

# Development of Tailorable Mechanical Design Support Software

by  
Ruan van der Merwe

*Thesis presented in fulfilment of the requirements for the degree of  
Master of Science in the Faculty of Engineering at Stellenbosch  
University*



Supervisor: Prof. A. H. Basson

December 2013

## DECLARATION

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: 26<sup>th</sup> of August 2013

## **Abstract**

A wide variety of design methodologies exist in literature and the methodologies employed may differ among companies and even among design teams. Therefore a software tool, called DiDeas II, is being developed for the early phases of mechanical engineering design. DiDeas II is customisable to accommodate various design methodologies.

An approach for customisability which allows the user interface and data structure to be customised without changing the source code has been implemented in previous developments via an approach combining ontology and conceptual graphs. This approach is expanded in this thesis to allow for the implementation of various design methodologies through the use of tables for the display of information with inheritance of data among these tables.

During groupwork, communication is both asynchronous and synchronous. DiDeas II has been developed in this thesis to facilitate and capture both asynchronous and synchronous communication between team members. Capturing such communications has the potential to provide insight into design decisions.

The communication functionality was assessed in case studies in an academic environment. DiDeas II proved to be effective at recording “soft” information during design and placing the information into context for future reference.

The degree to which DiDeas II could be customised to suit the design process at different companies was assessed through discussions with engineers in industry. These discussions showed that it was possible to customise DiDeas II according to the design processes followed by the participants.

## Uittreksel

'n Wye verskeidenheid ontwerpmetodologieë bestaan in die literatuur en die metodologieë wat gebruik word kan tussen maatskappye en selfs tussen ontwerpspanne verskil. Daarom word 'n sagteware-hulpmiddel, genaamd DiDeas II, ontwikkel vir die vroeë fases van meganiese ingenieursontwerp. DiDeas II is pasbaar om voorsiening te maak vir verskeie ontwerpmetodologieë.

'n Benadering vir pasbaarheid wat toelaat dat die gebruikerskoppelvlak en datastruktuur aangepas kan word sonder om veranderinge aan die bron-kode te maak, is geïmplementeer in vorige ontwikkelings deur 'n benadering wat ontologie en konseptuele grafieke kombineer. Hierdie benadering is in hierdie tesis uitgebrei om voorsiening te maak vir die implementering van verskeie ontwerpmetodologieë d.m.v. tabelle vir die vertoon van informasie, met data wat "oorgeërf" word tussen hierdie tabelle.

Kommunikasie is beide asinkroon en sinkroon tydens groepwerk. DiDeas II is in hierdie tesis verder ontwikkel om beide asinkrone en sinkrone kommunikasie metodes te bemiddel en daarvan rekord te hou. Die rekordhouding van sulke kommunikasie het die potensiaal om insig te bied aangaande ontwerpbesluite.

Die kommunikasie funksionaliteit is geassesseer in gevallestudies in 'n akademiese omgewing. DiDeas II was effektief in die rekordhouding van "sagte" informasie tydens ontwerp, sowel as om sulke informasie binne konteks te plaas vir latere verwysing.

Die mate waartoe DiDeas II aangepas kan word om voorsiening te maak vir die ontwerpprosesse van verskillende maatskappye, is geassesseer deur gesprekke met ingenieurs in industrie. Hierdie gesprekke het getoon dat dit moontlik is om DiDeas II aan te pas volgens die ontwerpprosesse wat die deelnemers gebruik.

## Acknowledgements

I would like to express my sincere gratitude to the following people:

- Prof Basson, my supervisor, for his crucial guidance and motivation.
- Prof Scheffer, for his assistance and advice.
- My parents, for their support and encouragement.
- All the participants that took part in the case studies, for their time and consideration.

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>1</b>
1.1	Background.....	1
1.2	Research Objective.....	3
1.3	Thesis Outline .....	3
<b>2</b>	<b>User Needs .....</b>	<b>5</b>
2.1	Literature Survey.....	5
2.1.1	Design Research Methodology .....	5
2.1.2	Groupware and CSCW.....	8
2.1.3	Concept Design Methodologies.....	10
2.1.4	Customization .....	15
2.1.5	Networking Architecture .....	18
2.1.6	File Access Control .....	20
2.2	Survey of Existing Software.....	22
2.2.1	Microsoft SharePoint Workspace (previously Microsoft Groove) .....	22
2.2.2	Microsoft OneNote .....	23
2.2.3	IBM Leaf Notes/Domino .....	23
2.2.4	Commercial CAD Packages Integrated Collaboration Tools.....	23
2.2.5	DropBox.....	23
2.2.6	Skype (VoIP).....	24
2.2.7	Online Collaboration Tools .....	24
2.3	Targeted Survey of Engineers.....	24
2.3.1	Work Related .....	24
2.3.2	Communication .....	25
2.3.3	Design Habits.....	26
2.3.4	Methodology .....	27
2.3.5	Needs Identified .....	27
2.4	Interviews.....	27
2.4.1	Systems Engineering Company .....	28
2.4.2	Product Development Company.....	30
2.4.3	Consulting Engineer.....	31
2.5	Development of Specifications .....	32

2.5.1	Important Features Not Included for Case Studies .....	33
2.5.2	Requirements for Case Studies.....	33
<b>3</b>	<b>DiDeas II Development.....</b>	<b>35</b>
3.1	Previous Developments .....	35
3.1.1	DiDeas I .....	35
3.1.2	DiDeas II .....	35
3.2	New Developments .....	36
3.3	Software Implementation.....	38
3.3.1	System Architecture .....	39
3.3.2	Communication .....	40
3.3.3	Data Structures .....	42
3.3.4	User Interface .....	51
<b>4</b>	<b>Case Studies.....</b>	<b>66</b>
4.1	Academic Environment Case Studies .....	66
4.1.1	Setup of Academic Case Studies .....	66
4.1.2	Setup for Synchronous Case Studies.....	68
4.1.3	Synchronous Case Studies Observations .....	69
4.1.4	Synchronous Case Studies Feedback.....	72
4.1.5	Synchronous Case Studies Conclusions.....	75
4.1.6	Asynchronous Case Study Set-up.....	75
4.1.7	Asynchronous Case Study Observations .....	76
4.1.8	Asynchronous Case Study Feedback.....	76
4.1.9	Asynchronous Case Study Conclusions.....	78
4.2	Industry Expert Case Study.....	78
4.2.1	Participant Company Profiles .....	79
4.2.2	Observations, Participant Feedback and Interpretation .....	80
<b>5</b>	<b>Conclusions.....</b>	<b>85</b>
<b>6</b>	<b>References.....</b>	<b>87</b>
<b>Appendix A: Web-Survey.....</b>		<b>90</b>
<b>Appendix B: Synchronous Case Study Design Brief.....</b>		<b>98</b>
<b>Appendix C: Asynchronous Case Study Design Brief .....</b>		<b>100</b>
<b>Appendix D: Student Questionnaire .....</b>		<b>102</b>
<b>Appendix E: Industry Questionnaire.....</b>		<b>120</b>

**Appendix F: Original DiDeas II Design Interface..... 128**

## List of figures

Figure 1: Time/Space Matrix Showing Collaboration Modes.....	1
Figure 2: QFD House of Quality .....	11
Figure 3: Traceability of Requirements through a "Family of Houses" .....	13
Figure 4: DiDeas II Architecture.....	39
Figure 5: Client-Server Communication Sequence.....	40
Figure 6: DiDeas II Data Distribution Diagram .....	41
Figure 7: Data Structure .....	44
Figure 8: Row Header Inheritance.....	46
Figure 9: Column Header Inheritance.....	46
Figure 10: Column and Row Header Inheritance.....	46
Figure 11: Column Header to Row Header Inheritance .....	47
Figure 12: Row Header to Column Header Inheritance .....	47
Figure 13: Local Column Header Inheritance .....	47
Figure 14: Local Row Header Inheritance .....	47
Figure 15: Local Column to Row Header Inheritance .....	48
Figure 16: Local Row Header to Column Header Inheritance.....	48
Figure 17: Alternatives Comparison .....	48
Figure 18: Original Data .....	48
Figure 19: Local Compare Column Headers to Self.....	49
Figure 20: Local Compare Row Headers to Self.....	49
Figure 21: Server Main Window .....	52
Figure 22: User Names Editor .....	52
Figure 23: Client-side Main Window .....	53
Figure 24: Edit Project Templates .....	53
Figure 25: Project Template Editor.....	54
Figure 26: Project Template Editor Tab: Edit Node & Relation Types.....	55
Figure 27: Project Template Editor Tab: Edit Table Types.....	56
Figure 28: Tab-Pages Layout Editor.....	57
Figure 29: Table Type and Inheritance Source Selection Window.....	57
Figure 30: Node Type Right Panel Style Assignment .....	58

Figure 31: Tree View Menu Editor .....	58
Figure 32: Tree View Node Display Editor .....	59
Figure 33: Tree View Relationship Display Editor .....	59
Figure 34: Project Creation Window .....	60
Figure 35: Project Window .....	61
Figure 36: Typical Tab-Page Layout.....	62
Figure 37: Data Table Example One .....	63
Figure 38: Data Table Example Two .....	63
Figure 39: Related Information Tab-Page.....	64
Figure 40: Related Files Tab-Page.....	65
Figure 41: Messenger Window .....	65
Figure 42: Input Devices.....	66
Figure 43: Autodesk Sketchbook Pro Interface Elements .....	67
Figure 44: Concept Sketches Produced During First Experiment .....	71
Figure 45: Concept Sketches Produced During Second Experiment. ....	71
Figure 46: Selected Sketches from Asynchronous Case Study .....	76

# 1 Introduction

## 1.1 Background

Johansen (1988) defined four collaboration modes among groups in terms of location and time (see Figure 1). This thesis involves the further development of a software tool called Distributed Design Assistant II, or DiDeas II for short, which facilitates communication and information sharing, in order to allow engineering design team members to collaborate. These team members can be geographically dispersed, either across different locations, countries or even continents and time-zones, and are either working simultaneously or at different times. DiDeas II therefore aims to facilitate all modes of collaboration described in Figure 1 during group work.

The first development of the system was called DiDeas I and was a web-based implementation developed by Schueller (2002). The HTML interface of this system was, however, too restrictive and, following from the concepts presented by Basson et al. (2004), the system was developed further by Liu (2007) into DiDeas II which consists of two separate programs: a server program and a program running on the user- or client-side. Both the server-side and client programs were implemented as desktop applications. These previous developments of DiDeas are briefly presented in Section 3.1.

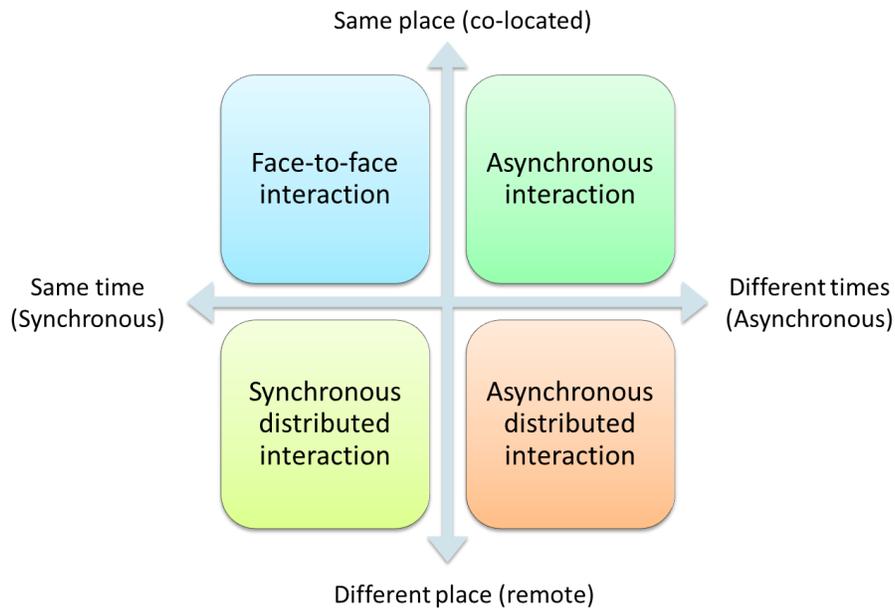


Figure 1: Time/Space Matrix Showing Collaboration Modes (Johansen, 1988).

In this thesis, where a software tool was developed that aids engineering teams during the design process, design methodology is a key concept. According to Pahl and Beitz (2007):

*“A design methodology is a concrete course of action for the design of technical systems that derives its knowledge from design science and cognitive psychology and from practical experience in different domains”.*

A design methodology includes a working plan that specifies a set of activities or working steps that are structured into design phases. The process is to be adapted appropriately for each project and does not replace intuition or experience. The plan includes strategies, rules and principles aimed at achieving general or specific goals, as well as methods to solve design problems and specific tasks (Pahl & Beitz, 2007).

A design methodology is a prerequisite for flexible and continuous computer support (see Section 2.1.2 for a description of the terms Computer-Supported Collaborative Work (CSCW) and groupware) of the design process using product models stored in the computer (Pahl & Beitz, 2007).

A wide variety of design methodologies exist in literature and various companies follow diverse design methodologies. Certain aspects of design also vary from team to team and even between different projects. One aspect is design terminology where one term can be used for various notions or different terms can be used for similar notions (Basson et al., 2004). Kamrami & Nasr (2010) also state that software vendors may provide "custom" software packages for individual firms, yet different industries have different product development strategies, which demand a generic framework that assist in efficient collaboration irrespective of the product, organizational structure and/or geographical location.

Small to medium sized companies can, however, not afford to have design support software customised to their specific needs. DiDeas II is therefore aimed at such engineering enterprises and has the goal of allowing companies to easily change DiDeas II to reflect their design styles as well as the terminology they employ.

During the first development of DiDeas II, an approach was therefore sought to enable DiDeas II to be customisable without changes to the source code of the software. This was accomplished by combining an ontology approach with conceptual graphs to create a database structure for DiDeas II where all information is classified as either an element or a relation (i.e. information is structured in conceptual graphs) with the set of available element types and relation types making up the ontology. Changes to the design process or terminology can be made by changing the ontology, without changing the structure of the database. This approach is further explained in Section 2.1.4.1.

DiDeas II is focussed on the early phases of mechanical design which is difficult to realise since the knowledge of design requirements and constraints are imprecise

during these early phases. The conceptual design phase is, however, very important as about 75 per cent of the manufacturing cost of a product has been committed by the end of this phase (Wang et al. 2002).

## **1.2 Research Objective**

The objective of this thesis is to investigate the usefulness of various communication methods within a collaborative design environment, as well as the use of ontology and conceptual graphs for customisation of design support software.

An important aspect of collaborative design is communication and many stand-alone computer mediated communication tools, which can be used during design, exist (see Section 2.1.2 for examples of such communication mediums). DiDeas II aims to integrate communication tools into the collaborative environment to allow for the documentation of communication along with other design information in DiDeas II for future reference and to capture the context for design decisions. Since the team members may work simultaneously (synchronously) or at different times (asynchronously), both synchronous and asynchronous communication methods are to be investigated. The level of acceptance of such integrated tools, as opposed to more familiar commercial tools, is to be evaluated.

In addition to the documentation of communication, DiDeas II is to provide designers with the ability to record their work, including information gathered during a project, and DiDeas II should structure and display this information in a logical and transparent manner that communicates the context of the information.

Each project is to be customisable to such a degree that it can be tailored to the specific design style of a variety of engineering companies and should allow for the specific terminology and design methodology used by each company.

Specific methodologies such as the QFD house of quality were implemented through a purpose-made interface (a QFD window) in the previous development of DiDeas II. The aim of this research is to further expand the customisability of DiDeas II by providing the means for users to create a project according to their needs through the use of a generic set of user interface elements and without prescribing any methodology to the user.

This tool is to be developed to a level where it can be used by students during case studies and demonstrated to engineers in industry for evaluation.

## **1.3 Thesis Outline**

In Chapter 2, the user requirements for a design support tool are identified and the specifications for the desired design support tool are developed. The specifications are developed through various means including a literature study (Section 2.1), by surveying existing software applications for collaboration (Section 2.2), a web-based

survey (Section 2.3) and interviews (Section 2.4) with engineers in industry as participants with the aim of investigating their design habits.

The implementation of the design support tool, DiDeas II, is discussed in Chapter 3. The previous developments of DiDeas are discussed briefly (Section 3.1) before the development of DiDeas during this thesis is discussed (Sections 3.2 and 3.3). The system is conceptually divided into networking and communication (Section 3.3.2), the data structure (Section 3.3.3), and the user interface (Section 3.3.4).

Chapter 4 describes the case studies performed to assess the usability and usefulness of DiDeas II. The academic case studies are presented in Section 4.1 and the case studies involving participants working in industry are presented in Section 4.2.

Finally the conclusions are drawn regarding the success of the research presented here and are presented along with ideas for the future development of DiDeas II (Chapter 5).

