be chosen for illustrating the product system. This system must also be scalable and support subdivision of the design tasks at the requirements and specifications development stages. The general process of a synthesis-analysis-evaluation loop should be repeated and should cover each level of the project design. It should have the ability for subdividing the design problem into subtasks, which can be handled concurrently by distinct teams if need be. To ensure that the overall objectives are met, each subtask must depart from a "baseline" common between all subtasks and the evaluation activity should determine to what extent the solution synthesised for each subtask meets the requirements set in the baseline. In addition to reference to the baseline, some subtasks will have to interact with other subtasks. The design information system should therefore support the management of the baselines, as well as the interfaces between a subtask, the baseline and other subtasks. The user interface should therefore provide the designers with a clear view of the project structure with the requirements and specifications linked to each level of the system and also to the levels of the subsystems.

A diversity of design methodologies and design modes are available in various companies, especially in the small and medium enterprises. Design information systems that aim to be generally applicable cannot enforce a prescriptive design model or any particular overall structure. It must allow the team to use the design procedure and terminology most appropriate to the particular context, whether top-down, bottom-up, or a combination of the two. It must facilitate frequent revisions of previous design decisions, but in a controlled and traceable manner. There need interpret a flexible user interface so that a company can adapt the terminology that used in the company on this interface.

3. CASE STUDIES ON DESIGN INFORMATION DURING CONCEPTUAL DESIGN

3.1. INTRODUCTION

In this chapter, the general design information management of the early design phases will be discussed first with reference to four case studies, i.e. a student project and the design projects undertaken by three industrial companies. Different information representation formats are also illustrated. The diversity of the design process as well as an approach to handle it in a design support system will also be addressed.

3.2. SPECIFICATION DEVELOPMENT AT EARLY DESIGN PHASES

Design is a procedure during which information is searched for, retrieved, transformed, transported, represented and interpreted. Ullman [2003] pointed out that three kinds of documents are produced during the design process:

1. A design record
2. Communications between designers and managers
3. Documents communicating the final design

Court *et al*. [1998] report that in a survey of designers in the United Kingdom and the United States of America it was found that 50% of design decisions are recorded in hard copy design notebooks. The other major decision repositories were found to be diaries, memos, reports, general notebooks, data/calculation sheets, project files, contracts, design documents and drawings. These repositories need to be managed by an information management system in a systematic way so that it can be clear how and why these decisions were made. However, the design know-how used in the conceptual design phase, including the functional knowledge, is usually left implicit in spite of the fact that the advancement of computer technologies has enabled easy access to information related to structure and/or shape of artefacts. General knowledge, the so-called "design catalogue", also focuses mainly on the mechanism concerning shape and link. Even if such knowledge is found in the documents, it is often scattered around technical domains and is improperly categorised.

A requirement specification could be a description (usually a list) of the desired properties from the viewpoint of the behaviour of the product. In product development, the specification can be seen to be a description of a goal to achieve (i.e. to develop a product, or to solve a problem) while considering certain limitations. The overall goal is to create a product that suits the market strategy of the business. The specification will guide the designers towards the correct solutions and provide the base for solution verification. During the conceptual design phase, the project needs or the customer requirements and the design specifications are regarded as a statement of the input and output of the system or subsystem being designed. It is necessary that this information is handled systematically.

Four case studies were carried out to identify how the requirements and specifications would be developed, and the difficulties encountered in developing these specifications. These studies will be discussed in the following section.

3.3. CASE STUDY OF THE STUDENT PROJECT

3.3.1. Case Study Purpose

The aim of this case study was to identify how the design specifications were handled by the participants in a design project as part of a course, the factors affecting these participants during their design process and the difficulties they encountered in handling specification development.

3.3.2. Case Study Setup

The whole design process lasted a month and the scheduled meetings were observed and recorded by a video camera. All the documents exchanged between the participants such as e-mails, drawings and notes were collected for analysis. At the end of the study, each participant had to complete a questionnaire (Appendix B) and was interviewed individually. These interviews were audio-taped in order to ensure accuracy of the data.

3.3.3. Participants

All the participants were fourth-year students. They were required to design a mash cooler system for a winery in a design course. The design team consisted of six members and they were familiar with Ullman's [2003] design methods which were presented to them in a preceding design course at the Department of Mechanical and Mechatronic Engineering at Stellenbosch University. Except for the one that came from the Netherlands, the students were from South Africa. The foreign student had a different educational background. Because of this, the case study could also be used to observe some of the effects that group members with different backgrounds and fields of expertise would have on a design group.

3.3.4. Design Process

At the beginning of the project, the customer requirements were given by the lecturer. The participants were required to complete the customer requirement refinements and the specification development. They had to do the concept generation and the decision making, and at the end, they had to submit a report of the specifications of each subsystem as well as the system layout drawings.

3.3.5. Case Study Results

This design was divided into subsystems that different people worked on, with the result that the communication between different participants became very important. As the different subsystems were developed separately, reporting of each subsystem's development was the first issue of the regular meetings, followed by requirements and specification modification negotiation as the main aim of the meetings. It was clear that keeping track of the team members' progress and sharing with all team members is an essential issue for improving the progress of a design team.

Group meetings were the main team activity, where the participants exchanged their design information and ideas. These meetings can be divided into two types: class meetings and private meetings. The private meetings were the meetings which were arranged in the participants' spare time and there were no fixed schedule and normally no fixed place. However, in order to inspect and record these private meetings, the meeting place was fixed. The class meetings were the meetings in the classroom that the lecturer would join and where he would give advice. During the project, there were six private meetings and six class meetings in total.

This project took 26 day and the participants had a total of 8 meetings. Figure 3-1 illustrate the time that the participants spent in each of these eight meetings. It can be seen that the students spent about 170 minutes on private meetings and 63 minutes on class meetings. About 62.9% of the total meeting time was therefore spent on private meetings. From the interviews with the students, it was gathered that they all regarded the purpose of the class meetings to be for the exchange of the design parameters and for getting advice from the lecturer. Little idea generation was achieved in the class meetings due to environmental effects. When using a distributed design system, such meetings will be more focused on problem solving and less influenced by environmental effects such as the attention of the participants being distracted by the noise from other group discussions and the participant's feeling of pressure. The designers also spent time outside the meetings on calculations and function analysis but that was not recorded.

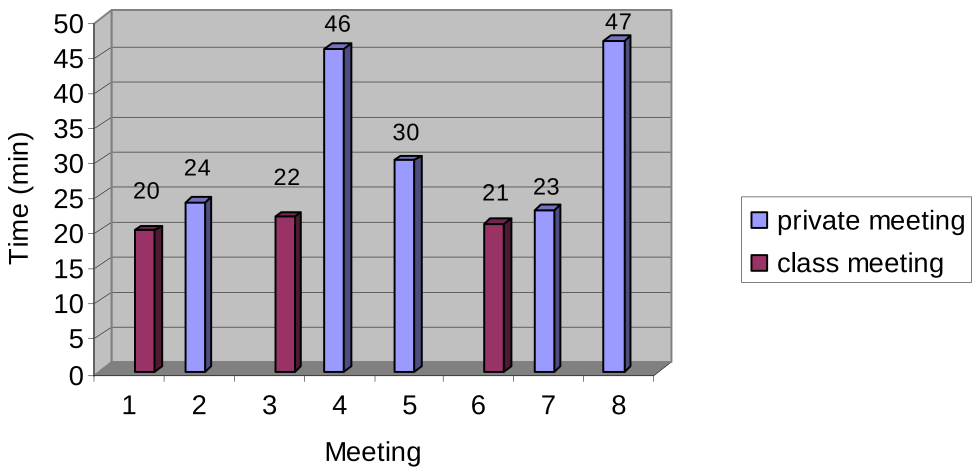

Figure 3-1 The Meeting Time and the Meeting Date

Figure 3-1 sets out the time that was spent on each meeting. From this it is clear that the meetings were longer in the middle of the design process and at the end. The longer meetings in the middle of the process can be attributed to the fact that the team had to resolve the interface problem between different subsystems of the design project after all the design parameters were set, and the longer meetings at the end indicates that they had to finish the project by a certain deadline.

From the author's industrial work experience, the time duration corresponds with that of designers working in a commercial company. In order to improve the design quality, the early design information needs to be clear and exchangeable so that the "midway meeting" can be more effective. The "end meeting" was longer because the designers felt the work pressure and tried to finish the project by the deadline.

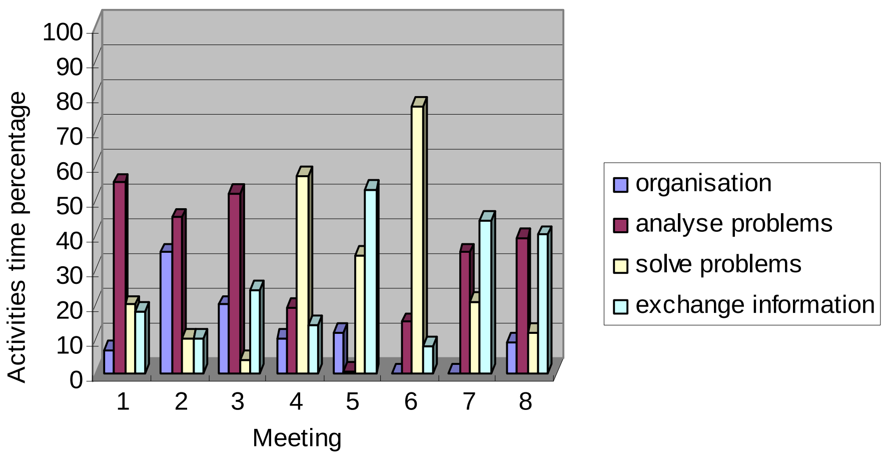

Figure 3-2 The Activity List of the Meetings

For each meeting, the activities of that meeting, according to duration and content, are illustrated in Figure 3-2. The vertical axis shows the consumed time percentages of the four major activities (other non-academic activities are not accounted for) that were performed in these eight meetings:

- Organisation: the team leader assigns designers to be in charge of the different subsystems, schedules the meetings, clarifies meeting topic, etc.
- Analyse Problems: the designers discuss the problems encountered in satisfying requirements and with interfaces between subsystems
- Solve Problems: a group discussion to solve problems such as resolving conflict about subsystem interfaces

- Exchange Information: the designers report their work progress at each meeting; the parameters and requirements of different subsystem interfaces are presented

These four activities consumed time percentages as shown in Figure 3-3. The project customer requirements and the design parameters were established in the first two meetings. Due to the given design parameters from the project requirements, the design group had to select different products from different quotations in these regular meetings, and their selection was dependent on their knowledge and design experience.

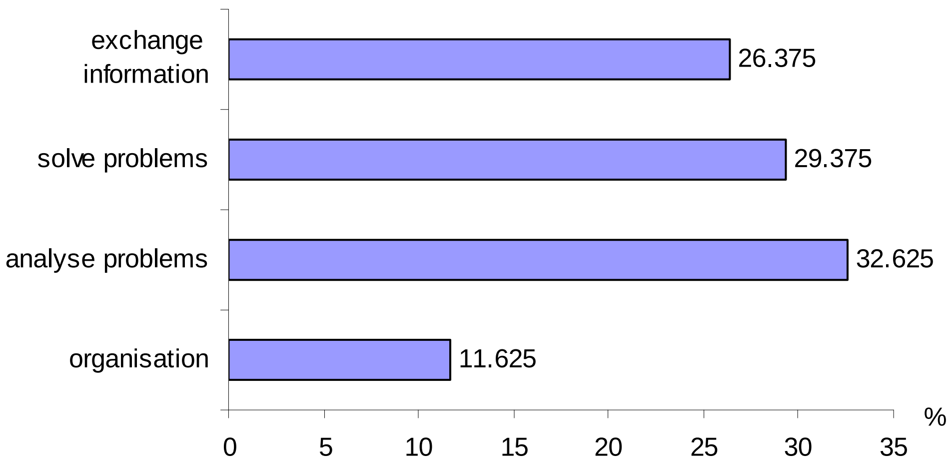

Figure 3-3 Percentage of Total Time Spent on Four Activities

From Figure 3-3, it is clear that most of the activities were geared towards Analyse Problems. Information Exchange also took considerable time, but Analyse Problems sometimes included elements of Information Exchange as well, e.g. when details about a team member's progress were mentioned during the discussion. Even though different designers progressed at different rates with their subsystem designs, some key interface parameters (roughly 15%) between subsystems had to be reconfirmed at beginning of each meeting. This took the main part of the Information Exchange time. If the time spent on the information exchanging can be reduced, the design effectiveness can be improved.

One of the team members was assigned as a secretary to keep minutes so that the results of each meeting could be recorded. However, the minutes only showed modified requirements or specifications of subsystems since the secretary participated in the discussions. Notably, however, how and why the specifications were modified, were not recorded in the minutes.

It was observed that most of the specifications came from the brainstorming discussions. The brainstorming methods were heavily dependent on the participants' design experience. About 36% of the consumed time was used to analyse new specifications, which were not identified at the beginning.

It was also observed that participants spent a considerable amount of time searching for various manufacturing companies and suppliers, in able to obtain component quotations. However, some of this information could have been obtained from previous projects. Interviews conducted with the team members show that considerable time would have been saved if the participants had access to a list of manufacturing companies and suppliers that are related to the project.

Seven e-mails, two AutoCAD drawing files, one Microsoft Excel file and five Microsoft Word documents were collected. These documents contain different design information and indicate that different file formats need to be handled in a support tool.

From the interviews it was clear that textbooks and design catalogues were the major sources that helped designers to solve the problems especially during specification development.

An important factor emerged in this case study: the minutes of meetings did not record all the reasons for modification of system requirements and specifications and even some modifications were not recorded. The completeness of the record of changes was highly dependent on the secretary's style. The incompleteness of the design records severely limits its suitability for design reuse.

In summary, this case study provided the following insights:

1. A clear presentation of requirements and specifications is required to improve the design efficiency
2. In addition to text format, the design information can be in the form of sketches, drawings, analysis reports, etc.

3. The information of different manufacturing companies and suppliers should be collected so that it can be reused in other projects.
4. Keeping record of design modifications could potentially prevent the same mistakes being made again and improve the design efficiency.
5. A checklist can be provided from textbooks, design catalogues and other design resources. These resources should, however, be at an abstract level to allow the opportunity for brainstorming by the design team members. The resources can also be in line with a company's design style.

3.4. CASE STUDIES IN INDUSTRIAL COMPANIES

3.4.1. Introduction

Three companies were involved in the case studies: CAE (a division of Stellenbosch Automotive Engineering), TF Design (Thermodynamic Fluid Design) and Molenaar. All three companies are located in the Western Cape in South Africa.

3.4.2. Case Study in CAE

3.4.2.1. Company Profile

CAE (a division of Stellenbosch Automotive Engineering (Pty) Ltd) is a spin-off company of the Department of Mechanical and Mechatronic Engineering at Stellenbosch University. It provides research, design, development, prototyping and testing services to the automotive industry. CAE conducts and supports projects in which new automotive products are developed and manufactured. Capabilities include the development and evaluation of petrol and diesel engines, the development and evaluation of fuel and lubricant products, and the development and manufacturing of specialised testing equipment for the automotive industry.

3.4.2.2. Case Study Scope

The aim of this case study was to identify how design specifications are handled in a commercial company. In the case study, the focus was on a project to redesign a marine engine for a high-speed boat. The whole design process lasted three months. It was, however, cancelled by the client during the conceptual design phase. This case study focused on the design team's experience during specification development. No detailed project information may be reported in the case study due to confidentiality requirements.

The team used DiDeas I (Appendix A) as an aid to generate design documentation. This case study therefore acted as a way to determine further development directions for DiDeas I.

3.4.2.3. Case Study Method

The technical documents could not be analysed, due to the confidentiality requirements of the contract. Each designer completed a questionnaire and was interviewed. A tape recorder was used to tape the interviews to ensure the accuracy of the data. The detailed questionnaire is presented in Appendix B.

3.4.2.4. Case Study Participants

All five participants were designers working in the same office. Four of them had more than one year's experience in mechanical design work. Two of the designers had been involved in design work for five years, and there was one participant at entry level. Four of them were familiar with Ullman's design methods and one was familiar with VDI. Four of the participants had prior-experience with using DiDeas I.

3.4.2.5. Results

Based on the questionnaires and an interview with the design manager, the following steps for the design process could be set out:

1. Customer requirements identification and refinement
2. Development of overall technical specifications
3. Division of the design problem into sub-problems according to the system functions, and assignment of these sub-problems to different team members
4. Development of possible solution principles for sub-problems individually
5. Evaluation of these possible solutions and their integration into the overall solution system

These steps were generalised from the designers' own points of view. In fact, the project was more specifically aimed at engine development instead of being set out in such general terms. It also brought to the fore the problem that designers could have with communication as the one designer had a different understanding of some technical terms at the beginning due to his different background.

Another observation from this case study is that the product designers in CAE would normally have more responsibilities than their counterparts in big companies: they would not only be responsible for product development but also for aspects such as marketing and product support. This meant that the design progress of different designers were not the same. From the interview during the specification development phase, it was clear that the communication on the work progress between team members was the main aim of their group meetings. The participants got about 40% of their design information from their own experiences from past projects. This is an indication that finding a solution for many design problems can be seen as researching the solution sub-space near existing designs that are close to the current design problem. From the point of view of these experienced designers (participants), such a case-based approach is important in mechanical design where there does not have to be a direct mapping from the functional requirements to the system components.

There were only two participants who used the checklist that was provided by DiDeas I. The rest of the participants felt that the checklist was too abstract for them, and some of it was not relevant for their project and distracted them from the task at hand. They preferred using design catalogues or their company guidelines.

It was concluded from this case study that a design information system was required that could provide the designers with the opportunity to organise the steps with their own words and to form their own design guidelines. This system should provide an interface to show the dynamic system structure of the project that different team members work with. It could improve the design efficiency and is a necessity for a distributed team. This design information system should have the potential to provide case-based knowledge that can assist the designer to solve design problems. Such a design information system, should also have the ability to be extended to include a company-defined guideline.

3.4.3. Case Study at TF Design

3.4.3.1. Company Profile

TF Design is a product design and development company in the fields of heat transfer and thermodynamics, general mechanical design and electric/electronic control, employing about 15 engineers. The company is involved with a wide variety of products, including timber drying kilns, materials handling equipment, machine vision applications, and PLC and PC based control systems. TF Design provides engineering services such as CFD analyses of flow phenomena, hydraulic analyses of pipe systems, specialised design and project management.

3.4.3.2. Case Study Scope

This case study focused on a heat exchange system upgrade project that was carried out by TF Design for a Mozambique company. Three companies were involved in this project, and this provided the chance to analyse a common case in a commercial product design company.

The project manager was interviewed but there was no opportunity to interview any of the other designers. The project documents were collected and analysed.

3.4.3.3. Case Study Objective

The objectives of this case study were to analyse the design documentation of TF Design in order to determine the related requirements for DiDeas II.

3.4.3.4. Results

From the interview, it was found that for this company the model design process was focused more on identifying problems and redesign than on innovational design. The design approach in this company is heavily dependent on the character of the project, and there is no fixed approach. The design process is usually also different from other companies. A design information tool to support such a company must therefore cater for the company's specific design approach and be easily adaptable according to the needs of the project.

The design documents collected from TF Design were faxes, e-mails, an analysis data sheet, a technical report, meeting notes, and drawings. It consisted of a total of 102 hard copy files - the composition of these design documents are given in Table 3-1.

File format	Percentage in total documents	Technical information percentage
Faxes	68%	22%
E-mails	15%	20%
Technical report	5%	100%
Meeting note with sketching	5%	100%
Financial letter	3%	0%
CAD Drawings	2%	100%
Analysis data sheet	1%	100%
Contract document	1%	60%

Table 3-1 TF Design Project Design Document List

It was found that 27% of the design documents contained technical information. However, about 63% of these documents were used for negotiation between the manager and the client. The CAD drawing files contained the assembly drawings, and the analysis data sheets the results of analyse using FEA (Finite Element Analysis) software. The contract documents gave a view of the design task. The faxes, e-mails and meeting notes contained no index information and were not clear for somebody else to understand. It was very difficult to clarify how the customer requirement refinement and design specifications were generated. This information would have provided a useful design repository for similar design. Recording the customer requirement refinement and design specifications is one of the roles that DiDeas II can fulfil. Some of the documents were technical support results.

From Table 3-1, it is clear that the faxes and e-mails were the main formats that were kept in the design documents. This can be attributed to the fact that the project involved one company which was located in one place and another company which was located in another place. About 20% of the e-mails were sent to ensure that the faxes were in fact received, these e-mails did not attribute to any design work and were used for business purposes.

The product purchase documents, such as product quotations, product descriptions and contact information, were also an important part of the design documents. From the interview, it was gathered that working with product information took about 26% of the design time.

Most of these design documents were kept for business purposes, and it was very difficult to build a design repository based on such limited design information. From the interview, it also became clear that there were several reasons why it was difficult to record the design process: the team had quick impromptu meetings instead of regular scheduled meetings and these quick meetings normally have few design records. The designers were distributed in different work sites and most of the conversations were carried out via telephone, which brought the problem of recording the design information. The recording of the results of meetings was dependent on a designer's personal habits.

Some design modifications were found in the design documents, but since the documents were not kept in a systematical order and lacked an index, they could not provide useful design information to indicate how and why the changes had been made.

From this case study it can be concluded that small companies have a problem to keep design records due to their limited human resources. Many of the records do not provide sufficient design information to build a repository or contribute to the understanding of other designers. A design tool should therefore include functionality that helps a designer to record more complete design information especially at the conceptual design phase, which has few other support tools.

3.4.4. Case Study at Molenaar

3.4.4.1. Company Profile

H.G. Molenaar & Co. (Pty) Ltd. is a South African based company and operates as manufacturer, engineering and equipment supplier in the food, wine and beverage industry. It can provide upgrading and refurbishing services to food processing factories or it can build machinery from scratch. Its primary expertise lies in the fields of fruit and beverage engineering, but a recent project was in the field of meat processing in the form of pet food.

3.4.4.2. Case Study Scope

A project entailing the design of rotary pressure cookers and cooler systems was used for the case study. This project was a complex system compared to the previous case study project. The approximately 1024 paper files were kept in the following folders:

1. Contract documents (such as company proposal, contract item list, etc.)
2. Financial documents
3. Purchase documents (contain product description and product quotation, etc.)
4. Client communication documents
5. Drawings
6. Technical reports (such as analysis report, product purchase such as proposals etc.)
7. Notes

To stay within the scope of this research, only the documents that were relevant to conceptual design were focused on. Two interviews were conducted with the design manager of the project.

3.4.4.3. Results

The contents of the design files were classified as notes, drawing files, financial documents or contracts between different companies. Although the design files were collected in such way, it still could not be assumed that they were kept systematically since it was difficult to find the links between the different document files. An assembly drawing provided the chance to clarify the structure of system and subsystems, but it was still difficult to identify the link between the requirements and specifications that were contained in these documents. This brought the problem of understanding how and why the design specifications were developed. Many documents came from the outside companies that participated in the project, but these documents only contained the solutions of the subsystems they were in charge of; there were no information that indicated how and why these results were achieved.

The documents all came from different departments and from different external companies located in different offices and places. The design process could be regarded as a distributed design environment linked with internal and external communications. From the interview, it was gathered that the reason for most of these communications was to keep track of the design process that was carried out in different departments. The design efficiency was decreased due to the delay in these communications.

The design reviews of some design phases were found to be contained in the design notes. However, the design reviews of all phases were not found in these notes. From the interview, it was gathered that the design review format was dependent on the project manager's personal habits. The information reflected in the design review mainly represented the design progress. There was limited information that indicated the design modifications. The design modification could however be found by comparing the different notes and the design process reports, but the documents without index made it difficult to find the documents that belong to the different design phases.

From the many design sketches in the design files, the characteristics of the sketches in this case study could be listed as:

1. Immediacy and speed of capture of visual representations of ideas and concepts
2. Usage of spatial structure to represent relationships
3. Evolution of form by facilitating cognitive activities of restructuring and combining

All the sketches were found to be contained in the design notes. These sketches had no links to subsystems and it was difficult to determine which subsystem a specific sketch belonged to. This limited their value, although they had the above mentioned beneficial characteristics.

In conclusion of this case study, it can be noticed that due to the project design having different stages of progress in different the subsystems, a design information system is needed to handle the different levels of the subsystems. The reuse of past solutions is an essential feature of an effective design system. The knowledge that is stored, and past successful solutions that are readily accessible, are the key to ensuring that the new problem can be solved effectively and that the solutions generated are likely to be of practical significance. An information system that can systematically provide shared different design files, such as design reviews and sketches, etc., to all the designers would therefore be of value.

3.5. CASE STUDIES SUMMARY

Small and medium enterprises are usually characterised by well-established knowledge of specific components but, usually, do not have all the capacities internally to develop a complete product and to lead a whole project. It is in their nature then to adopt co-design principles while maintaining full autonomy [Giannini *et al.*, 2002]. Co-operation with other enterprises working in the same industrial sector has become part of the small and medium enterprises design activities and the co-operation implies not just a simple made-to-order development but also a real collaboration among the companies in the definition of the new product.

Although these case studies were carried out in only a few companies instead of a large number of companies, the results could still indicate the directions in which a design information system for small and medium enterprises should develop:

1. An information system for small and medium enterprises should support distributed design. These companies do not have the complete knowledge for developing all the components and need to outsource the development of specific parts, and this often includes cooperation between different companies.

2. Since the designers work on various system/subsystem levels at various times, as shown by the disorganized assembly of the design documents in the case studies, the design information system must be able to handle different levels of design information systematically and it must allow the designer to enter or find information at any level of the system/subsystem hierarchy.

3. The provision of a systematic representation of design requirements and specifications is needed. It must be clear and be shared by all the design team members.

4. Different formats of design information such as notes, design reviews and sketches must be handled in a systematic manner.

5. Design information history should be recorded, in particular when modifications are made.

4. DIDEAS II DEVELOPMENT

4.1. INTRODUCTION

In this chapter, the DiDeas I (Distributed Design Assistant), an Internet-based distributed product development system, is introduced briefly and the directions in which DiDeas I can be developed are presented. Ontology technology is further illustrated, and the way in which it can be applied in the extended development of DiDeas (called DiDeas II in following section) is discussed. The DiDeas II database that is built by using ontology is described in detail.

4.2. DIDEAS I

This section describes DiDeas I as it was developed by Andreas Schueller [2001]. This is the starting point for the work presented in this dissertation.

DiDeas I is an Internet-based system that allows simultaneous multi-user collaboration in the early design phases. It aims to provide a collaborative environment to enhance the design process in the early design phases for medium and small enterprises. Also, the DiDeas I intends to support project reuse and negotiation activity by automatically keeping track of all the modifications proposed. Since most of the new projects and decisions are based on knowledge from previous experiences, it is important to define methods to maintain and support historical data.

A relational database is located on a central web server and stores all design information entered into the system. The user interface was realised in the form of a collection of Microsoft Active Server Pages (ASP), which can be accessed platform-independently via a standard web browser. A DiDeas I interface screenshot is shown in Figure 4-1.

DiDeas I classified the design process into the following phases:

1. Specification development
2. Functional analysis and concept generation

3. Concept selection
4. Concept evaluation

During each phase, the designer can input textual design information into the database in a server through the web browser. DiDeas I guides the designer to go through these steps of the design procedure and enables him to create/manage the design information in a concurrent environment. A detailed description is given in Appendix A.