## 2 User Needs

A central attribute that determines a product's acceptability is usefulness, which measures whether the actual use of a product can achieve the goals the designers intend it to achieve (Microsoft Corporation, 2000). The concept of usefulness can be broken down into utility and usability. Although these terms are related, they are not interchangeable. The term "usability" in the context of creating software represents an approach that puts the user, rather than the system, at the centre of the process. According to Nielsen (2012), usability is a quality attribute that assesses how easy user interfaces are to use and the word "usability" also refers to methods for improving ease-of-use during the design process. This is distinct from the related concepts of utility and likeability. Utility refers to the ability of the product to perform one or more tasks. The more tasks the product is designed to perform, the more utility it has. Likeability is a factor that determines users' willingness to use a product, which might also be for reasons unrelated to usability and utility (Microsoft Corporation, 2000).

With the aim of producing a system that is useful in its purpose and usable by the users within the target market, the identification of user needs for the software tool was regarded as a high priority. These needs were obtained through a combination of methods including a literature survey (Section 2.1), an evaluation of commercial collaboration software packages currently on offer (Section 2.2), a web survey (Section 2.3) and interviews with engineers in industry (Section 2.4).

### 2.1 Literature Survey

In this section the concepts, terminologies and relevant technologies that are important for the research are introduced. A framework for conducting design research is discussed after which the concepts "computer supported collaborative work" (CSCW) and "groupware" are presented. An overview of the various methodologies for the early stages of design that are presented by three well-known authors is then given. Some approaches that allow for the customisation of software by the end-users are then presented and finally networking architectures and data access concepts are discussed.

#### 2.1.1 Design Research Methodology

The methodology for conducting design research, as proposed by Blessing and Chakrabarti (2009) and which includes the development of a design support tool serves as a reference for the research conducted in this thesis and is presented in this section. They describe the term support in this context as: *"the possible means, aids and measures that can be used to improve design and includes strategies, methodologies, procedures, methods, techniques, guidelines information sources, software tools, etc."* In the case of this thesis, the support is the software application DiDeas II.

The existing and desired situations in design play a central role in the methodology and are described by a reference and an impact model respectively. The reference model is the reference against which the intended improvements are benchmarked, while the impact model shows the assumed impact of the support to be developed. Initial versions of these models are developed during the early stages of the research and are developed further during the latter stages of the research.

The development of success criteria is also proposed. Such criteria aid in focussing the investigation and relates to the ultimate goal to which the research project or programme intends to contribute, thereby providing a means to determine whether the results achieve this aim. Success is, however, difficult to determine according to specific metrics and it might not be possible to determine during the timeframe of a given research project. It might therefore be necessary to develop measurable success criteria that can be applied to judge the outcomes of the research with the given resources.

The research methodology is divided into the following four stages:

- 1) Research Clarification (literature analysis). This stage aims to support researchers formulating a clear, challenging but realistic overall research plan. During this stage the initial reference and impact models are developed.
- 2) Descriptive Study I (empirical data analysis). During this stage the understanding of design and its success factors is increased by investigating the phenomenon of design through reviewing the literature about empirical research, undertaking empirical research and through reasoning. The reference model is completed and the impact model is updated.
- 3) Prescriptive Study. During this phase, the design support is developed and the impact model is completed. An outline plan for the evaluation of the support is also developed.
- 4) Descriptive Study II. This stage focuses on the evaluation of the developed support to determine whether it can be used for the intended task and whether it addresses the success factors.

Data collection is an important aspect of this research, both during the initial survey and interviews, and the case studies conducted at the end of the thesis. Blessing and Chakrabarti (2009) provide some guidelines for data collection and present various methods. A distinction is made between quantitative and qualitative research, with a quantitative approach being applied to investigate or measure the degree in which phenomena occur (the methods used include experiments, observations, closed questionnaires, etc.), while a qualitative approach is used to investigate the nature of phenomena (the methods used include interviews, observation and hand-written documents such as open ended questions on questionnaires and diaries). The research can either be conducted in a laboratory or an industrial environment (termed field research) and the way in which the researcher is involved in an empirical study can influence the outcome, even in pure observational studies. The following methods of data collection during empirical studies are described:

- Observation. Observation methods involve the researcher recording what is actually taking place either by hand or using measuring or recording equipment and takes place in real-time.
- Simultaneous verbalisation. This refers to the situation where participants speak aloud while working. Participants may have been asked to do so, or this may be a natural part of their work. The aim is to provide insight into the cognitive behaviour of participants, which may not be obtained through normal observation.
- Case Study. The term case study is often used to describe a study that involves data from a real setting and is seen as equivalent to an observational study in which one or a very few cases are involved. A one-shot case study cannot be used for exploratory research or for pre-testing some research hypothesis.
- Collecting Documents. Retrieving documents related to a particular project, topic or product from a variety of sources can be very useful as an additional data-collection method.
- Collecting Products. Physical products and any mock-ups, prototypes and other physical models can be part of the collected data e.g. to trace the development of a product.
- Questionnaires. Questionnaires are used to collect thoughts, beliefs, opinions, reasons etc. from people about past, present or future events by asking questions and is focussed on data that cannot be captured using observation or simultaneous verbalisation and on data about the past that was not captured. A potential bias is introduced in the results as people are forgetful, see things from their own perspective or provide answers that are coloured by what they conceive as more desired with respect to the purpose of the interview, social standards or their own behaviour and that of others.
- Interviewing. Interviews are carried out face to face with similar purpose to that of questionnaires. The interview should provide a framework within which respondents can express their own understandings in their own terms.
- Action research. This is an approach to introducing and evaluating change, originally in organisations and programmes, but increasingly in design. It has the aims of research and action. Through cycles of action and research a better understanding is obtained, while at the same time the organization or programme under investigation is gradually changed.

### 2.1.2 Groupware and CSCW

Some authors, such as Blessing and Chakrabarti (2009), use the terms Computer Supported Collaborative Work (CSCW) and groupware interchangeably; however, many authors make a distinction between these concepts and use the term CSCW to refer to a scientific discipline (Greenberg, 1991) or a generic term combining the understanding of the way people work in groups with the enabling technologies of computer networking and associated hardware, software, services and techniques (Wilson, 1991). The term groupware is then used specifically to refer to software that enables users to work collaboratively on projects or files via networking ("groupware", 2012) and encompasses many technologies and business process areas.

Wilson (1991) states that the aim of CSCW is to facilitate group work effectiveness and has two major areas of concern: the group working process and the technology that might be used to support it. Wilson (1991) then divides the aspects affecting the group work process and other more people-oriented aspects, central to the problem of improving group efficiency, into the following four categories:

- Individual human characteristics such as conversation patterns and commitment making.
- Organizational aspects such as organizational structure and culture.
- Group work issues such as user involvement in the design process, rapid prototyping and usability testing.
- Group dynamics such as group decision-making and collaboration.

Wilson (1991) also divides the technology employed (groupware) into the following four categories:

- Communication mechanisms enabling people at different locations to see, hear and send messages to each other – for example, video conferencing and electronic mail.
- Shared workspace facilities that enable people to view and work on the same electronic space at the same time – for example, remote screen sharing.
- Shared information facilities allowing people to view and work on a shared set of information – for example, multi-user databases.
- Group activity support facilities to augment group work processes such as the co-authoring of documents and idea generation.

User interfaces are important aspects in CSCW. According to Heer et al. (2007), "*Information visualization leverages the human visual system to improve our ability to process large amounts of data*". Visualization supports the process of sense making, in which information is collected, organized and analysed to form new knowledge and inform further action. Sense making is often a social process. As participants build consensus or make decisions they learn from their peers.

Interactive computer systems are built in order to help people achieve some goals as efficiently as possible. The user interface, which is often the yardstick by which a system is judged, causes at best a high level of user errors to be incurred and at worst users being unwilling to use the software irrespective of its functionality (Blessing & Chakrabarti, 2009).

In groupware, *what you see is what I see* (WYSIWIS) is a term that is often used and refers to an interface where information is presented to all participants in a consistent manner to provide a shared context (Schlichter & Borghoff, 2000). WYSIWIS might be implemented in a strict manner where all participants are displayed identical information simultaneously or a more relaxed form might be implemented such as the following examples:

- Information is divided into public and private areas. In the public areas, the information is available to all users for viewing and manipulation, while the private workspace is invisible to other users.
- Information is displayed or represented differently to different users, who are presented with different parts of a shared context and can manipulate these different parts independently.
- Time divergence where the shared context is only updated after a certain time delay and might either be initiated explicitly by the user or implicitly by the system.

Various computer mediated communication tools are in existence and can be divided into asynchronous communication and synchronous communication methods. Asynchronous communication is considered first:

E-Mail is the most widely used collaborative software application. Features include forwarding and archiving of messages and the attachment of files to messages (Typical Collaborative Software Applications, 2012).

Group calendars allow scheduling, project management and coordination among many people. Such applications can typically detect when schedules conflict or organise meeting schedules to accommodate everyone (Typical Collaborative Software Applications, 2012).

Collaborative writing systems can allow users to track changes and make annotations to documents. Authors collaborating on a document may be allowed to lock parts of the document or link separately-authored documents (Typical Collaborative Software Applications, 2012).

Newsgroups and mailing lists are intended for exchanging messages among large groups. The main difference between newsgroups and mailing lists is that newsgroups are an “on-demand” service that only show messages to a user when they are explicitly requested, while mailing lists have an “interrupt driven” interface that delivers messages as they become available (Typical Collaborative Software Applications, 2012).

Wikis. A Wiki is an internet-based platform that allows users to update the contents of web-pages without having to use a programming language such as HTML. A Wiki can be accessed from any computer with internet access and by any user with sufficient access rights (Walthall et al., 2009).

Now consider synchronous communication methods:

Shared whiteboards allow two or more people to view and draw on a shared drawing surface even from different locations. Most shared whiteboards are designed for informal conversation, but they may also serve structured communications or more sophisticated drawing tasks, such as collaborative graphic design, publishing or engineering applications (Typical Collaborative Software Applications, 2012).

Chat systems permit many people to write messages in real-time in a public space. As each person submits a message, it appears at the bottom of a scrolling screen. Text-based communication provides a direct transcript of the conversation, which has both long-term and immediate value as participants can join a conversation and quickly understand the context (Typical Collaborative Software Applications, 2012).

Video communication systems allow for one-on-one or conference calls. Cost and compatibility issues limited early use of video systems to scheduled videoconference meeting rooms. Video is advantageous when visual information is being discussed, but may not provide substantial benefit in most cases where conventional audio telephones are adequate (Typical Collaborative Software Applications, 2012).

Decision support systems facilitate meetings and decision making in groups and provide tools for brainstorming, critiquing ideas, putting weights and probabilities on events and alternatives and voting. Such systems are supposed to enable more rational and even-handed decisions and also encourage equal participation during meetings by, for example, providing anonymity or enforcing turn-taking (Typical Collaborative Software Applications, 2012).

### **2.1.3 Concept Design Methodologies**

This section provides an overview of a selection of design methodologies for the early phases of engineering design that are presented by three well-known references, i.e. Ullman (2003), Blanchard and Fabrycky (2006) and Cross (2006). Quality function deployment (QFD), functional blocks, the morphological method and Pugh's method (decision matrix method) are all described by more than one of the above mentioned authors and are presented first. The QFD and decision matrix methods form an integral part of the case studies and evaluation of DiDeas II and are therefore discussed in detail.

#### **2.1.3.1 QFD**

According to Cross (2006), the QFD method recognizes the person who buys (or influences the buying decision) as the most important person in determining the

commercial success of a product and provides a comprehensive method for matching product attributes (customer requirements) to engineering characteristics (physical properties). The relationship is close and should be clearly understood to avoid confusion.

The QFD method for generating engineering specifications includes the development of a house of quality or QFD diagram as shown in Figure 2:

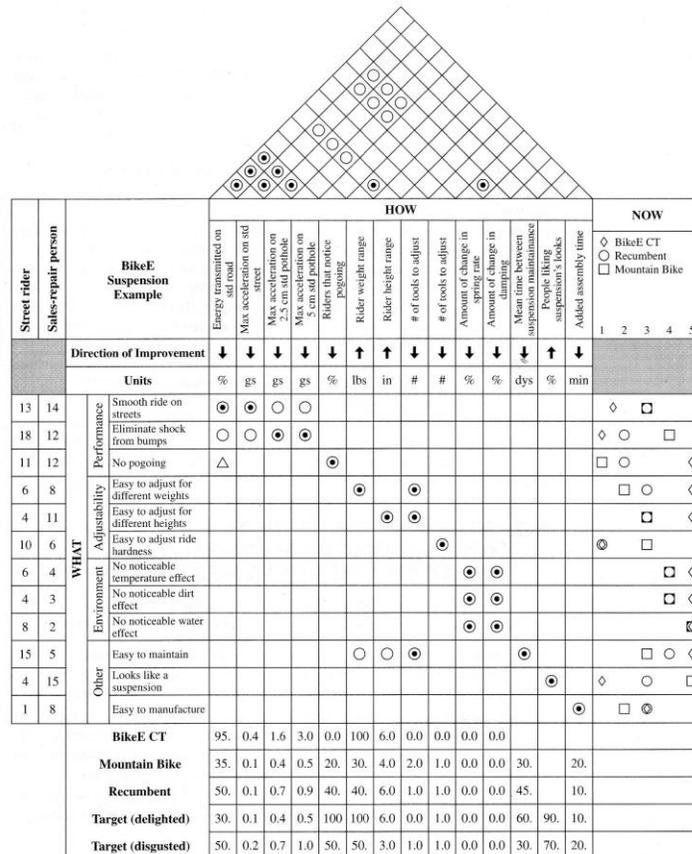


Figure 2: QFD House of Quality (Ullman, 2003)

Ullman (2003) and Cross (2006) describe the following steps for completing the house of quality diagram:

- Identify the customers. The customers can include people involved in manufacturing and assembly who are also regarded as customers.
- Identify the customers' requirements and desired product attributes. These include functional performance requirements, human factor requirements,

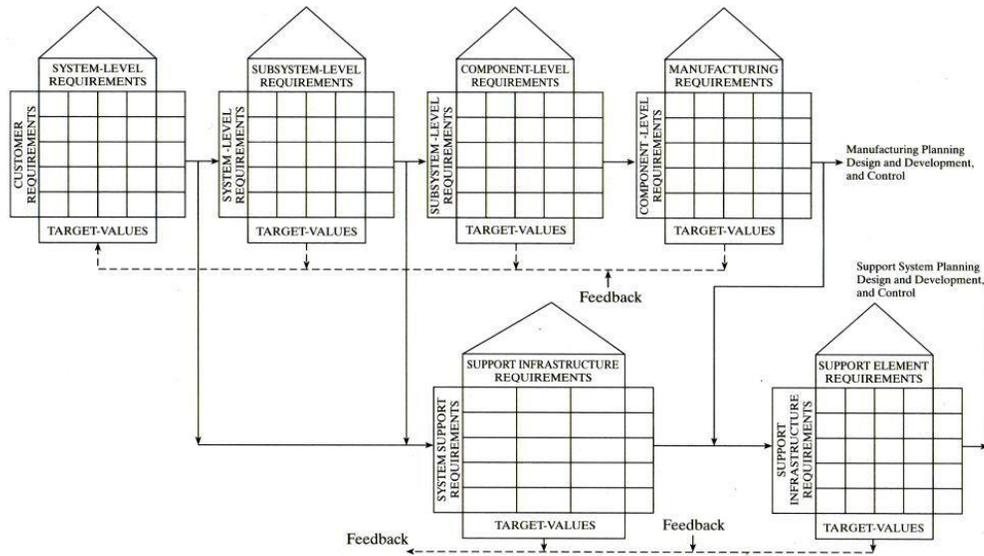
physical requirements, reliability, resource concerns, life-cycle concerns and manufacturing requirements.

- Rank the requirements or product attributes according to importance and assign a relative weight to each. This might require that the attributes of the team's own existing product be compared to that of other products if a product is being redesigned or improved (see below).
- Identify and evaluate the competition. Competitor benchmarking may be used where each competing product is evaluated according to customer requirements.
- Generate measurable engineering specifications or engineering characteristics from the customers' requirements.
- Relate customers' requirements to engineering specifications. A matrix is drawn to display the relationships between the engineering specifications and the client requirements. The client requirements are listed along the left edge of the matrix and the engineering specifications are listed along the top edge. The relationship is then indicated by a value in each cell where a relationship occurs in the matrix.
- Set engineering targets. Determine how the competition meets the engineering specifications and establish a target for the new product, in order to satisfy customer requirements or to improve the product over its competitors.
- Identify relationships between engineering requirements. A triangular roof shaped section is added to the matrix to show interactions between engineering characteristics and how dependent they are on each other.

Blanchard and Fabrycky (2006) also state that the QFD method is used to facilitate the translation of a prioritized set of subjective customer requirements into a set of system-level requirements during conceptual design. A similar approach may be used to translate system-level requirements into a more detailed set of requirements at each stage in the design and development process and involves the construction of one or more matrices where the "hows" (column headings) of one matrix become the "whats" (row headings) of a succeeding matrix (see Figure 3).

#### *2.1.3.2 Functional Decomposition*

Functional modelling has the aim of decomposing the problem in terms of the flow of energy, material and information. Functional modelling forces a detailed understanding at the beginning of the project as to what the product is to do. Firstly, the overall function that needs to be accomplished by the system is identified and stated as a single clause. The system is treated as a "black box" with inputs and outputs across the boundary of the system (Ullman, 2003). This overall function is then broken down into essential sub-functions. A block diagram is used to show the interactions between sub-functions and displays the system boundary (Cross, 2006).



**Figure 3: Traceability of Requirements through a "Family of Houses" (Blanchard and Fabrycky, 2006)**

In the systems engineering approach described by Blanchard and Fabrycky (2006), functional analysis is initiated during the latter stages of conceptual design and is facilitated through the use of functional flow block diagrams that include block numbers showing sequential and parallel relationships, initially for top-down traceability of requirements and later for a bottom-up traceability and justification of physical resources required to accomplish these functions. The analysis is extended from the system level down to the sub-system level and below as required during the preliminary system design phase.

### 2.1.3.3 Concept Generation

Concepts are the means for providing function and can be represented as verbal or textual descriptions, sketches, paper or clay models, block diagrams or any other form that gives an indication of how the function can be achieved. Various methods for concept generation are presented by Ullman (2003), including the following methods:

- **Brainstorming.** Each group member contributes ideas from his/her own viewpoint. As many ideas as possible are generated. These ideas are verbalised and all ideas are recorded. Ideas are not to be evaluated.
- **The 6-3-5 method.** Team members are arranged around a table with the optimum number of members being six. Each takes a clean sheet of paper and divides it into three columns. Each team member then has five minutes to write down three ideas on how to perform a specific function before passing the piece of paper on. The previous ideas are then studied before

being added upon or ignored in the next round. At the end of the session, team members discuss the possibilities.

- The morphological method (described in more detail below).
- Logical methods for concept design. The theory of inventive machines (TRIZ) developed by Artshuller (2002) is based on the idea that many of the problems engineers face contain elements that have already been solved, often in a completely different industry and for a totally unrelated situation that uses a totally different technology to solve the problem. Another method, axiomatic design, is based on the relationships between four design domains: customer, function, physical and process (Suh, 2001).

The morphological method exploits the phenomenon that the re-ordering of even a small number of components can lead to a very large number of combinations. The aim is to widen the search for possible new solutions. The features that are essential to the product are listed to try to establish the essential aspects that must be incorporated in the product or that the product must be capable of doing (Cross, 2006). As many concepts as possible are developed for each function in the functional decomposition. These concepts should be kept as abstract as possible and at the same level of abstraction. The concepts for each function are then combined into complete conceptual designs. This can lead to a large number of concepts and not all concepts are practical. Only the ideas that are reasonably possible are to be considered (Ullman, 2003).

#### *2.1.3.4 Concept Evaluation*

Evaluation implies both comparison and decision making. There are two possible types of comparisons. The one is absolute in that each alternative concept is directly compared with some target set by a criterion, while the second type is relative in that alternative concepts are compared with each other using measures defined by the criteria.

Concept selection can be done by guesswork, intuition, experience or arbitrary decisions. However, a more rational or open procedure is preferable as the designer will feel more secure in the decision and others involved in the decision will be able to participate or assess the validity. The overall value or utility of a design proposal regarding the objectives is assessed in the evaluation. However, different objectives have different levels of importance and it is necessary to have the objectives weighted differentially.

Each of the above mentioned authors describe the method whereby alternatives are compared according to weighted objectives. For example, Ullman (2003) describes the decision matrix method (Pugh's Method) as an iterative method with the following steps:

Step 1: Choose the criteria for comparison. Usually the customer requirements are used as a basis for comparison. If they have not been developed, the first step should be to develop criteria for comparison.

Step 2: Develop relative importance weightings. To determine the relative weights, Ullman (2003) describes the fixed sum method, where 100 points are distributed among the requirements which means that certain criteria are to be rated low if others are to be rated high. Cross (2006) suggests assigning the weights by using an objectives tree. The highest level objective is assigned a value of 1.0 and at each lower level the sub-objectives are given weights relative to each other along with a 'true weight' which is in turn a fraction of the 'true weight' of the objective in the level above them.

Step 3: Select the alternatives to be compared. These include the different ideas generated which are to be compared at the same level of abstraction and represented in the same manner.

Step 4: Evaluate alternatives. The objectives are converted into parameters that can be measured where possible, while others are assigned a utility score estimated on a points scale.

Step 5: Compute the satisfaction. Finally the performance measures for each parameter are multiplied by the weight value and can simply be added up to allow for comparison among the alternatives.

Blanchard and Fabrycky (2006) also describe the weighted method for comparing alternatives and describe the *method of paired comparisons* for ranking criteria or objectives.

Another method described by Ullman (2003) is Go/No-Go screening. In this method, each concept is compared with the customer requirements in an absolute fashion and is also evaluated in terms of the maturity of the technology used. The method is most effective if each design team member performs it independently and the individual results are then compared. The results may vary widely since neither the concepts nor the requirements may be refined.

## **2.1.4 Customization**

As mentioned in the introduction (Chapter 1), different design styles exist and different designers and companies incorporate different strategies and methodologies into the design process. Therefore a design support tool should be customisable to enable different designers to follow their preferred approach. The specific approach for customization of software followed by Liu (2007) was the use of ontology and conceptual graphs.

### **2.1.4.1 Ontology and Conceptual Graphs**

Various definitions and goals of ontology are given in literature. Until a few years ago, ontology was seen as "the study of the kinds of things that exist" (Chandrasekaran et al., 1999) in the context of artificial intelligence and information systems. Recently, the phrase an ontology has come into use, and according to Neches et al. (1991), "An ontology defines 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 to the vocabulary". Similarly, according to Chang et al. (2008), "Ontology provides a shared conceptualization of a particular domain and captures

knowledge about concepts and the relations among them in the domain in order to formalise domain knowledge in a generic way and to provide a common understanding of a domain, which may be used and shared by applications and groups". Chandrasekaran et al. (1999) also state that the conceptualisations that the terms in the vocabulary are intended to capture, qualify as an ontology and not the vocabulary. Therefore translating these terms from one language to another does not change the ontology conceptually.

Ontology has grown past philosophy. It now has many connections to information technology where certain divergences occur among ontologies, yet the following agreements exist (Chandrasekaran et al., 1999):

- *Objects* exist in the model.
- Objects have *properties* or *attributes* capable of holding *values*.
- Objects have various relations among them.
- Properties and relations can change over *time*.
- *Events* occur at different *time instants*.
- There are *processes* in which objects participate and that occur over time.
- The world and its objects can be in different *states*.
- Events can *cause* other events or states as *effects*.
- Objects can have *parts*.

Basson et al. (2004) regarded an ontology as the conceptualization of design patterns according to different design methodologies applied in different companies.

Sowa (2008) developed a version of conceptual graphs as an immediate language for mapping natural language questions and assertions to a relational database and gave the following definition of a conceptual graph: "A conceptual graph is an abstract representation for logic with nodes called concepts and conceptual relations, linked together with arcs." Each concept has a type and can have a "name". The conceptual relations also have types.

Liu and Basson (2007) combined conceptual graphs with an ontology-based approach to create a database structure for DiDeas II in which all design information is classified either as an element or a relation of a specified type. The elements are only linked to relations that are in turn only linked to elements (a property of a conceptual graph). The set of available types of elements and relationships constitute the ontology. Two of the main databases in DiDeas II are the Ontology Database, in which the element and relation types are stored, and the Project Database, in which the design project's information is held. Each piece of project information is classified as an element or relation of a type in the Ontology Database and changes in the design process or terminology are implemented in the databases by adding to the Ontology Database, but the database structure does not change. This is the most significant difference between the approach implemented in DiDeas II and the typical relational database structure applied in DiDeas I, or even object

orientated database approaches, which require prior knowledge of the design process to set up the database.

#### 2.1.4.2 Tailoring

Tailorability is a property of software which allows for the change of certain aspects of the software in order to meet different user requirements. On a technical level, the software architecture has to provide means of changing system behaviour other than rewriting and recompiling source code. According to Slagter et al. (2001), tailoring refers to modifying the system within its context of use. Tailorable systems can be changed by users after implementation to specific preferences or different tasks.

Specifically the dynamics of the group, the performed task, and the context in which the task is performed, contribute to changes in requirements on the technical support. Evolution in the use of groupware is the main reason why it needs to be modifiable (Slagter et al., 2001).

Mørch (1997) classifies the following three levels of tailoring:

- Tailoring by customisation. Users select from a pre-configured set of configuration options.
- Tailoring by integration. Users select the functionality of the software from a list of available functions.
- Radical tailoring. The set of functions to choose from can be expanded by adding new building blocks (referred to as software components in software engineering (Slagter et al., 2001)) to the program.

The tailoring options in the first two types are said to be closed, while that of radical tailoring is said to be open. An example of open tailoring is where users can download building blocks from the Internet. Open standards are required as different building blocks can originate from different vendors. It is likely that only more skilled users will use open tailoring, nevertheless it needs to be as clear as possible. A control mechanism is also to be implemented to assign tailoring rights to different people (Slagter et al., 2001).

The following properties that play an important role in the component based design of groupware are described by Slagter et al. (2001):

- *Composability:* A system is considered composable if it can be composed out of separate building blocks. To allow for composition at run-time, the system should allow dynamic discovery of new components.
- *Extensibility:* A system is extensible if new functionality can be added to the system without changing the existing parts.
- *Vertical openness:* Allows a group of people to combine the services that best suit their needs.
- *Horizontal openness:* Allows different people to use different groupware systems to communicate.

## 2.1.5 Networking Architecture

A distributed application is an application that contains two or more software modules that are located on different computers. The software modules interact with each other over a communication network connecting the different computers (Verma, 2004).

### 2.1.5.1 Client-Server Architecture

The client-server model (Verma, 2004) is a configuration in which a distributed application is structured into two distinct software modules namely a server module and a client module. There is only one instance of the server module present and multiple instances of the client module.

In this architecture, communication only takes place between the client and the server and each client needs to discover the network address of the server.

For discovery, the server runs on a port and network address known to the client module and the client connects to this network address. Once a connection is made, the server and client modules are able to communicate. The server does not need to be configured with any information about the client and can communicate with any number of client modules. The only constraint is the resources available as the server needs to respond within a reasonable time to all connected clients.

The simplicity and ease of maintenance of client-server architecture are the key reasons for its widespread usage in the design of distributed applications at the present time.

### 2.1.5.2 Peer-to-Peer Architecture

In the peer-to-peer architecture (Verma, 2004), the distributed application is structured to consist of many identical software modules, each located on a different computer. These modules communicate directly with one-another to perform the processing of the distributed application.

Each module can be viewed as both a client and a server module and can access services from modules running on other computers, as well as provide services to them. The discovery process in this architecture is more complicated than that of the client-server architecture since each module needs to know the network address of the computers that the other modules are running on, or at least a sub-set with which it needs to communicate and propagating changes to the different software modules is much more difficult.

### 2.1.5.3 Hybrid Architecture

In real life, one could also use a hybrid approach that is a mixture between the client-server architecture and the peer-to-peer architecture. The hybrid approach places some software modules on a set of computers that can act as servers and others act as clients. The hybrid approach for some distributed applications can often result in a better balance between the ease of software maintenance, scalability and reliability.

In a pure peer-to-peer architecture, all nodes are identical. Yang and Garcia-Molina (2001) describe a subset of peer-to-peer systems where some functionality is centralised, such as file indexing which is performed at server nodes. Certain functionality can be performed more efficiently in a centralised manner.

#### 2.1.5.4 Comparisons between Client-Server and Peer-to-Peer Architectures

The following comparisons between the two architectures are made by Verma (2004):

- Ease of Development. There exists a large number of debugging and development tools for client-server systems, leading to lowered risk of undiscovered bugs. Debugging and testing of peer-to-peer applications is also more complicated as the interaction between several components is required.
- Manageability. With a centralised system, maintenance tasks, such as backup, upgrades and bug fixing, is easier than on a decentralised system such as peer-to-peer applications. Peer-to-peer applications might also need to run on multiple platforms, making maintenance difficult. With client-server systems, a single platform is required for the server-side application and very little maintenance is required for a standard client module.
- Scalability. This can be measured in terms of the number of user-level interactions the application can support while maintaining a reasonable performance level. The combined processing power of several large computers could easily surpass the processing power available from even the best single computer, and the peer-to-peer architecture could thus result in much more scalable applications. However, a centralised solution is more efficient than a distributed solution in most cases.
- Administrative Domains. A peer-to-peer system can be created by using computers from many different administrative domains. Therefore if the usage of the software requires that computers from many different administrative domains are used, a peer-to-peer system is preferable.
- Security. Generally, the security of a centralised system can be managed much more readily than that of a distributed system which has multiple sites that are vulnerable and require replicated security apparatus and mechanisms.
- Reliability. The reliability of a system is measured by its ability to continue working when one or more of its components fail. In the context of computer systems, reliability is often measured in terms of the ability of the system to run when one or more computers hosting the system are brought down. High-reliability computer applications can be developed by using either client-server or peer-to-peer architectures. Distributed peer-to-peer systems, for

most applications, use multiple computers to do identical tasks, and thus the system continues to be operational and available, even when a single computer fails or goes off-line. As in the case of scalability, the difference between the reliability of a server-centric approach and the peer-to-peer approach is that of the cost at which the reliability is achieved. The peer-to-peer approach provides for a much lower cost solution for reliability than the server-centric approach.

In summary it can be said that a client-server approach provides for better security, manageability, and ease of development, whereas the peer-to-peer approach provides for increased reliability and scalability in a more cost-efficient manner and allows for interoperation across multiple administrative domains.

### **2.1.6 File Access Control**

The purpose of access control is to limit the actions or operations that a legitimate user of a computer system can perform based on privileges and allows for the cordoning off of certain portions of a database, thus enabling users to access specific data (Oracle, 2003). The following concepts are relevant to file access control:

*Granularity* - the degree to which data access is differentiated or the extent to which the system contains discrete components of ever-smaller size (Miltchev et al., 2006).

*Authentication* - the process by which a user's identity is verified.

*Authorization* - the process by which a user's privileges are ascertained.

*Confidentiality* – the system only allows individuals to see the data they are allowed to see.

*Integrity* – the system ensures that the data it contains is valid.

*Availability* – data is available to authorised users without delay.

Collaborative systems contain information and resources with varying levels of sensitivity and the applications deployed in such systems create, manipulate and provide access to a variety of such protected information and resources. It is difficult to balance the competing goals of interaction and information security. The protection of contextual information and resources in collaborative systems entails addressing additional requirements not raised by traditional single-user environments, due in part to the unpredictability of users and the unexpected manners in which users and applications interact in collaborative sessions (Tolone et al., 2005).

There are various models for controlling users' access to data:

Access Matrix Model: this is a conceptual model in which subjects (the active entities which are the users or programs executing on behalf of the user) initiate actions on objects (the entities or resources that can be accessed). These actions are permitted or denied determining the authorization specified in the system. The access matrix specifies the access rights that each subject has for each object. In an access

matrix, the rows represent the subjects and the columns represent the objects (Tolone et al., 2005).

Access Control Lists (ACLs): Each subject is associated with an access control list that specifies the access rights each subject in the system has for the specific object. ACLs provide a convenient access review of an object (Sandhu & Samarati, 1994).

Discretionary access control: These policies govern the access of users to information on the basis of the user's identity and authorizations that specify for each user (or groups of users) and each object within the system the access modes the user is allowed on the system. Each request for a user to access a specific object is checked against the specified authorizations (Sandhu & Samarati, 1994).

Mandatory policies: Access is granted based on the classification of subjects and objects in the system. Each object and subject is issued a security level. The security level of an object relates to the sensitivity of the information, while the security level associated with a subject is meant to reflect the user's trustworthiness (Sandhu & Samarati, 1994).

Attribute based access policy (ABAC): In this policy access control, the attributes of objects and subjects are used as the basis for authorisation. These attributes can be static or dynamic (Priebe et al., 2006).

Role-Based Access Control (RBAC): A role can be defined as a set of actions and responsibilities associated with a particular working activity (Sandhu & Samarati, 1994). In this model, access rights are assigned to roles instead of individual users. Roles are assigned for different job functions and different users are assigned roles according to their rank and responsibilities. Users can easily be assigned from one role to another without changing the underlying access structure. RBAC is therefore more scalable than user-based security specifications. Role hierarchies define roles that have unique attributes and may implicitly contain the operations, constraints and objects associated with other roles (Ferraiolo et al., 1995).

Another aspect of multi-user file access is version control. Version control systems are used in software development and the basic idea of version control is to separate the working copies of files (the copies that users work on) from the master copy stored in the repository (Louridas, 2006). The following are key concepts of version control (Spinellis, 2005):

- *Version*: An identifiable instance of a specific file or release of a complete system.
- *Trunk*: The software's main line of development.
- *Branch*: A version of a file that has not been incorporated into the main line of development.
- *Merge*: To combine different changes to the same file.

When multiple users can change a given set of data, the situation can arise where two users attempt to update the same data, introducing a form of conflict. According to Louridas (2006), the following two models for conflict resolution exist:

- *Lock-modify-unlock*. Files cannot be modified by other users while locked, but can however be viewed. The lock-modify-unlock model is simple yet restrictive and is especially cumbersome in distributed development when people in different time zones work on a project.
- *Copy-modify-merge*. This method is used in version control systems (VCS) and allows users to check files out, make changes and check in (commit) these changes. Every time a file is committed, the VCS adds this file to the repository meaning that all versions of a file are stored. Users are free to edit the working copy. If conflicting modifications are made, users are alerted and must resolve the conflict by merging modifications.