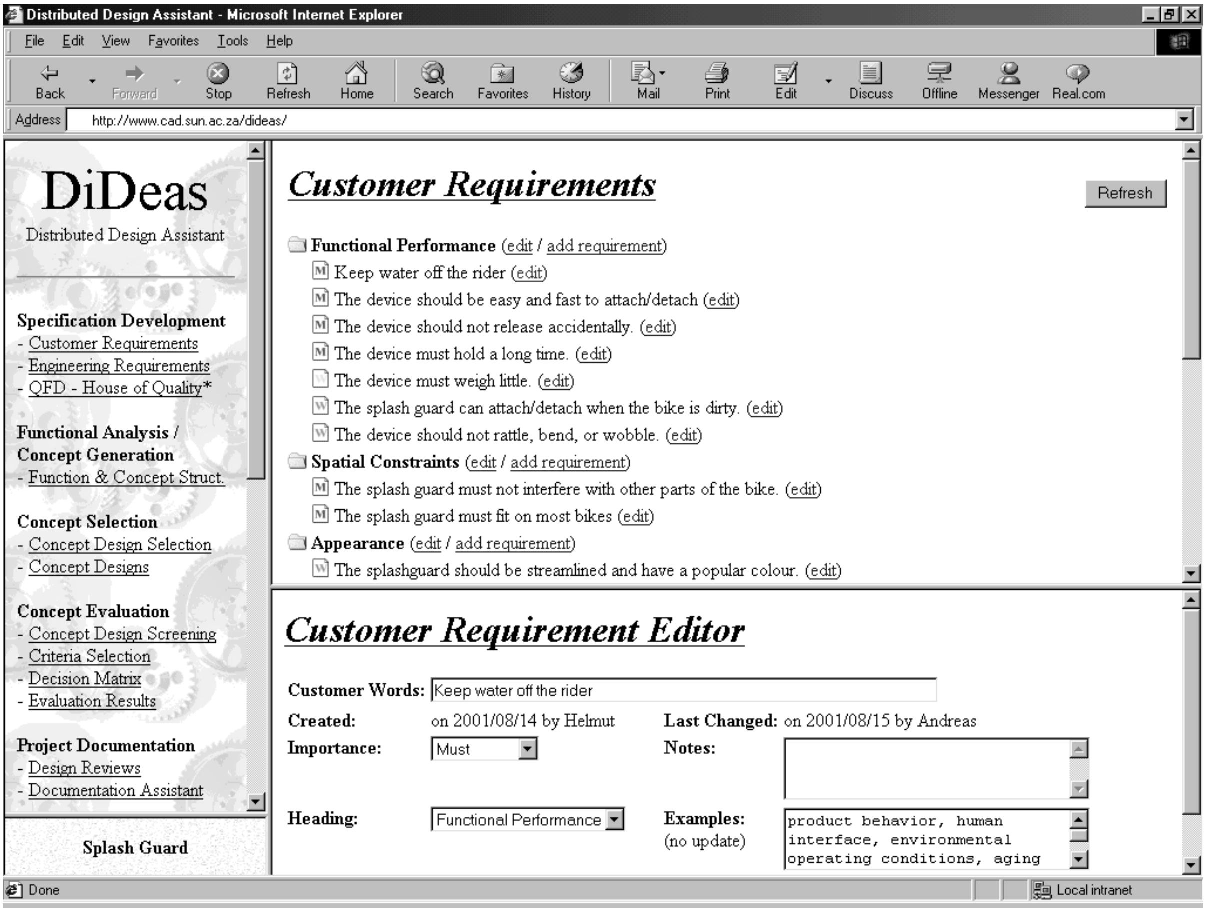

Figure 4-1 DiDeas I Interface [Schueller, 2001]

4.3. DIDEAS II DEVELOPMENT DIRECTIONS

4.3.1. Systematic Product Representation with the Link of Requirements and Specifications on Different Levels

For a general design process, Ullman [2003] identifies the following main steps; identify need, plan for the design process, develop engineering specifications, develop concepts, and develop product. These steps employ various methods such as functional decomposition, etc. Suh [1990] has a much simpler, but more abstract structure. He considers the design to be a mapping between the "functional requirements" in the functional domain and the "design parameters" of the physical domain, which occurs through the proper selection of design parameters that satisfy functional requirements. Blanchard and Fabrycky [1998] consider the design process for large engineering systems to comprise conceptual design, preliminary design, detail design and development. The differences between the design process structures of the various approaches are in some respects related to differences in scope (level of abstraction and size of the design project), but in many respects just show that there is no single structure that is being used in design processes in general. The commonalities are usually on a more abstract level, and are therefore not obvious to the design practitioners.

Ullman [2003], Pahl and Beitz [1997], and Ulrich and Eppinger [1995] all pointed out that the first step in solving a design task is to divide it into sub-tasks. Product information should therefore then be stored on the different levels associated with each sub-task, especially the specifications in the conceptual design phase. These sub-tasks are not independent, but are linked to each other. The design information system should therefore support the management of the baselines, as well as the interfaces between a sub-task, the baseline and other sub-tasks. It is important to note that a sub-task does not necessarily coincide with a subsystem or module.

Blanchard and Fabrycky [1998] classified baselines as follows:

- functional baseline: it is established following a review with the customer of a system specification. The system specification is then decomposed and allocated to lower level elements of the system, normally assigned to software and hardware configuration item specifications. These lower level elements comprise the allocated baseline.

- developmental baseline: it is established during top-level design. This baseline is maintained to ensure control of the internal development cycle and to incrementally achieve interim milestones.
- product baseline: the product baseline includes engineering drawings and their complementary documentation. For software, the product baseline includes the source code, compilers, and tools that assist in creating the software.

A functional analyses of the design process, taking the role of baselines into account, lead to the general process illustrated in Figure 4-2. When a designer works on a project, the overall function is identified first and then the objectives are formulated, followed by the generation of alternative concepts. Each of these alternative concepts will be divided into sub-concepts/subsystems. When designers work on each of these sub-concepts/subsystems, the design process will be repeated from identifying the function of the sub-concepts/subsystem until the sub-concepts/subsystem solution is achieved, such as purchase/subcontract decision can be made or when sub-concept detail design occurs. Figure 4-2 shows two typical design activities (component design, layout design) in sub-concept detail design, but there are many more possible design activities dependent on the design project and design resources.

The functional baseline (refined objectives) is the pivot for handling the system and subsystems. However, the baseline management from the system view is rarely encountered in recent research.

When a subsystem has more than one parent subsystem (note that some subsystem requirements and specifications are inherited from a parent system) then it is necessary to create a "duplicate" of the design information in a "one-to-many" information presentation. To ensure the consistency of the data, such duplicate entities will be highlighted using a different display format and when a designer edits or deletes them, a warning message should appear so that such operations can be confirmed.

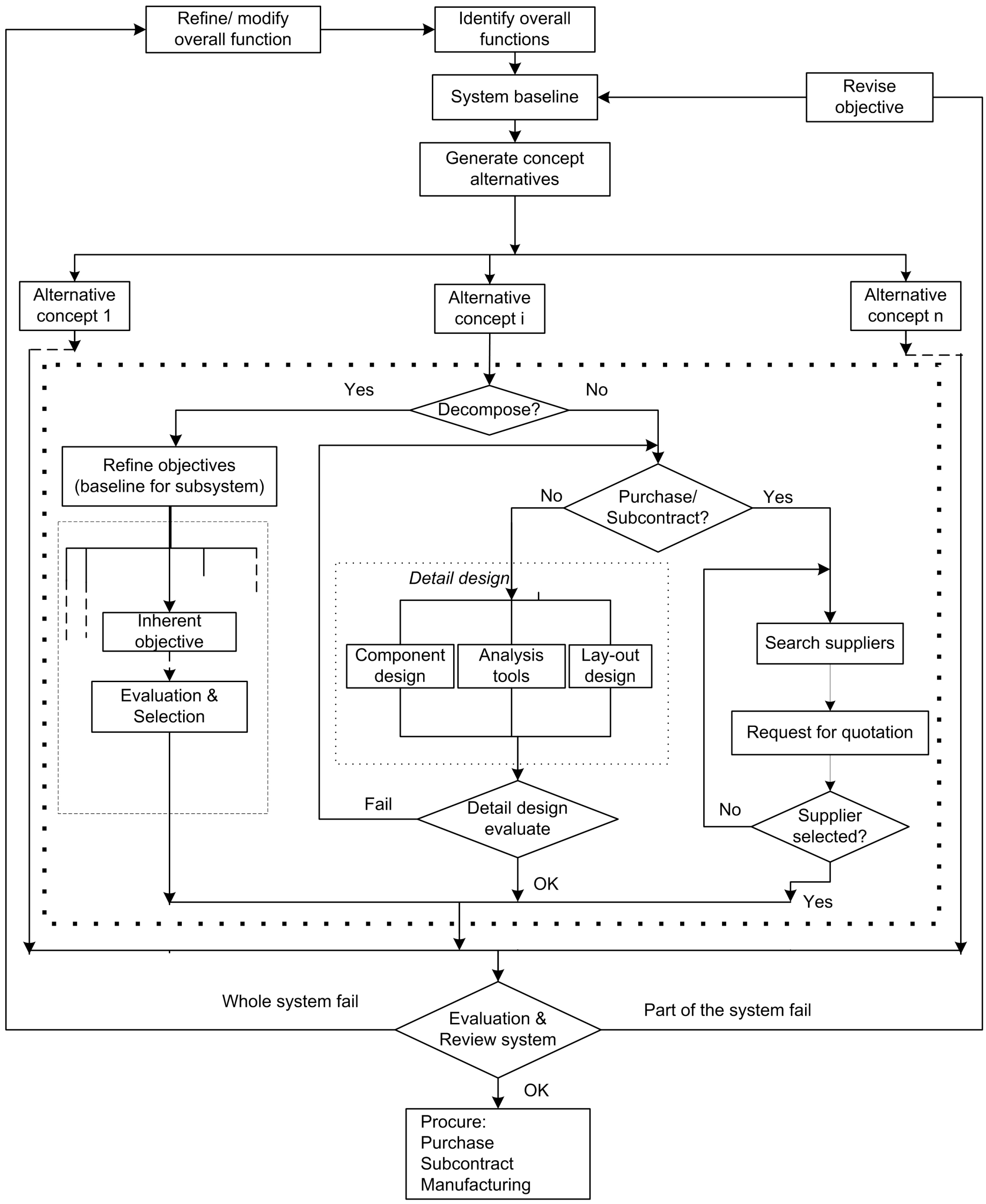

Figure 4-2 General Design Process

DiDeas I can only handle the requirements and specifications for the development of one component; it lacks the ability to handle both a system and a subsystem in a clear way. From the case studies presented in Chapter 3, the following objectives for handling requirements and specifications were deduced:

1. Clear, consistent and verifiable in a systemic manner
2. Traceable to an identified source requirements
3. Being integrated in the system and ensure that all product function performance are satisfied

From case studies in Chapter 3 it can be concluded that in many companies much of the design information are currently not being captured in structured or systematic ways and thus can only provide limited benefits to the companies. However, the final product definition is always presented in detail and fully structured because this final product is the main output of the design process. This phase is also assisted by many available tools such as CAD and CAE. It is clear that DiDeas II can be developed to handle the requirements and specifications of different systems and subsystems at different levels. DiDeas II also must provide a clear view of the structure of the system being designed with linking of requirements and specifications. This can help the design team to complete a design project in a systematic manner.

The DiDeas II must also be scalable and must support subdivision of the design tasks at the requirements and specifications development phases. The general process of a synthesis-analysis-evaluation loop should be repeatable and should cover each level of the project design. It should have the ability for subdividing the design problem into sub-tasks, which can be handled concurrently by distinct teams if need be. To ensure that the overall objectives are met, each sub-task must depart from a "baseline" common between all sub-tasks and the evaluation activity should determine to what extent the solution synthesised for each sub-task meets the requirements set in the baseline. In addition to reference to the baseline, some sub-tasks will have to interact with other sub-tasks.

4.3.2. Flexible User Interface for Diversity of Design Processes

As discussed in section 2.6, the diversity of design approaches, terminology and design models bring certain difficulties of communication in a design team which has team members with different education backgrounds. Building a “customer-defined” flexible user interface can allow users to use their own agreed design approach and terminology to form a base of shared understanding and thus to improve communication.

4.3.3. Design History Records

When a design engineer generates a set of prototype solutions, one important factor is that requirements or specifications must be negotiated in the conceptual design phase. As a result, some requirements or specifications will be abandoned. This deleted design information should be stored for potential future use. A designer can prevent the repetition of mistakes in a current design project by going through the design history of a previous design project. In this sense, the design history forms a design repository and is especially useful for the novice designer. In a small company, the designer would normally have other responsibilities as well, such as marketing, technical supporting, etc. He/she will then be more concerned with the main output of the project (such as drawing files) instead of how and why the specifications are generated. To form such a design history, the system will have to keep the deleted requirements and specifications in the database, and this information should then be available on request. In DiDeas I, the recorded design history distracts the designer because it takes up too much space on the limited view space of the computer screen. This history display should be shown in a controllable manner.

Applying ontology to information management in the conceptual design phase can bring big advantages to the designer to obtain the design knowledge (including the deleted design information) and reuse them in future design work. It can also apply in a distributed design environment, and it can extend the design knowledge resources in cases where there are designers from different domains.

4.3.4. Interface Utilities

Ease of use is an important factor for a successful system. Due to the limited programmability of HTML, the user interface of DiDeas I can provide limited functionality for design information management. More facilities are needed by the designer to create, edit, delete and maintain the design information in the system. A good example is the display of the QFD (Quality Function Deployment) information in DiDeas I. It is limited by the size of the computer screen, which limits the numbers of the engineering requirements (see Figure 4-3). It is not easy to establish the links between the customer requirements and the engineering requirements in this QFD page. These shortcomings necessitates recoding the system in another programming language, the detail of which will be described in Chapter 5.

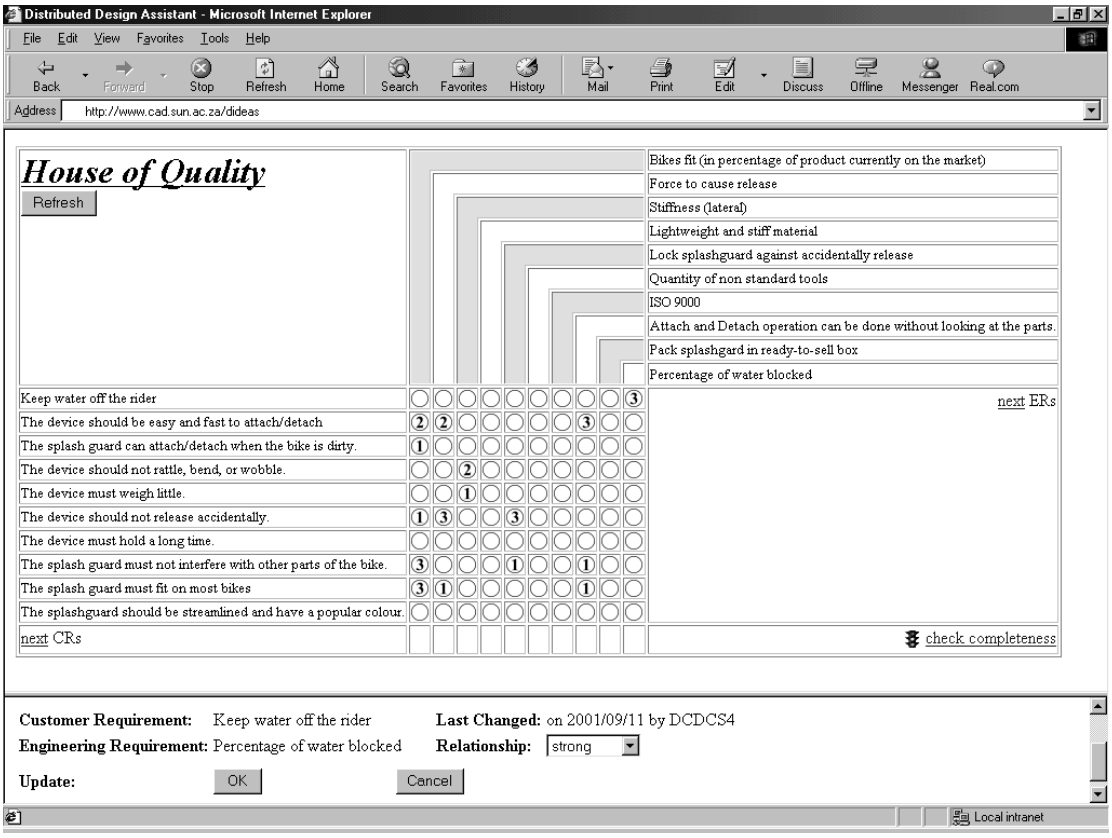

Figure 4-3 DiDeas I QFD View [Schueller, 2001]

4.4. DATABASE DEVELOPMENT USING ONTOLOGY

4.4.1. Design Information in Conceptual Design

Design can be viewed as a process of decision making, and design information can then be viewed as the knowledge that helps to make these decisions. In the same way, design information can be defined as the information that is used in generating, quoting, or in discussions in the process when a concept is turned into reality [Ulrich & Eppinger, 1995]. The design information usually includes different types and formats – in addition to regular text and drawings, design information can be represented as tables, graphs, pictures, audio, and video. While text, tables, drawings and graphs are static information, audio, video, and pictures are dynamic information that must often be compressed into compact forms. In the early design phases, the requirements and specifications normally are the main part of the design information.

Andreasen [1992] claims that for product information a distinction can be made between three types:

1. Specifications: define the intended behaviour of the product during the life cycle phases
2. Characteristics: structure, materials, geometry, dimensions, tolerances and surface quality
3. Properties: describe the product behaviours and can be derived from the characteristics

The manner in which the design information is handled in the early phase is very important. However, such design information is domain dependent and is difficult to retrieve and to reuse. It also presents the problem of communication due to the misunderstanding of different domains. Ontology is one of the keys for handling such information. This issue will be discussed in more detail in 4.4.3.

4.4.2. General Requirements for a Design Information System

The design information is an artificial representation of a real-world system, as perceived by somebody, and built to enable an information processing function [Wand, 1996]. Irrespective of the objectives of the design information system, the source of the data is situated in the design process. Since the information captured by the system will normally be recorded by the design team members, the system must aim to place the minimum additional workload on them and not impede their creative processes. The system should therefore provide the designers with an intuitive user interface that is easy to relate to their preferred way of working.

Since a design information system is not a goal unto itself, it is not influenced only by the sources of the information, but also by the objectives for a particular system. Some requirements are introduced by the information system's goal as outlined in the following paragraphs.

Some requirements common to many users of the information that are already captured by a design information system are facilities to extract information selectively, to record comments on the information and to communicate these comments to the appropriate team members.

If a design information system is aimed at making the information exchange during the design process more effective, the ease-of-use requirements mentioned above are particularly important. It is also imperative that the system responds virtually immediately to a user request or creative processes will be disrupted. All design team members, even if they are in different locations, should further have concurrent access over data networks like the Internet to the information relevant to their activities. The information system must ensure consistency of the information (e.g. it must prevent two users from editing the same information at the same time or users inadvertently using outdated information).

Since performance, cost and time are equally important in engineering design, design information systems should ideally assist design teams in managing information about all three these aspects.

The objective of a design information system will influence the extent of the information that must be captured, e.g. systems aimed at design reuse (also called a design repository [Szykman *et al*., 2000]) would require more extensive documentation of the context of the design task, and would have to provide tools for storing the design record for reuse.

Many design information systems must also work with legacy data. One of the side effects of globalisation of product realisation is that design information systems from various companies that have been combined or are collaborating, have to be made accessible through a common design information system.

4.4.3. Ontology Database and Project Database

In DiDeas II, ontology is applied in the user interface as well as for the design information management. This section considers the latter application, while the user interface aspects are considered in the next chapter. The design information ontology is also utilised in the user interface since the design information ontology indirectly specifies the design procedure and terminology of the design methodology.

As was discussed in Chapter 2, ontology can be combined with the conceptual graph concepts of elements, their relations and attributes to build information structures for different purposes. With the support of ontology, both user and system can communicate with each other by the shared and common understanding of a domain.

Creating an ontology for a domain provides an opportunity to analyse domain knowledge, to make domain assumptions explicit, to separate domain knowledge from operational knowledge, and to provide common understanding of the information structure. The key notions for the flexible design information system are the use of an ontology approach, coupled with a database implementation structured using conceptual graph – the objects under consideration in design information systems can be classified as design information elements, and their ties as relationships between the elements. The classification in terms of elements and relations is found in conceptual graphs [Sowa, 1992.]. Another aspect of conceptual graphs that is of importance for flexible design information systems is that elements only interact with each other via relations, and relations only interact with elements.

This use of ontology and conceptual graphs were identified by Lutters [2001] when seeking ways of integrating information systems over all product realisation phases. The diversity of information in that situation is even greater than in design alone. The main difference between Lutters' proposals and what is proposed here, is that the flexible design information systems will not try to get all designers to agree *a priori* on the ontology, but it will allow designers to adapt the ontology used at any particular instance even though this will impede data exchange outside the design context.

Classical relational databases are equivalent to pre-defined elements and relations. This approach fixes the element types and relation types, which limits the extension of the database to different applications in different domains. Significant changes in the elements and/or relations usually require changes to the database structure and source code.

The difference between the approach proposed here and object orientated design information systems (such as that by Szykman *et al*. [2000]) should be considered.

Object orientated information systems also use an abstraction of design information (in the form of objects), but such objects are substantially more complex than conceptual graphs, even though any object can in turn be described in terms of elements and relations. Working with objects more complex than merely elements and relations certainly hold intuitive advantages for the user, but significant changes or additions to the objects handled in the design information system may require database changes and therefore software code changes. Further, object orientation loses some of its appeal if the data cannot be structured hierarchically (e.g. in terms of parent and child objects), such as when a particular information element has relations with elements that would naturally be in unrelated objects.

In the mechanical design setting, previous designs are often used as a reference for the design of new products. One of the critical issues in such cases is how to understand the intentions and justifications of various decisions that were made by different designers. They are collectively called the "Design Rationale" (hereafter referred to as DR). DR information is often implicit and this implicitness often causes difficulties in reusing the designs. However, DR is as important as design drawings.

The ontology plays a role similar to DR in reusing design information. In order to reuse design information, the ontology builder has to know the underlying conceptualisation which reflects the assumptions and requirements made in the problem solving by using the design information. Ontologies as DR information will contribute to the reusing of design information and will play a role as the backbones of knowledge bases.

It is generally accepted that a product structure reflects a product by using its elements and the relations [Tichem & Storm, 1996]. In broadening this option, any definite information can be established by looking at elements and their relations to other elements. Furthermore, additional information can be related to attributes of either elements or relationships between these elements, Figure 4-4 shows the block that can principally be applied to any information structure.

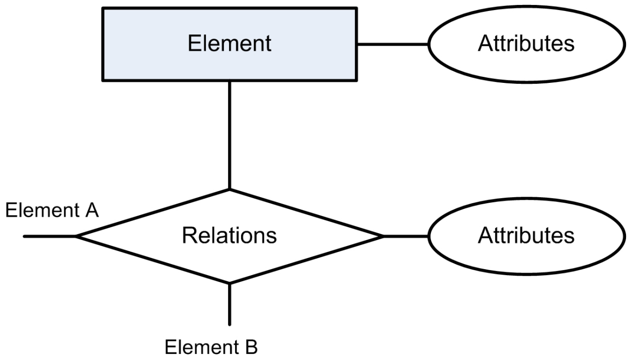

Figure 4-4 Element, Relations and Attributes

The element, the relation and the attributes can be defined as:

- Element: An entity represents an atomic object in order to describe the abstract concrete relationship among concepts
- Relation: A relation represents a relationship among entities to denote a static structure
- Attribute: An attribute is a concept attached to an entity and takes a value to indicate the state of the entity.

By using the elements, relations and attributes, the information can be represented in a mutually understandable way, and information can also be added by adding elements, relations and attributes without influencing the existing information structure. This helps to build a flexible information system. Using these concepts, an overall architecture for DiDeas II was conceived [Basson *et al.*, 2003]. This architecture has since been refined as part of the work presented in this dissertation, and is illustrated in Figure 4-5. Figure 4-5 shows the information flow of the main components - these different components will be illustrated in detail in the following section.

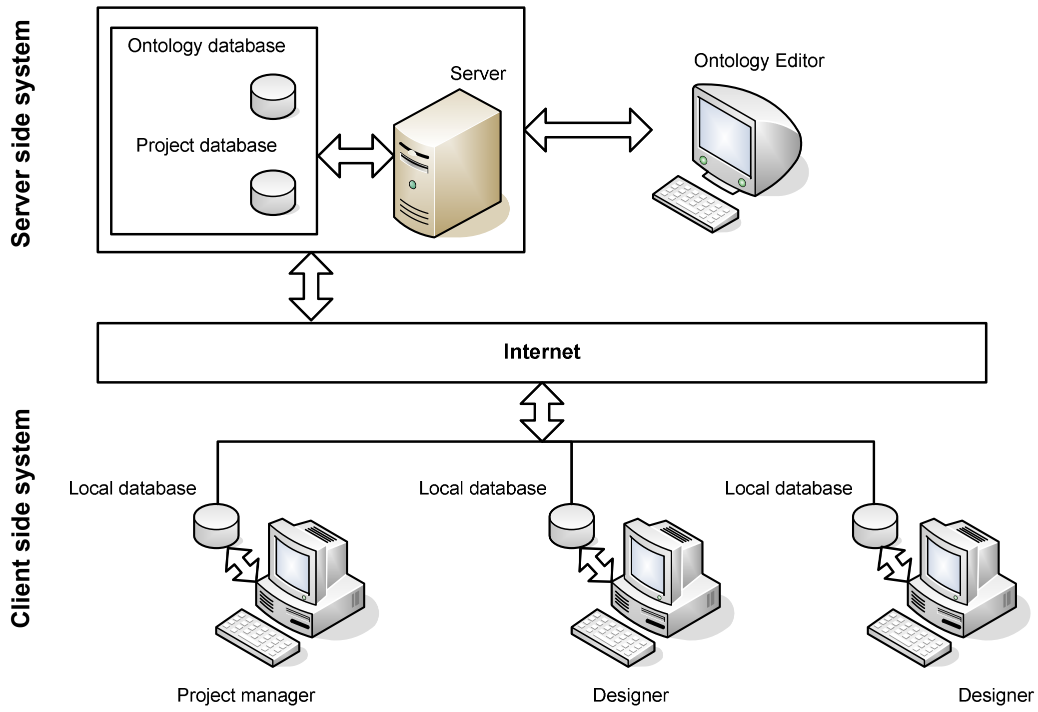

Figure 4-5 Main Information Flows in DiDeas II

DiDeas II contain two parts: the server side and the client side running separately. The server side saves the design information and designer works on the client side. The Ontology Editor is also running on the server side to setup the ontology database. These two systems transfer information via the Internet. The detail of this configuration will be discussed in detail in 5.3.

The Ontology Database and the Project Database are the core of the DiDeas II design information system. They are illustrated by the simple examples in Table 4-1, Table 4-2, Table 4-3 and Table 4-4.

All these tables have common items: The “Editor” is the designer who created/modified the element, the “Created Date” and “Modify Date” store the dates on which the element is created and modified. The “Lock” item indicates the status of the element: deleted (Y), undeletable (F) or active (N).

The Ontology Database (Table 4-1 and Table 4-2) contains a description of the available types of elements and relations that constitute the ontology. Each type has a unique ID, OEID and ORID which is referred to by the Project Database. The "Description" field explains the role of the element or relation for the particular design team. The "Show Type" is the attribute that determines the format of the element in the user interface.

The Relation Types table (Table 4-2) also shows the types of elements that may be associated with each relation (OEID(1^{st}), OEID(2^{nd})). The element types in the Relation Types table are redundant information, as the same information can be derived from the Project Database. Even though this permits potential inconsistency in the databases, this risk is accepted because the availability of the element type information in the Relation Types table makes the creation of generic user interface code (described in Chapter 5) possible and makes the meaning of each relation clearer to the user. The risk of inconsistency can be reduced by setting up a read-write checking mechanism. Future development of DiDeas II can reconsider whether it is necessary to include the element types in the definition of a relation type. It may be possible to omit this information, but further investigation is required.

Figure 4-6 illustrates the elements and relations in Table 4-1 and Table 4-2. In this figure, [] represents the elements and () the relations.