## 2.2 Survey of Existing Software

There are various commercial software packages available with groupware tools to support collaboration among distributed teams. These include applications that aim to facilitate multiple aspects of collaboration (e.g. Microsoft SharePoint Workspace) or might only be aimed at certain aspects thereof, such as Voice over Internet Protocol (VoIP) and videoconferencing (e.g. Skype) or file sharing (e.g. DropBox). Some applications are aimed specifically at engineering design and are integrated with commercial CAD packages. Some of these software applications were analysed to better understand which features different groupware applications employ to satisfy the needs of users.

### 2.2.1 Microsoft SharePoint Workspace (previously Microsoft Groove)

SharePoint Workspace is a desktop program that allows for collaboration among team members via various groupware tools. File sharing functionality allows users to open, edit and organize files directly within the program interface and updates to files are transmitted as update packets instead of entire files to improve bandwidth efficiency.

Various communication tools are incorporated for collaboration among team members and users can select which tools to incorporate into each workspace. These include meeting management which allows for the scheduling of meetings and the recording of meeting minutes, instant text messaging functionality and asynchronous communication for discussions about specific topics.

Users can work both online and offline. Data is synchronised dynamically among online collaboration points and updates are made as soon as team members go online. Users can “check-out” (lock) files and changes will only be uploaded to the workspace when the file is “checked-in”. Other users can access such a file in read-only mode.

### **2.2.2 Microsoft OneNote**

OneNote provides a single place to gather notes and information which can also be shared by multiple users in shared notebooks and can be saved either locally, on a network drive or on the web. It includes collaborative tools that allow teams to work together both offline and online. Information can be gathered in multiple formats including text, pictures, digital handwriting, audio and video recordings, and more.

Tabs on the left-hand side of the interface can be used to select different note-books; while tabs at the top can be used to select different sections within each notebook and tabs on the right-hand side can be used to select individual pages within each section of the notebook.

Users can attach information in various formats such as images or tables anywhere on a page which provides for a less structured interface with information being displayed in a visual manner.

### **2.2.3 IBM Leaf Notes/Domino**

Lotus Notes and Lotus Domino is a client-server platform that provides a single point of access to information and incorporates various collaboration functionalities such as e-mail, calendars, feeds, instant messaging, contact management, discussion forums and blogs while allowing users to create, open, share and access documents. Lotus Domino is the server program, while Lotus Notes is the client application. The client can also be tailored through the use of 3<sup>rd</sup> party or custom “widgets” or modules that can be added to the environment. Users can work while disconnected from the server and changes are automatically synchronised once they connect.

### **2.2.4 Commercial CAD Packages Integrated Collaboration Tools**

There exist various data management tools that are developed by CAD software companies and are integrated in their CAD packages. These include packages such as Autodesk Vault and Siemens Teamcenter.

Autodesk Vault is a product data management (PDM) system that allows engineering and design workgroups to organize, manage and track data creation, simulation and documentation processes. Vault is integrated with other Autodesk CAD, simulation and product lifecycle management (PLM) packages. Teamcenter is a product lifecycle management (PLM) suite that incorporates different applications. Features include collaboration via data sharing and in real-time, document and record management, virtual conferencing and the recording of requirements that are traceable.

### **2.2.5 DropBox**

DropBox is a service that allows users to backup files on online storage space. The service can be accessed via devices such as computers and smartphones that have

the DropBox application installed or from the DropBox website. The files are automatically saved on all devices connected to the specific account.

When multiple users edit a file at the same time, the first copy is saved while changes made by other users will be saved as “conflicted copies” along with the name of the person or computer responsible and the date.

### **2.2.6 Skype (VoIP)**

Skype is a peer-to-peer, Voice over Internet Protocol (VoIP) client that allows communication over the internet via voice, video and instant messaging (Baset & Schulzrinne, 2004). Skype also incorporates other features such as video conferencing, conference calls (voice), file transfer and screen sharing.

The Skype network consists of ordinary and super nodes and a login server. The ordinary nodes are client applications while the super nodes are host endpoints and can be any node with a public IP and sufficient resources (Baset & Schulzrinne, 2004).

### **2.2.7 Online Collaboration Tools**

Various online collaboration tools exist which allow teams to access and edit documents from anywhere in the world. One such example is the Google Drive functionality which allows users to log into a Google account over the Internet and edit documents that are shared among various members which might constitute a team. Documents can be saved locally in different formats. Users are provided with online storage space and can upload files or create files online.

## **2.3 Targeted Survey of Engineers**

A web-based questionnaire was used with the aim of gathering information on the communication, design habits and design preferences of engineers during the conceptual or early phases of engineering design. The details of the survey are provided in Appendix A. Various engineers in South Africa were approached, but only twelve completed the survey, providing feedback from a very limited group. However, these participants fall into the specific target group the software is aimed at and their feedback is therefore valued. The twelve participants that completed this survey have varying levels of education, as well as experience which ranges from six months to 27 years. They are mostly employed at small to medium sized companies, with two from larger companies.

### **2.3.1 Work Related**

A set of questions in the survey were aimed at understanding the typical nature of the work the participants are involved in. This section summarises the results.

All participants are typically involved in more than one project at any given time, while nine participants stated that they work on more than one project in a typical day.

All participants listed the nature of their design work as involving smaller production volumes of 100 systems or less, with five participants listing their work as involving prototypes.

Eight of the participants stated that more than half of their time is spent on technical work, as opposed to management related work.

Collectively the participants are more involved in the design of products that are original or new to the company or section and are less involved with configuration design. They are also more involved in the design of products for clients, as opposed to equipment to be used by the company.

### **2.3.2 Communication**

Some questions in the survey were aimed at understanding the different communication media that the participants employ in specific situations and the different formats used to convey information while communicating.

Free-hand and formal sketches form a large part of the communications among engineers, with eight participants claiming to use free hand sketches more than five times per week during communication and nine participants claimed to use formal sketches more than once a month. CAD models and drawings are also used often with some participants using 3D CAD models more than twice per day. Other information such as catalogues (electronic or paper), information on websites and diagrams or flow charts are also used frequently.

Various sources of information such as websites, catalogues etc. are also used regularly during typical design projects.

For communication among members of the same team, e-mail, phone and ad-hoc meetings are used most often, followed by fax, Skype as well as memoranda and reports. For communication with other teams outside the section or company, e-mail, phone and previously arranged meetings is used most often. For communication with clients, e-mail, phone, previously arranged meetings as well as memoranda and reports are used most often.

User requirements are communicated to team members either during team meetings (where minutes might be taken) or as documents that are available to team members, either electronically or as hard copies. The electronic files are stored on the company network where they are accessible to all team members.

Project progress is communicated to people outside the design team by various means including meetings (telephonic, video conference and other) and e-mails, or through electronic reports or other files such as CAD files.

Project progress is reported to design team members relatively often, with eight participants stating that progress is reported between two and five times per week. It is reported through impromptu meetings and e-mails, and also through more formal meetings and status reports with one participant mentioning CAD files.

Three participants responded that there is no means for project members in their company to view project progress on demand. Others reported that they use project management software such as Microsoft Project, while others use repositories for the data or even a physical white board.

### **2.3.3 Design Habits**

These questions aimed to determine the resources employed for design by the various companies, and the manner in which information is stored.

All participants store free-hand sketches in a physical file with two mentioning that they sometimes scan these to obtain electronic copies.

All but one participant keep project notes in a physical design file, while five document the notes in electronic files on the computer.

Five participants record notes on phone conversations in a design file, while five do not record anything about phone conversations. Two use confirmation e-mails.

The participants store e-mails in various electronic means, such as copying them to a design folder or using the e-mail client or other software to archive the e-mails. Two participants print e-mails and store them as hard copies.

Only one participant does not store design decision documentation. The rest use a project file or electronic copies stored in the project folder, a repository or a document management system.

Eight participants state that they re-use design information gathered in previous projects more than once a month, while all participants stated that they re-use information. The majority of participants store this information both electronically and in a physical format, with some participants storing it only electronically and two participants only storing it physically.

Various CAD programs are used for layout designs and there is no clear support for a single CAD package.

Seven participants stated that they use software such as database management systems for documentation of projects.

Project planning software is used by eleven of the twelve participants.

Web searches and supplier websites are used often as sources of information. Personal and company catalogue libraries are also used often, as well as textbooks. In-house design procedure documents are used by eight participants.

### **2.3.4 Methodology**

These questions were aimed at determining the level to which the participants employ formal design methodology during design.

All but one of the participants make use of project planning software during the early phases of the project. All participants use the Gantt charts feature with one participant using spread sheets to create Gantt charts. Eight participants use Work Breakdown Structures. Six participants use Resource Allocation.

Two thirds of the participants use formal methods for concept generation, with all but one of those participants using brainstorming. One participant uses patent searches and paper searches.

Only a third of the participants use any formal means of concept evaluation. Three use concept comparison matrices (Pugh's method), while one uses other formal methods (Kepner Tregoe Potential Problem Analysis, Decision Analysis, System Performance Analysis, Hazard and Operability studies, etc.).

### **2.3.5 Needs Identified**

Sketches are used often for communication during design and it should therefore be possible for users to input sketches for the sharing of ideas.

Various file formats are used frequently in communications, including various CAD file formats and it is therefore important that all file formats are supported. An open workspace where users can click on files, which will be opened in their native programs, would therefore be attractive.

Various sources of information are also referenced during projects and it should be possible for users to attach this information to projects in their desired format and within the correct context.

The use of in-house design procedure documents indicates that there is a need for software that guides users through the process followed by the company.

## **2.4 Interviews**

Three brief interviews were conducted with engineers in industry that are involved in different market segments and follow differing design approaches. Their respective design processes were analysed and the different requirements for a design support

tool that these individuals have, were identified. The following subsections were named according to the dominant character of the context within which each of the persons interviewed, operates.

### **2.4.1 Systems Engineering Company**

The person interviewed is the chief mechanical engineer at a company with clients from various sectors in industry including aerospace, military, commercial, industrial etc. The person is familiar with systems engineering and makes use of certain design methodologies described in Section 2.1.3.

#### *2.4.1.1 Product Development Process*

A systems engineering approach is followed at the company and there is a strong correlation between this approach and that described by Blanchard and Fabrycky (2006). Each project commences with client consultation during which the client's needs are obtained. Thereafter a quotation is compiled, accompanied by a time schedule for the project. The client then requests changes and milestones are agreed upon with the client after which a contract is drawn up and an order is issued.

A requirement analysis is then performed according to the user needs, followed by a functional analysis (see Section 2.1.3.2) during which a flow diagram is created. The system specifications are then obtained from the functional and requirement analysis.

A project team is gathered and the team leaders break the specifications down into a sub-system specification and draw up project plans. Reviews are done occasionally throughout the project to assess whether the specifications accurately address the client's needs.

After the specification development, concept design commences and the system architecture is designed. From the functional analysis, it is determined which blocks can physically be grouped together. The documentation that has been created so far contains specifications, project plans and also testing instructions in order to determine from test results if client needs, i.e. the system and sub-system specifications, are met. The documents are stored electronically and a program is used which links the related paragraphs of different documents to allow for the extraction of information. When Design Review is done, a print-out of what should have been considered, what has been tested and the results is made to ensure that nothing has been skipped when the system is finally integrated. After the final qualification tests, a document is created.

Once the physical architecture is documented, technical engineering work relevant to the project such as CAD, structural analysis, FEM and electronic design is performed and a design review is done. Manufacturing drawings, quotes and components are received and the various elements of the system are integrated. Off-the-shelf items are specified, e.g. the power consumption (volts, amps) of a fan. Before assembly, parts are inspected by the quality insurance department. All manufacturing documents and drawings have numbers for change control since changes without

the knowledge of team members cause major problems. Parts are then assembled after which sub-systems are tested and integrated and tests are documented in a report. Thereafter there is an important meeting with the client where progress is reported and it is determined if the client's needs are met. This leads to more technical and payment milestones. Once-off or different development models are used, such as a concept phase model, pre-qualification models, production readiness models or re-qualification models if the production process changes. The process may, however, vary from project to project

For consumer systems there is a parallel leg for logistics. Manuals and repair manuals are created and an analysis is performed on what spare parts are to be kept for the client and what the scheduled maintenance is. Depending on the lifecycle, an agreement on de-commissioning is to be made.

For quality control it is determined whether the company's prescribed process was followed, whether the correct documents were generated, how valid the contract is, and what the risks were. Parts are inspected, assembly and inspection instructions are provided and every step of the process is documented. Systems also have to meet external specifications such as ISO 9001.

#### *2.4.1.2 Design Support Tool Needs Identified*

The company performs brainstorming during concept development. Multiple concepts are developed and evaluated according to weighted criteria. Concepts can be combined to obtain the best concept according to the weighted criteria. Various people are involved such as quality control, the project manager and the technical team. Everything is documented including what has been decided. Risk analysis is performed and a report is generated which is also used for concept evaluation. A design support tool should therefore provide for concept comparison matrices with weighting, and the facility to capture further discussions and considerations.

The importance of communicating changes to all parties involved was stressed and it is therefore required that the user interface displays any changes to users in a transparent manner.

All work is documented including all decisions that have been reached and it is therefore necessary that users are able to record all aspects of their design work.

Multiple concepts are developed at each level of the system structure and it should be possible for teams to develop alternatives for the entire system, the various sub-systems and components or even for manufacturing processes.

The specifications are broken down to the sub-system level and even for off the shelf items. It should therefore be possible to develop specifications at all levels of the system and trace these back to the systems level as described in Section 2.1.3.1.

## 2.4.2 Product Development Company

The person interviewed is the owner of a company that is involved in all phases of product development, from research and development to manufacturing and commissioning. The company sells its services for the entire development process or might only be involved during certain stages of the process or with specific design tasks. The company possesses various resources including mechanical engineers, electronic engineers, industrial designers, a prototype shop, model makers and technicians for testing, measuring, building, assembly and machining.

### 2.4.2.1 Product Development Process

The process that is followed varies according to the type of product and the project can be entered and exited at specified stages. Sometimes the concept phase and feasibility study falls away as is the case with corporate clients that have a good understanding of the market.

The specific methods employed are determined by the skills and resources available, as well as the cost involved. With larger products, small design iterations of design and analysis are performed and compared to competing products. Some clients accept the risks involved and want to build moulds and manufacture immediately to obtain the end result as quickly as possible.

Corporate and small businesses have a good idea about the market; however some clients do not understand the market for their product. The problem is finding a balance between performing research, such as a feasibility study and market analysis, versus believing what the client says with the risk of having to start over at the end of the project.

This person prefers to start with market research and work towards a focussed goal with the shortest process possible as opposed to following a set of processes where the goal is discovered during the development process.

Often, short briefings are received from the client and specifications are added during the project.

In product development, innovation is the goal and the user requirements are written with the market in mind with changes being made constantly throughout the project. User requirements often include "must have" aspects, "nice to have" aspects or "what can be achieved" aspects. A strict set of user requirements will lead to products that comply with specifications, but do not exceed any expectations.

The project is broken up into small steps and design reviews are performed to confirm that decisions are still valid, that the project is still viable during the design review and that no new technology has been employed by the competition which will change the user requirements. When the aim is to develop as quickly as possible, new technology does not affect the design process. In product development,

decisions can be made towards the end of the project and certain decisions are postponed to ensure they are relevant towards the end of the project.

Decisions are recorded in an informal manner via tools such as Microsoft One-Note. Information such as web research is forwarded to the client or other team members. Sharing information amongst more people builds momentum.

In product development clients are consulted informally at the start of a project, whereafter the client specifies any desired changes or gives consent to proceed with the project. The client remains involved throughout the process.

#### *2.4.2.2 Design Support Tool Needs Identified*

As research is frequently performed on the market for the product, the technologies to be employed or feasibility of a specific concept, information gathering is an important part of the design process and the ability to capture such information in the design support tool is important.

User requirements are dynamic and the system should allow users to edit these requirements throughout the project.

The option to change decisions should remain open during the course of the project with reviews being performed all the way through the project. The ability to retain concepts throughout the project and review the decision as to which alternative is to be selected should be included.

The company already makes use of Microsoft One-Note for information gathering and prefers a less structured interface where information can also be attached in any format and at the desired location.

### **2.4.3 Consulting Engineer**

The person interviewed is a consulting mechanical engineer who works alone and is involved in various projects for clients. No formal methods such as brainstorming or decision matrices are used for the documentation of concept generation and evaluation.

#### *2.4.3.1 Product Development Process*

Each project is divided into stages with deliverables at the end of each stage for which payment is received. Firstly a meeting is held with the client where the client provides specifications and/or user requirements (that can be very abstract) after which a proposal is made to the client explaining what the aim of the project is and the scope of the work that is to be done along with the different phases and milestones, and the budget of the project. Traditional design methodologies are not followed strictly and projects can vary greatly depending on the goal of each project. He is not often involved in hardware development and only occasionally has to provide documentation.

Iterations are done throughout the project and there is continuous conversation with the client to ensure that requirements are met or to determine whether a proposed solution will comply with specifications as compromises might have to be made, for example, when a strict budget is to be adhered to. The client gives feedback on proposed changes and may specify additional requirements during the project, such as cost or working environment.

After detail design, a data pack is provided to the client and a review is done to ensure the client is satisfied with the documentation. The relevant electronic files are provided along with hard copies of the documentation.

Multiple prototypes are manufactured and checked before being used in testing, which is done in the field rather than in a laboratory.

Industry projects have to be completed quickly as clients are only interested in a working solution that conforms to specifications and not necessarily the optimal solution. The person interviewed states that he is therefore sceptical about the demand for formal methods for specification development in industry.

#### *2.4.3.2 Design Support Tool Needs Identified*

The reasoning behind design decisions are to be documented for future reference and it should be possible to attach notes to design data to provide the context of the decisions, thus allowing a person reviewing these decisions to understand the thought process without having to read a time consuming report.

The requirements are dynamic and some may clash with others. Provision should therefore be made for different versions of the user requirements and specifications list.

Users should be allowed to choose the appropriate process to follow from a list of formal methods from literature. The need for the inclusion of the QFD method and decision matrix method (Pugh's method) was, however, specifically mentioned.

The need to teach users in industry about the available methods was expressed. This can be achieved by providing a software framework such as DiDeas II that allows users to follow a specific methodology.

## **2.5 Development of Specifications**

Developing DiDeas II to meet all the needs derived from the previous sections in this chapter, is not feasible within the scope of this thesis. A selected set of features therefore had to be identified. In selecting which features to include, preference was given to features that could be evaluated during the cases studies envisaged as part of this thesis' research. However, for the sake of completeness, some other features that would have a high priority in industry, are also pointed out.

### 2.5.1 Important Features Not Included for Case Studies

- *Block diagram editor.* During the Needs Analysis for DiDeas II, the ability to draw up functional analysis diagrams was identified; however widely used tools such as Microsoft Visio exist for creating block diagrams. The decision was therefore taken not to include this functionality into DiDeas II, since the user is able to attach such diagrams to a project as a file.
- *Role-based data access.* As pointed out above, different data access schemes exist that provide different levels of access to various users. This is an important aspect of collaboration tools. However, access rights for different users were not implemented in DiDeas II since it was not necessary for the case studies.
- *Re-use of design information.* Users should be able to re-use design information. The case studies will however not allow users the opportunity to re-use information and this is not a priority.
- *Scheduling and meeting management.* Users should be able create and organize teams. The ability to schedule meetings is a functionality that allows for this; however for the case studies, this is not a priority.

### 2.5.2 Requirements for Case Studies

- *File attachment.* It was evident that designers attach various forms of media to projects, and in various file formats. Users should therefore be able to attach and open any file format in DiDeas II. This should also allow users to easily attach concept sketches, which are often used for design communication.
- *Customisable.* The system is to be customisable to such an extent that users can implement their own design styles, which include the language and terminology they use and the methodologies or elements thereof that they follow.
- *Documentation of communication.* The system should allow for communication among users which can then be documented along with the project data. This includes communication while working synchronously, such as instant text messaging or chat, and discussion tabs for asynchronous communication.
- *The ability to work on multiple projects within a given day.* However, it is not necessary that users are able to have multiple projects open simultaneously on the same PC. Users should be able to choose the project they wish to open from a list.

- *Shared interface.* There should be no difference in the way information is displayed to different users (WYSIWIS). Information is to be updated to the user interfaces of all users within a reasonable time.
- *Specification development.* The user should be able to input user requirements into the system and translate these into system specifications. Users should be able to change user requirements which are dynamic. It should also be possible to create a project template that includes the various elements of the QFD method.
- *Development of multiple alternatives.* Users should be able to develop multiple concepts for a given system, sub-system, component or any other element. The user should be able to document all alternatives that were developed for future reference. The decisions are to remain open and the users should be able to change their decisions at any point during the project.
- *Concept evaluation.* It should be possible for users to perform concept evaluation through the use of decision matrices (Pugh's method). Users should also be able to add weights to the criteria against which alternatives are compared.
- *Conflict resolution.* A data access scheme is to be implemented to prevent data conflicts where users overwrite each other's work. This access scheme should also provide a balance between preventing conflicts and providing users access to data.
- *Transparency of project information.* Users are to be able to view project progress on demand. This is to be displayed in a transparent manner and users should be able to navigate to desired information quickly.
- *Non-restricting.* The user interface should be "open" in the sense that it does not force the user to input information in a specific place or specified format and allows users to navigate to desired information.
- *Context of information.* Information that is recorded is to be put into context and related to specific elements to allow others to quickly understand the reasoning behind design decisions.

## 3 DiDeas II Development

This section describes the development of the design support tool which aims to improve the existing situation in design.

### 3.1 Previous Developments

There have been two previous developments of DiDeas namely DiDeas I which was implemented as a web-browser-based system, and DiDeas II which was implemented as a desktop application. The latter is further developed as part of this research.

#### 3.1.1 DiDeas I

Schueller (2002) identified three elements for a design support system: Design Methodology, Communication and Information Transfer, and Input Devices for Conceptual Design. A framework was then developed to integrate these three elements into one support system. The 'Distributed Design Assistant', abbreviated 'DiDeas', was developed as a web-browser-based system that allows simultaneous multi-user collaboration with a relational database (implemented in Microsoft Access) which is located on a central web-server and which stores all design information entered into the system. The user interface was realized in the form of a collection of Microsoft Active Server Pages, which could be accessed platform-independently via a standard web browser.

Case studies showed that DiDeas I provided a suitable context for information exchange in synchronous and asynchronous collaboration scenarios. However, the use of HTML for the user interface severely restricted its flexibility and it was prohibitively complex to adapt the relational database to a large variety of design styles (Liu & Basson, 2007).

#### 3.1.2 DiDeas II

As a result of the issues experienced with DiDeas I, DiDeas II was implemented as a system consisting of two separate programs in a client-server architecture, with each designer running his/her own client side program. Several programming languages were considered for the development of DiDeas II, and Visual C++ with .NET was chosen since it imposed the fewest limitations (Liu & Basson, 2007).

The overall layout for the user interface of the client side of DiDeas II as implemented by Liu (2007) was retained for this thesis' research (see Figure 1 in Appendix F).

The layout uses an expandable tree-view of the system/subsystem/component structure on the left-hand side of the screen as a means of navigating through the data. The tree was implemented using an in-house graphics library named GTek.

Some functionality was however added to the tree-view in this thesis' research, such as the "greying-out" of discontinued branches of development and the display of meta-data in the tree nodes.

When a node in the tree on the left-hand side of the screen is selected, information about that node is displayed in a panel on the right-hand side. Information in the right-hand panel was structured in a similar fashion as the previous development of DiDeas II with information organised into two levels of tabbed pages that can be customised by the user creating the project template (see Section 3.3.4.3). However, the type of information that can be displayed and linkages between information sets were significantly expanded in this research. A customised set of tabbed pages for the right-hand panel, i.e. the specific selection of tabs and tables on each tab page, is called a "right panel style". Each element type in the tree view is associated with a right panel style. Different tree element types may use the same right panel style, but each tree element type can only be associated with one right panel style, since the associated right panel style determines what DiDeas II displays in the right-hand panel when the particular tree node is selected by the user.

The ontology and conceptual graphs approach for the data structure developed by Liu (2007) was retained to allow for a data structure that can be customised during run-time by the user without altering any source code. After Liu completed his research, Basson continued work on this data structure and a user interface for editing it. These interfaces were incorporated into the project template wizard along with new features (see Section 3.3.4.3). Basson also added the facility to link attributes to elements because such attributes can be used to enrich the information displayed in the tree view and to facilitate other unrelated research using DiDeas II.

## 3.2 New Developments

DiDeas II was further developed during this thesis, with new features being added to the existing framework, as well as additions made to the framework itself. The new features that were introduced provided more communication functionality within DiDeas II, as well as additional customisation functionality and a different structuring of the information displays. The following sections briefly describe the features that were added to DiDeas II as part of this thesis' research.

- Custom Tables and Inheritance Allowing Specification Development and Tracking  
The aim was to extend the customizability of the data structure to allow users to customize tables according to their needs from a generic set of elements, as opposed to using pre-configured tables that impose a specific methodology on users. Whereas the previous development of DiDeas II included pre-configured tables for specific design methodologies, such as the QFD House of Quality table that was displayed in a separate window, it is now possible to create such tables with the required functionality from the project template editor (see

Section 3.3.4.3). The column and row header names for each type of table can be customized according to the terminology employed in a specific company. In the data structure implemented, data can also be inherited between adjacent levels of the tree hierarchy, allowing for the creation of the various tables as encountered in literature. This includes the QFD method which is implemented through multiple tables, creating a “baseline” as with the systems engineering approach of Blanchard and Fabrycky (2006), where specifications on each level of the system can be traced back to the user needs at the top level. User needs therefore can be developed into system specifications, which in turn can be translated to sub-system specifications, component specifications, etc.

Concept evaluation tables also use the inheritance capabilities. In the tree view, a sub-system's concepts are listed as children of the sub-system node. Properties associated with each concept node can be accessed by the sub-system node to populate a concept comparison table, thus facilitating concept evaluation and selection.

The inheritance among tables is determined during the project template setup phase, in accordance with the selected design methodology for the project.

- **Data Locking**  
A data access scheme (see Section 2.1.6) was implemented to allow for synchronous collaboration and to prevent multiple users from altering the same data and over-writing each other's changes when uploading their work.
- **Project Template Wizard**  
A wizard to assist users when creating project templates (ontologies) is accessible from any DiDeas II client. The wizard takes the user through a series of steps that are organised using tab pages.
- **Concepts/Alternatives**  
The possibility of having multiple alternative concepts for a given node, as children of that specific node in the system structure, has been introduced. This allows various ideas to be developed and kept as a record of the design work.
- **Real-time Chat (Synchronous Communication)**  
The introduction of real-time text-based communication within DiDeas II allows designers working synchronously to coordinate their efforts and keeps a record of this communication.

Since a major focus of DiDeas II is capturing the design decisions of designers, a means to capture the communication between designers was identified as a requirement. An instant messaging tool was therefore included in DiDeas II to allow communication to be recorded in a specific project. Designers can select the usernames of the relevant recipients for the particular message from a list. In the Messenger window, a user will see a list of all the messages for which he

was a recipient. Each message is displayed along with the name of the sender and the exact time of the message. Messages are stored in a queue on the server and downloaded to the relevant team members.

- **Context-based Discussion (Asynchronous Communication)**  
To allow team members to understand the context of other designers' work, a discussion tab is added to the right panel of each node. The discussion tab allows users to attach information they deem important to each node. Each posting can be viewed by all team members and also includes the username of the person that posted the comment, as well as the exact time at which it was sent. This allows team members to find the relevant information quickly as the information is stored in a structured manner.

- **Attach Files**  
Users can attach files that are relevant to a specific node. The files are then uploaded to the server. When a user requests to open such a file, it is downloaded from the server. After changes have been made, this updated version of the file is then automatically uploaded to the server. This ensures that the latest version of the file is always accessed.

The files related to a node are displayed in a list along with a short description of each file and the name of the person that posted the file.

- **Node Meta-data Display**  
As described in the previous section, attributes can be associated with specific elements. In DiDeas II, these attributes hold values which can be entered in the right-hand panel associated with the specific node and are used to display information about the specific node (element) in the tree structure. The goal is to promote information transparency by allowing users to scan the tree structure and thereby quickly obtain information about specific nodes and their status, which includes the overall project status at the top-level node.

- **Ad-hoc Team Formation**  
Project templates and projects can be created in a decentralised manner from any DiDeas II client application and the creator of a project can select the team members involved in the project. The project is then uploaded to the server and will become visible on all the client modules of the selected team members from where the project can be accessed.

### **3.3 Software Implementation**

As with the previous version of DiDeas II, C++/CLI (employing .NET) was used for implementation and Microsoft Visual Studio was used as the development environment.

### 3.3.1 System Architecture

The network communication had to be re-designed and implemented during this thesis' research since the previous version could not transfer large files. This presented the opportunity to reconsider the most appropriate network architecture for DiDeas II. Two network architectures, namely the client-server (where communication among client nodes are centralised through a server) and peer-to-peer (which is de-centralised and with communication taking place directly between identical nodes) are discussed in Section 2.1.5 along with a hybrid peer-to-peer architecture where certain components are centralised. Section 2.1.5 compares the two architectures in terms of reliability, security, scalability and maintainability. As seen from the comparison, the main advantage of peer-to-peer is that a distributed application is more scalable; however DiDeas II is aimed at smaller enterprises where scalability is not expected to be an issue. A data access scheme (see Section 2.1.6) also had to be implemented to control the access to data and files among multiple users and the assumption was made that it would be easier to implement such a scheme in a client-server architecture, where data is located in a central location. This does however provide a system with a single point of failure.

As can be seen in section 2.1.5, both architectures have merit; however from the discussion it can be assumed that client-server architecture is superior in terms of ease of development and was selected for the network architecture of DiDeas II.

Both the server and client applications are divided into three conceptual layers, i.e. the user interface (UI), background layer and the data layer as illustrated in Figure 4. Communication between the client and the server application, and the access of data is handled on background threads on both applications (shown in Figure 4). This separates the user interface from processes such as searching the data structure or network communications, with the aim of providing a more responsive user interface.

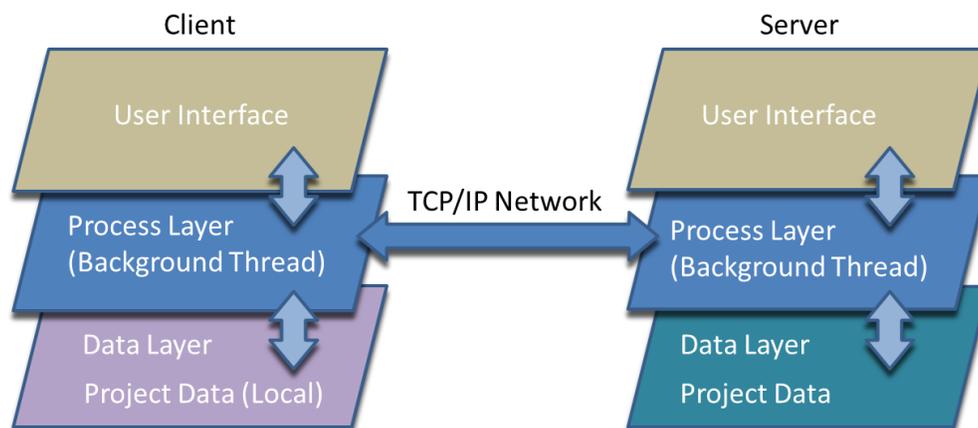


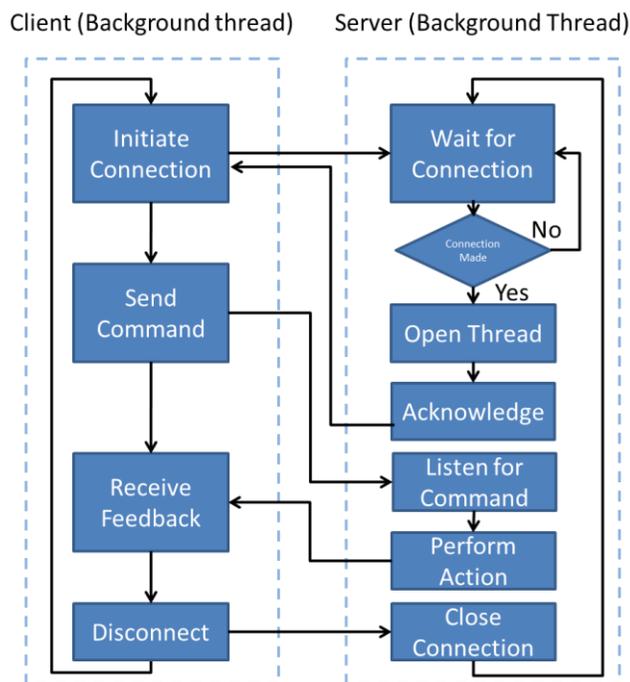
Figure 4: DiDeas II Architecture

### 3.3.2 Communication

DiDeas II uses the TCP/IP protocol for network communication. This was implemented through socket programming via the networking classes in the Microsoft .Net libraries.

#### 3.3.2.1 Communication Sequence

During the early stages of development, network communication was implemented where the connection between the client and server remained open continuously. Problems were however experienced with this configuration since the connection between the server and client was often lost, even when idle. The decision was therefore taken to adopt the process illustrated in Figure 5. In this process, the server continuously listens for connections from clients.



**Figure 5: Client-Server Communication Sequence**

Each client periodically initiates a connection with the server, in the client's background communication thread. The client then determines whether there were any changes to the tree structure or other data, such as the tables or discussion tabs and attached files, and whether there are any new messages available. The thread might also receive commands from other processes, such as an action initiated by the user.