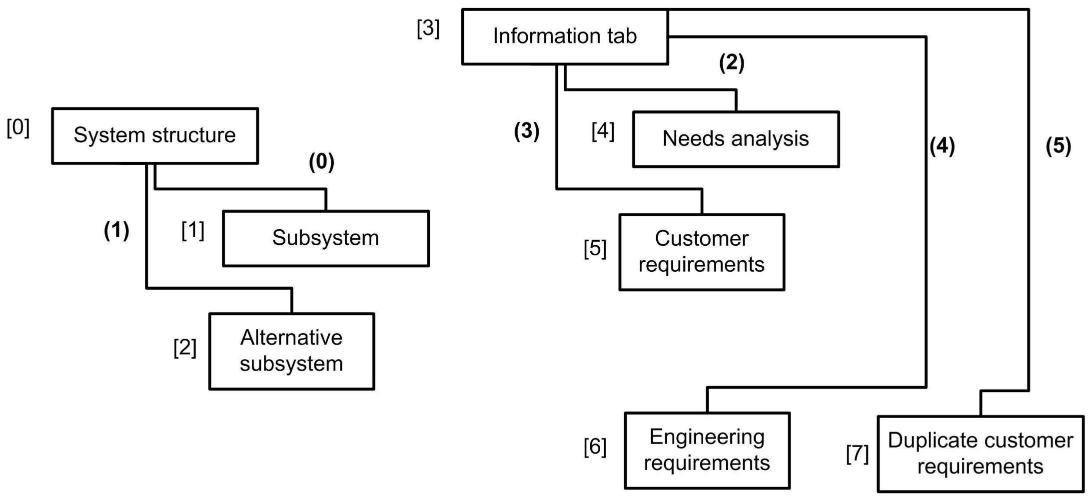

Figure 4-6 Ontology Database Example

OEID	Name	Description	Editor	Show Type	Created Date	Modify date	Locked
0	System structure	System structure view	Peter	N	2006-05-02	2006-05-02	F
1	Subsystem	The subsystem in the system structure view	John	N	2006-05-02	2006-05-04	F
2	Alternative subsystem	The alternative subsystem in the system structure view	Peter	N	2006-05-02	2006-05-04	F
3	Information tab	The information edit interface	Peter	N	2006-05-02	2006-05-04	N
4	Needs Analysis	The name of the design phase	Peter	N	2006-05-02	2006-05-02	N
5	Customer Requirement	The name of the design phase	Peter	T	2006-05-02	2006-05-02	N
6	Engineering Requirement	The name of the design phase	Lee	T	2006-05-02	2006-05-02	N
7	Customer requirements	Duplicate customer requirements	Lee	N	2006-05-02	2006-05-02	Y

Table 4-1 Example Element Type Table (Ontology Database)

ORID	Description	OEID (1^{st})	OEID (2^{nd})	Display Type	Editor	Created Date	Modify date	Locked
0	System has subsystem	0	1	0	John	2006-05-02	2006-05-02	N
1	System has alternative subsystem	0	2	1	Peter	2006-05-02	2006-05-02	N
2	Information has needs analysis tab	3	4	1	Peter	2006-05-02	2006-05-02	N
3	Needs analysis contains customer requirement tab	4	5	0	Peter	2006-05-02	2006-05-02	N
4	Needs analysis contains engineering requirement tab	4	6	1	Lee	2006-05-02	2006-05-02	N
5	Needs analysis contains duplicate customer requirement tab	4	7	1	Lee	2006-05-02	2006-05-02	N

Table 4-2 Example Relation Type Table (Ontology Database)

The Project Database contains the actual design information associated with one or more projects. The element table (Table 4-3) contains cross-references to the element types in the ontology (OEID), while the relation table (Table 4-4) cross-references the type of relation in the ontology (ORID), as well as the two elements associated with the relation (PEID(1^{st}), PEID(2^{nd})).

PEID	OEID	Content	Editor	Created Date	Modify Date	Locked
0	0	Bike	John	2006-05-02	2006-05-02	N
1	1	Support system	John	2006-05-02	2006-05-02	N
2	2	Attachment system	John	2006-05-02	2006-05-02	N
3	3	Bike needs analysis	John	2006-05-02	2006-05-02	N
4	4	Can support people and cargo	Peter	2006-05-02	2006-05-02	N
5	5	Support weight 160 – 200Kg	Peter	2006-05-02	2006-05-02	N

Table 4-3 Example Element Table (Project Database)

PRID	ORID	PE(1st)	PE(2nd)	Editor	Created Date	Modify Date	Locked
0	0	0	1	John	2006-05-02	2006-05-02	N
1	1	0	2	John	2006-05-02	2006-05-02	N
2	4	3	4	John	2006-05-02	2006-05-02	N
3	3	3	5	Peter	2006-05-02	2006-05-02	N
4	4			Peter	2006-05-02	2006-05-02	N

Table 4-4 Example Relation Table (Project Database)

The user interface titles (indicating the design process using certain technical terms) can also be built by using elements and relations. It provides the chance for the ontology editor to use his/her terms so that different design methodologies can be applied. It can help different companies to create their own "style" in the design process and improve the communication that is based on the same domain. Figure 4-7 shows the example of the project example to illustrate the elements and relations in Table 4-3 and Table 4-4. In this figure, 1 represents the physical structure of the bike system and subsystem, 2 stands for the step of design and 3 stands for the requirements of the subsystems.

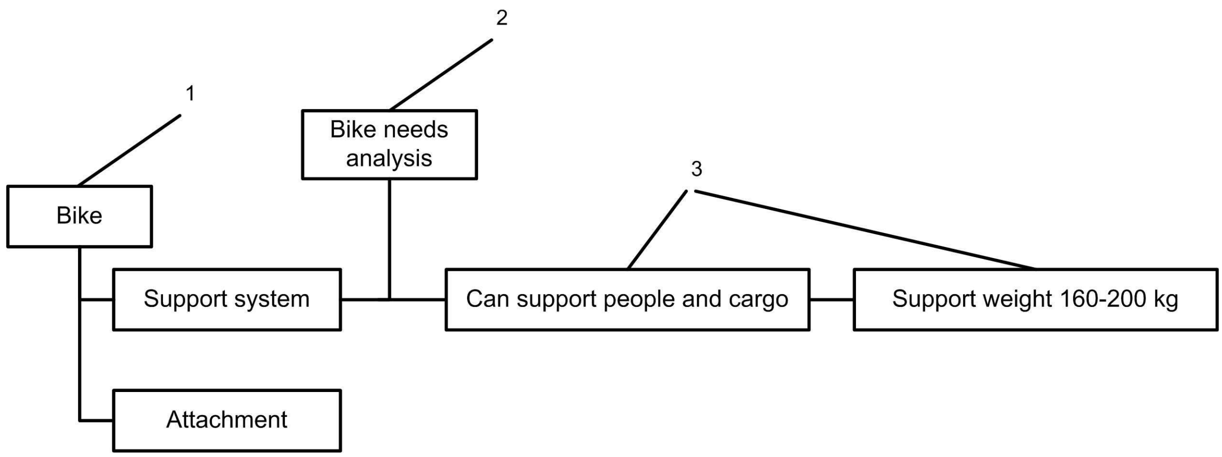

Figure 4-7 Project Database Example

To allow designers the freedom to brainstorm, without having to worry about classifying information in terms of the ontology at the same time, "unclassified" element and relationship types can be provided. The information system can record such unclassified elements and later present the designer with a "to-do" list of elements that still have to be classified in the same format (using a certain colour for the unclassified items). The flexibility of the proposed approach extends to not fixing a specific sequence in which data has to be entered, except that a relation normally has to be added after at least one of the elements to which it refers. "Incomplete" relations can also be added to a designer's "to-do" list.

Information that can be shared between different projects (such as the team member contact information, etc.), is stored in another database. Different projects can then read the same data from this shared information database to reduce the redundant information.

A disadvantage of the database structure described above is that every time when searching for all of the occurrences of a particular relationship, the whole relations table in the project database has to be searched through. In a classical relational database, many of the relationships would have been explicitly formed in the database structure, thus reducing the amount of searching that has to be done. The increased searching effort causes the data reading speed to be more dependent on the size of the project. However, the development of computer technology will improve the high speed data processing and that can reduce the reading time.

5. DIDEAS II IMPLEMENTATION

5.1. INTRODUCTION

In this chapter the system development platform is introduced. The server side system and the client side system are described separately. The functions of each user interface will be discussed in detail.

5.2. PROGRAM PLATFORM AND DEVELOPMENT ENVIRONMENT

Several programming languages such as HTML (Hyper-Text Markup Language), ASP (Active Server Page) and Java Applet were considered, but they could only provide limited functionality to display the design information structure view.

Visual Studio 2005 is a complete set of development tools for building ASP.NET Web applications, XML (Wireless Markup Language) Web Services, desktop applications, and mobile applications. Visual Basic, Visual C++, Visual C#, and Visual J# all use the same integrated development environment (IDE), which allows them to share tools and facilitates in the creation of mixed-language solutions. In addition, these languages leverage the functionality of the .NET Framework, which provides access to key technologies that simplify the development of Web applications and Web Services [Microsoft Visual Studio 2005 home page, http://msdn.microsoft.com, 2006]. Visual Studio 2005 Professional Version was therefore selected as the programming platform to develop the DiDeas II system.

C++ is a powerful programming language that can provide rich functions for different fields and is enhanced in Visual Studio 2005 to be able to develop .NET programs. With features such as garbage collection, a unified type system, networking and multithreading support, Visual C++ with .NET was chosen as the programming language for the DiDeas II development.

GDI+ (Graphics Device Interface) is a set of integrations that provide for the rendering of 2D graphics, and it provides support for colours, pens, images transformations and anti-aliasing [Fraser, 2003]. By using GDI+ and Gtek (a library of in-house developed graphics utilities) design information can be displayed as hierarchical trees in a graphic format and it can be viewed at different levels in a scalable manner. The user can also create, edit and delete the content of the tree.

DiDeas II was developed using Windows XP Professional with Service Pack 2. The DiDeas II would display a runtime error in previous versions of Windows and in other operating systems such as UNIX and Linux.

5.3. DIDEAS II INTERFACE DEVELOPMENT

5.3.1. DiDeas II Framework

DiDeas II has two separate programs running on the server side and the client side respectively. The flow of information in DiDeas II is illustrated in Figure 5-1. The server side database contains the ontology database and the project database. The ontology editor interface also runs on the server side and is used to create, edit and delete ontology elements and relations.

The designer works on the project editor interface on the client side. The Project Management block (through the Project Information Window) in Figure 5-1 is used to allow the project manager to set up project meta information such as project name, start date, end date, etc. as shown described in section 5.3.3.2. The design team assignment and involved companies information are also handled here. By using the Internet, the client side computer gets the data from the server side and saves it into the local database. Any information creation and modification are also sent to the server instantly and are saved on the server side database so that all the designers can work on the same project simultaneously.

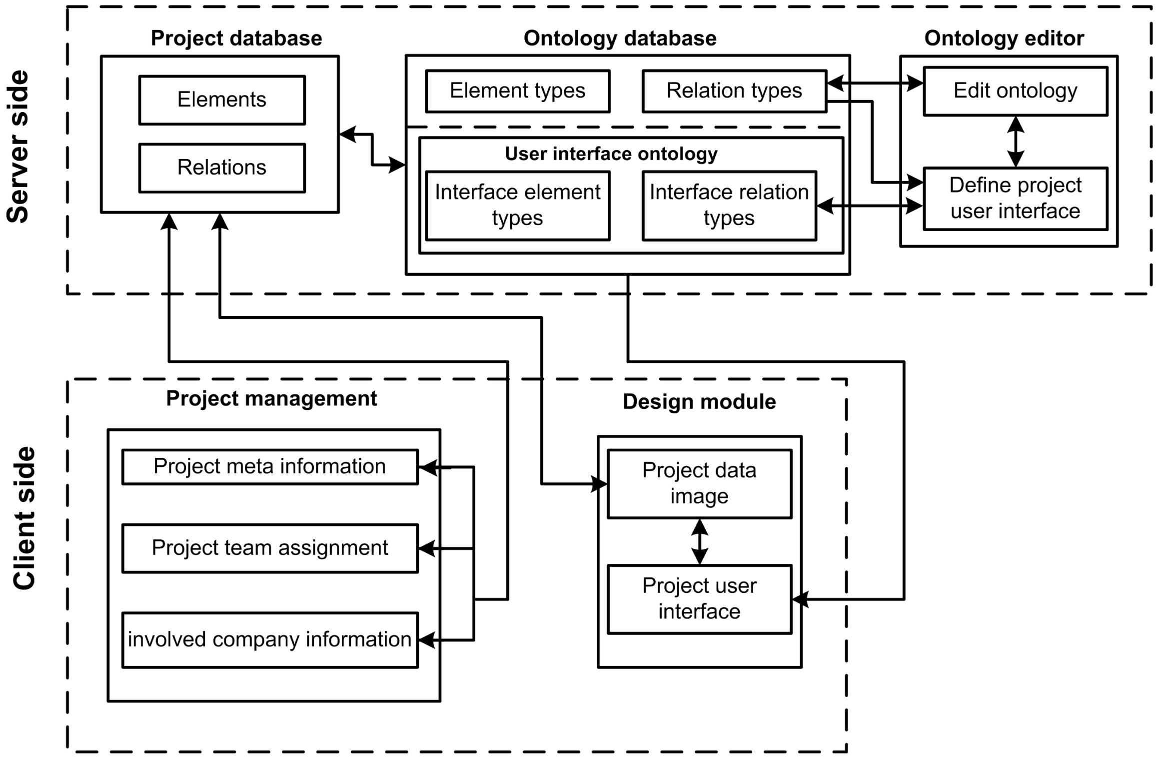

Figure 5-1 DiDeas II Information Flows

The typical sequence of operations in the interface of DiDeas II is presented in Figure 5-2.

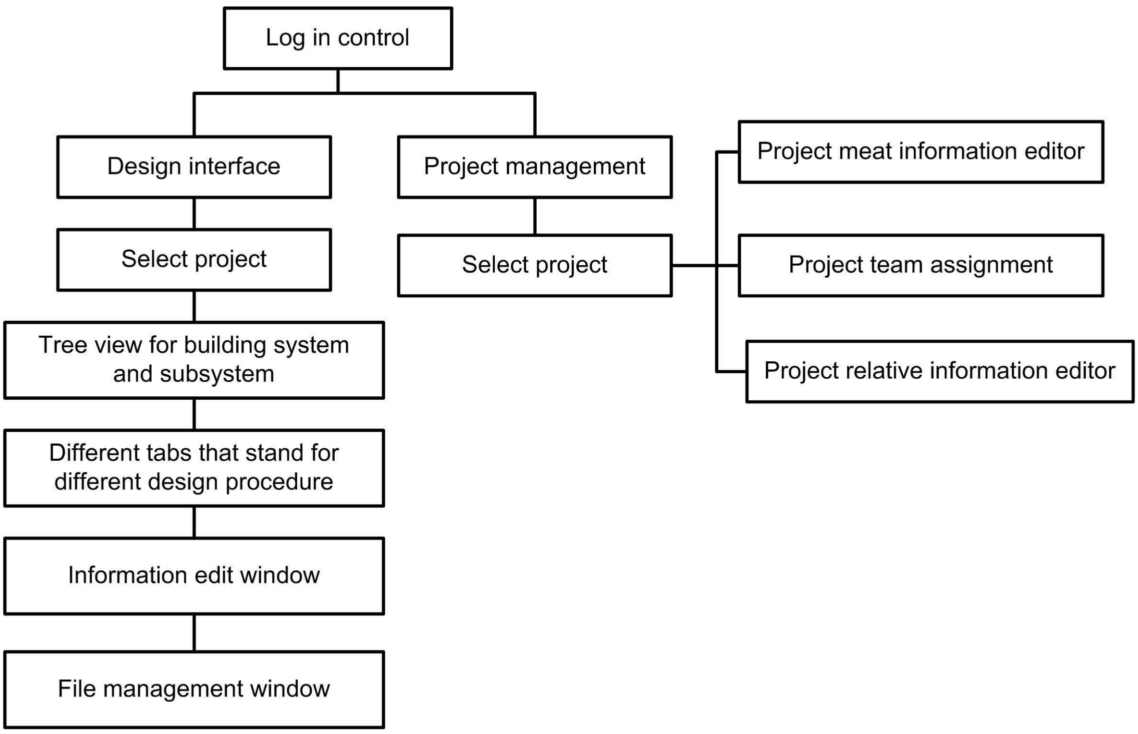

Figure 5-2 DiDeas II Sequence of Operations

5.3.2. User Roles

As shown in Figure 5-1, there are different databases in DiDeas II. Three roles can be identified for the maintenance of the information in these databases, i.e. that of the normal design team member, that of the project manager and that of an "Ontology editor" who can be the project manager, systems engineer and/or chief designer.

The ontology editor is the one that sets up the ontology database, thereby determining the design procedure and associated terminology that will be used. The ontology editor does this by maintaining the "Element types", "Relation types" and "User interface ontology" in the "Ontology database", shown in Figure 5-1.

The project manager maintains the overall project information and the team member allocation. This information is kept in the "Project database".

The design team member can only access a project if permission is given by the project manager. Through the "Design module", the designer can enter project specific information in the "Project database", using the design procedure and terminology set up in the user interface by the "Ontology editor".

The following sections describe the typical user interface screens through which the persons fulfilling the respective roles, maintain the data in the various tables.

5.3.3. Client Side System

5.3.3.1. Log-in Window

When the DiDeas starts up, a log-in window will appear to provide access control to the system - the interface is shown in Figure 5-3.

The “Server IP” is the IP address of the computer operating as the server. “User Name” and “Password” provide the access control. A user can get the password by answering certain security questions that are assigned by the project manager. A user has two log-in options: as project manager or as designer. The project manager can assign a new project or edit existing project information in the Management Window (described in 5.3.3.2) and a designer can work on the designer interface (described in 5.3.3.4).

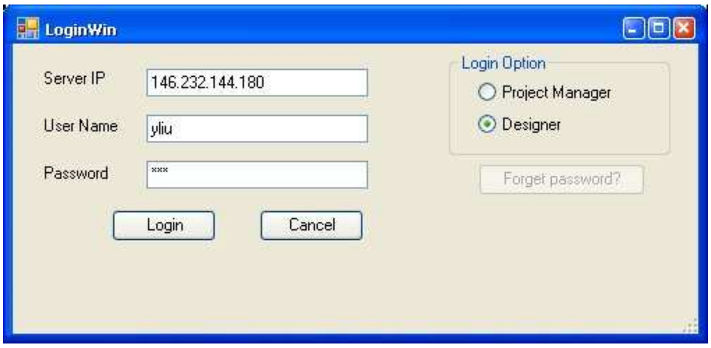

Figure 5-3 Log-in Window

5.3.3.2. Project Management Window

After a login as project manager, the project management window will appear. It has three main functions:

- Project information management
- Assigning of team members and editing their personal information
- The management of the company information involved in the selected project

These three main functions are performed by three windows shown in Figure 5-4, Figure 5-5 and Figure 5-6, and these three windows can be reached by clicking the relevant tab. When the project is selected, the last two windows will show the information of the selected project.

The project information window (Figure 5-4) is used for inputting the overall information of the project, such as the project name, the start date, the end date and a short description.

The team member window (Figure 5-5) is used to assign the team members and to manage the team member's information as follows:

- User ID: automatically generated number for identifying different users
- User Name: user’s full name
- Nickname: user’s login name
- Expertise Field: the designer’s field of experience
- Type Index: it is used for access control such as ontology editor, project manager or designer
- Location: where does the designer stay

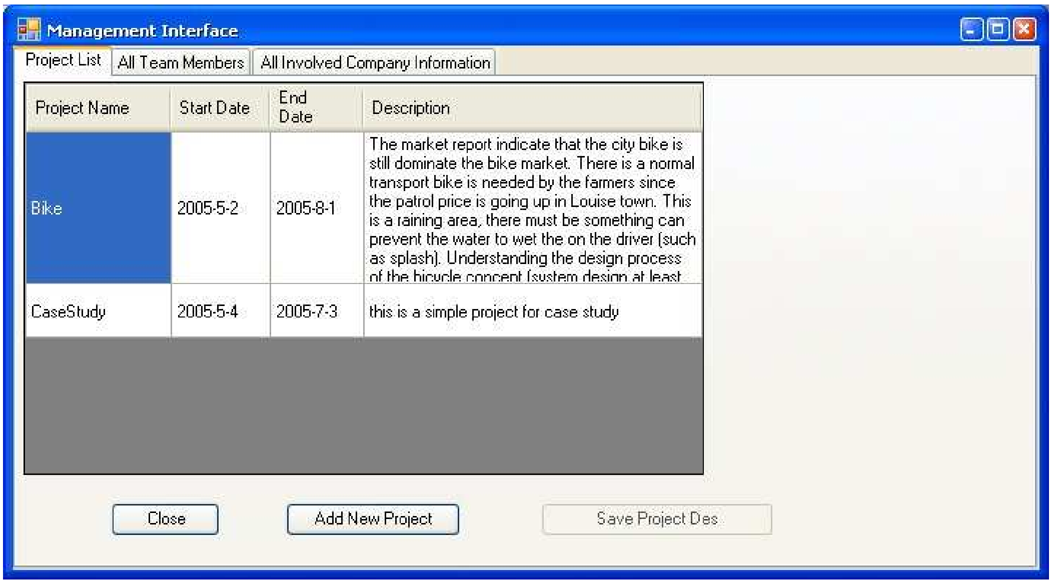

Figure 5-4 Project Information Window

- Address: the mail address of designer
- Telephone Number: designer’s telephone number
- Fax Number: designer’s fax number
- E-mail Address: e-mail address of the designer
- Password: login password, it can only changed by project manager
- IP Number: the IP address of the designer’s computer
- Cell Number: mobile phone number of the designer
- Question: when a designer forgets the password, this question is used to help him/her to retrieve the password. It was setup by the designer and passed on to the project manager. It is ready-only on the screen
- Answer: the answer to above question

The design team members’ personal information can provide the various members with a first impression of each other in a distributed design environment, and it also provides contact information.

The involved company information window (Figure 5-6) is used to assign, create, edit and delete the following company information:

- Involved Company's ID
- Company Name
- Contract Start Date

- Contract End Date
- Contact Person
- Company Capabilities
- Telephone Number
- Fax Number
- Cooperation Field
- Company Address
- Company Description
- Contract Information
- Web Site

The Involved Company's ID above is auto generated for distinguishing between different companies. Company Capabilities provide the product field of the specific company.

The involved company's information can be used as a "contact memo" and is useful when the designer needs to contact such a company.

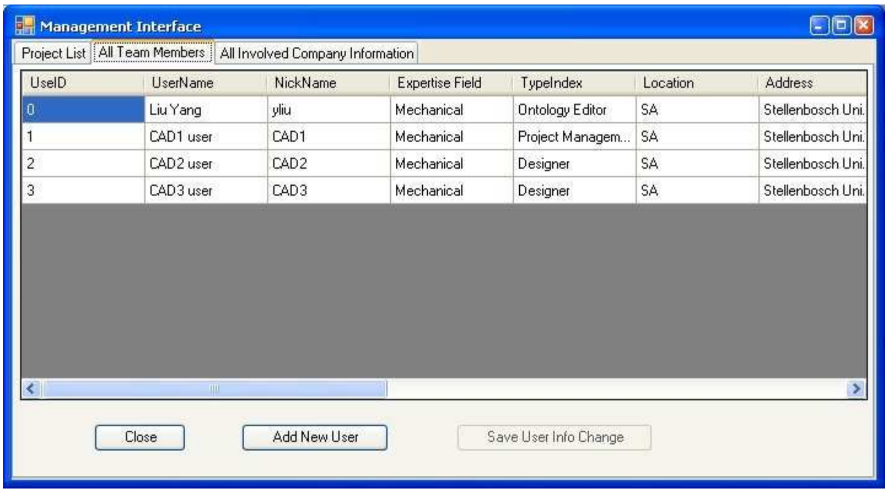

Figure 5-5 Team Member Information Window

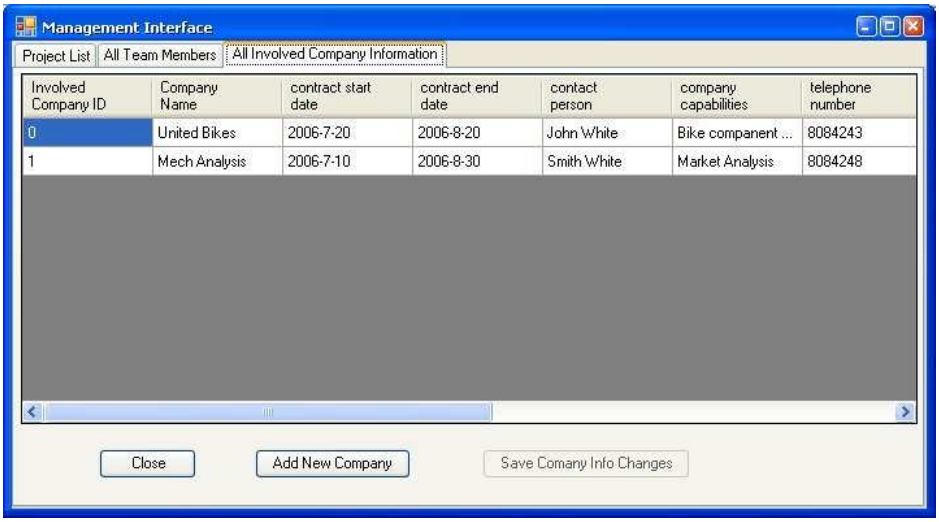

Figure 5-6 Involved Company Information

5.3.3.3. Updating Interval Setup Window

When the project management window is closed, a dialog box will ask whether the project manager would like to carry on with the design work; the "yes" option can bring the Updating Interval Setup Window. If a user directly logs in as a designer, this window is the first window to be opened.

During the design process, all the design data must be read and saved into the server. To ensure that client side's data remains up to date, it must periodically request updates from the server (in case another team member has made changes to the project). The window shown in Figure 5-7 enables the user to determine the data updating interval.

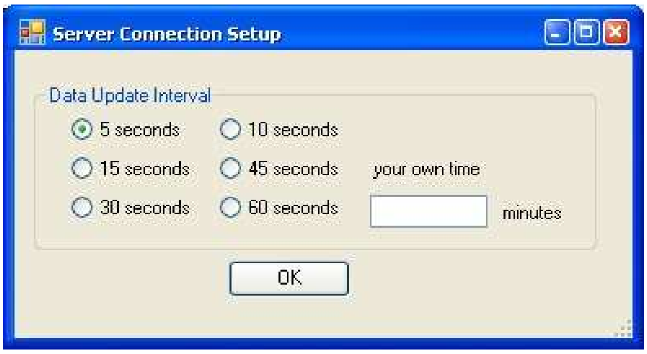

Figure 5-7 Data Updating Interval Setup Window

5.3.3.4. Designer Interface

The designer interface in DiDeas II is responsible for reading and storing the design information during the design process. It provides the design support capabilities for refining, elaborating, and editing the evolving project. It is the main window that the designer would work on. A project is selected from the list on the far left of the screen (the list disappears after the selection), and the design information will be managed in this window as shown in Figure 5-8.

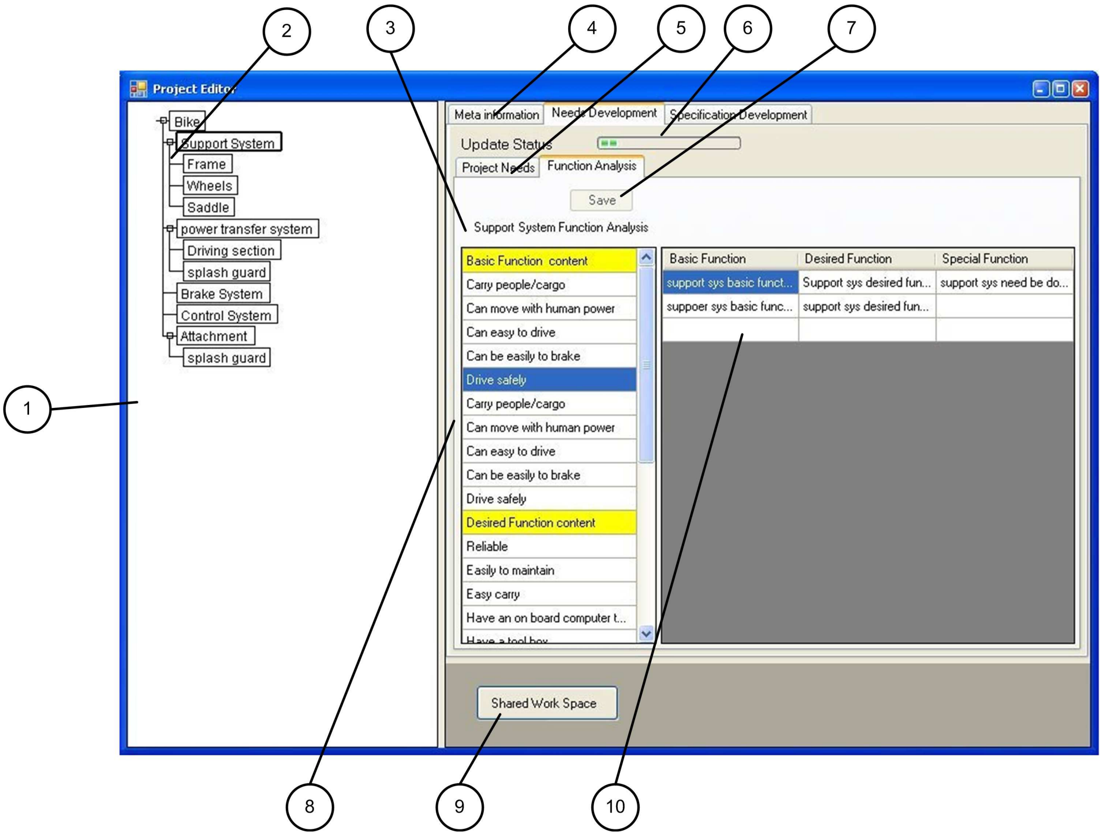

Figure 5-8 Designer Interface

On the left-hand side (1) is the tree view which shows the structure of the system and subsystems (2). On the right-hand side (3) is the design information edit window linked to the node selected in the system/subsystem tree. By using the Alt key and the mouse, the user can pan and zoom the tree view according the size of the tree view – the tree view can also be printed. The information edit window (3) contains the tabs (4) and sub-tabs (5), the data updating bar (6), tables (8 and 10) and function buttons (7 and 9).

The left section (1) displays the current design project, which is broken down into its subsystems and constituents according to their different functions. This represents the design slate on which designers decompose problems and detail constituents. The tree view can be collapsed and expanded to show the hierarchical pattern of the system and the subsystem in a clear and understandable manner. The nodes of the tree are either concepts or subsystems, depending on the design project.

This gives a global view of the current problem, and allows the designer to focus on any level of the decomposition, and provides several designing, editing, and browsing functionalities. The user can create, edit and delete the tree node (subsystem) by selecting the right click menu.

The tabs (4) in Figure 5-8 indicate the design phase that is defined by the ontology editor, while the sub-tabs (5) are the design steps in this design phase. The table heading reflects what kind of information will be shown. This design style in DiDeas II is suited to small and medium enterprises that normally focus on small projects which are not very complicated.

The flow chart of the designer interface building process is shown in Figure 5-9. Before the interface window is loaded, a request is first sent to the server to get the ontology data. DiDeas II then uses this data to build the user interface window that uses the terms that are defined by the ontology editor. After the user interface window is formed, the user must select a project from a list of projects on the most left-hand list (see the Tree view area (1) in Figure 5-8). The project list window will disappear after the user has selected one project – only one project can be accessed at a time.

After the project has been selected, the relevant project information will be sent from the server to the user interface as in Figure 5-8.

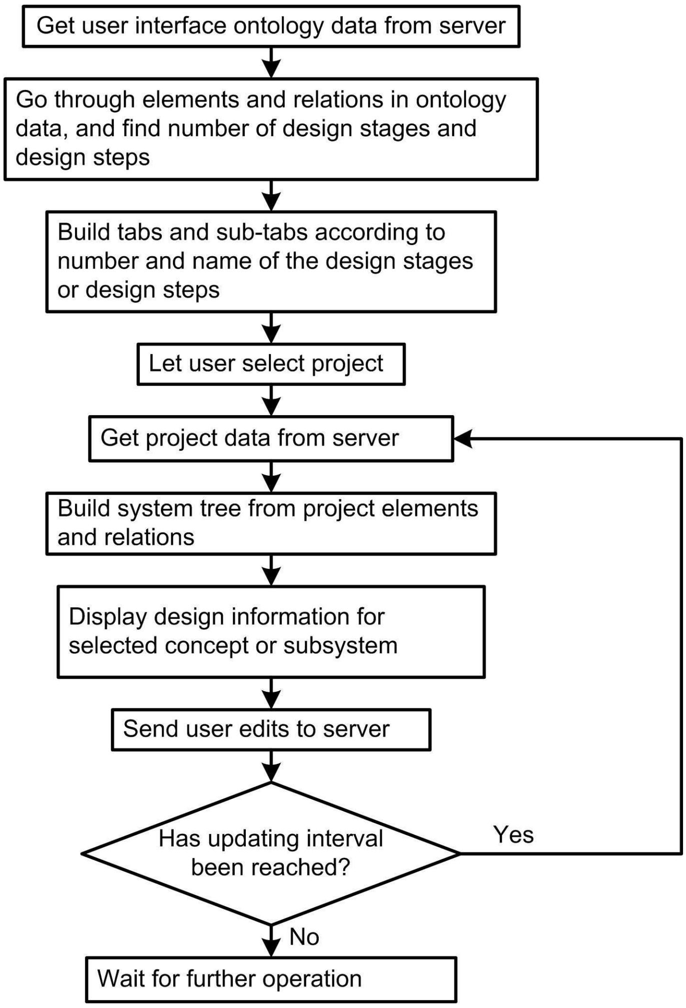

Figure 5-9 Flow Chart of the Data Reading Process in Designer Interface

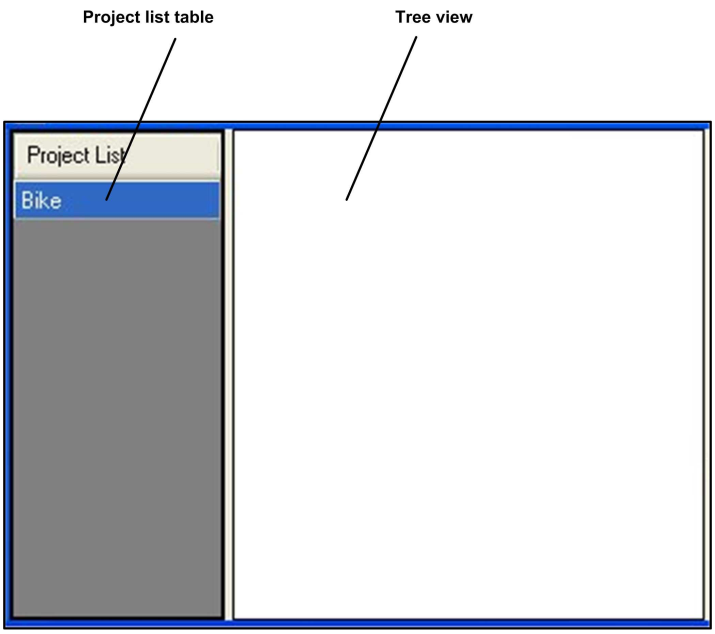

Figure 5-10 Project List Table

The right-hand side section of the window in Figure 5-8 is the design information management interface. The first tab (4) is the meta-information, such as project general information, team member personal information and involved company information as is shown in Figure 5-11. The information in the tables under the meta-information tab is read-only to the designer and can only be modified by the project manager on the Project Management Window (as was discussed in 5.3.3.2).

When a node in the tree view is selected, the relevant design information is shown in, or can be input into, the tables under the second and later tabs on the right-hand side. The tab shows the design phase that the designer is working on and for each tab there are sub-tabs that indicate the steps of each design phase.

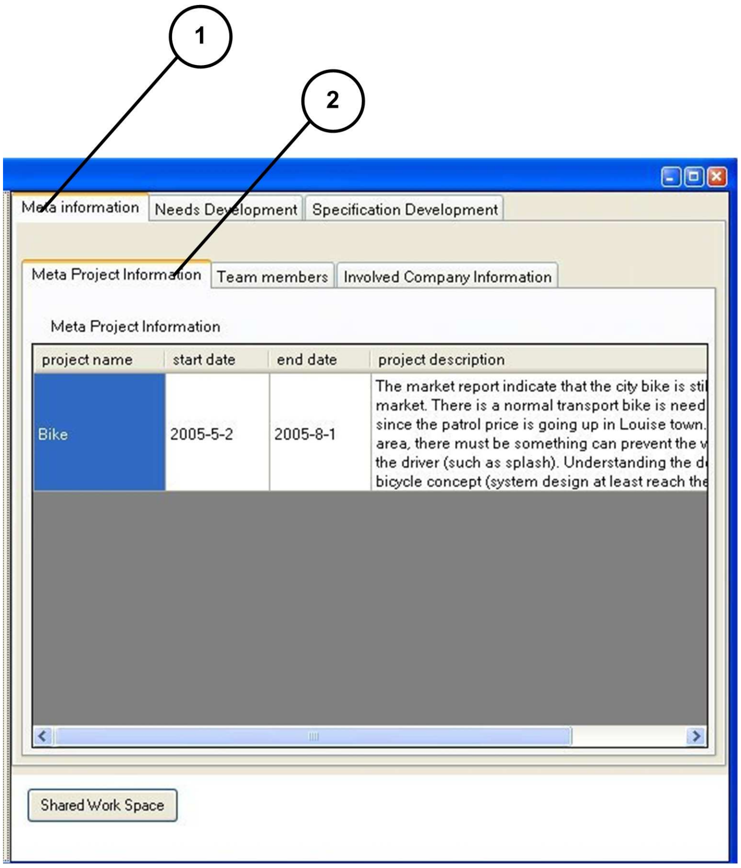

Figure 5-11 Project Meta Information Tab

The second and later tabs in the examples were setup, through the ontology editor, to reflect the specification development process shown in Table 5-1.

Design Phase (table)	Design Step (sub-tab)	Involved information (table)
Needs Development	Project Needs	list all project list
	Function Analysis	function classification
Specification Development	Customer Requirement Development	list customer requirements
	Engineering Requirement Development	list engineering requirements
	QFD Analysis	Quality Function Deployment table

Table 5-1 Design Process for Case Studies

However, all these design phase names, design step names and information type names (tab, sub-tab and table) come from the ontology database and can be changed to suit a certain design methodology. The ontology editor can only create and modify them through the ontology edit window; such tab, sub-tab titles and table headings are read-only to the designers.

A progress bar indicates the progress of the updating process (item 6 in Figure 5-8). The design data is updated after a previously specified period (an updating interval as is assigned in the Updating Interval Setup Window, section 5.3.3.3), so a designer can always get the latest information. When a user starts to edit the information in a table, no update request will be sent to the server so that the designer can work without interruption. Only after the save button is clicked, the information changes will be sent to the server to ensure that the other designers can also see the changes when the database is updated. There is a simultaneous server accessing protection running on the server side to prevent a network block and system running error. A check request is sent to the server before the client sends the changed information, and only when there is no request being processed on the server side, the changed information request will be sent to the server side.

Using the tables on the information sub-tab, the designer can input or modify the information linked to the currently selected subsystem node. The left-hand side table shows the same type of information in a read-only format for the parent subsystem in the tree so that the designer can easily compare the information at different levels. If the project node in the tree (the root node in Figure 5-8) is selected, there is only one table for managing the information.

If two subsystems share the same function, this function needs to be shown in two different tables as these two subsystems are selected. However, this function should be saved once in the project database, since any change in this function must be shown in both tables. This kind of design information is called duplicated information here. The duplicated information that exists on the different levels is highlighted in a certain format (such as using a different background colour for a cell compared to the other items). The designer will receive a warning message when modifying such information to make sure that the designer also checks it in the other subsystems. The current implementation does not provide the facility to break the link between instances of duplicated information, but this can easily be added in future.

Normally more than one solution for one system/subsystem in the design process must be analysed. In order to distinguish alternative subsystems, these alternatives in the tree view can be highlighted to remind the designer to evaluate this subsystem and to make decisions.

5.3.3.5. QFD Analysis Window

For a distributed design environment, the concept of QFD (Quality Function Deployment) is particularly valuable because of its high information content and the comprehensible graphic representation in the form of the QFD-matrix "House of Quality". A properly developed "House of Quality" can ensure the completeness of customer requirements and engineering requirements, and it can also ensure that the problem to be addressed is understood by all team members. The latter is a vital aspect, especially for a distributed design team.

A QFD analysis window in the DiDeas II system (Figure 5-12) can be invoked from a right click menu in Figure 5-8. By using programming with GDI, this view provides a better view that can help a designer to identify the relations between the customer requirements and the engineering requirements. If the view in DiDeas I in Figure 4-3 is compared with the view in DiDeas II as shown in Figure 5-12, it is clear that the latter view is more convenient for designer to work on.

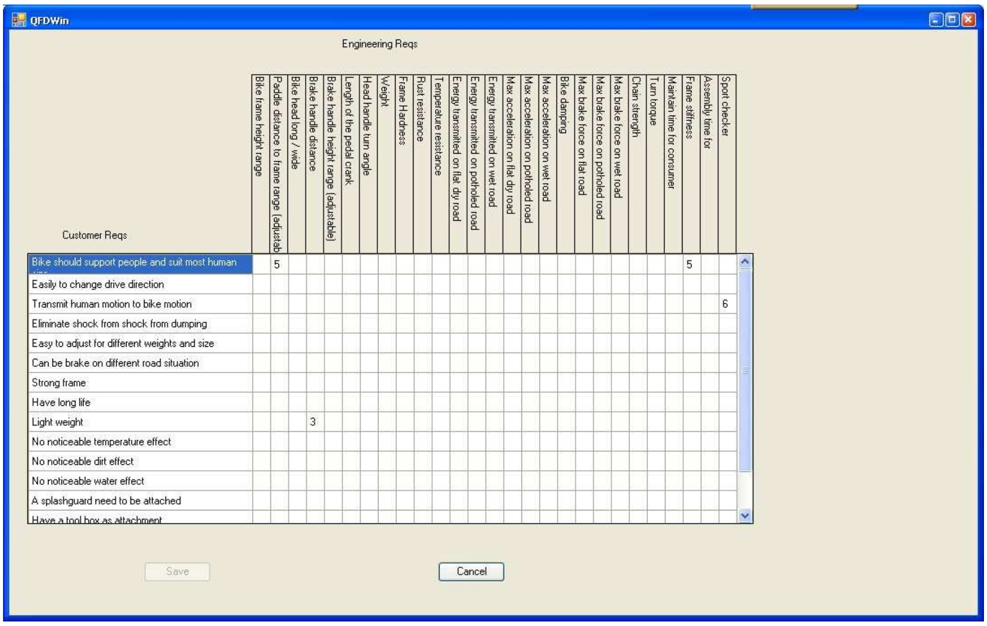

Figure 5-12 QFD Analysis Window

This QFD analysis window is independent of the designer interface and is created dynamically by using the content from the customer requirement table and the engineering requirements table in the Specification Development tab in Figure 5-8. If one of these two tables is empty, a warning message will appear to prompt the designer to enter information, instead of showing the QFD window.

This designer interface can be further developed to handle the concept generation and concept evaluation by using an ontology setting.

5.3.3.6. Shared Work Space

During the design process, many types of files, such as technical reports, CAD files, and Excel data sheets, are generated for different purposes. By clicking the "shared work space" button (item 9 in Figure 5-8), a shared work space window (see Figure 5-13) will show up. DiDeas II provides this interface to handle the non-textual files in the system.

The designer can download and upload the files to the server through this interface. The table on this interface provides information such as the filename, the upload time, the person who created the file, which subsystem it relates to and a short description of the file. Although there is no direct link to the subsystem in the tree view, the information on this table can guide the designer to find the required file.

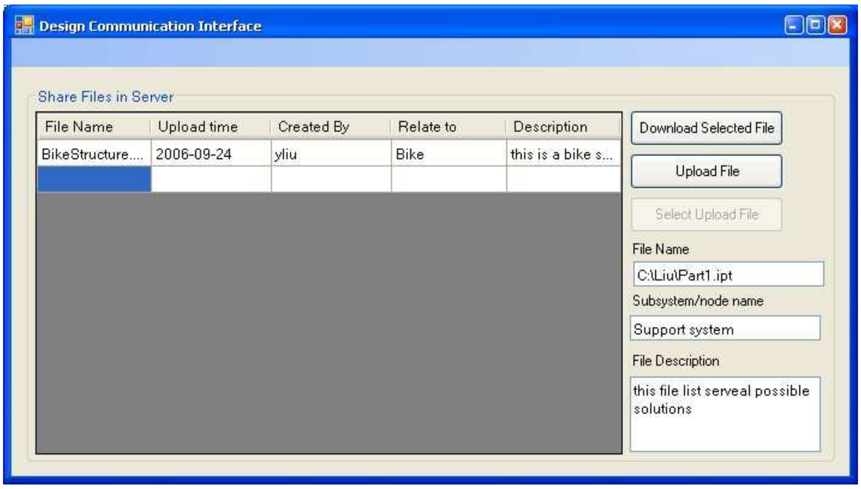

Figure 5-13 Work Space Window

This workspace can be accessed by all the team members and show only the files relevant to one project.

5.3.3.7. Design Review Window

In order to manage the documentation of design reviews, the Design Review Window (Figure 5-14) can be loaded through a right click menu on the designer interface.

Figure 5-14 Design Review Management Window

This window is used for managing the design reviews. The designer can read the text file directly from the server side by clicking the read button. This interface will download the selected file from the server and open it.

When a designer generates a design review or a design note, he/she must input the Design Stage and Filename text boxes in Figure 5-14. DiDeas II will combine these two into a filename that can contain the design phase information. After the filename is assigned, the Create button will open a blank Microsoft Notepad file. After the designer finishes input/editing the content of this file, the “Save to server” button will be enabled and the file can be sent to server by clicking it. Microsoft Notepad is a common editor in the Microsoft Windows, so it was selected as the design review editor and if there are files in other formats, they can be handled in the shared workspace discussed in 5.3.3.6.

5.3.4. Server Side System

5.3.4.1. Ontology Editor

The ontology edit interface running on the server side provides the facilities to manipulate the ontology elements and relations. It has two modes: Element Edit Mode (given in Figure 5-15) and Relations Edit Mode (given in Figure 5-16). The first one is used to directly create, edit and delete elements. The second mode is specifically for the relation management. In addition to the general open, save and edit functions, this mode also provides utility functions such as when a user is editing a relation, the two elements that are linked by this relation can be highlighted by clicking the “show relation” menu.

Using the element table (item 1 in Figure 5-15), the ontology editor can directly enter or edit the element name, the element description, the element show type, the created date, changed date, etc.

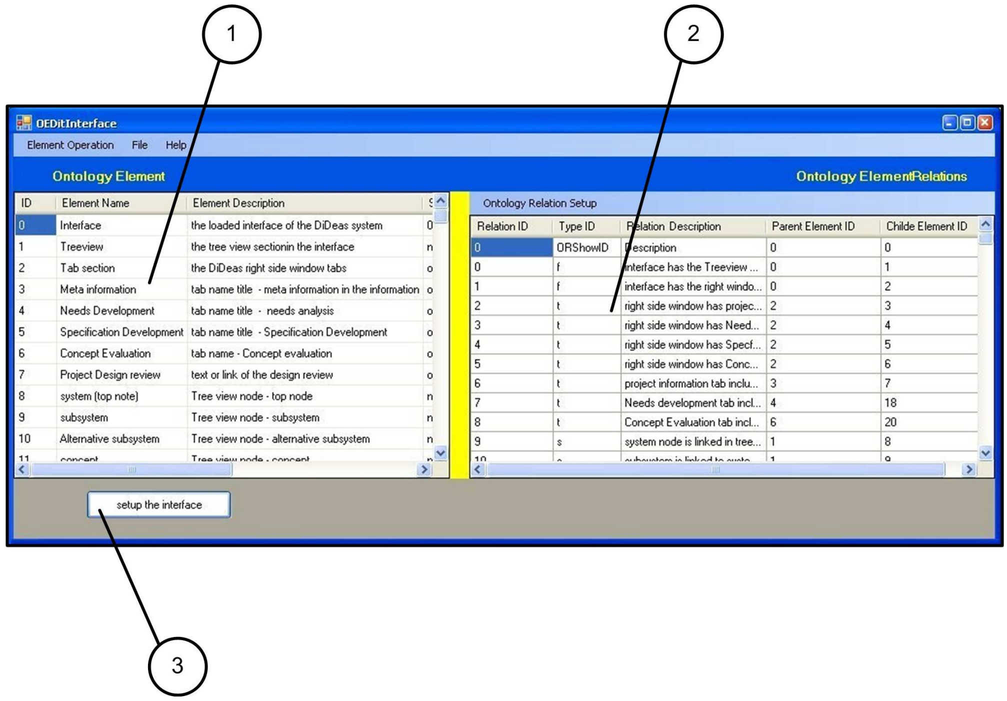

Figure 5-15 Ontology Edit Interface (Element Edit Mode)

The new relation can either be entered in the relation table (item 2 in Figure 5-15) or on the bottom edit box (item 3 in Figure 5-16), into which a user can input information according to different types of text boxes. In order to see the relation between two elements, there are two tables of the parent element (1st element) and the child element (2nd element), shown as 1 and 2 in Figure 5-16, While the information is saved by clicking the OK button (item 3 in Figure 5-16), a check mechanism will check that the information is consistent and ensure that the same information is not saved onto the server twice.

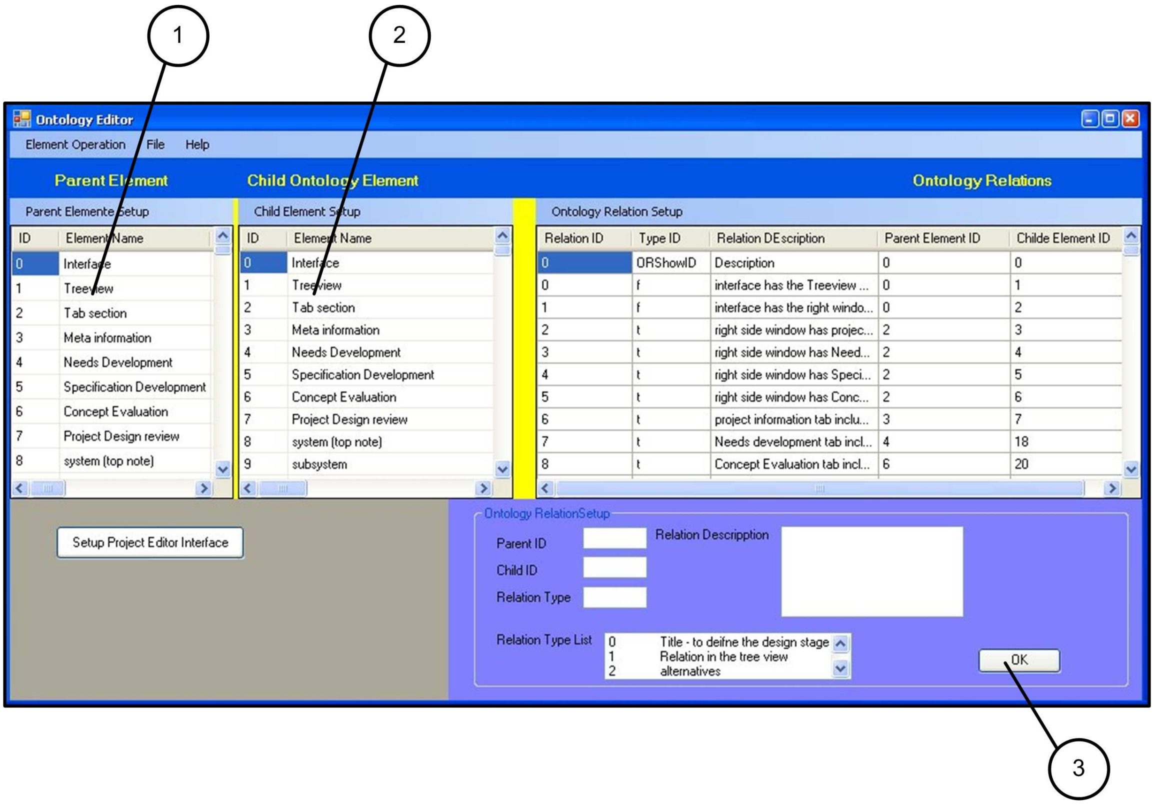

Figure 5-16 Ontology Edit Interface (Relation Edit Mode)

Clicking the "setup the interface" button (item 3 in Figure 5-15) gives the interface setup window (Figure 5-17), which is another interface especially for setting up the tabs, the sub-tabs and the tables in the Designer interface (items 4, 5, 8, 10 in Figure 5-8).

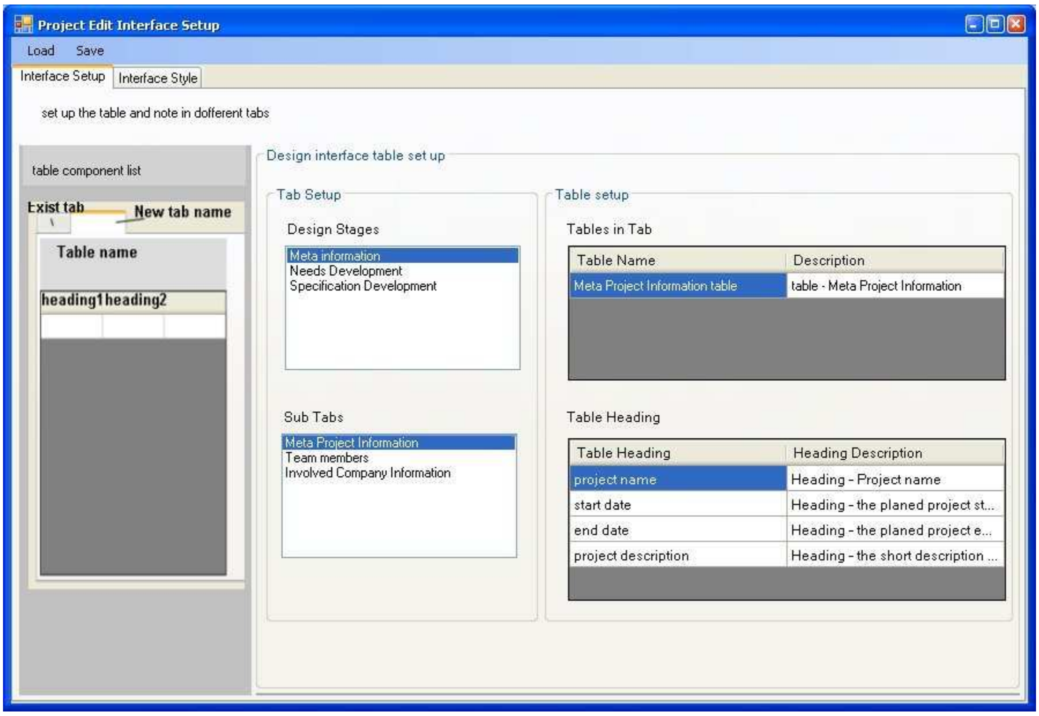

Figure 5-17 The User Interface Setup Window

A view of the tab settings of the user interface is shown on the left side of the window. It helps the ontology editor to understand which title and heading are going to be shown in the designer interface. The terms can be input via the text boxes on the right-hand side, through which the design process can be organised.

The design technical terms on the interface are chosen by the ontology editor, who is normally an experienced designer or the project manager. In this way the company design style is recorded, and the design methodology that applies in the design project is also implicitly represented. When a new designer views a previous project, it would be easy to learn from the example and to integrate his/her own knowledge into the company. By using the same technical terms, a new designer's ability to communicate with other designers in the company can also improve.

The "interface style" tab in the setup interface (Figure 5-18) is used to define the format used for a tree node in the tree view.

The Element Style defines the text colour and the node box background colour of the tree view (in Figure 5-8). The Relation Type defines the line colour and line style of the line that connects different subsystem nodes in the tree view.