The server continuously listens for connections from clients. Once a successful connection between a client and the server is established, the socket is passed to a background thread that is dedicated to that socket, thus enabling the server to serve a number of clients with minimal delays. On the background thread, the server interprets the client's commands and takes the appropriate action. Once all actions have been completed, the socket is closed and the thread is terminated.

### 3.3.2.2 Communication Events

Since a client initiates communication with the server, a number of communication events can be distinguished. This section describes the communication that occurs with each event.

Project templates and projects can be created by any user from the client-side. After creation, the project and/or template is uploaded to the server, before being downloaded to the PCs of the team members that were selected for participation in the specific project, by the project creator, as soon as they log onto the server.

Every time a user starts a session in a specific project, the latest project data is downloaded from the server, ensuring that the most up-to-date information is available. Each client works on a local copy of the data, as illustrated in Figure 6. When a user leaves the session of a specific project, all background actions are first completed and the project data on the server is updated to ensure that the latest project data is saved.

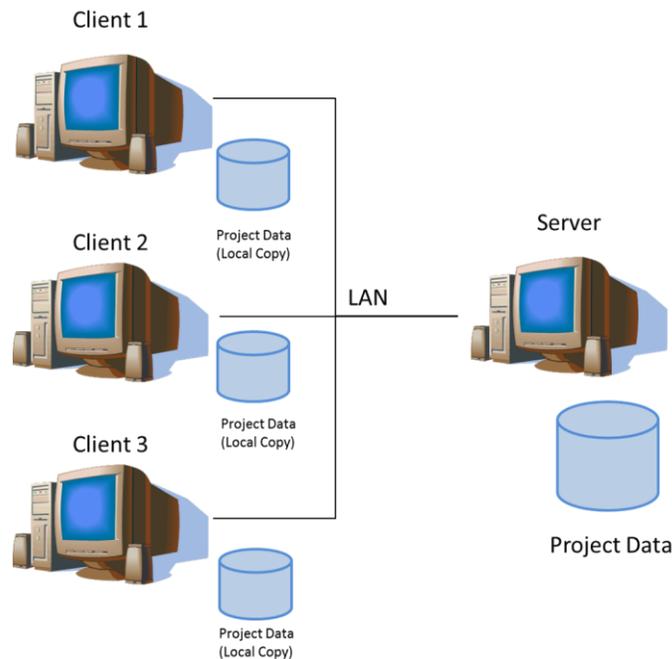


Figure 6: DiDeas II Data Distribution Diagram

When a user attaches files to a project, these files are uploaded to the server where they can be accessed by all clients. Each time a user wishes to edit a file, it is downloaded from the server. After a file has been edited, when the file is closed, the local copy is sent to the server to overwrite the master copy on the server. Once a file is ready for sending, the receiving end is notified and waits for the file to be sent.

At the receiving end, data packets are loaded into a buffer before being reassembled. As problems arose when sharing large amounts of data across the network, files are broken into small data packets before sending files across the network, and reassembled at the receiving end. Each packet is assigned a sequence number, which is used to reassemble the packets in the correct order. After all packets have been received, the receiver acknowledges this to the sender.

As mentioned in Section 3.2, users can send instant text messages to one-another, providing synchronous communication in DiDeas II via the DiDeas Messenger window. Messages are stored in a queue on the server along with a list of all recipients. Each client regularly polls the server for new messages. When a connection is made to the server, the message queue is scanned and the first relevant message in the queue is sent to the client. Once all recipients have received a message, it is removed from the queue. A message history for each project is also stored along with the project data.

The tree structure display is implemented as a WYSIWIS interface (see Section 2.1.2) and changes are to be visible to other clients with as little delay as possible. When a client performs an action that will alter the tree structure, the server is polled to determine if the action can be performed. For instance, a node cannot be deleted if it has child nodes, in which case a message is sent back and the client is notified. If the change is permitted, it is added to a queue which is checked periodically by all clients. After the specific change has been received by all active clients, it is removed from the list.

Changes to the information related to specific nodes are uploaded to the server once the user chooses to save them. This includes the data in tables, the discussion tab and the list of related files along with the files themselves. Only incremental changes are sent across the network in order to minimise network traffic. These changes are also stored in queues in the same fashion as messages or changes to the tree structure and are sent to other clients as soon as these clients poll the server for changes.

### **3.3.3 Data Structures**

#### *3.3.3.1 Data Storage*

As shown in Figure 6, project data is stored on the server and a local working copy (the entire data file) is downloaded to each client at the start of a session every time the client opens the specific project. Data is stored in a self-implemented relational database built using text files to simplify development and setting up the test cases.

If the implementation needs to be converted to using database software (such as MySQL), only one low level class will be affected.

As mentioned in Section 2.1.4.1, the approach used to enable customization for DiDeas II is the use of ontologies and conceptual graphs. Figure 7 shows the overall data structures for each project. The Project Template (indicated in green in the figure) stores the ontology (see Section 2.1.4) which is used to describe the different element types, which include the types of nodes in the tree structure, the different row and column header types in the tables and the data table types.

The project specific data in the DiDeas II data structure is contained at two levels namely the “elements and relations” level and the “table data” level. These levels are indicated in blue and orange respectively in Figure 7. The first level is the elements and relations level. These elements hold the information of the nodes in the left-hand view tree-structure. Each element has a specified type and can be identified by a Unique ID (UID). Each element can also have various attributes that are used to hold specific values about the element that are displayed in the tree node.

The relations between these elements are stored in relation classes. Each relation class stores the UIDs of the parent and child nodes. Each relation can also have a specific type. Although the relations in the tree-structure are always of a parent-child nature, the relation type can be used for data inheritance among nodes, such as the concept-of relation type used for inheritance in the case of concepts where the parent node inherits information from the concept nodes.

The second level is the table data level. This level consists of data elements and data relations that are used to build up the information in the tables. To ensure customizability, instead of using pre-configured tables, users can create tables according to their requirements by selecting the element types for row and column headers and selecting the inheritance among them.

The data in the tables are stored as data elements with each cell in the table being associated with a specific data element. Row headers and column headers are also associated with a specific element and the cells are located via a relation to both a column header and a row header.

Row and column headers in a table can be inherited from other tables. In such instances, the cells are located by relations to row and column headers in the table from which the data is inherited.

The expansion of the conceptual graph and ontology approach of Liu (2007) to include this second or data level with the inheritance of data among tables has been introduced in this research with the aim of allowing users to customise the user interface for different project types to display information according to the methodology followed.

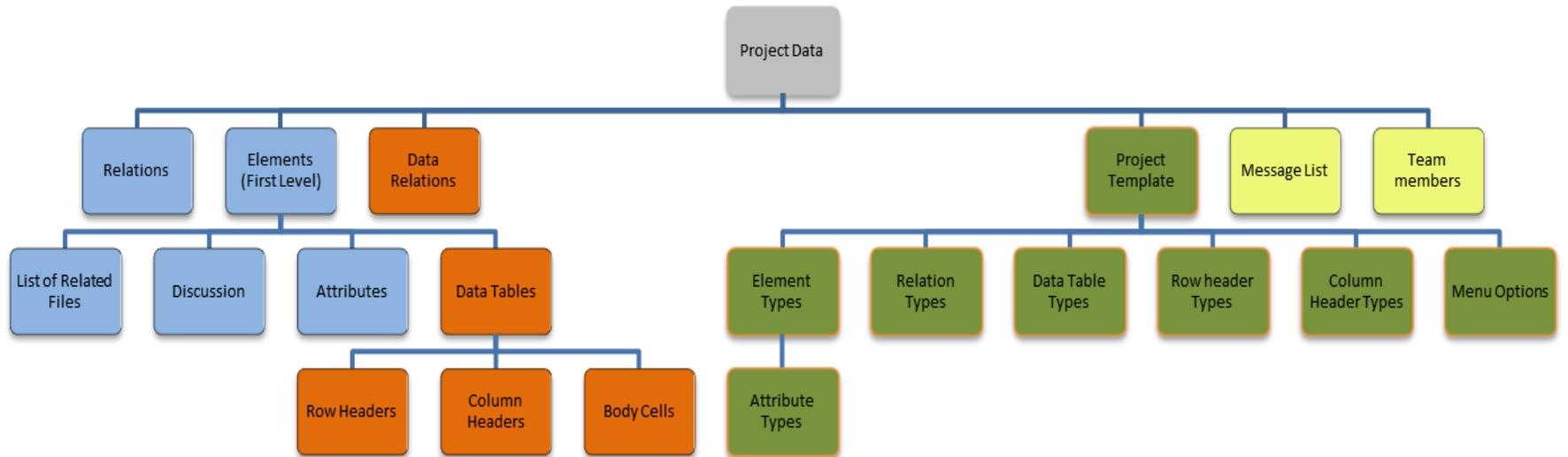


Figure 7: Data Structure

### 3.3.3.2 *Data Inheritance*

Data can be inherited from one table to another. The ontology editor provides the user with various forms of inheritance among data tables. The person setting up the project template can then select the type of inheritance for each table according to the requirements for the specific project type.

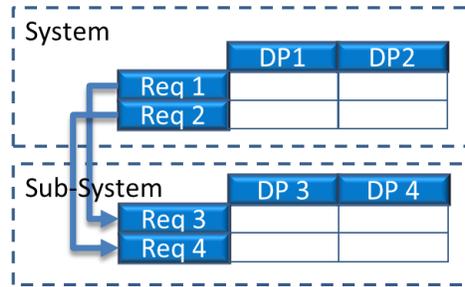
The following forms of data inheritance among tables are possible:

Five types of inheritance from parent nodes to child nodes (one level below the parent) are provided. This type of inheritance will be useful when relating requirements for a sub-system with reference to the next higher level sub-system's requirements. Figures 8, 9 and 10 show how row headers and/or column headers can be inherited from parent to child, remaining respectively row or column headers in the child's tables. Figures 11 and 12 show that row headers and column headers in the parent node's tables can be inherited, but are converted respectively to column headers and row headers in the child node.

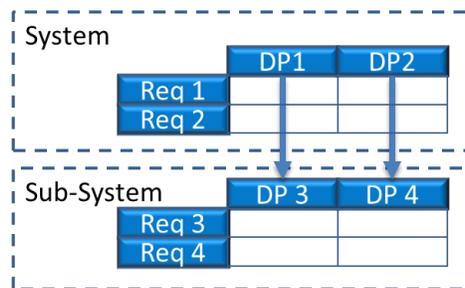
Provision is also made for inheritance between different tables of the same node (local inheritance). Figures 13 to 16 show row and column headers can be inherited to either other row and column headers or to column and row headers, respectively.

To make provision for evaluating concepts, the notion of "alternatives" was introduced in the development of DiDeas II presented here. A given sub-system node in the hierarchy in general will have a number of child nodes that are all possible ways of implementing that sub-system. Since only one of the concepts can eventually be used, the nodes representing concepts are called "alternatives". To facilitate the comparison of alternatives, row and column headers from children nodes and their parent node, can be "inherited". In this case, also the data in the relevant column of the parent and child node can be shared. Data in the body cells can be edited at both the parent node and the child node (see Figure 17).

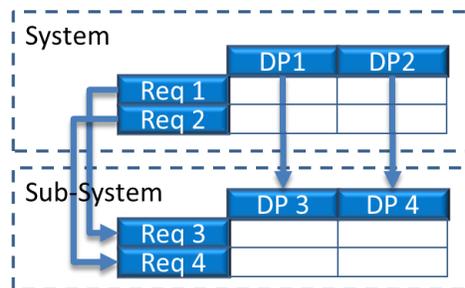
Three further types of inheritance are provided: firstly, no inheritance need be used (see Figure 18), where neither the data in the table nor the column or row headers are shared with other tables. The other two types allow the creation of tables that have identical row and column headers, and these headers can either be inherited from another table's column (see Figure 19) or row (see Figure 20) headers, in the same node. The last two types of inheritance were introduced to provide for inter-relationships of parameters in QFD.



**Figure 8: Row Header Inheritance**



**Figure 9: Column Header Inheritance**



**Figure 10: Column and Row Header Inheritance**

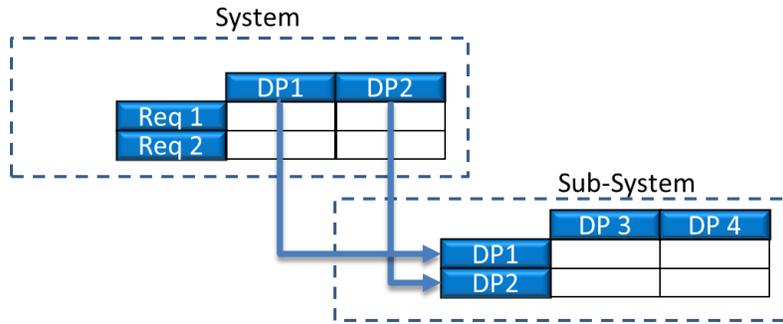


Figure 11: Column Header to Row Header Inheritance

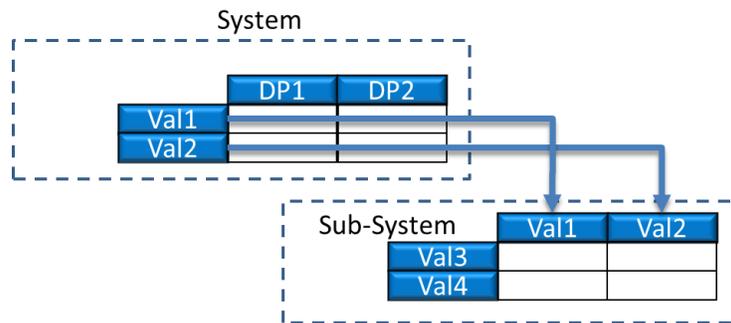


Figure 12: Row Header to Column Header Inheritance

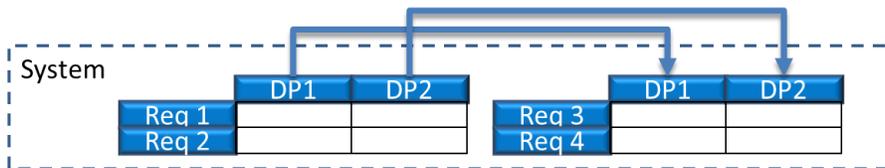


Figure 13: Local Column Header Inheritance

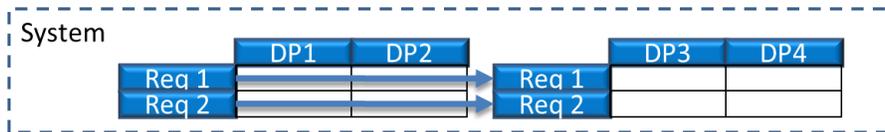


Figure 14: Local Row Header Inheritance

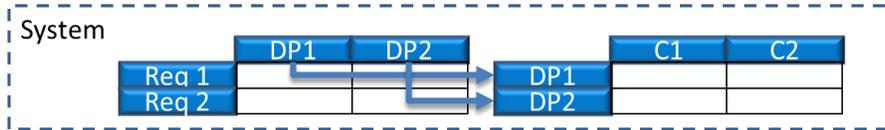


Figure 15: Local Column to Row Header Inheritance

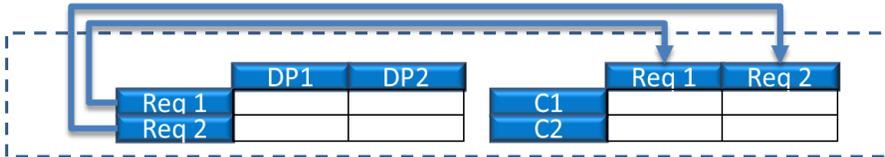


Figure 16: Local Row Header to Column Header Inheritance

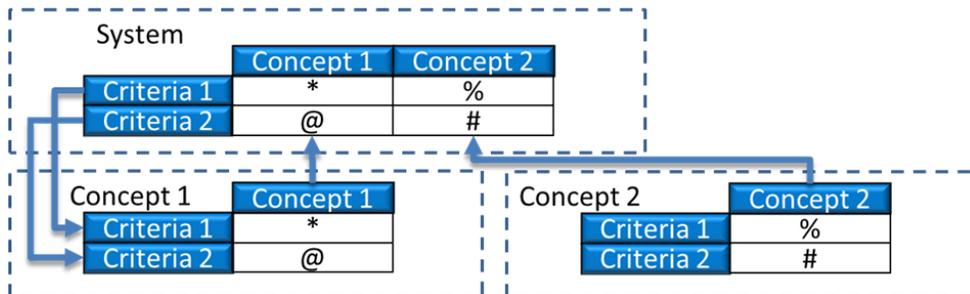


Figure 17: Alternatives Comparison

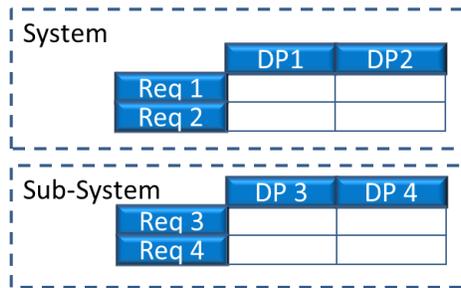


Figure 18: Original Data (No Inheritance)

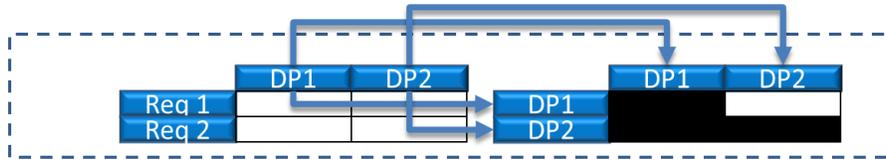


Figure 19: Local Compare Column Headers to Self

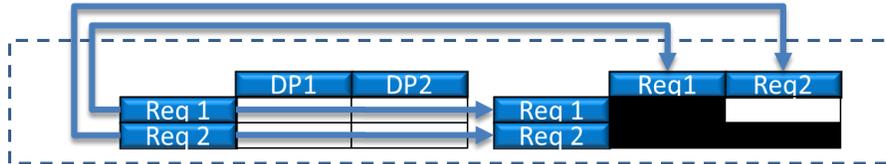


Figure 20: Local Compare Row Headers to Self

To allow for flexibility, the twelve types of inheritance among tables described above are possible. The only limitation is that data can be inherited among nodes only if a parent and a child node has the same right panel style (defined in Section 3.1.2).