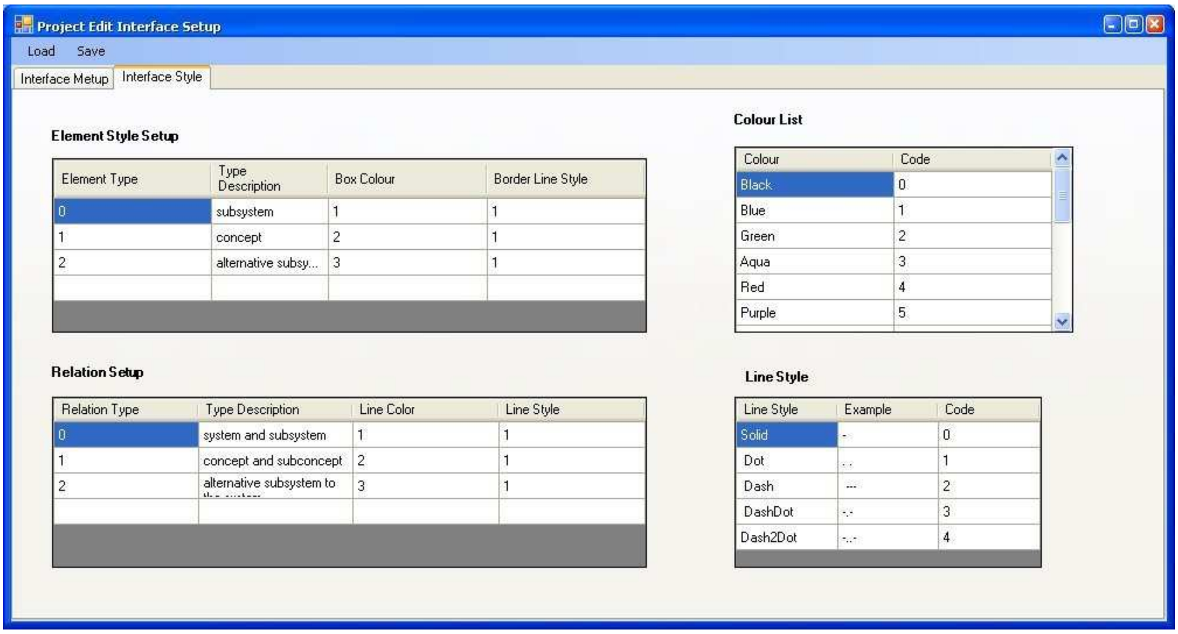

Figure 5-18 Information Setup Window

The right-hand side table shows the colour codes and the line style codes that this system can provide. The style information is saved as a separate file on the server side and when the system starts, this file will automatically be sent to the client side computer.

5.3.4.2. Server Window

The DiDeas II server is shown in the Server Window (in Figure 5-19). It provides the function that starts, stops and closes the DiDeas II server. A text box in this window shows the running status of the DiDeas II server.

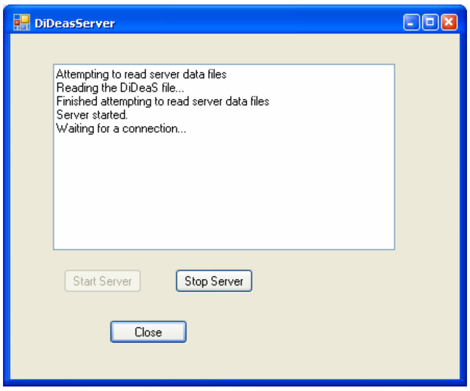

Figure 5-19 Server Window on a Server Side

The Server Window has the responsibilities of receiving requests from the client side and sending related data according to these requests from the data base on the server side. These functions were programmed based on TCP/IP (Transmission Control Protocol over Internet Protocol) technology. It is one of the protocol suites used most commonly on the Internet. IP is a network layer protocol that handles packet switching, fragmentation, and routing. TCP is a transport layer protocol built on top of IP and it handles flow control, multiplexing, and error control. By using the TCP/IP functions that were provided by .NET, the data can be sent via the Internet to the required IP address.

If there is more than one designer at work on the same concept, the last changes saved will be kept. This concept or function will then be read-only for the other designer. When a request is received by the server, a global flag is used to ensure that no other request can be processed until the required data is sent to the client side computer. If a request cannot be received, a server busy message will show up on the client side computer to prompt him/her to send the request again later. Each request process has a running time limit (500 ms). If there is an unsuccessful data operation causing a runtime error and making the process run over the time limit, the server program will cancel this process and prepare the server to receive a new request. Meanwhile an error message will be sent to the "request-send" computer on the client side to prompt the designer to make a further decision.

6. DIDEAS II CASE STUDIES

6.1. INTRODUCTION

Two case studies were carried out to identify whether DiDeas II can handle the design information during a design process by using ontology. It was evaluated based on system functionality and performance. In this chapter, the configurations of the case studies are first described, and then the data analysis and the results are presented.

In the first case study, DiDeas II was used in a number of sessions by small design teams. In the second case study, DiDeas II was evaluated by three participants from industrial companies. They provided valuable comments, and a new design interface was built based on these comments.

The potential users of the DiDeas II system were identified as small and medium enterprises. The case studies therefore simulated the work environment of a small group of people working on the same project in a distributed environment.

6.2. CASE STUDIES OBJECTIVES

The objectives of these case studies were to evaluate the DiDeas II system in terms of a number of aspects in the following two categories:

Functionality

1. DiDeas II can be accessed synchronously in a distributed environment.
2. DiDeas II can apply different design procedures and different terminologies in a flexible manner for different purposes.
3. DiDeas II can enhance a designer’s productivity during design specification development.
4. DiDeas II can handle different system/subsystem levels in the same project.
5. DiDeas II can easily manage the design documentation.

Performance

1. DiDeas II is reliable and user-friendly.
2. The tree view and table views are suitable for handling the design information.
3. DiDeas II can process different types of design information.
4. The ontology editor can manage the setup of the user interface in a flexible manner.

Both the DiDeas II client-server system and the ontology editor were evaluated in the case study.

6.3. CASE STUDY OF SMALL DISTRIBUTED DESIGN TEAMS

6.3.1. Case Study Setup

6.3.1.1. Case Study Environment

Three offices at the Department of Mechanical and Mechatronic Engineering at Stellenbosch University were used for the case study. The offices were adjacent so that the study could be supervised easily. The offices also provided a spatially distributed environment. Group meetings were recorded by a video camera for later analysis.

6.3.1.2. Case Study Participants

A total of 12 postgraduate and undergraduate engineering students with different design educational background and experience participated in the case study. The chosen students were volunteers from the Department of Mechanical and Mechatronic Engineering and the Department of Process Engineering. Two of the students were from different nationalities, and so their languages and academic backgrounds differed from that of the other participants. These differences can be valuable when the influence of different technical backgrounds is investigated. The students were divided into four groups with each group consisting of members with varying levels of experience. In the different sessions, each member of a specific group worked in a different location to simulate functioning as a distributed design team:

Session 1: One postgraduate student acting as team leader with two second-year students as team members (all were from the Department of Mechanical and Mechatronic Engineering).

Session 2: One postgraduate student acting as team leader with two third-year students as team members (all were from the Department of Mechanical and Mechatronic Engineering).

Session 3: Two fourth-year students from the Department of Mechanical and Mechatronic Engineering and one postgraduate student from the Department of Process Engineering (with a different nationality and academic background).

Session 4: Two postgraduate students from the Department of Mechanical and Mechatronic Engineering and one postgraduate student from the Department of Process Engineering (with different nationality and academic background).

All the participants were volunteers and the remuneration for the student participants were a pizza meal.

6.3.1.3. Case Study Hardware and Software

All case study participants used similar computer hardware: the computers ranged from Intel Pentium 4 1.2 GHz to 2.8 GHz, 256 Mb to 521 Mb RAM.

The operating system on each of the computers was Windows XP Professional with Service Pack 2. The network that connected the computers to each other was the Stellenbosch University Network, a 100Mbps Ethernet network.

DiDeas II client side software was installed on each of the participant's computers providing a user interface for the project manager and the designer to work on. The DiDeas server side software and ontology were installed on another computer. This computer acted as the server workstation for receiving client requests and sending data to the client computer.

Vypress Chat was installed for local network chatting and file transfer if necessary. It is shared real-time chatting software that can be used in local networks, and was developed by VyPRESS Research LLC (VyPress company home page, http://www.vypress.com, [2006]). Figure 6-1 shows the user interface of this software. Vypress Chat requires a local network or the Internet, and all users have equal access to the application's tools and functions. Vypress Chat enables individuals to communicate instantly about joint projects, and managers can send instantaneous memos to individuals.

The communication between team members through Vypress Chat was captured but not analysed in the case study, since that communication was unstructured and therefore of limited long term value.

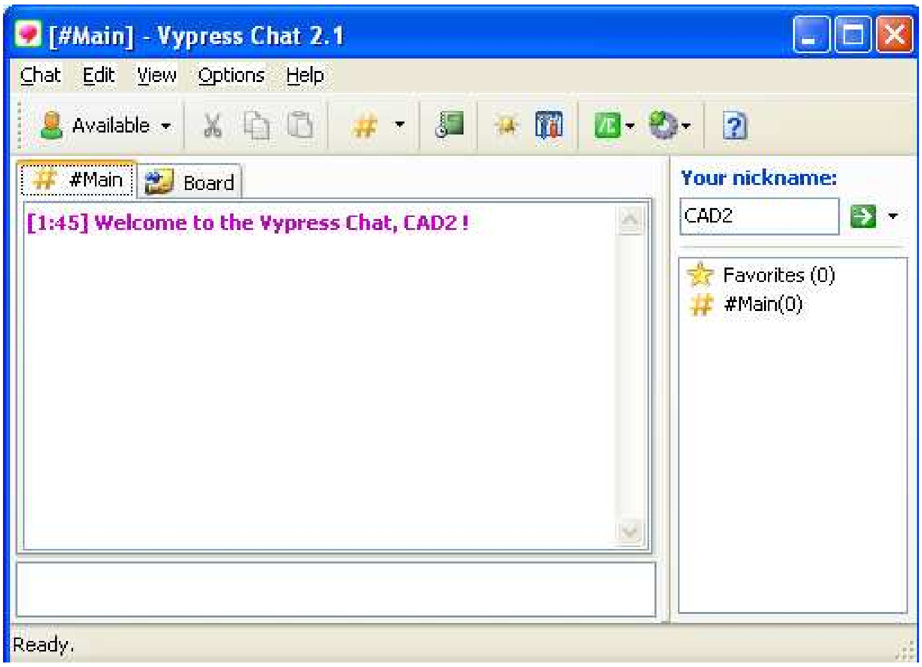

Figure 6-1 Vypress Chat 2.1 User Interface

ACDSee is a comprehensive application ideal for acquiring, organising, viewing, enhancing and sharing your images, developed by ACD System Ltd. (http://www.acdsystems.com, [2004]). In this case study, it was used for viewing image files.

Microsoft Notepad is a standard text editing accessory tool included in Windows XP Professional. In this case study, it was used for design review and design note editing.

Each participant had a WACOM Graphic Tablet with a special pen to create images. Figure 6-2 shows the Painter Classic accompanying software and Figure 6-3 shows the tablet itself. This system consists of a tablet with a sensitive area of approximately 127 mm by 102 mm, and a pen with a touch-sensitive tip and a control button.

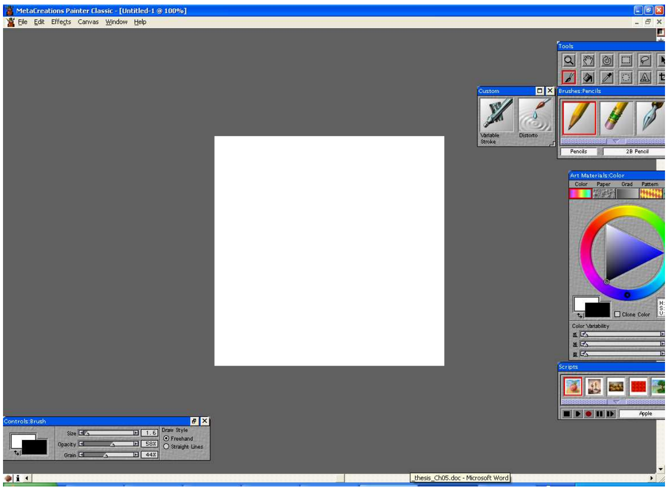

Figure 6-2 User Interface of Painter Classic

Figure 6-3 Paint Classic Sensitive Touch Board and Pen

Painter Classic was used as sketch generation software and images were created by using a mouse or a graphic tablet. The user can create graphics by drawing on the sensitive board with different drawing options that is shown on the Paint Classic user interface.

6.3.1.4. Data collection

All the data were collected from interviews, memo papers, design notes, questionnaires and observations (memo papers were used to record the user's feeling and to write down some advice during the design process). The interviews with the postgraduate participants who had more design experience were carried out individually. Group interviews were held with the undergraduate participants who had less design experience in order to extend the scope of their experience. A tape recorder was used in each interview to assist data recollection and accuracy. Each participant had a memo book in which comments or design notes could continually be written so that they could be analysed after the case study.

Each participant had to complete a questionnaire after the case study. An example of the questionnaire is given in Appendix C. All group meetings were recorded on video tape and were analysed after the case study.

All the sessions of the case study were conducted under supervision of the author, and all group meetings were recorded by a video camera.

6.3.2. Case Study Procedure

The duration of each case study was limited to three hours during which the users had to create system and subsystem customer requirements, engineering requirements and a quality house (QFD) in order to manage the different levels of design information at the early design phase.

6.3.2.1. Short Introduction Presentation

At the beginning of each session of the case study, the participants received an introduction to the project as well as a presentation on the operation of the DiDeas II system. All the software that would be needed for the case study were also introduced. There was also time for the participants to get answers to any questions that they might have. The time for this introduction ranged from 20 minutes to 40 minutes depending on the levels of experience of the participants with the hardware and the software. Participant were each given a free time of up to 40 minutes to edit an example project on the system to be familiarise themselves with the design process and the functions that the system could provide.

6.3.2.2. Design Methodology

All the participants had knowledge of the design methodology recommended by Ullman [2003]. The following design procedure to be applied to the DiDeas II was communicated to each participant:

1. Project needs collection and refinements
2. Engineering requirements generation
3. Project functions analysis
4. System breakdown to subsystems according the functions analysis
5. Engineering requirements generation
6. QFD analysis for ensuring the developed specifications cover all the customer requirements

The main focus here was to try to identify whether the ontology could handle different levels in specification development in a flexible manner. Due to the limited time, only one or two of the subsystems were developed.

6.3.2.3. Design Tasks

It was not possible to carry out all the design steps in full detail as would be the case during the design process in a commercial company. The duration of the case study was limited to three hours due to practical considerations.

The participants in Session 1 had to finish the specification development of a bicycle system and subsystem. The support subsystem specifications were developed until the phase where the QFD was built.

The participants in Session 2 continued the work done by the group in Session 1. The main aim was to see whether the system could assist participants in understanding a design project that had been carried out by another group and if they could then develop another subsystem.

Each group in Sessions 3 and 4 had the task of designing two similar projects: a bottle and can separator, and a coin separator. The first task of one group (in Session 3) was to have a group meeting for developing the bottle and can separator specifications, while the first task of the other group (in Session 4) was to develop the coin separator project by using the DiDeas II system. The second task of the Session 3 group was to design the coin separator by using the DiDeas II system, while the second task for the Session 4 group was to hold a meeting to accomplish the bottle and can separator specification development. For each of the two groups, the outcomes for the meeting as well as for using the DiDeas II system were to finish the possible subsystem specification of the project and to accomplish the QFD matrix for the specific subsystem. These two projects were similar in terms of their functions. They were selected so that the group that were using the DiDeas II system could be compared with the group that were not using the DiDeas II system.

The first project was a subsystem of the refuse recycling project, a fictitious project with the aim to recycle useful materials from the rubbish bins in the Stellenbosch University Student Centre. The requirements of the first project were as follows:

1. Ten rubbish bins can be used for collecting the recyclable materials.
2. The recycle process will be daily except for the weekends.
3. All the recyclable materials should be put into different containers for the process that is to follow.
4. The entire possible subsystem can be operated or controlled by a human.

The bottle and can separator was assigned as one subsystem of the whole recycling process, and the technical specifications for this subsystem had to be finished.

The second project was to design a coin separator, which also was a subsystem of a bank note and coin sorting system. The overall requirements had to be completed and refined. The coins separator system was assigned as one of the subsystems, and the specifications of this subsystem had to be developed and the QFD matrix had to be completed to ensure that all the customer requirements were considered. The requirements of this project were the following:

1. Try to separate daily the bank notes and coins for one regional bank.
2. List all the possible subsystems solutions.
3. One hundred thousand notes and coins have to be processed.

4. The process will be daily except for weekends.

5. The entire possible subsystem can be operated or controlled by a human.

6. The power for the possible devices that would perform the subsystem functions would be electricity and human power.

7. R5, R2, and R1 coins have to be separated into different containers.

The system reliability and the server synchronous access were also investigated by the groups in each session.

6.3.2.4. Completion of the Questionnaires

After each case study, all the participants were required to complete a questionnaire and to hand in their notepaper and sketches for later analysis.

6.3.2.5. Interviews with Participants

After each case study, each group member was either interviewed or was part of a group discussion depending on the level of their design experience. A tape recorder was used to ensure the accuracy of the data.

6.3.3. Case study results

DiDeas II was evaluated in terms of its functionality and the interface performance to ascertain whether the research objectives were achieved.

6.3.3.1. System Functionality

The time consumed by these four case studies is listed in Figure 6-4. Session 1 took more time compared to the other sessions – it was a new project and the participants don’t have enough design experience. Session 3 took a slightly longer time than Session 4 – the participant from process engineering in session 3 did not have enough background of system engineering and design methodology, while the participant from process engineering in session 4 had certain knowledge of system engineering.

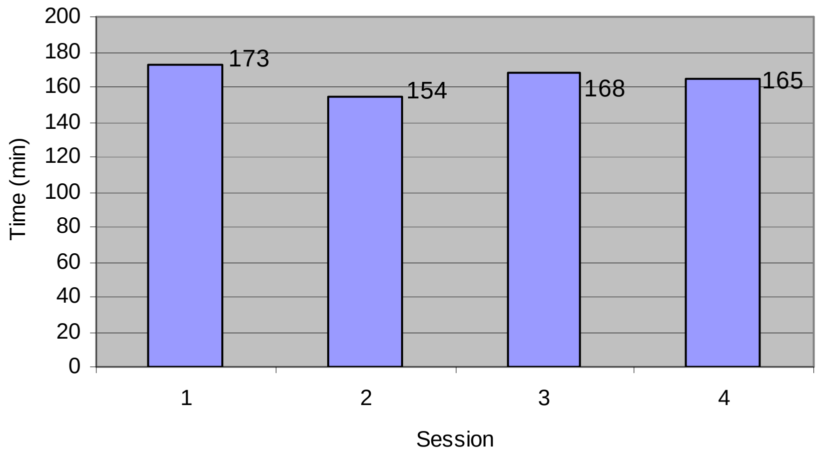

Figure 6-4 Time Consume of Case Studies

The time that the team leader in each session spent on the four activities are shown in Figure 6-5, as a percentage of the total time he spent in each session.

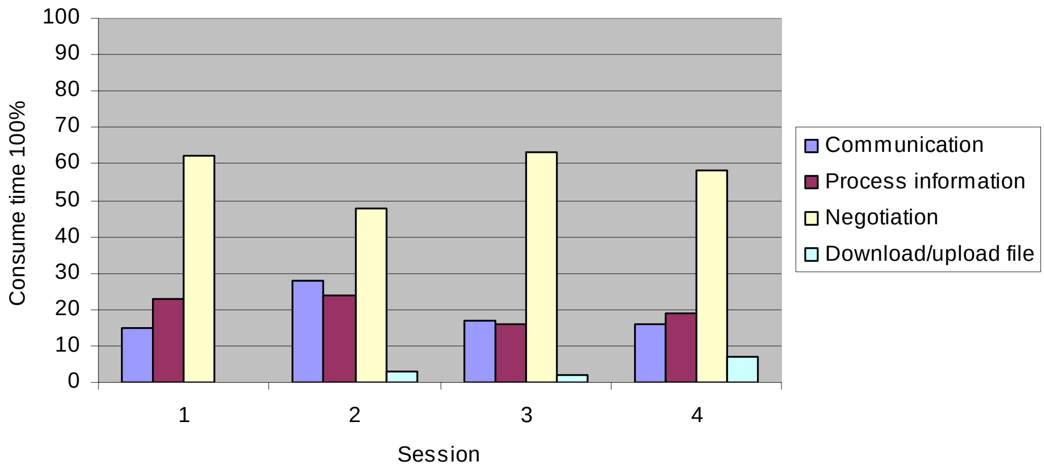

Figure 6-5 Different Activities in Case Studies

There were four main activities during the design process:

- Communication: including a designer assigning different tasks to different people, free chat with designer etc.
- Process information: read the data and enter the design information, for example create/edit tree node (subsystem).
- Negotiation: explain how and why one node or specification is created.
- Download/upload file: some drawing files can assistant the negotiation.

Figure 6-5 clearly shows that Negotiation is the major activity when using DiDeas II. "Process Information" in Figure 6-5 is equivalent to "Exchange Information" in the case study in Section 3.3. It is notable that the percentage of time spent on these aspects in Figure 6-5 is more or less the same as in Figure 3-3, even though the case study in Section 3.3 was done co-located and the one presented here was done in a simulated distributed context. The two case studies are, however, not directly comparable because the one in Section 3.3 extend over a number of meetings, while in the one discussed here each team worked only for a single session.

Session 2 continued the design from Session 1, and had to design another subsystem. Interviews with the participants clearly showed that DiDeas II's ability to capture the linkages between the subsystems and their specifications, navigated through the tree view, was an important factor in their understanding of the work done in Session 1. Session 2 took a shorter time than Session 1, even though Session 2's participants had to review the work done in the previous session. Comments in the interviews indicate that having an easily understandable example helped them to understand how to do the specification development. All of them agreed that project ontology could handle the information and can save and show the data associated with any subsystem.

The project requirements and specifications were developed and saved on the server (see Figure 6-6), and the QFD analysis was carried out to check that the requirements were satisfied (see as Figure 6-7).

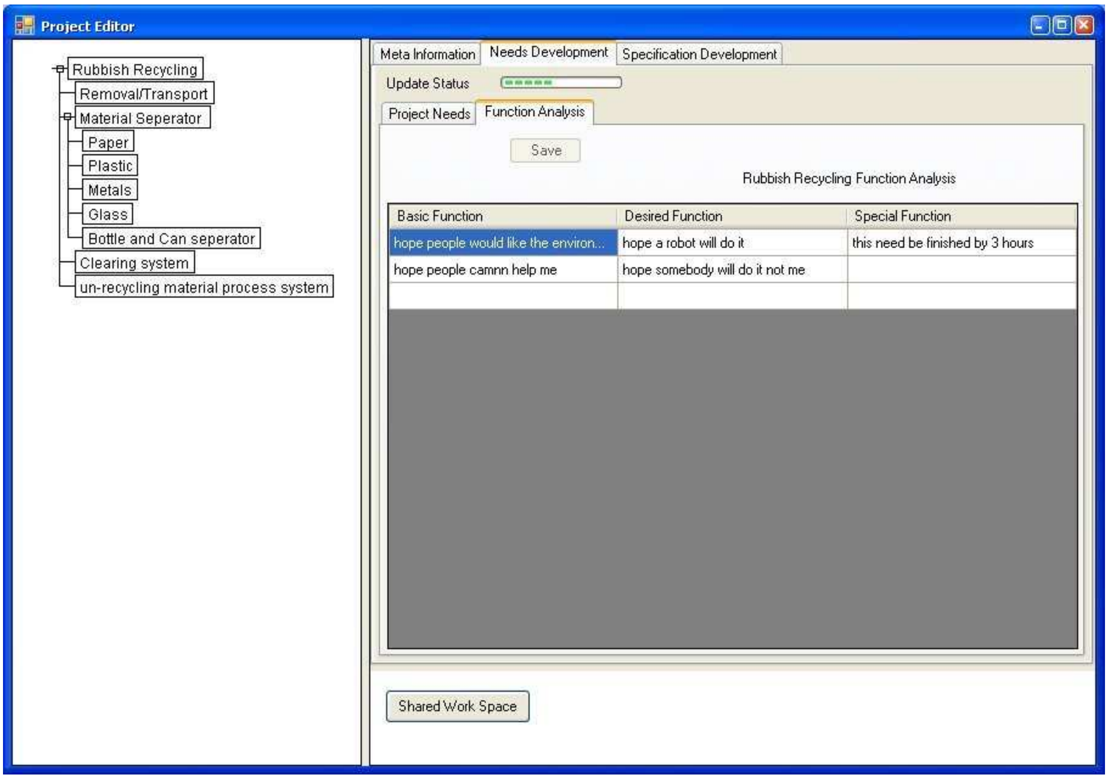

Figure 6-6 The Case Study Project Design Interface

From the interviews with the participants in Session 2, it was found that the participants could in fact easily understand the previous work done by the participants in Session 1. By reading through the design history that was recorded in the project database, a clear view of the system structure could be captured from the tree view and all the requirements and specifications were also heuristic in nature to the participants since they were linked with different nodes of the system-subsystem tree.

Figure 6-7 compares the time spent in the group meetings (without using DiDeas II) in Sessions 3 and 4. It was found that after using the DiDeas II system, the meeting for the team in Session 4 lasted 125 minutes. However, the meeting of the participants in Session 3 that lacked the experience of using DiDeas II lasted 146 minutes for the same project. Because of the previous experience from using the DiDeas II system, the meeting time was shortened by 16.7%. From the interviews, it became clear that the team formed their own understandable terms after using DiDeas II to improve their communication.

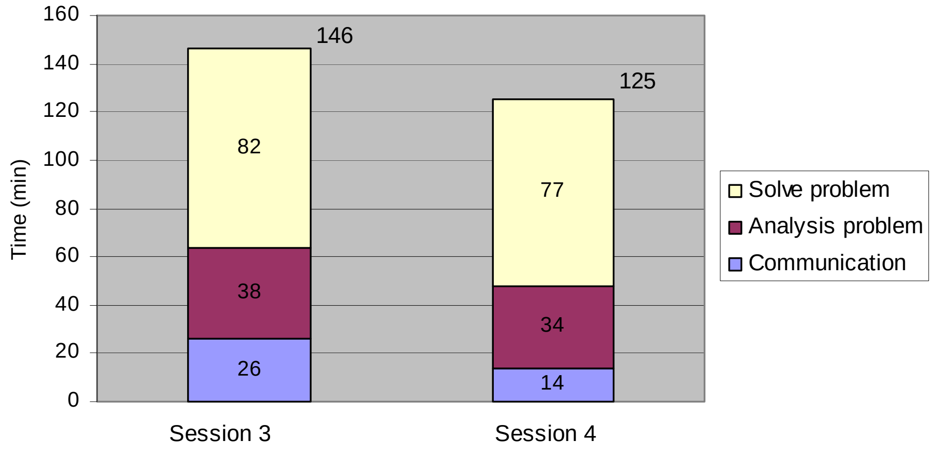

Figure 6-7 Time Consume in Session 3 and 4

The group that had the experience of using DiDeas II produced clear specification documentation following the development process (especially with the QFD analysis), and also generated three concepts according to the systematic 14 specifications. The group that had not used the DiDeas II system before developed two concepts, and it was difficult to find how these concepts were generated due to the unclear documentation of the specification development. The design notes of the group without previous experience of using DiDeas II and the group that did have experience of using DiDeas II are shown in Figure 6-8 and Figure 6-9 respectively. Personal style is one aspect that plays a role in a design note. Another aspect, deduced from the interviews, is the experience with using DiDeas II help to create a systematical record, which leads to a clear and useful system structure.

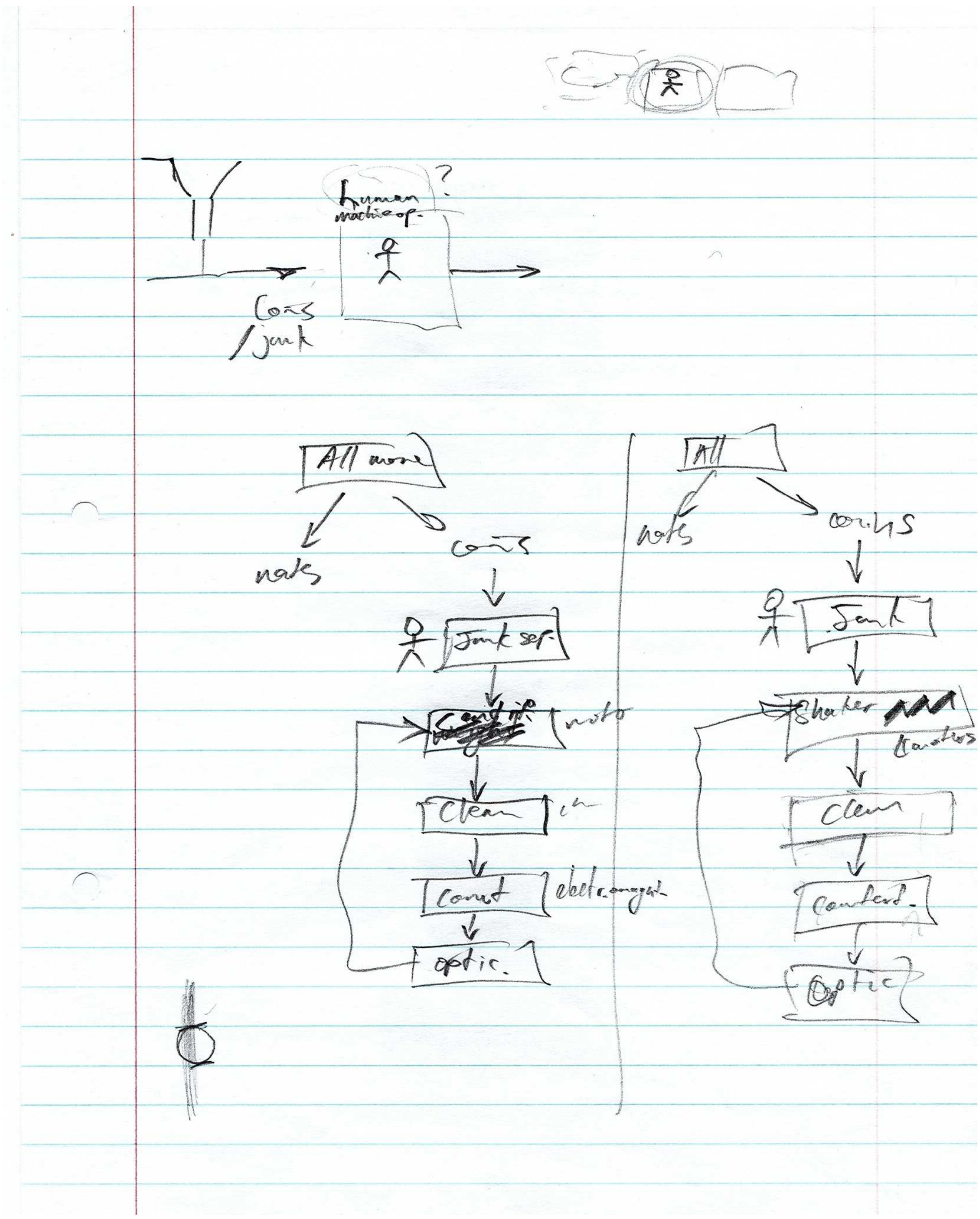

Figure 6-8 Design Note of the Group without DiDeas II Experience

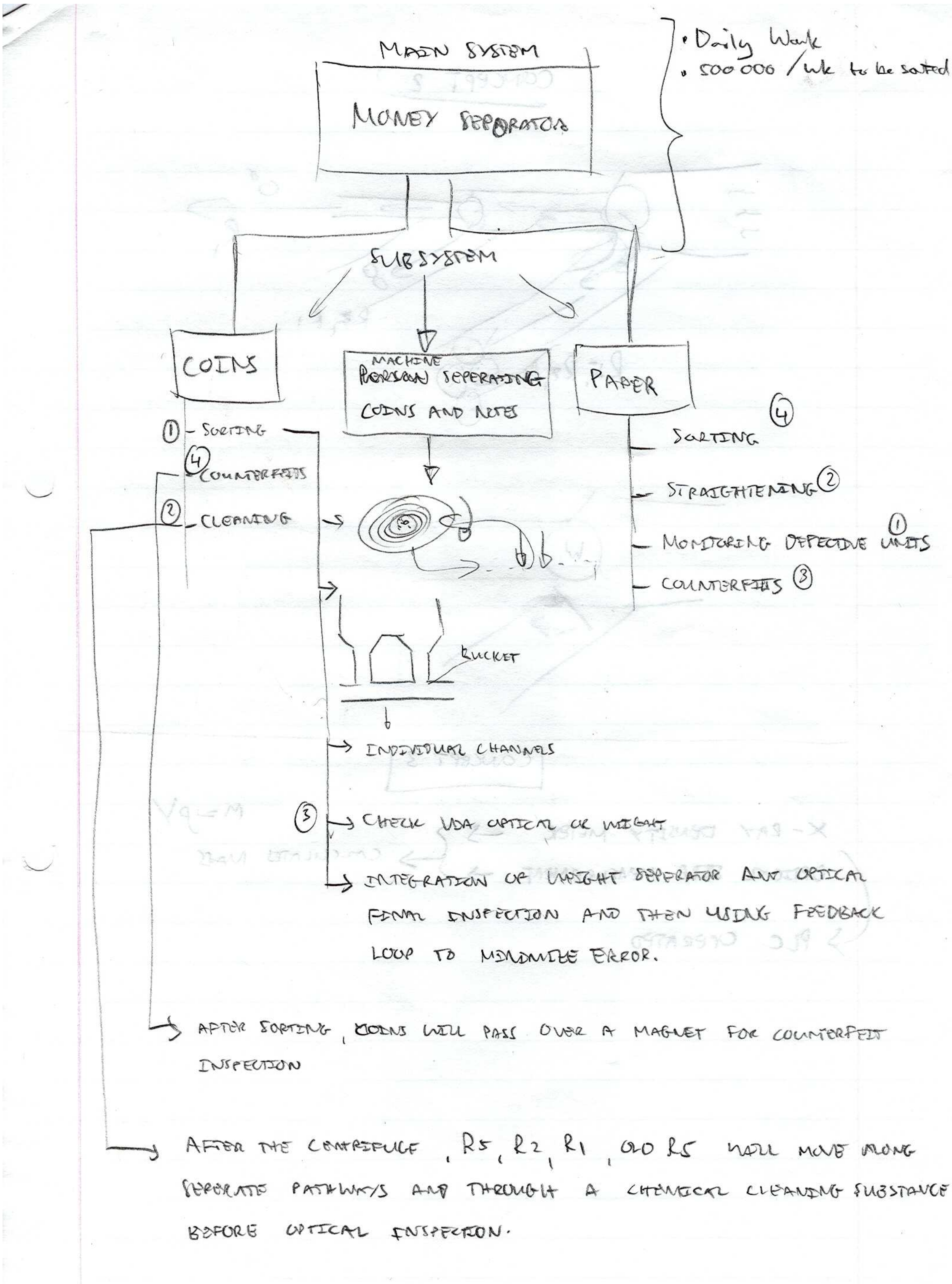

Figure 6-9 Design Note of the Group with DiDeas II Experience

All the participants felt that the QFD analysis table could help them when they are inspecting the development of the engineering requirements against the customer requirements. However, the direction of the text on the screen made it difficult to read.

This can be attributed to a limitation in the programming with GDI and unprofessional programming skills. Figure 6-10 shows one of the QFD windows in the case study.

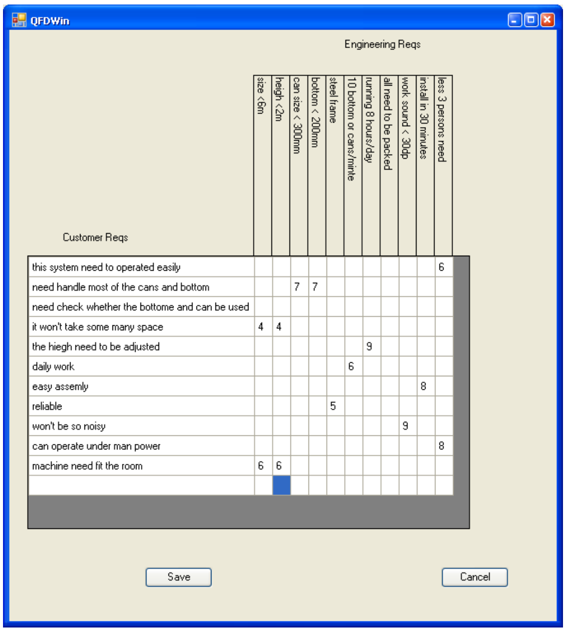

Figure 6-10 The Case Study QFD Window

6.3.3.2. System Performance

All the participants who had previous experience of being part of a design team took less than 15 minutes to understand the system. Only the two participants who came from different fields had problems understanding the system and it took them 30 minutes to get familiar with the system. However, none of the participants had difficulty in understanding how to approach the design process after they had read through the system structure and the different levels of design information.

Some participants experienced problems in getting the newest data from the server side because of the different settings for the data updating interval setting. The time interval of data updating to the server was set according to personal preference. Different data updating interval settings lead to time being consumed by conversions between participants to confirm whether the data was updated. This shows that having an immediate updating option is required.

Another experience came from the team leader; it was difficult for him to determine the progress of every team member. A set of progress bars that shows different people's work progresses can assist the team leader in this regard.

None of the participants had any problems with the Design Review Interface, and they could easily read and create a design review or design notes through the Design Review Interface. Through the Share Work Space Interface, pictures and Word format files could be uploaded and downloaded between the server and the local computer. However, this system was only successful when dealing with files with a size of less than 100 kb due to programming problem. Participants had to use the Vypress Chat (the chat tool illustrated in Figure 6-1) to download and upload files with a size of more than 100 kb.

Eighty five percent of the participants preferred to draw and sketch with a pencil and paper instead of using Paint Classic due to the inconsistent operations when drawing and viewing. This would indicate that a scanner would be useful additional equipment for the DiDeas system.

In summary, the DiDeas II system has the capability to aid a design distributed team during the requirement and specification development phase and to record design information in a systematic manner.

6.4. CASE STUDY OF POTENTIAL INDUSTRIAL APPLICATIONS

This case study was aimed at the possible application of DiDeas II in a real industrial company. One research engineer from Stellenbosch University who does mechanical design on a part time consulting basis and two other experienced designers from small manufacturing companies also investigated the interface's ability to accommodate different design procedures.

6.4.1. Case Study Procedure

DiDeas II was not directly applied in the small manufacturing companies due to the heavy work load of the participants. DiDeas II was demonstrated to the participants using the data from the previous study (discussed in section 6.4) and all the participants were interviewed to investigate to what extent DiDeas II could be adapted in real industrial companies.

6.4.2. Case Study Results

The interviews with all the participants indicated in general that showing the physical system/subsystem structure in a tree format was considered to be a valuable feature as it provides a clear and understandable view of the system. The ability to adapt the style and terminology used in DiDeas II was also seen as helpful for designers to communicate effectively with each other.

The first participant came from a small company designing and manufacturing cooking equipment for the canning industry. The products in this company are relatively simple in terms of designing and manufacturing. However, their design records were not well organised with the result that this company had problems with design data reuse e.g. some of the product modifications required product redesigns due to the insufficient historical design information. This often occurred when the product was designed by another designer or the product was designed a long time ago. Using DiDeas can help to record such how and why design information and aid in product modification especially for a designer who was not involved before. This participant also indicated that he would consider using DiDeas II as a tool for process planning. Since DiDeas II allows designers to use their own terminology to define the interface, this opens the opportunity to adapt DiDeas II to be used not only in mechanical design.

The other two participants do detail design and development of subsystems for other companies. Most of their projects are consulting projects and seldom the completed design of a new product. The comments from these two engineers clearly indicated that DiDeas II can provide a flexible way to handle the system-subsystem structure and can provide a clear view of how specification of a subsystem was developed. Both participants spent considerable time in meetings aimed specifically at reporting progress to a project manager or a client. In a distributed environment that DiDeas II provides, the amount of time can be reduced, and the design efficiency can be improved. The participants also expressed interest in that DiDeas II allows designers to use their own terminology when defining the user interface, thereby forming a company-style.

These companies also would have interest in using DiDeas II as a training tool for novice designers – in DiDeas II, the design information can be saved in a systematic way and they can adopt their own terminology, which helps to familiarise novice designers with the company's design process and design terminology.

Another design interface was produced based on the comments from two of the participants (see Figure 6-11). This design interface contains the following design processes and design steps:

- Problem Identification
  - Problem List
  - Problem Classification
- Problem Analysis
  - Problem Formulation
  - Problem Reason Analysis
- Problem Solution
  - Principles Solution
  - Solution Specifications

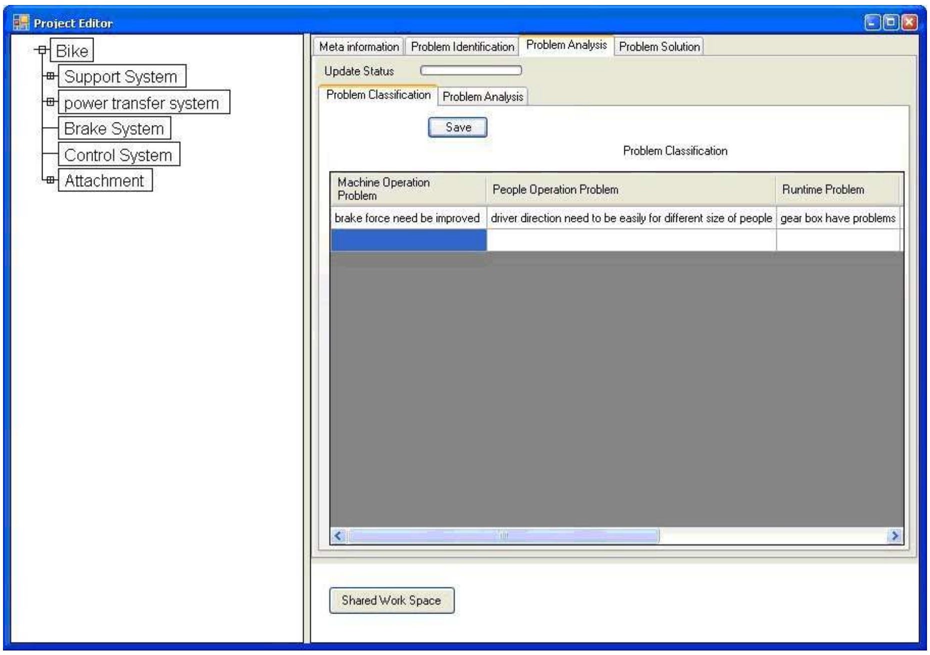

Figure 6-11 Problem-Based Approach Design Interface

First, the system is divided into subsystems and all the problems need to be solved are listed with linking to the top node. The sub-problems are also linked to the nodes in the tree view. In the Problem Classification process, a problem catalogue (like a checklist) is used to classify different problems into different groups such as classifying the problem according to the required rank.

In the Problem Analysis phase, there are two possible design steps:

Problem Formulation: the essential problem that needs to be solved (the highest rank in problem classification) is defined in the form of design goals to be achieved.

Problem Reason Analysis: from the participant's comments, the industrial companies mentioned above always encounter the situation that the client requires a clear view of what caused the problem. This step will provide such information especially to the client and focus on the interfaces between different subsystems.

As a project output, Problem Solution here will provide the solution principles and solution specifications for further design or manufacturing.

This approach can be applied easily by changing the ontology elements and relations in the Ontology Editor. Although this approach is simple in the current implementation and can be improved, it showed that, in practice, this system can implement other design methodologies and terminologies. The terminology applied in a single team or company must be understood by all of the team members, but different terminologies can be applied in different design teams or companies.

7. CLOSURE

7.1. CONCLUSION

The economic success of manufacturing is dependent on the ability to identify the needs of customers and to create quickly the products that meet these needs and can be produced at a low cost. It is generally accepted that the majority of the product cost is committed by the end of the conceptual design phase. The design of complex engineering systems is increasingly becoming a collaborative task among designers or design teams that are physically, geographically and temporally distributed. Many tools are available to support the final design phases, such as CAD, CAE and FEA. However, few existing tools support the early design phases since the information at this phase is fuzzy and incomplete, which makes the design process quite difficult and challenging.

DiDeas I was developed by the Department of Mechanical and Mechatronic Engineering at Stellenbosch University to support the capturing and exchange of design information during the early phases of design. As an extension of the development of DiDeas I, the present research, as implemented in DiDeas II, was intended to provide a distributed, flexible, early design assistant system for specification development and concept generation in small companies which normally have small budgets.

Four case studies were conducted using DiDeas I to determine the development requirements for DiDeas II. They showed that the design records in small and medium companies are difficult to understand. One approach for capturing such fuzzy and implicit design information is to use an ontology and conceptual graphs to formulate elements and relationships adapted to the particular application domain. Ontology has received considerable recognition as a useful mechanism in many different fields. DiDeas II was therefore developed using these notions in order to achieve flexibility, i.e. the ability to adapt to a company's unique design procedure and terminology, and the ability to extend the design procedures without affecting data already captured. To achieve such flexibility, an ontology-based approach was used for the database structures in DiDeas II and to adapt the user-defined interface style to different design procedures and terminologies.

DiDeas II uses the Internet as communication medium to allow geographically dispersed team members to work together. DiDeas II is also aimed at improving design productivity by promoting design reuse and teaching a new designer the design procedure of a company, through the customizable user interfaces. The project manager can adapt the technical terminology in DiDeas II to that used in his/her own organisation, and by doing so can improve the communication between different designers with different educational backgrounds.

Two case studies were carried out to evaluate the system. These case studies showed the following: DiDeas II can handle design information, especially the requirement and specifications, on different levels of the project system/subsystem hierarchy. By using a tree view, tabbed views and tables, the design information in the early design phases can be displayed in a systematic and understandable manner. This leads to an improvement in the designer's work efficiency, especially during redesign.

The case studies indicated that DiDeas II potentially has the flexibility to adopt different design procedures and terminologies used in different companies, but it was not feasible to test this fully within the scope to this dissertation. This capability of DiDeas II is expected to give a design team the freedom to use the terminology that they are familiar with, when communicating with each other.

The research presented here therefore has shown that a flexible design information system to manage the early design information can be built by using an ontology-based approach. Further evaluation in real industrial applications is, however, required to assess whether the approach is commercially viable and whether a company or design team will find it worth the effort to set up the ontology to reflect its design style.

7.2. FUTURE WORK

Based on the development experience and the results of the case studies, several possibilities for future development have been identified:

- Professional programmer cooperation is required to improve the reliability of the system.
- Providing for relations between the different subsystems that have one parent (i.e. inter-sibling relations) should be handled by an extended development of DiDeas II. This requires extensions to the user interface, but not the ontology database or project database.
- Investigating whether it is feasible to omit the element types in the definition of a relation type, which currently is a potential source of inconsistencies in the database.
- The ability for more than one team member to edit the same subsystem at the same time should be developed.
- More reliable and versatile file upload and download facilities should be provided.
- Extending DiDeas II to handle concept generation and evaluation can be investigated.
- DiDeas II allows designers to work on only one project at a time. It could be developed to allow designers to access different projects on the server side.
- Other possibilities for using DiDeas II, such as manufacturing planning, could be investigated.
- Report generation facilities, e.g. for printing the tree view and the table information in different layers, could be provided.

8. REFERENCES

ACD System Ltd HomePage, 2005, http://www.acdsystems.com, visited 15.08.2004.

Akiyama, K., 1991, Function Analysis: Systematic Improvement of Quality Performance, Productivity Press, Cambridge.

Anderl, R., Ott, T., Lindemann, U., Gaul, H.-D. and Gierhardt, H., 1999, 'Planning and Improvement of Distributed Product Development Processes by using a Taxonomy System', *Concurrent Engineering: From Product Design to Product Marketing*, 6th European Concurrent Engineering Conference, Erlangen, Germany, pp. 69-75

Andreasen, M.M., 1992, *Designing on a Designer's Workbench*, Proceeding of the 9th WDK Workshop, Rigi, Switzerland.

Babu M. Joglekar N. Ganiji A. Ramani K., 2003, *Flexible Software Framework for Collaboration Systems*, Collaborative Design Tools, International CIRP Design Seminar, Grenoble, France, May 12-14, 2003

Basson, A.H., Bonnema, G.M. & Liu, Y., 2003, *A Flexible Design Information System*, 2003 International CIRP Design Seminar Laboratoire 3S, Grenoble, France, May 12-14.

Begelinger, R.E., Post, E., Van Houten, F.J.A.M. & Kals, H.J.J., 1999, *Product Architecture Development Enabling Integrated Re-design of Mechanical Products*, Integration of Process Knowledge into Design Support Systems, 1999 CIRP Design Seminar, March 24-26, pp. 189-199.

Brickley, D and R.V. Guha, 2007, http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/, visited 05, 2007

Bentley, R., Appelt, W., Busbach, U., Hinrichs, E., Kerr, D., Sikkel, K., Trevor, J. & Woetzel, G., 1997, *Basic Support for Cooperative Work on the World Wide Web*, International Journal of Human Computer Studies, Vol. 46, No. 6, pp. 827-846.

Bernaras, A., Laresgoiti, I. & Corera J., 1996, *Building and Reusing Ontologies for Electrical Network Applications*, Proc. European Conference on Artificial Intelligence ECAI 96, Budapest, Hungary, pp. 298–302.

Blanchard, B.S. & Fabrycky, W.J., 1998, Systems Engineering and Analysis, Prentice Hall, Upper Saddle River, NJ.

Bonnema, G.M. & Van Houten, F.J.A.M., 2003, *Conceptual Design in a High-Tech Environment*, Proc. International CIRP Design Seminar 2003. Laboratoire 3S, Grenoble, France, May 12-14.

Boujut J.F. and Laureillard P., 2002, *A co-operation framework for product–process integration in engineering design*, Design Studies, Volume 23, Issue 6, November 2002, Pages 497-513

Bracewell, R.H. & Sharpe, J.E.E., 1996, *Functional Descriptions used in Computer Support for Qualitative Scheme Generation – "Schemebuilder"*, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 10, No. 4, pp. 333-345.

Chen, K.H., Chen, Si., Lin, L. & Changchien, S.W., 1998, *An Integrated Graphical User Interface GUI for Concurrent Engineering Design of Mechanical Parts*, Computer Integrated Manufacturing System, Vol. 11, pp. 91-112.

Cherng, J-T., Chiou, S-J. & Hsu, Y-L., 1997, *Information Service and Sharing for Distributed Collaborative Mechanical Design*, Advances in Concurrent Engineering, August 20-22, pp. 199-206.

China XML, 2007, http://bbs.w3china.org/dispbbs.asp?boardID=2&ID=44566, visited 05, 2007

Corazzon, R., 2000, *Descriptive and Formal Ontology*, [Online], Available: http://www.formalontology.it/, [20 August 2003].

Court, A.W., Ullman, D.G. & Culley S.J., 1998, *A Comparison Between the Provision of Information to Engineering Designers in the UK and the USA*, International Journal of Information Management, Vol. 18, No. 6, pp. 409–425.

Cross, N., 1994, Engineering Design Methods: Strategies for Product Design, 2nd ed, Wiley., NewYork

Curry, A. & Stancich, L., 2000, *The Intranet – an Intrinsic Natural Component of Strategic Information Management?*, International Journal of Information Management, Vol. 20, pp. 249-268.

Ding Y., Fensel, D., Kevin, M. & Omelaynko, B., 2002, *The Semantic Web: Yet Another Hip?,* Data & Knowledge Engineering, Vol. 41, pp. 205–227.

Eekels, J. & Poelman, W.A., 1995, Industriële Productontwikkeling, Deel 2: Methodologie, Lemma, Utrecht.

Fraser, S.R.G., 2003, Managed C++ and .Net Development, Apress, NewYork.

Fruchter, R., Reiner, K.A., Toye, G., & Leifer, Li., 1996, *Collaborative Mechatronic System Design*, Concurrent Engineering: Research and Applications, Vol. 4, No. 4, pp. 401-412.

Giannini, F., Monti, M., Biondi, D., Bonfatti, F. & Monari, P.D., 2002, *A Modelling Tool for the Management of Product Data in a Co-design Environment*, Computer-Aided Design, Volume 34, Issue 14, pp.1063-1073.

García-Sánchez, F., Martínez-Béjar, R, Contreras, L., Tomás, J., Breis, F. & Castellanos-Nieves, D., 2005, *An Ontology-based Intelligent System for Recruitment*, Expert Systems with Applications, Vol. 31, pp. 248–263.

Giannini, M., Monti, D., Dionti, F., Bonfatti, P.& Monari, P., 2001, *A Product Model Enabling SMEs to Co-operate in a Distributed Engineering Environment*, Proc. of ICED 2001, Lyon France, July.

Guarino, N., Carrara, M. & Giaretta, P., 1995, *Ontologies and Knowledge Bases: Towards a Terminological Clarification*, in: N. Mars (Ed.), Towards Very Large Knowledge Bases, Knowledge Building and Knowledge Sharing, IOS Press, Amsterdam, pp. 25–32.

Gruber, T.R., 1993, *A Translation Approach to Portable Ontology Specification*, Knowledge Acquisition Vol. 5, pp. 199–220.

Hague, M.J., Taleb-Bendiab, A. & Brandish, M.J., 1996 *An Adaptive Machine Learning System for Computer Support Conceptual Engineering Design*, in J. Sharpe (Ed.), AI System Support for Conceptual Design - Proceedings of the 1995 Lancaster International Workshop on Engineering Design, Springer , London.

John F. Sowa, 2006, http://www.huminf.aau.dk/cg/Module_I/1002.html visited in Nov 2006

Kitamura, Y. & Mizoguchi, R., 2003, *Ontology-based Description of Functional Design Knowledge and its Use in a Functional Way Server*, Expert System with Application, Vol. 24, pp. 153-166.

Kitamura, Y., Kashiwase, M., Fuse, M. & Mizoguchi, R., 2004, *Deployment of an Ontological Framework of Functional Design Knowledge*, Advance Engineering Informatics, Vol. 18, pp. 115-127.

Lee, T., Lee, I.H., Lee, S., Lee, S.G., Kim, D., Chun, J.H., Lee, H. & Shim, J., 2005, *Building an Operational Product Ontology System*, Electronic Commerce Research and Applications, November, pp. 169-181.

Lee, T. B. Handler, J., & Lassila, O., 2001, *The Semantic Web*, Scientific American, May.

Lutters, E., 2001, Manufacturing Integration Based on Information Management, Ph.D. Thesis, University Twente.

McNeil, T., Gero, J.S. & Warren, J., 1998, *Understanding Conceptual Electronic Design Using Protocol Analysis*, Research in Engineering Design, Vol. 10, pp. 129-140.

Neches, R., Fikes, R.E., Finin, T., Gruber, T.R., Senator, T. & Swartout, W.R., 1991, *Enabling Technology for Knowledge Sharing*, AI Magazine 12 No.3 pp. 36–56.

O'Grady, P., Pamers, D. & Bolsen, J., 1988, *Artificial Intelligence Constraint Nets Applied to Design for Economic Manufacture*, Computer Integrated Manufacturing, Vol. 1, pp. 204-210.

Pahl, G. & Beitz, W., 1997, Engineering Design – A System Approach, 2nd ed., Springer, London.

Pahng, G.D.F., Bae, S. & Wallace, D., 1998, A Web-based Collaborative Design Modelling Environment, Proceedings of the IEEE Workshops on Enabling Technologies Infrastructure for Collaborative Enterprises WET ICE' 98, pp. 161-167.

Pels H.J., 1996, Computers in Industry, Elsevier. NewYork.

Poltrock, S.E. & Engelbeck, G., 1999, *Requirements for a Virtual Collocation Environment*, Information and Software Technology, Volume 41, Issue 6, pp. 331-339.

Riboulet, V., Marin, P. & Léon, J., 2003, *Distributed Design – Current Approaches and Evolution Tendencies*, 2003 International CIRP Design Seminar Grenoble, France, May 12-14.

Robert, M.C. & Dampney, C.N.G., 2005, *An Approach to Ontology for Institutional Facts in the Semantic Web*, Information and Software Technology, Vol. 47, pp. 775–783.

Rodgers, P.A., Huxor, A.P. & CaIdwell, N.H.M., 1999, *Design Support Using Distributed Web-based Al Tools*, Research in Engineering Design Vol. 11, pp. 31-44.

Roy, U. & Kodkani, S.S., 1999, *Product Modelling within the Framework of the World Wide Web*, IEEE Transactions Vol. 31, No. 7, pp. 667-677.