### 3.3.3.3 Data Inheritance Application Examples

To implement, in a more flexible way, tables that were pre-configured in previous developments of DiDeas, such as the house of quality table in the QFD method, the types of tables described above were introduced. Here follows two examples, namely the house of quality and Pugh's method (decision matrix for concept evaluation).

The House of Quality (HOQ) formed part of the conceptual design methodology of various authors. To allow the various parts of the HOQ table to be implemented, various forms of inheritance among tables were used.

In the body section of the HOQ table (see Figure 2), the user requirements are tabulated against a set of design parameters. In the “roof” of the table the degree of inter-correlation among the design parameters is shown. At the bottom of the table, the target values for each design parameter are entered.

To implement these aspects in DiDeas II, three tables are used. The first table tabulates the user requirements versus the design parameters. This table uses the type of inheritance as shown in Figure 11, with the user requirements inherited from the design parameters of the parent node and the design parameters defined in each local node.

Design parameters are also tabulated against themselves by using the inheritance type shown in Figure 19, which represents the “roof” section of the house of quality. The table type as shown in Figure 13 is used for the target values section at the bottom of the HOQ.

For Pugh's method of concept evaluation, the case where the information stored in a table's body cells is inherited from a child's table (see Figure 17) is used. A set of criteria is added in the node for which alternatives are to be compared. The values for each concept can then be entered and edited either at the concept node or at the parent node and the values for all the different concepts are compared in the parent node.

#### 3.3.3.4 *Data Locking*

Data access schemes and file access control are an important factor in CSCW and was investigated in Section 2.1.6, where it is mentioned that a balance has to be found between providing users with access to data and preventing conflicts such as users accessing files simultaneously and overwriting each other's changes in the master copy. Since it was not needed for the case studies, user roles were not taken into account and all users have equal access rights in DiDeas II.

Conflicts can clearly arise when users work synchronously and wish to edit the same information. To avoid these conflicts, data-locking was added to DiDeas II. Data-locking occurs at the node level, which means that only one user at a time can edit the information attached to a specific node including associated files, table data and discussions. Users can, however, view the information associated with the specific node while in "read-only" mode, except that associated files cannot be accessed. Further, nodes that are being edited or have child nodes cannot be deleted to avoid conflicts.

To implement this approach, each node has two modes, namely a read-only mode and an edit mode. When a user wishes to edit a specific node, he must first click on the "edit" button in the user interface. A message is then sent to the server where a list of all the nodes currently in edit-mode is kept. If the node is not being edited by another user, it is added to the list on the server and a message that the node is available for editing is sent back to the client. If the node is being edited by another user, a message is sent to the client and the user is informed of this status. Once a user releases a node that he has been editing, it is removed from the list of nodes in "edit-mode" and will become available for editing by other users.

The provision for inheritance of data among nodes required a decision whether nodes that inherit table data from a specific node would also be locked when the node is being edited. Owing to the inheritance among tables, information can be lost when the row or column is deleted within a table from which row or column headers are inherited by other tables. Another possible conflict is when the same information is edited in the parent node and concept nodes by multiple users at the same time. One possible solution would be to lock all the nodes that inherit data; however this will place severe restrictions on access to data.

Due to the tight time-scales envisaged for some of the case studies, having multiple locked nodes would however severely hinder designers wishing to edit these nodes.

The decision was therefore taken that parent and child nodes of the node to be edited are not locked.

### **3.3.4 User Interface**

#### *3.3.4.1 Overview*

The user interface has a strong influence on perception of the users of software, and can even be the determining factor for users' willingness to use the system at all. The system should therefore provide designers with an intuitive and transparent user interface that allows users to quickly assess project progress, navigate to the desired information and insert data or information in their desired format.

DiDeas II is aimed at smaller enterprises and from the survey (Section 2.3) and interviews (Section 2.4) it could be seen that designers working at such companies often do not follow formal methods during the design process. It was also evident during the interviews that participants are concerned about being forced to follow specific methodologies or processes that will increase their workload. DiDeas II therefore aims to give users freedom to enter information in any format and does not dictate the sequence of events during the design process. Users are ultimately responsible for the structure and content of information captured in DiDeas II.

The user interface has been implemented via the Windows Forms libraries in .NET and since distinct applications are used on the server and client sides, their user interfaces will be considered separately.

The user interface of the server-side application requires limited functionality and is primarily used to monitor the connection status of clients, as well as to manage a list of usernames. The main window on the server-side (Figure 21) contains a table displaying the usernames registered on the server along with their connection status (online or offline).

When the user clicks on the "Edit Users" button, a window appears (Figure 22) from which usernames can be added or removed from the server. The intent is to allow for the capture of user information (such as e-mail address and phone number) in this list. This information can then be shared among team members during projects, however only user names were used for the application of the case studies.

The client application has four main user interfaces:

- A start screen or main "dashboard" window from where users can start all processes.
- The project template creation window where ontologies or project templates are created.
- The window used to create a new project.
- The design window, which is the interface from where an existing project's data can be edited.

These user interfaces are considered in the following sections.

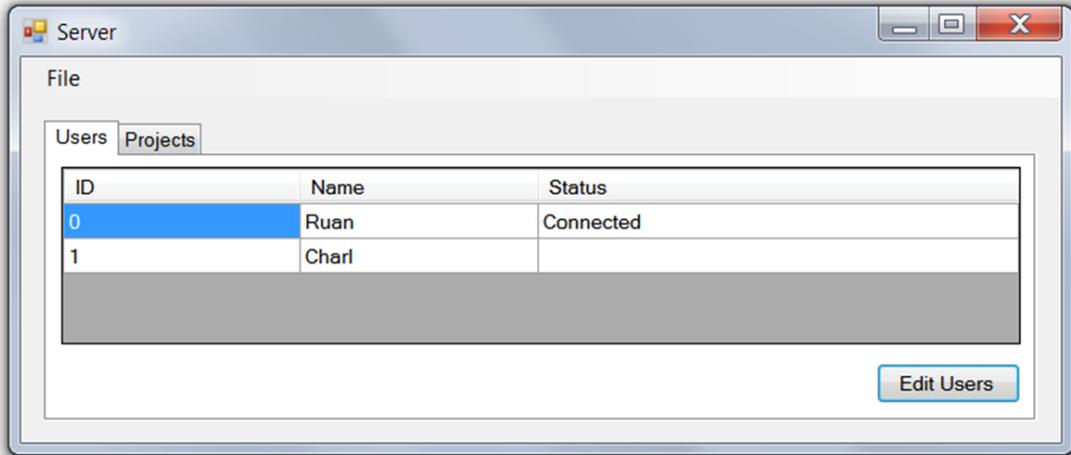


Figure 21: Server Main Window

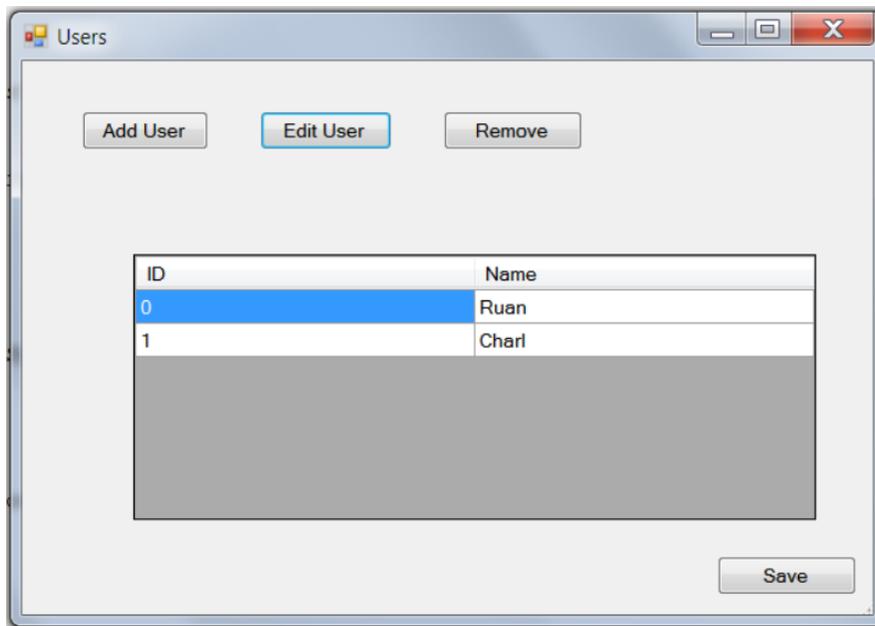


Figure 22: User Names Editor

### 3.3.4.2 Start Screen

When the client application is launched, the first window that the user sees is the start screen (Figure 23). From this window the user can create or open project templates (effectively the ontology blueprint for a given project) as well as create projects from existing templates or open existing projects. The user can also monitor the connection status and connect or disconnect from the server.

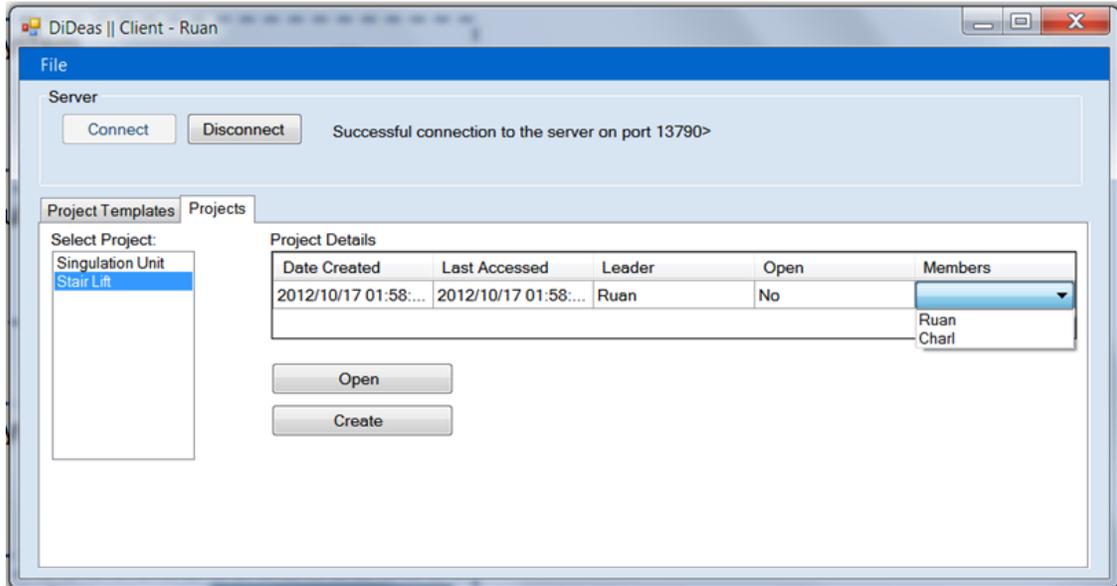


Figure 23: Client-side Main Window

On the "Projects" tab, a list of projects is displayed and the user can view some information related to a specific project, such as the creation date, the last date the project was accessed and the team members and team leader of the project. Similarly, on the "Project Templates" tab (Figure 24), a list of project templates is displayed along with the completion status of each template. From here, the user can create new project templates or edit existing ones.

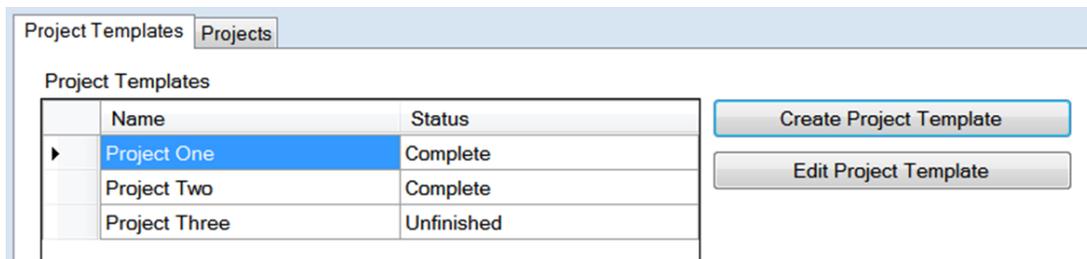


Figure 24: Edit Project Templates

### 3.3.4.3 Project Template Editor

Tailoring DiDeas II to a specific design style is done by creating a project template and can be performed without any knowledge of ontologies or conceptual graphs. Initially, the creation of a project template would require training, however after experience with this process, it can be performed in a relatively short time.

Once the user selects to edit or create a project template, the project template wizard is displayed from where a template can be created via a series of steps, each of which are displayed on a separate tab-page (Figure 25). If the user aborts the process before the template is completed, he/she can choose to save the project template in its current state for completion at a later time.

The respective tab pages perform the following functions:

#### 1) Project Details

The first tab-page is where the project template is given a unique name (Figure 25).

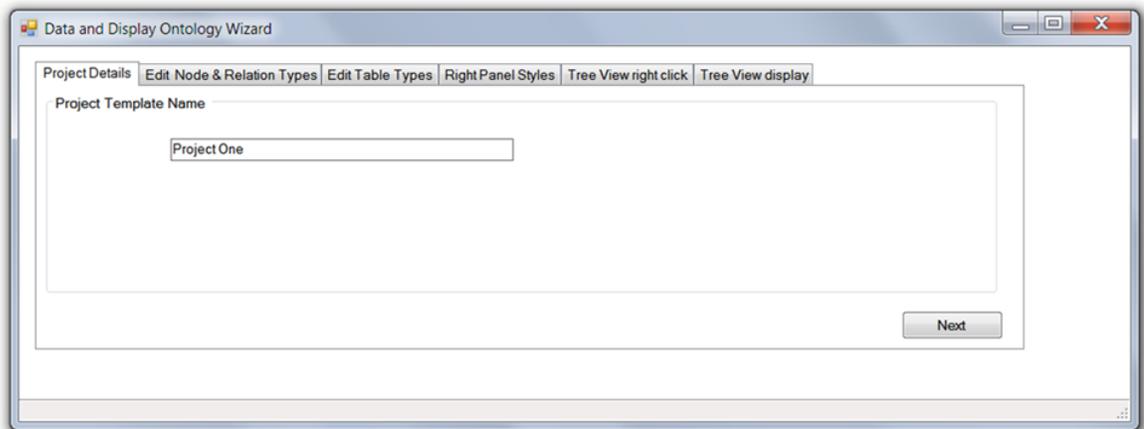
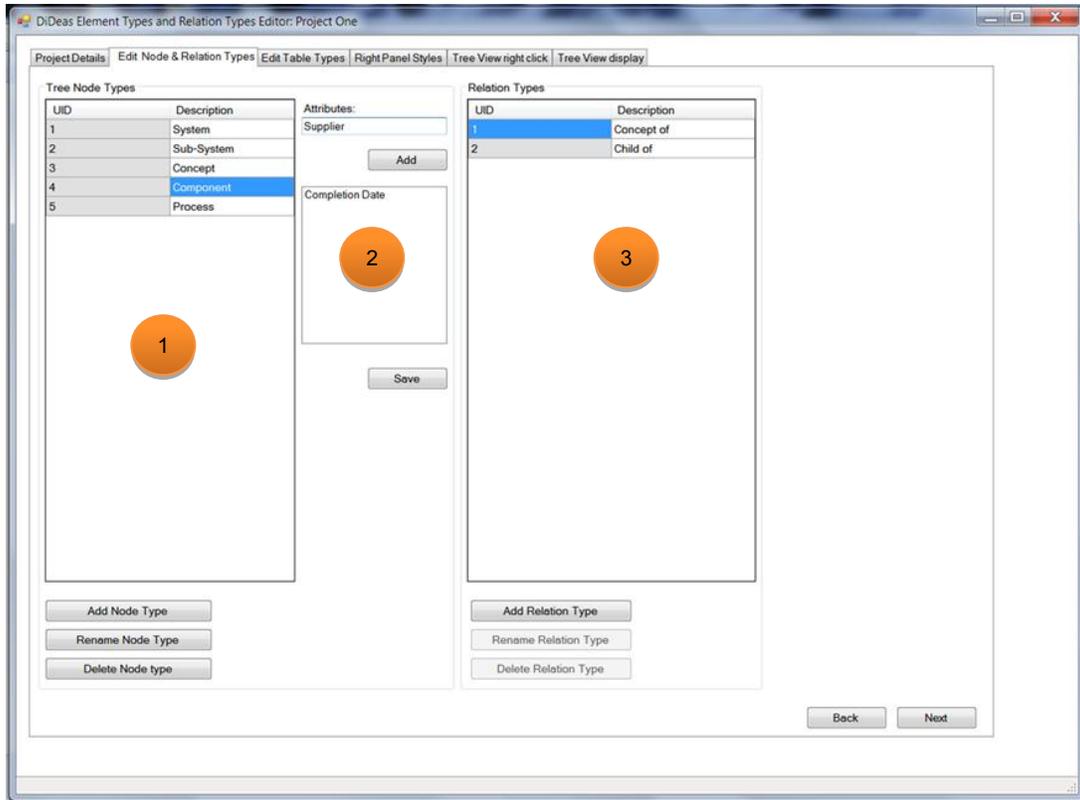


Figure 25: Project Template Editor.