Rozenburg, N.F.M. & Eekels, J., 1995, Product Design: Fundamentals and Methods, John Wiley & Sons, Chichester, U.K.

Schueller, A. & Basson, A.H., 2001, *A Framework for Distributed Conceptual Design*, Design Management - Process and Information Issues, Proceedings of the 13th International Conference on Engineering Design ICED01, Glasgow, U.K., pp. 385-391.

Schueller, A., 2002, Aspects of Distributed Conceptual Design Support, Ph.D. Thesis, Stellenbosch University.

Sowa, J.F., 1992, *Conceptual graphs summary, Conceptual Structure*, in: Nagle, T.E., Nagle, J.A., Gerholz, L.L. & Eklund, P.W (Ed.), Current Research and Practice, Ellis Horwood Ltd., Chichester.

Steven E. P. and George E., 1999, *Requirements for a virtual collocation environment*, Information and Software Technology, Vol. 41, Issue 6, 25 April, pp. 331-339

Storey, V.C., Dey, D., Ullrich, H. & Sundaresan, S., 1998, *An Ontology-based Expert System for Database Design*, Data & Knowledge Engineering, Vol. 28, pp.31-46.

Studer, R., Benjamins, V.R. & Fensel, D., 1998, *Knowledge Engineering: Principles and Methods*, Data and Knowledge Engineering, Vol. 25, pp. 161–197.

Suh, N.P., 1990, The Principles of Design, Oxford Univ Press., London.

Sugumaran, V.C. & Storey, V.C., 2002, *Ontologies for Conceptual Modelling: their Relations, Use and Management*, Data & Knowledge Engineering, Vol. 42, pp. 251–271.

Swartout, B., Ramesh, P., Knight, K. & Russ, T., 1997, *Toward Distributed Use of Large-Scale Ontologies*, AAAI Symposium on Ontological Engineering, Stanford.

Szykman, S., Racz, J., Bochenek, C. & Sriram, R.D., 2000, *A Web-Based System for Design Artefact Modeling*, Design Studies Vol. 21, pp 145-165.

Tichem, M. & Storm, T., 1996, *Issues in Product Structuring*, in Proceeding of the 2nd WDK Workshop on Product Structuring, Delft.

Toye, G., Culkosky, M., Leifer, L., Tenenhaum, J. & Glicksman, J., 1993, *SHARE: A Methodology and Environment for Collaborative Product Development. In Proceeding of Second Workshop on Enabling Technologies: Infrastructure for*

Collaborative Enterprises, IEEE Computer Society Press, Vol. 31, No. 7, pp. 33–47.

Ullman, G.D., 2003, The Mechanical Design Process, Chapman& Hall, New York.

Ulrich, K.T. & Eppinger, S.D., 1995, Product Design and Development, McGraw-Hill, Inc., New York, U.S.

Van Elst, L. & Abecker, A., 2002, *Ontologies for Information Management: Balancing Formality, Stability, and Sharing Scope*, Expert System with Application Vol. 23, pp. 357-366.

Varma, A., Dong, A., Chidambaram, B., Agogino, A. & Wood, W., 2003, *Web-based Tool by Engineering Design*, Journal of Integrated Design and Process Science, Vol. 7, No. 3, pp. 95-108.

VDI (1993). Effects and Assessment of Odours. Determination of Annoyance Parameters by Questioning. Repeated Brief Questioning of Neighbour Panellists. VDI-guidline 3883 Part 2.

Vinney, J.E., Blount, G.N. & Noroozi, S., 1999, *A Conceptual Design ASsitant CODAS*, Proceeding of the 1999 CIRP International Design Seminar, Enschede, the Netherlands, 24-26, March.

Visual Studio Help HomePage, 2006, http://msdn.microsoft.com, visited 11.02. 2006.

Vypress CompanyHome Page, 2006, http://www.vypress.com, visited 12.05. 2006.

Wand, Y., 1996, *Ontology as a Foundation for Meta-modelling and Method Engineering*, Information and Software Technology, Vol. 38, pp. 281-287.

Wang, Q., Rao, M., Zhou, J., Mital, A & Anand, S., 1994, Intelligent Systems for Conceptual Design of Mechanical Products - Handbook of Expert System Applications in Manufacturing: Structure and Rule, Chapman& Hall, New York.

Wang, L.H., Shen, W.M., Xie, H., Neelamkavil, J. & Pardasani, A., 2002, *Collaborative Conceptual Design – State of the Art and Future Trends*, Computer-Aided Design Vol. 34, pp. 981-996.

Wiegers, T., Horváth, I., Vergeest, J.S.M., Opiyo, E.Z. & Kuczogi, G., 1999, *Requirements for Highly Interactive System Interfaces to Support Conceptual Design*, Proceedings of the 1999 ASME Design Engineering Technical Conferences September 12–15, Las Vegas, Nevada.

Yi, G., 2003, Visual C++.Net Network Programming, China Railway Publishing House, BeiJing.

Yoo, S.B. & Kim, Y., 2002, *Web-based Knowledge Management for Sharing Product Data in Virtual Enterprise*, International Journal Production Economics, Vol. 75, pp. 173-183.

Yoshioka, Y., Umedab, Y., Takedac, H., Shimomurad, Y., Nomaguchie, Y. & Tomiyama, T., 2004, *Physical Concept Ontology for the Knowledge Intensive Engineering Framework*, Advanced Engineering Informatics, Vol. 18, pp. 95–113.

Zwicky, F., 1966, New Methods of Thought and Procedure: Contributions to the Symposium on Methodologies. Springer, Berlin.

APPENDIX A DIDEAS I

A.1. INTRODUCTION

The Distributed Design Assistant (DiDeas I), which was developed by Schueller [2002] in the Department of Mechanical Engineering at Stellenbosch University, is a design support system that guides distributed designers through the design methodology of specification and concept development. It is expected to relieve the users of the problems associated with the hierarchical approach and the exchange of design information, thereby making more time available for the intuitive tasks.

The user interface aspects of the implementation are also discussed in this appendix.

A.2. THE MAIN WINDOW OF DIDEAS I

The main window is divided into four areas, so-called "frames", as seen in Figure A.1.

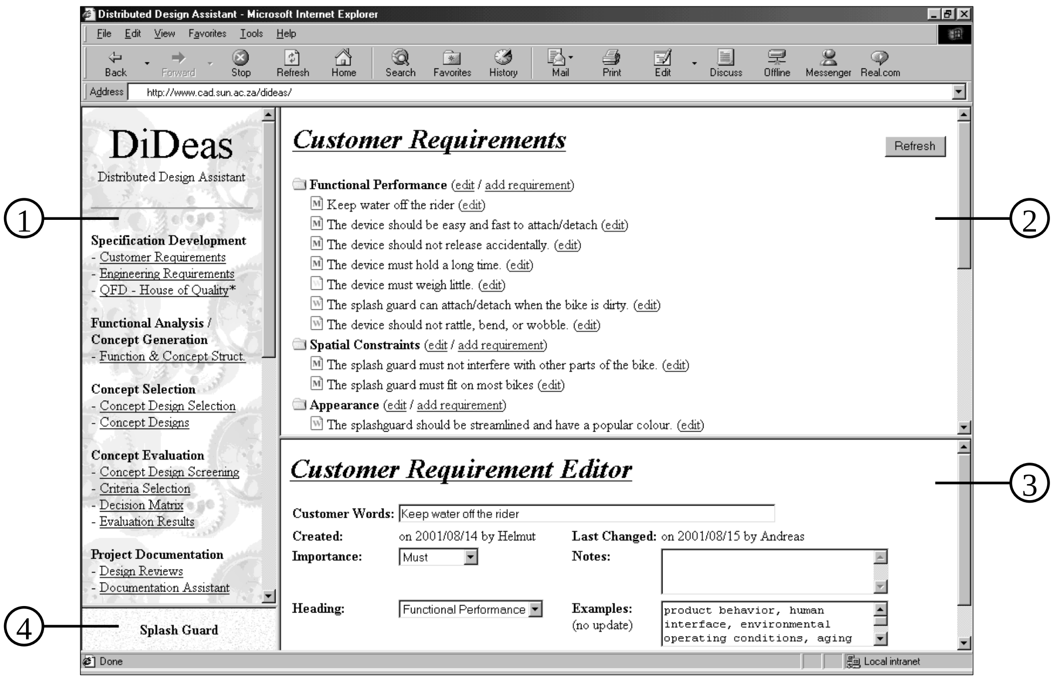

Figure A-1 DiDeas Main Window

The menu in the long frame on the left (1) provides hyperlinks to all elements of the DiDeas, from the components of the methodical design procedure to the communication tools and the help file. The content and size of the two frames on the right (2 and 3) vary, depending on the feature selected in the menu. The small frame in the lower left-hand corner (4) serves two purposes. It displays the name of the present project and it informs the user of new "Short Messages", similar to a pop-up of an e-mail or an instant messaging program. Some features, e.g. the "House of Quality" and the message board, are opened in separate windows.

All information entered into the DiDeas is stored in the DiDeas database on the DiDeas web server.

A.3. SPECIFICATION DEVELOPMENT

A.1.1. Customer Requirements Development

The design process starts with the customer requirements phase. Each team member can make contributions to these lists. During both steps, checklists (Figure A-1) can propose common requirements to guide the designers.

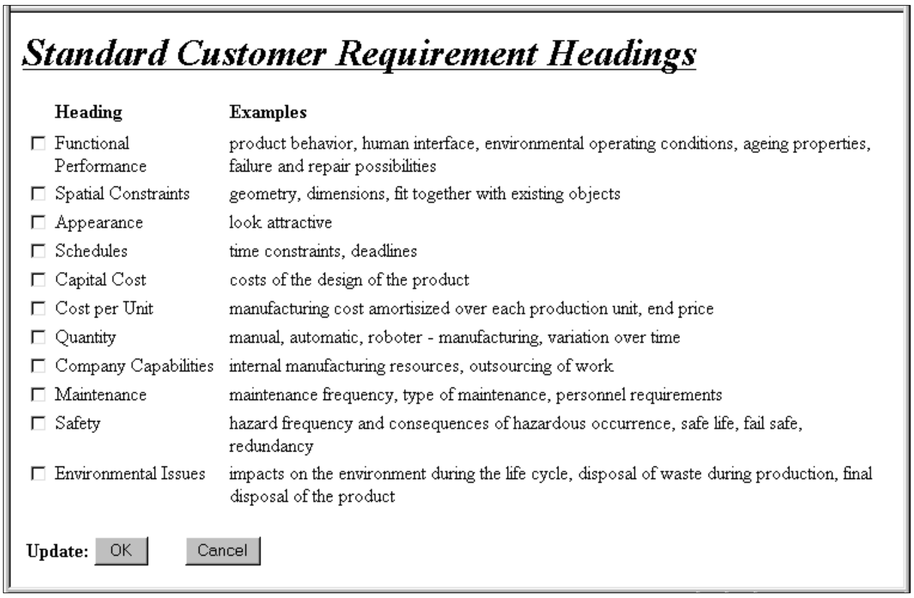

Figure A-1 Standard Customer Requirement Headings

Once a heading is available, customer requirements can be added. This is done by entering the data in the customer requirement editor, as shown in Figure A-2

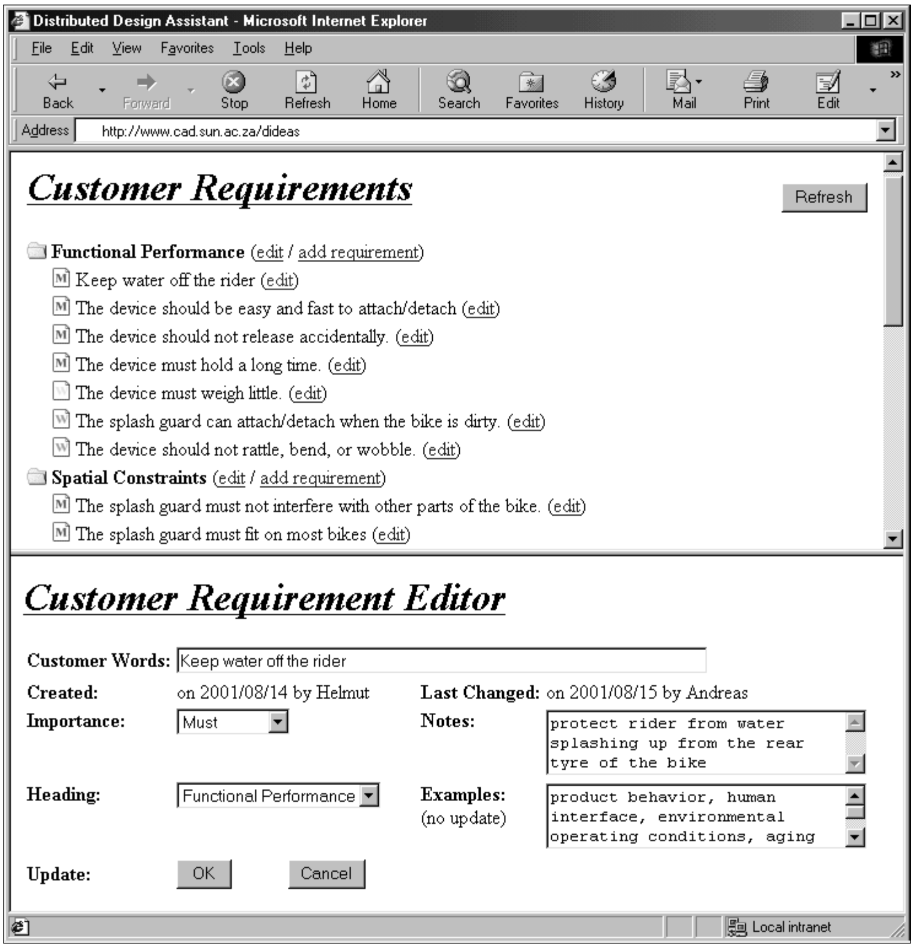

Figure A-2 Customer Requirements and the Requirement Editor

The designers can also enter unspoken requirements, i.e. requirements that the customer may not have explicitly mentioned. By clicking on the "OK" button, the entered information will be added to the database. The new customer requirement can now be displayed in the customer requirement list. The entries are sorted according to the requirement heading and their relative importance. The importance is indicated by coloured icons; a red "M" for a "Must", a yellow "W" for a "Major Wish" and a green "W" for a "Minor Wish".

A.1.2. Engineering Requirements Development

The following phase after the customer requirement development is the engineering requirement development phase, where the customer's words are translated into measurable engineering terms. The engineering requirements are entered and displayed in a similar way to the customer requirements.

The engineering requirements are entered in the engineering requirement editor, which has a similar structure to the customer requirement editor (Figure A-3).

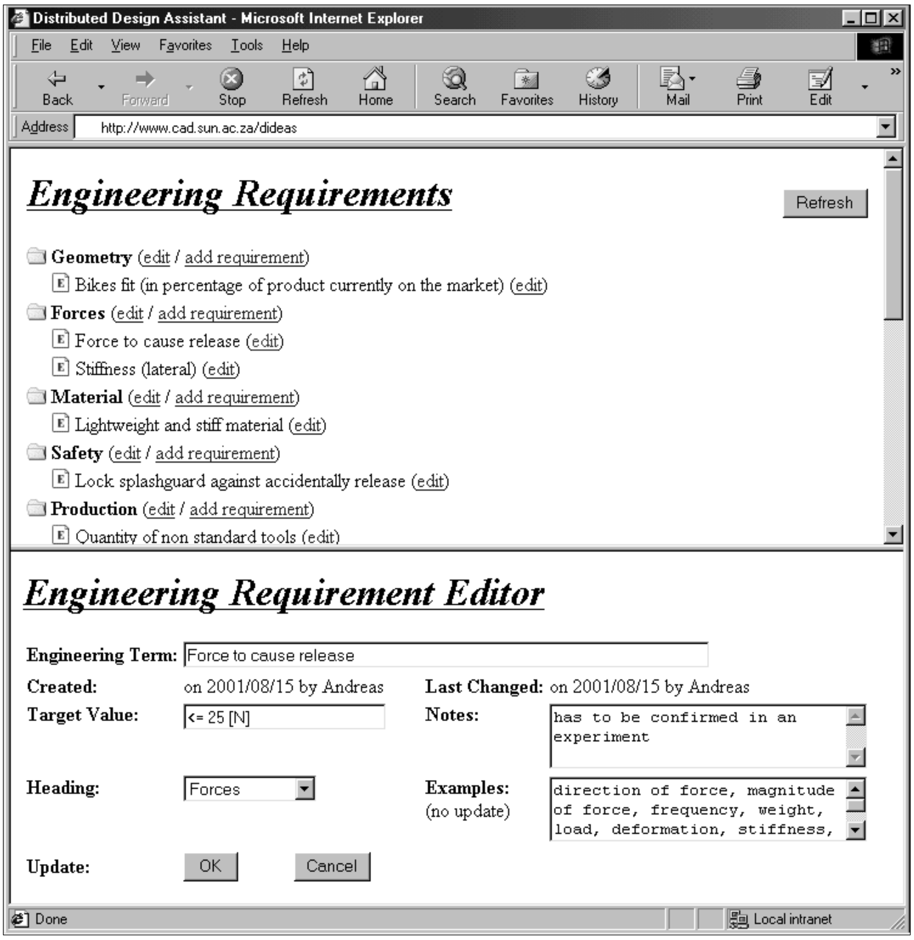

Figure A-3 Engineering Requirements and the Requirement Editor

The engineering term for the requirement is recorded in the form of a short sentence, and additional information can be entered as notes. As with the customer requirements, some examples are given for guidance. The target value should comprise a value and a unit, either as a nominal value, an upper or lower limit, or an interval.

By clicking on the "OK" button, the entered information will be added to the database and the new engineering requirement will be displayed in the engineering requirement list.

The designers can and should refresh the lists of customer requirements and engineering requirements regularly, in order to check their own entries, as well as to display the contributions of the team members. All customer and engineering requirements and customer and engineering requirement headings can be edited by anybody, at any time. This is important for dealing with inaccurate or duplicate entries. Duplicates are especially likely to appear when more than one designer are working on the specification development simultaneously. In addition to the changeable requirement information, the requirement editor also displays when and by whom the entry was created and last modified.

A.1.3. Quality Function Deployment and the "House of Quality"

A properly developed "House of Quality" can ensure the completeness of customer requirements and engineering requirements, and that the problem to be addressed is understood by all team members.

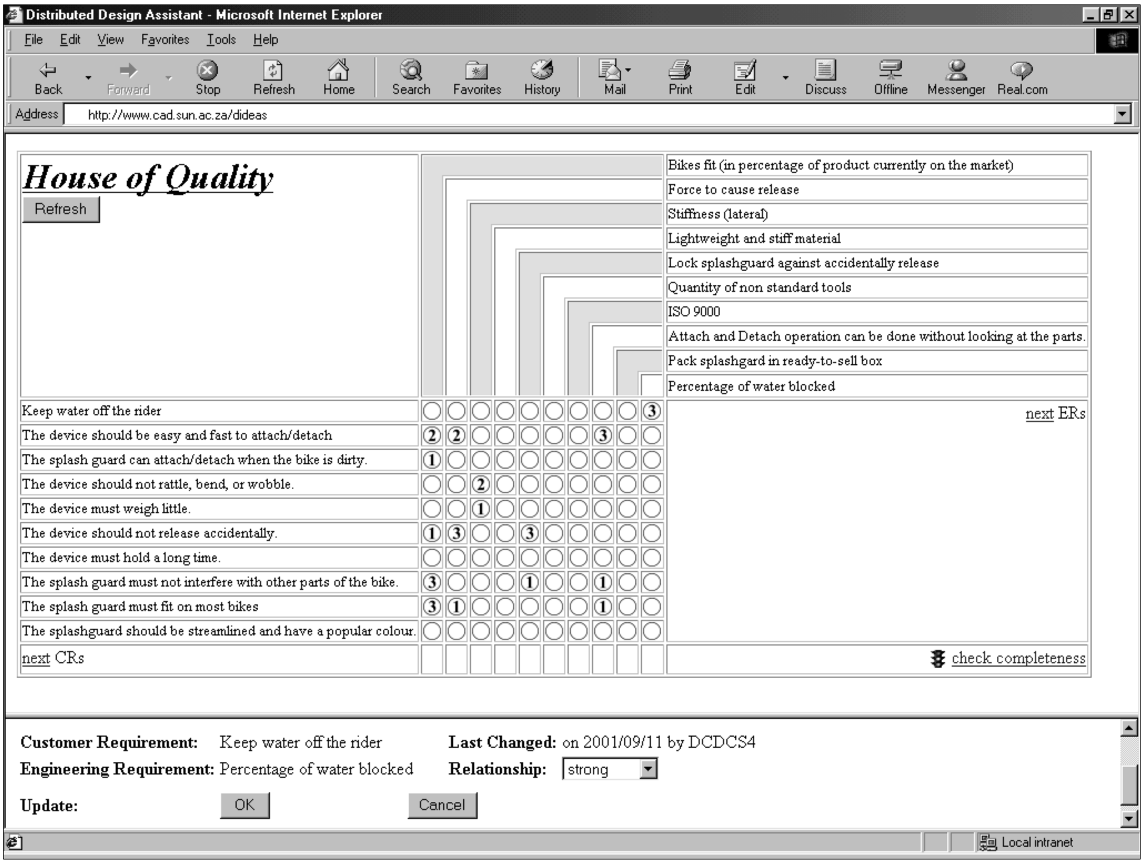

Figure A-4 "House of Quality" with Matrix and Editor

Due to its size, the matrix opens in a separate window. The customer requirements are located on the left, and the engineering requirements on the top of the matrix. Grey and white indicator lines facilitate the tracing of the correct engineering requirement column. For clarity, only groups of ten requirements are shown, arranged in the same sequence as during their development in the previous sections. To move to the next or previous group of customer or engineering requirements, the designer clicks on the "next" or "prev." hyperlinks below the group.

To add or edit a relationship value between a customer requirement and an engineering requirement, the user clicks on the corresponding cell in the matrix. The editor in the bottom frame displays the two requirements and the user can select the strength of the relationship from the pull-down menu: "weak" (1), "medium" (2) or "‘strong" (3). The fourth option is to delete an existing relationship ("no relation").

By clicking on the "OK" button, the relationship value will be added to or deleted from the database. The relationship values can be edited by anybody, at any time. The designers can and should refresh the "House of Quality" regularly, in order to check their own entries, as well as to display the contributions of the team members.

A.4. FUNCTIONAL ANALYSIS AND CONCEPT GENERATION

A.1.4. Function and Concept Structure

The function and concept structure displays all functions and concepts for the current project as a tree structure. The first step in developing the function and concept structure is the definition of the overall function. Figure A-5 shows the overall function editor, where one designer describes the essential problem in a short sentence.

Figure A-5 Overall Function Editor

The notes can contain details of the problem or a comment by the designer. It is also possible to add a description of the system boundaries and of the flow of energy, material and information.

Once the overall function has been created the main concepts can be generated and broken down into their individually determining functions. These steps of function and concept generation are repeated until a satisfactory level of decomposition is reached for each function-concept branch of the tree structure. Each function, including the overall function, has to have at least one concept, i.e. a solution to satisfy it.

Figure A-6 shows a part of the function-concept structure for the "Splash Guard" project.

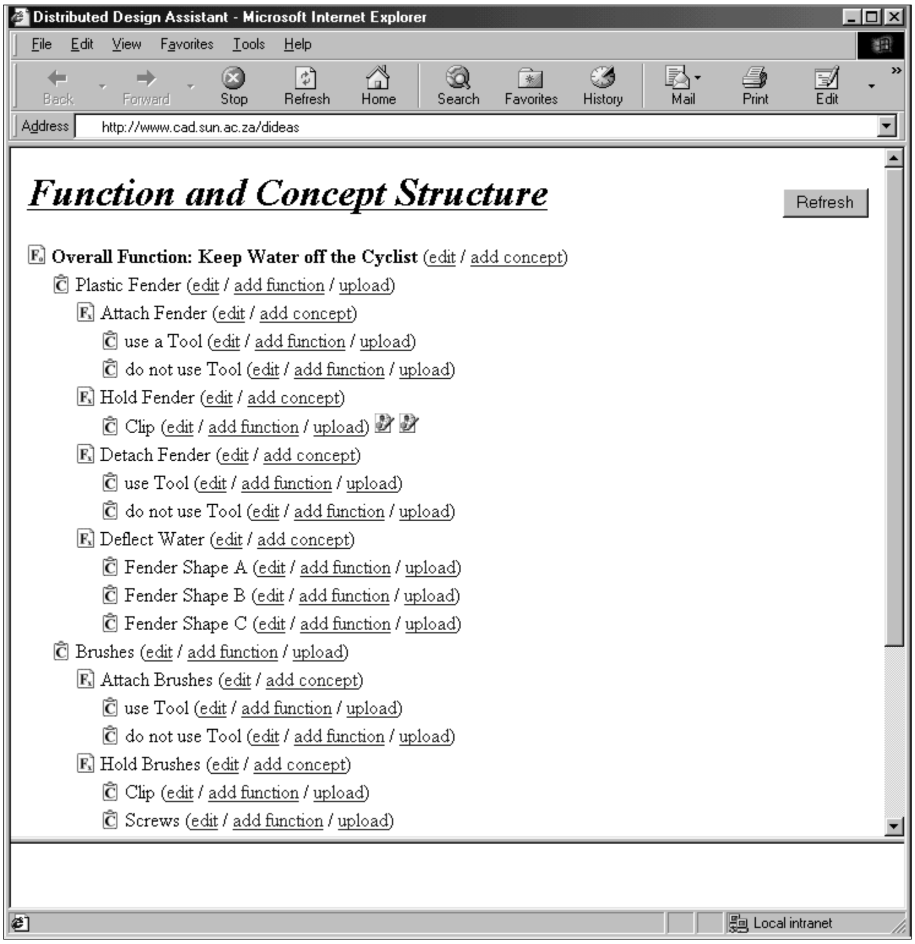

Figure A-6 Function and Concept Structure

In the structure, functions are represented by a "F_0" icon (overall function) or "F_x" icons (other functions), and concepts by "C" icons.

A.1.5. Concept Upload

The DiDeas offers the possibility to attach information to the concepts of the function-concept structure in the form of files. The file format is not limited to images; all file types are permissible, such as pictures, text, movie clips and CAD files. To upload a file to the function-concept structure, the designer has to click on the "upload" hyperlink behind the concept and specify the location of the desired file. This can be done by direct input into the text field or by browsing for the file on the user's computer or local area network, as shown in Figure A-7.

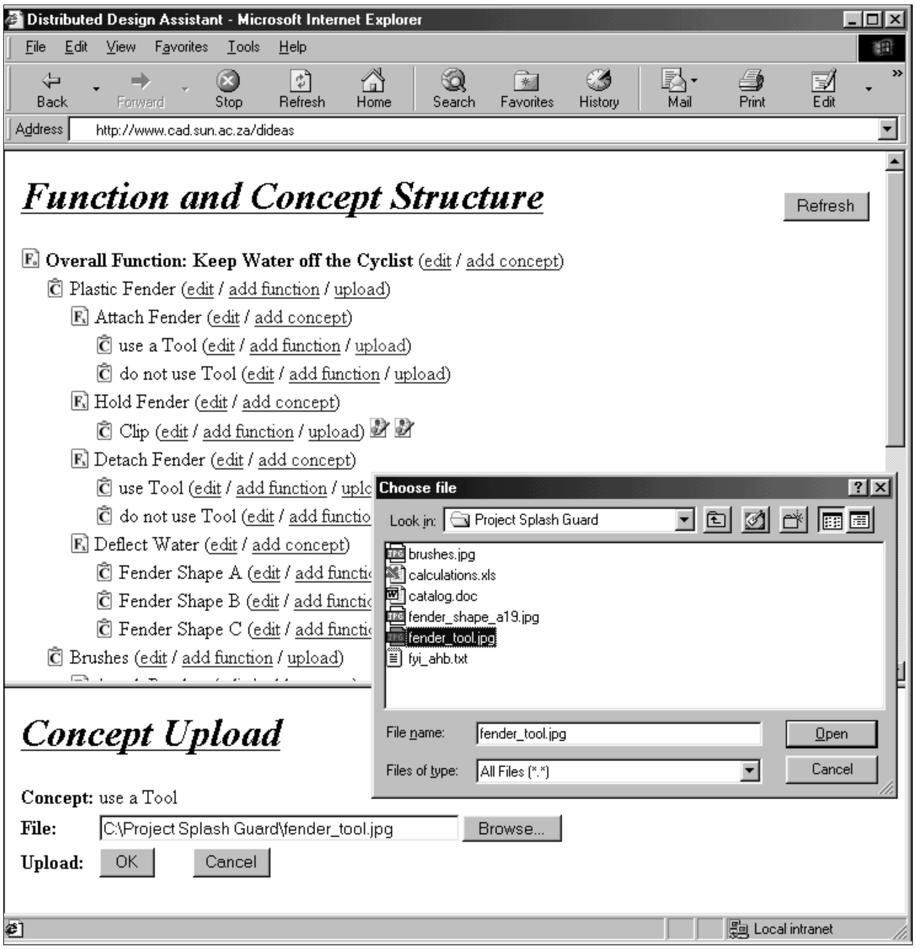

Figure A-7 Function-Concept Structure and Concept-Upload

When the file is uploaded, an image icon is displayed behind the concept to indicate an attachment. Two concept attachments have already been uploaded for the concept "Clip", as shown in the figure above. These icons are linked to the uploaded files, and by clicking on an icon the corresponding information will be displayed in a new web browser window. In case the file format is unknown to the web browser, e.g. an

AutoCAD drawing, the user can download the file to his/her computer and open it using the proper application.

A.1.6. The Concept Design Selection

For the concept design selection a tree-structure similar to the function-concept structure is displayed, as shown in Figure A-8. Since the function-concept structure has already been developed, no hyperlinks for adding, editing or uploading are necessary.

Figure A-8 Concept Design Selection in DiDeas

The designers select concept designs by combining different sub-concepts of a main concept. To do this, the user ticks the radio button in front of the desired concept. He/she starts with a main concept, i.e. a concept for the overall function, on the far left. Working one concept level at a time, from left to right, a concept for each function and sub-function of the main concept is selected. Within the concept group of one function only one concept can be chosen. If a function is only satisfied by one concept, this concept has to be picked. Figure A-8 shows one possible complete selection of a concept design for the Splash Guard project.

The concept designs created in this manner cannot be checked automatically by the system regarding their completeness or feasibility. The DiDeas can only record the concept designs, but not interpret them. However, it is possible to continue the design process with an incomplete concept design, if a designer intents to.

The concept designs page displays a table with all concept designs and some basic information, i.e. date and time the concept design was selected and the nickname of the responsible designer. By clicking on the concept design name, all functions and corresponding concepts of the particular concept design are shown, as illustrated in Figure A-9.

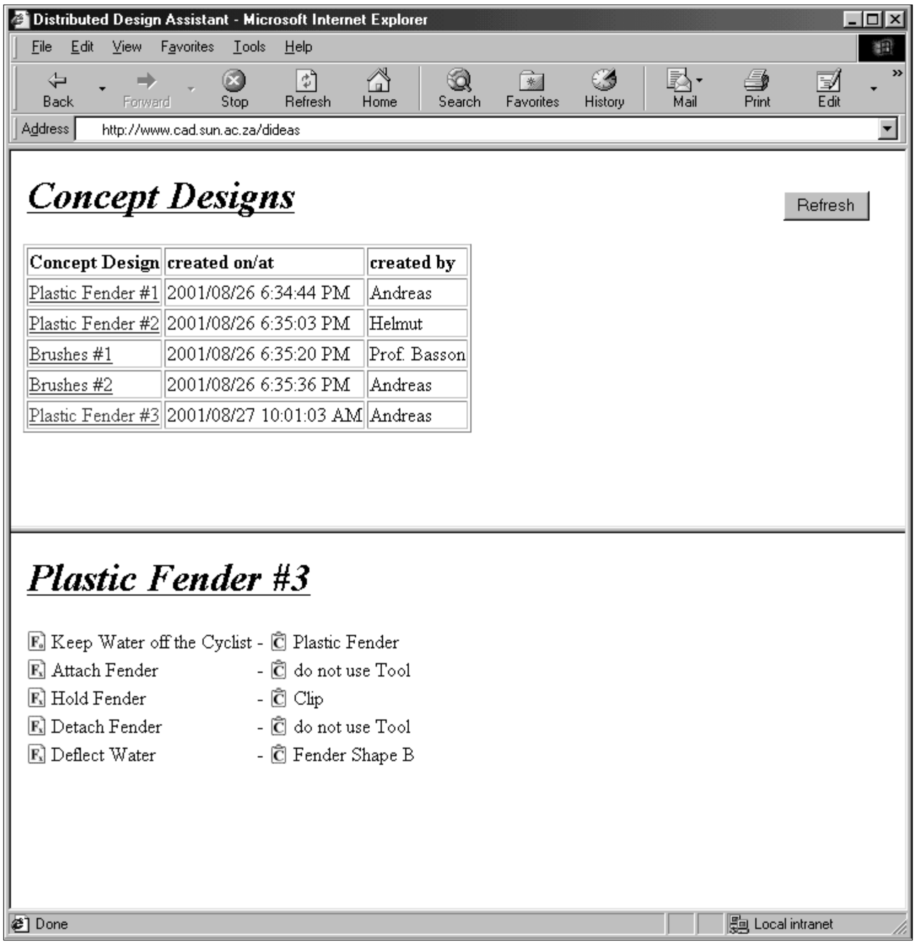

Figure A-9 Table of Concept Designs and List of Functions and Concepts

A.1.7. Concept Evaluation

Before comparing the concept designs, a rough concept design screening takes place. The intention is to eliminate those concepts from the evaluation process that do not comply with the customer requirements or engineering requirements and to reduce the number of potential winning concept designs to a more manageable number. In some cases it will not be possible to judge concept designs against all requirements, e.g. because not enough information is available during this early phase of the development process. Figure A-10 shows the two frames of the concept design screening of the DiDeas.

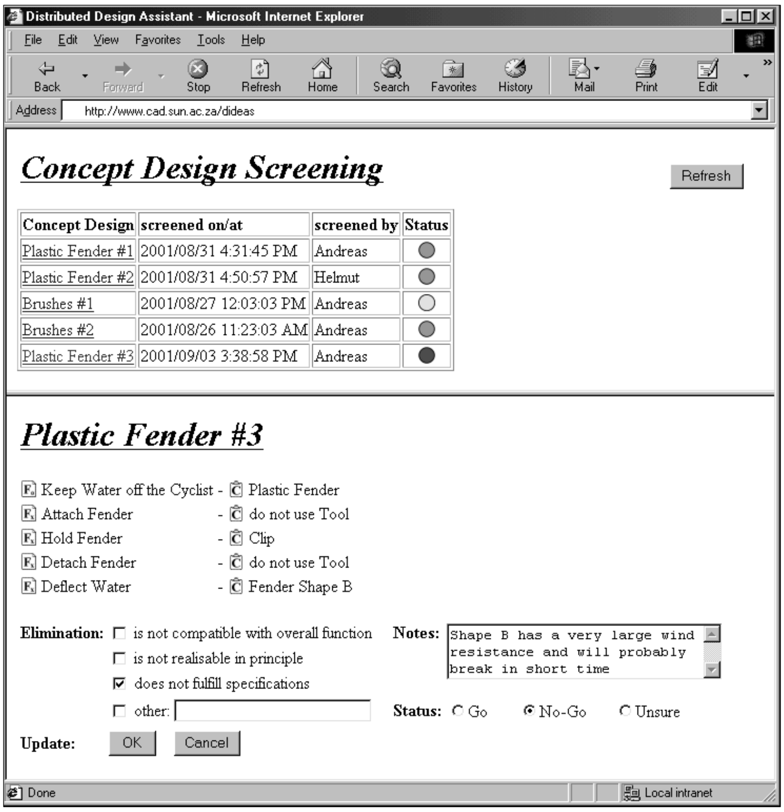

Figure A-10 Concept Design Screening in DiDeas

In the top frame a table with all concept designs is displayed. The most important field of the table is the "Status" field. A coloured dot indicates the status of the concept design as 'Go" (green), "No-Go" (red) or "Unsure" (yellow). Additionally, the date and time of the screening, as well as the nickname of the screening designer, are shown.

If only one of these items is ticked, the concept design will automatically receive the status "No-Go". Otherwise the user can decide on the status by selecting one of the three status radio-buttons. In case the designer is not convinced of either a positive or a negative judgment, he/she can choose the "Unsure" status to indicate a decision. A notes field provides space for comments or additional information.

All concept design screenings can be edited by anybody at anytime and the frame with the screening results should be refreshed regularly.

Of all concept designs only the ones marked as "'Go" will be considered for evaluation in the decision matrix.

A.1.8. Decision Matrix

Each designer performs an individual evaluation of the concept designs using the decision matrix. This process can be divided into the following four steps.

A.1.8.1. Criteria Selection

The criteria are based on the customer and engineering requirements, and will be used to compare the alternative concept designs with each other. Figure A-11 shows the two frames of the criteria selection.

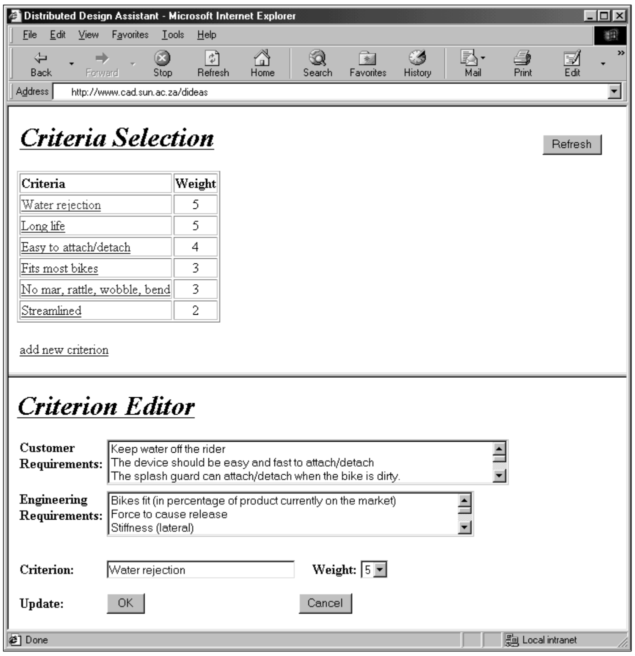

Figure A-11 Criteria Selection in DiDeas

In the top frame all criteria and their corresponding weight factors are displayed. A new project does not contain any criteria. To add new criteria, the designer clicks on the "add new criterion" hyperlink at the bottom of the criteria list. In the bottom frame the customer requirements and engineering requirements are shown, to guide the user in establishing new criteria. The criterion is entered as a short phrase and a weight factor between one and five is selected from the drop-down menu. The default value is three. By clicking on the "OK" button, the entered criterion information will be added to the DiDeas database.

There is no limit to the number of criteria for the decision matrix. The team or team-manager has to decide when the criteria selection process is completed.

A.1.8.2. Selection of a Datum Concept Design

The designer selects a personal favourite as the "datum", i.e. the concept design he/she regards as the best. All concept designs are listed in the datum selection frame, and the datum is selected by clicking on the concept design name.

A.1.8.3. Individual Evaluation

The next frame shows the decision matrix with the criteria and the weight on the left and the concept designs on top, represented by numbers. Figure A-12 illustrates a decision matrix for the "'Splash Guard" project.

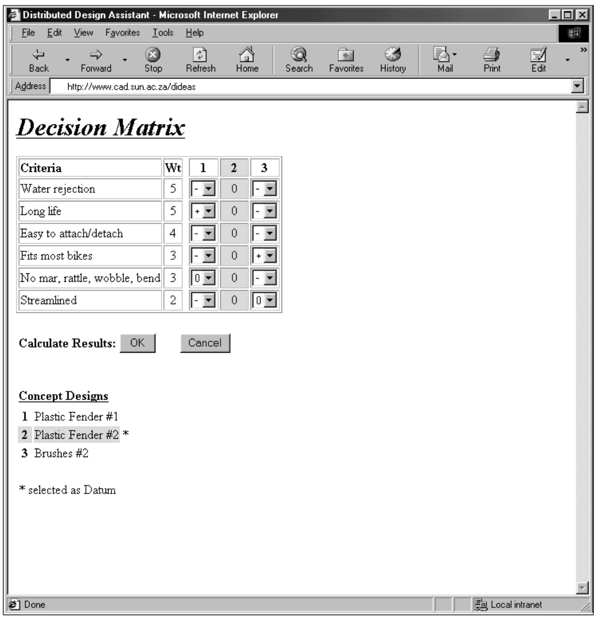

Figure A-12 Individual Decision Matrix

For each criterion, the designer judges each concept design as either: better ("+") than, equal ("0") to, or worse ("-") than the datum. This is done by selecting the corresponding value from the drop-down menu in each cell. The values in all cells are preset to "0". The cells of the datum column cannot be changed, as the datum cannot be compared with itself.

Once all concept designs are compared with the datum for every criterion, the personal decision matrix results can be calculated. Four scores are generated for each concept design:

- *Total "+"*: the number of "+" scores
- *Total "-"*: the number of "-" scores
- *Overall Total*: the difference between the number of "+" scores and the number of "-" scores
- *Weighted Total*: the sum of each score multiplied with the corresponding weight factor.

A.1.8.4. Combining Individual Evaluation Results

Finally the individual decision matrix results can be combined to produce the group results. The results for each user will be stored in the database, together with a code ("ControllID") for the concept designs and criteria incorporated in the decision matrix. In this way the adding up of user results with different numbers of concepts, different criteria and different weight factors, which would completely distort the group results, will be avoided.

A.1.8.5. Evaluation Results

The evaluation results are displayed in the two frames of the Evaluation Results page, as shown in Figure A-13.

In the top frame the combined evaluation results of all designers who have already completed the decision matrix are displayed. This identifies the winning concept design of the design team. In the bottom frame the individual results are shown. They are useful if there is disagreement between the designers or if no undisputed winner can be identified.

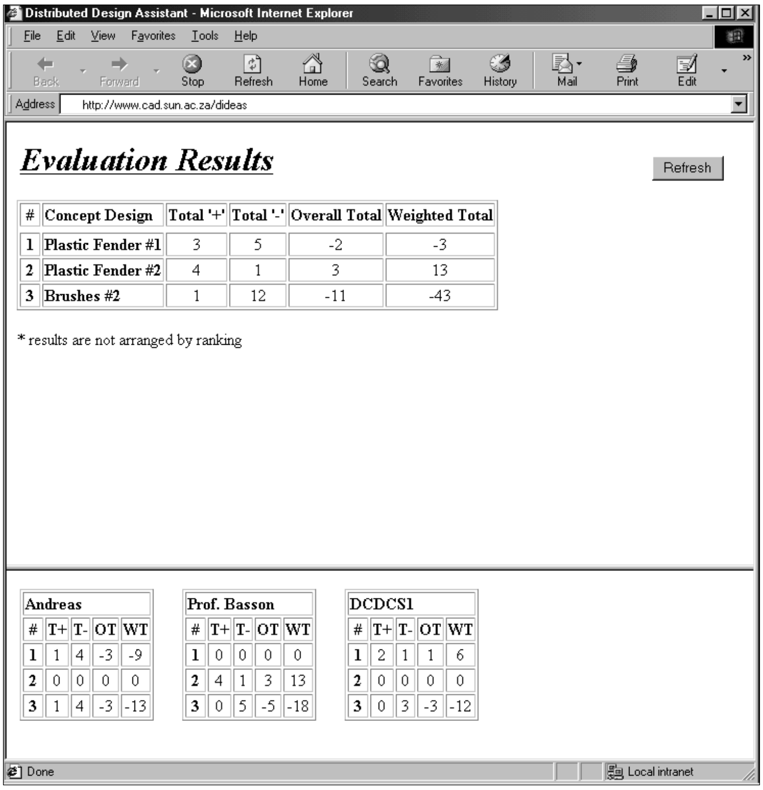

Figure A-13 Combined and Individual Evaluation Results

In this case the design team can add new criteria or edit the existing ones, e.g. by modifying the weight factors, and repeating the evaluation process until a clear winner can be determined.

A.5. PROJECT DOCUMENTATION

A.1.9. Design Reviews

Design reviews can be written by anybody, at any time. To create a new design review, the user can start the Design Review Editor by clicking in the "New Design Review" hyperlink in the top frame of the design review window, as illustrated in Figure A-14.

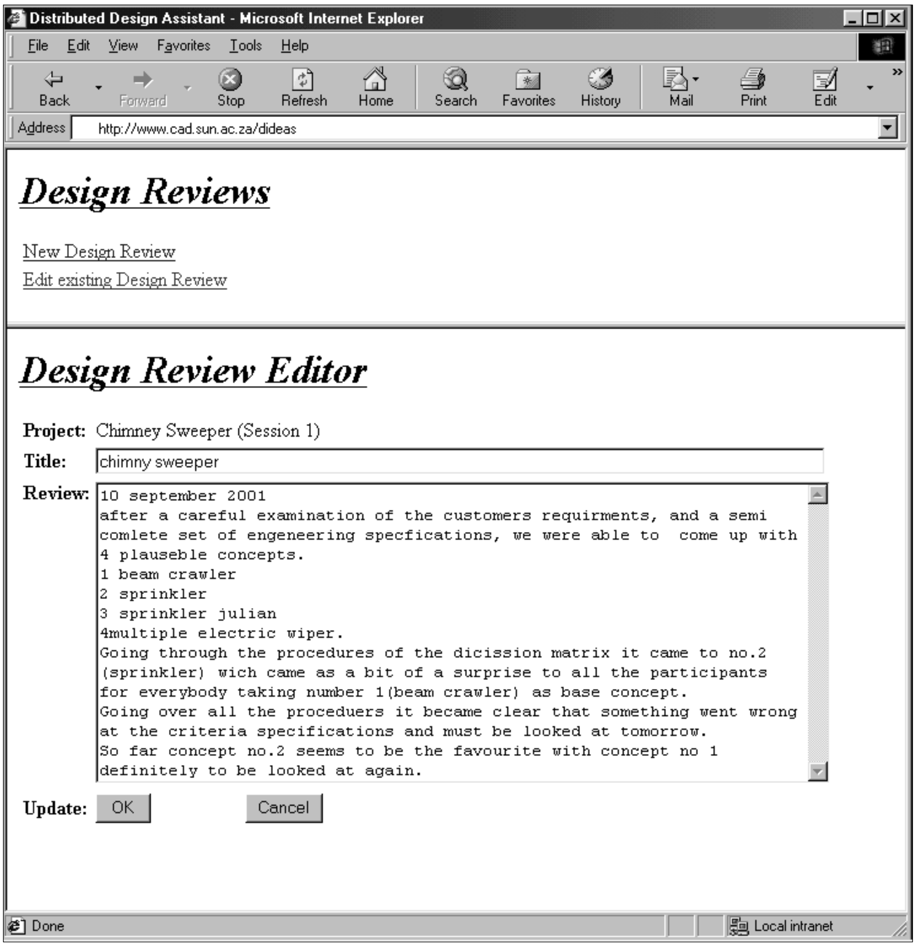

Figure A-14 The Design Review Window of DiDeas

The new design review is linked to the current project. The user has to specify a title for the design review and can then add the review. By clicking on the "OK" button, the entered information will be added to the DiDeas database.

To edit or view an existing design review, the designer clicks on the hyperlink "Edit existing Design Review". A list of all design reviews of the present project will be displayed, including information on when the review was changed last and by whom, as well as whether the design review is currently in use and by whom. A design review "in use" cannot be accessed by another designer until the present user has finished working on it.

A.1.10. Documentation Assistant

The documentation assistant allows for printing of all elements of the design procedure discussed in the previous sections, from the list of customer requirements to the design reviews. Figure A-15shows the documentation assistant with the printable documents.

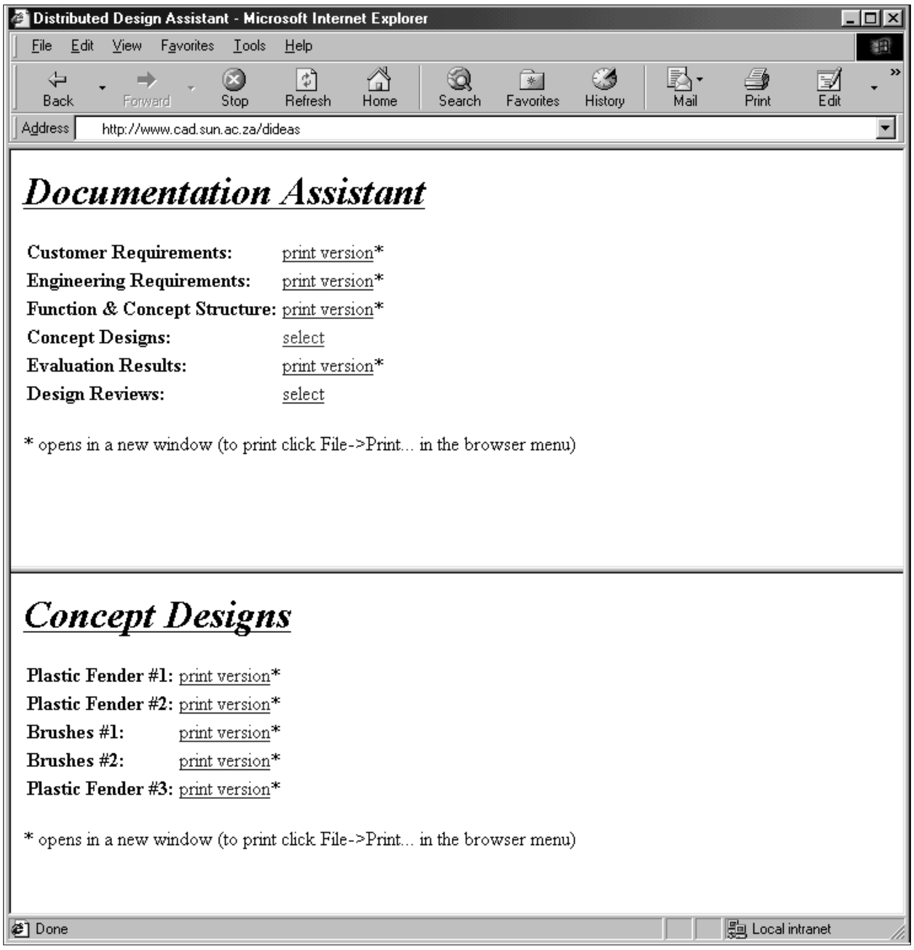

Figure A-15 The Documentation Assistant of DiDeas

In the cases of the concept designs and design reviews, a list of available print versions is displayed for selection in the bottom frame.

The print version of a document consists of the same elements and has the same layout as the original page in the DiDeas, but does not display any hyperlinks. Also the coloured icons used for the customer requirements are replaced by greyscale icons. The print version is displayed in a new browser window and can be printed directly from there via the print function of the web browser.

APPENDIX B DIDEAS II DEVELOPMENT REQUIREMENT QUESTIONNAIRES

Student Design Project Case Study Questionnaire

Name ____________________ Position in the group____________________

E-mail address _________________ Phone _______________________________

Instructions

- Please answer all questions of the questionnaire.
- Answer the questions by ticking the corresponding box and/or with a short text.
- If a question is not applicable, please tick the corresponding box (n/a).
- If you need more space for an answer, you can use the back of the paper (please write down the question number).
- Don't hesitate to contact me if you have a question (phone: 808-4242and Email: yliu@sun.ac.za, Liu).
- All information will be kept confidential. The name on top is used only for follow-up questions.

1) Did you participate in any design project in group?

   ☐ Yes, is that the same design period? For how long?

   ☐ No

2) Which of the following design methodologies do you know about or have you used before?

Ullman	know ☐	used before ☐
Pahl & Beitz	know ☐	used before ☐
Hubka	know ☐	used before ☐
VDI-2221	know ☐	used before ☐
other ____________________	know ☐	used before ☐

3) Where did you get the specifications in this design project?

   ☐ Form class assignment, ____%

   ☐ From somebody you visit ____%

4) How did you manage the specifications in this design project?

   ☐ From your own experience? _____%

   ☐ From the meeting discussion? _____%

5) Did you personally experience difficulties in distinguishing between customer and engineering requirements?
   - ☐ Yes
   - ☐ No

6) Did you personally like to use standard headings for customer and engineering requirements?
   - ☐ Yes
   - ☐ No

   Do you like the other method like brain storm? or you like other method?

7) Did you personally develop a 'House of Quality' in this projects?
   - ☐ Yes

     How useful was the 'House of Quality' for this project?

     Did you personally experience difficulties? What that would be?

   - ☐ No

8) Did you personally create or edit a design review in this project?
   - ☐ Yes
   - ☐ No why not?

9) Which way you would to be in a meeting?
   - ☐ Regular meeting
   - ☐ Instant meeting

10) Which of the following sketch input devices did you personally use for this project?
    - ☐ pen & paper
    - ☐ computer mouse
    - ☐ scanner
    - ☐ graphic tablet & pen

    Other ________________

11) Do you prefer the sketch input via pen & paper over electronic devices?

- ☐ Yes why?

- ☐ No why?

12) Which program did you personally use to create graphics/drawings?

13) Which program did you personally use to view graphics?

14) Any additional comments on this questionnaire (criticism, suggestions, etc.)?

APPENDIX C DIDEAS II EVALUATION QUESTIONNAIRES

Student Design Project Case Study Questionnaire

Name ____________________ *Position in the group*____________________

E-mail address ____________________ Phone ______________________________

Instructions

- Please answer all questions of the questionnaire.
- Answer the questions by ticking the corresponding box and/or with a short text.
- If a question is not applicable, please tick the corresponding box (n/a).
- If you need more space for an answer, you can use the back of the paper (please write down the question number).
- Don't hesitate to contact me if you have a question (phone: 808-4242and Email: yliu@sun.ac.za, Liu).
- All information will be kept confidential. The name on top is used only for follow-up questions.

1) Did you participate in any design project in group?

☐ Yes, is that the same design period? For how long?

☐ No

2) Which of the following design methodologies do you know about or have you used before?

Ullman	know ☐	used before ☐
Pahl & Beitz	know ☐	used before ☐
Hubka	know ☐	used before ☐
VDI-2221	know ☐	used before ☐
other ____________________	know ☐	used before ☐

3) Do you like the other method like brain storm? or you like other method?

4) Did you personally develop a 'House of Quality' in this projects?

☐ Yes

How useful was the 'House of Quality' for this project?

Did you personally experience difficulties? What that would be?

- ☐ No

Project Documentation

5) Did you personally create or edit a design review in this project?
   - ☐ Yes
   - ☐ No why not?
6) Do you like the way of managing the design review in this system?
   - ☐ Yes
   - ☐ No why not?

Communication and Information Transfer

7) Which way you would to be in a meeting?
   - ☐ Regular meeting
   - ☐ Instant meeting

Input Devices

8) Which of the following sketch input devices did you personally like to use?
   - ☐ pen & paper
   - ☐ computer mouse
   - ☐ scanner
   - ☐ graphic tablet & pen

   Other _______________

9) Do you prefer the sketch input via pen & paper over electronic devices?
   - ☐ Yes why?
   - ☐ No why?

10) Which program did you personally use to view graphics?

System Interface

11) Do you think this system use interface is friendly?
   - ☐ Yes
   - ☐ No what's your suggestion?

12) How long did you take to learn to use the system?
- ☐ Short time (<15mins)
- ☐ Medium time (15-30 mins)
- ☐ Long time (>30 mins)

13) Do you like the tree view in this system to handle the system and sub-system?
- ☐ Yes why?

- ☐ No why?

14) Is the tab clear for you to understand the design process?
- ☐ Yes

- ☐ No why?

15) Do you think the table format can handle the design information when you doing editing?
- ☐ Yes

- ☐ No why? And what format you like(like normal notepad etc.)?

16) Do you think the information updating time set interface is clear for you? What update interval you would like?

17) Do you like the parent table stay beside the child table (you are working on) or stay as a separate table?

18) Do you like the QFD interface? What you think need to be improved?

19) Do you think the download table in work space can provide enough information to describe the file?

Miscellaneous

20) After using the software, do you have a more clear understand of the design process?

21) Any additional comments on this questionnaire (criticism, suggestions, etc.)?