#### 2) Edit Node and Relation Types

On the second tab, the types of elements (marked 1 in Figure 26) that describe the nodes in the tree-structure on the left-hand side of the design window (described in section 3.3.4.5) can be added to the project template along with the types of relations (marked 3 in Figure 26) among these nodes. Each node type can be assigned multiple attribute types (marked 2 in Figure 26) which are used to display information related to the specific node within the tree structure. Each type acts as a blueprint for the related element, relation or attribute and dictates the behaviour and information to be displayed by a specific node or attribute.



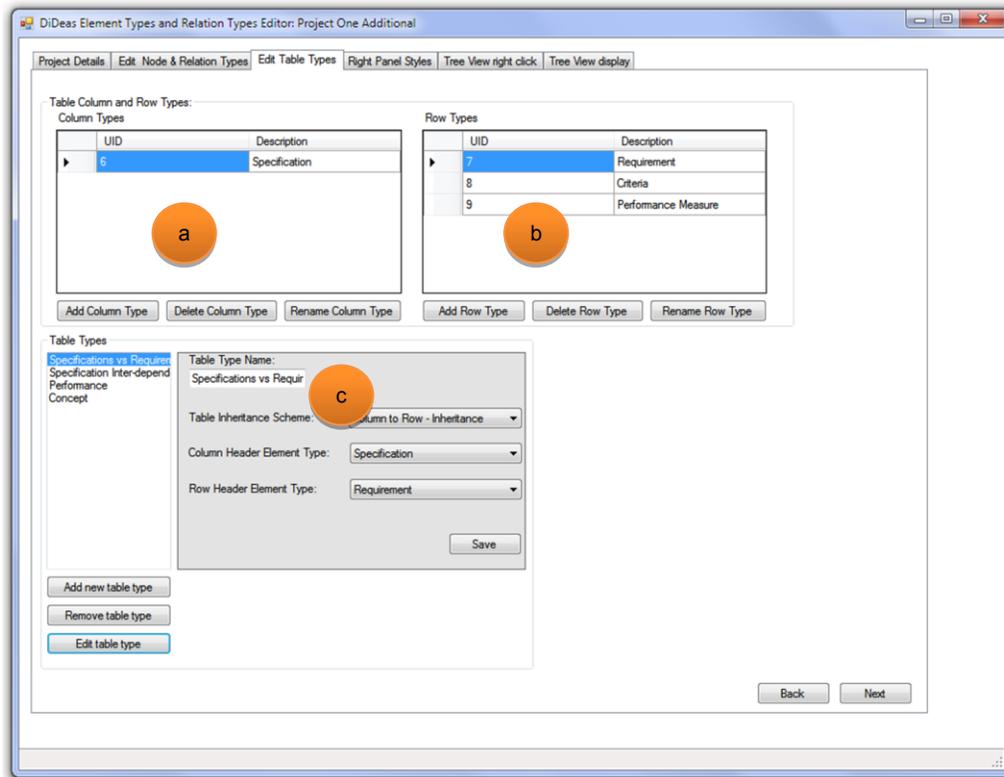
**Figure 26: Project Template Editor Tab: Edit Node & Relation Types**

### 3) *Edit Table Types*

On the third tab, data tables types, which will be used to display information in the right-hand pane of the design window (described in section 3.3.4.5), are defined. Once the user has created the various table types for the project template, he/she can advance to the next tab page where the table types can be allocated to specific tab pages.

Element types for the column and header cells of the tables must first be determined (marked a and b in Figure 27). The description can be selected by the user. This allows the user interface to use the terminology familiar to the specific team.

After defining the column and row types, the user can create a table type (marked c in Figure 27) and specify the type of data inheritance (see Section 3.3.3.2) to be employed in the table. The user can also select the column and row types to be used for the specific table type.



**Figure 27: Project Template Editor Tab: Edit Table Types**

#### 4) Right Panel Styles

On the fourth tab, the right panel styles are created. Each style specifies a way in which tab pages, sub-tab pages and tables are displayed in the right-hand panel of the designer window (see section 3.3.4.5).

A tree-view (see Figure 28) is used to show the definition of the right panel style. The top node represents the right panel style itself. The first level of child nodes represents the top level tab pages of the right-hand panel. A top level tab page can contain either another set of tab pages (here called sub-tabs) or a table. A sub-tab page can only contain a table. The hierarchical structure provides the user with a convenient graphical representation of the right panel style.

When a table type is added, a window (see Figure 29) is displayed where the user can specify the table type from the list of table types created in the previous step. From this window, the user can then select the tables from which row or column headers are to be inherited.

Each right panel style can then be assigned to a specific type of node (see Figure 30).

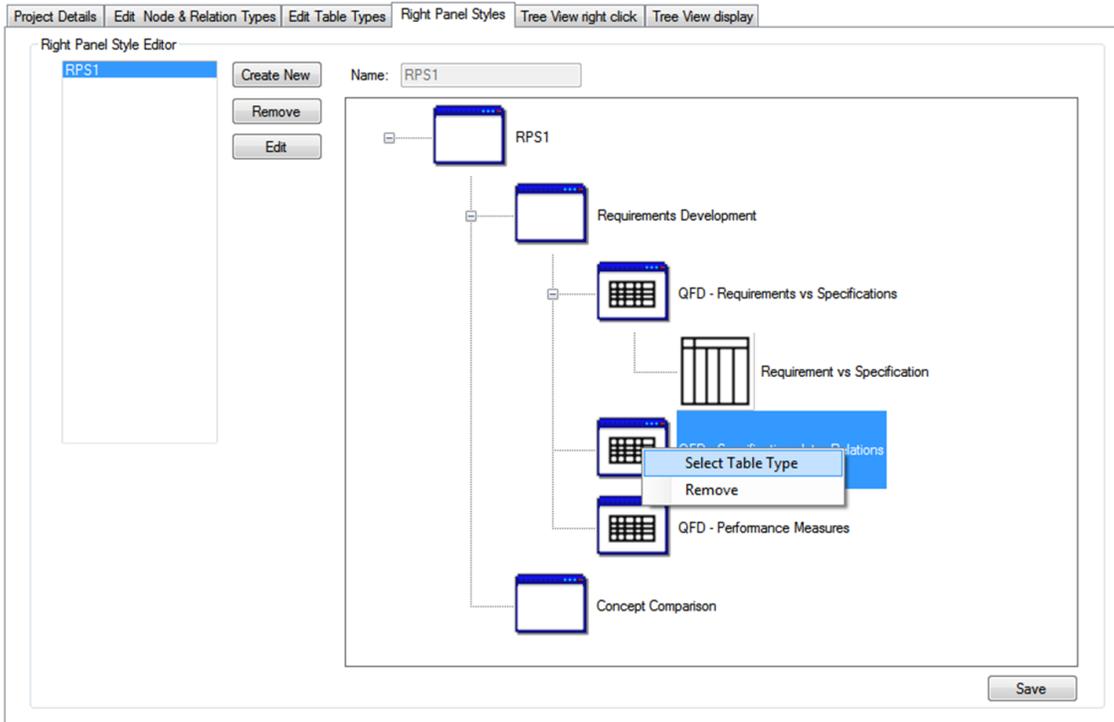


Figure 28: Tab-Pages Layout Editor

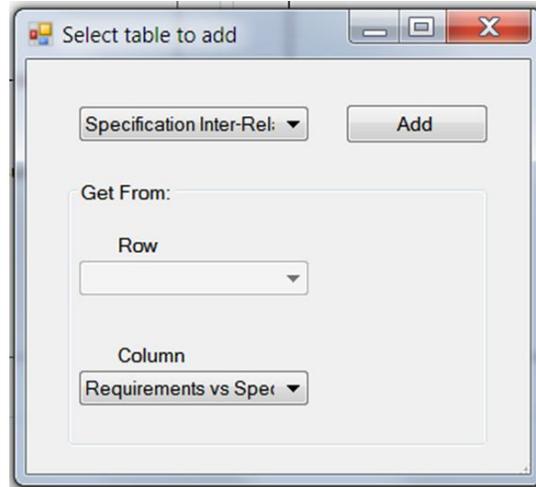


Figure 29: Table Type and Inheritance Source Selection Window

Assign Right Panel Style to Element Type

UID	Element type	Menu references	Right panel style
2	System	0	RPS1
3	Sub-System	0	RPS1
4	Component	0	RPS1
5	Process	0	RPS2
6	Concept	0	RPS1

**Figure 30: Node Type Right Panel Style Assignment**

5) *Tree-View Right-Click*

The tree-view in the designer window is edited by right-clicking on nodes in the tree. On the fifth tab, the right-click menu for the tree structure is edited, along with the text to be displayed for each option. Some menu options are fixed and cannot be edited by the user, for example to delete the selected node. The menu options to add nodes, that are presented when the user right-clicks on a node, are created and assigned to relevant nodes using the form shown in Figure 31.

Here the element types that can be added as child nodes, according to the selected parent node's element type, are defined. The menu can be different for each element type in the tree. This menu determines the rules for the tree structure and determines which types of nodes the user is allowed to add as children for each specific node type.

In addition to the element type to add for each option, the relation type that the new element will have to its parent is selected as well.

Project Details | Edit Node & Relation Types | Edit Table Types | Right Panel Styles | Tree View right click | Tree View display

Right click is used to change the nodes shown in the tree view

Right-click menu add option

UID	Menu Description	Menu order	No. of parent types*	Element type to add	Relation type to add
0	Add System	0	0	1) System	Root
1	Add Sub-System	0	1	2) Sub-System	2) Child of
2	Add Concept	0	2	3) Concept	1) Concept of
3	Add Component	0	1	4) Component	2) Child of
4	Add Process	0	1	5) Process	2) Child of

Eligible parent element types

UID 2 Description Add Concept

- 1) System
- 2) Sub-System
- 3) Concept
- 4) Component
- 5) Process

**Figure 31: Tree View Menu Editor**

6) *Tree-View Display*

On the sixth and final tab, the display style of each node- and relation-type in the tree structure can be edited. The line-style and colour of the border of each node can be edited along with the colour and size of the text in each node (see Figure 32). The colour and line-style of each relation type can also be edited (see Figure 33). This

allows for a customization of the display style of the tree-view and specifies how each type of node and relation is to be displayed.

Tree node type display format						
UID	Element type	Text size	Text colour	Box line style	Box line colour	Menu references
1	1  System	5.0	Black	Solid	Black	1
2	2  Sub-System	5.0	Blue	Solid	Blue	1
3	3  Concept	5.0	Green	Dash	Blue	1
4	4  Component	5.0	Navy	Solid	Navy	1
5	5  Process	5.0	Maroon	Solid	Maroon	1

Figure 32: Tree View Node Display Editor

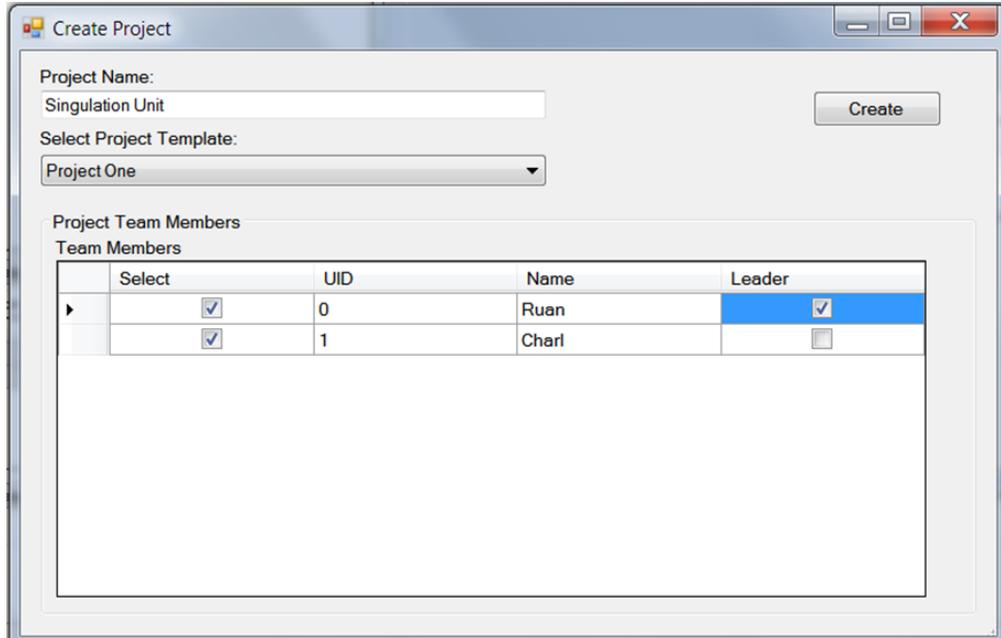
Relation type display format				
UID	Relation type	Leader line style	Leader line colour	Menu references
2	2  Child of	Solid	Black	3
1	1  Concept of	Dash	Blue	1

Delete selected      AddTree View Menu item to add new relation type

Figure 33: Tree View Relationship Display Editor

#### 3.3.4.4 Create Project Window

After one or more project templates have been created, the user can select to create a project by clicking on the "Create" button shown in the client-side main window (Figure 23). The Create Project window (Figure 34) then appears where the project is given a name and the template for the specific project is selected. The team members for the project are selected from a list of user names stored on the server and the team leader is selected.

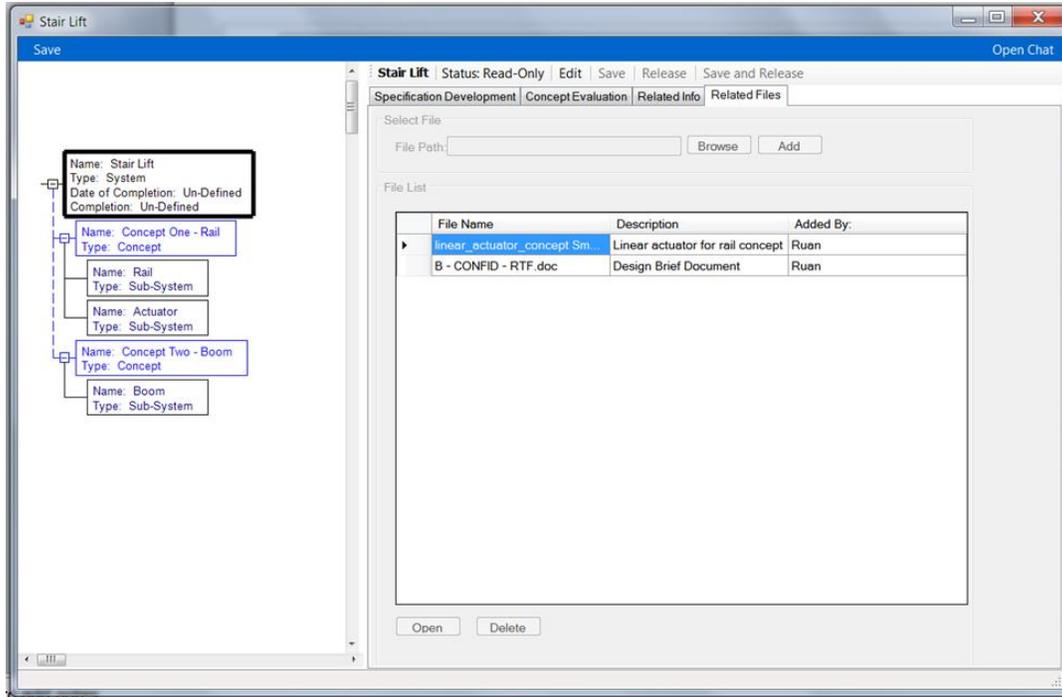


**Figure 34: Project Creation Window**

#### 3.3.4.5 Design Window

Figure 35 shows a typical design window of DiDeas II which provides the working environment during a project for each user. The particulars of what is displayed are determined by the project template definition (which, for example, determines the names of the tabs) and the project data. The window is divided into two sections, namely the tree-view and the right-hand side panel. The tree-view in the left-hand panel of the project interface provides a visual means to display the overall system structure of the system being developed, as well as a means of navigating through the project data. In Figure 35, the system node was selected in the tree-view, and the information associated with that node is displayed in the right-hand panel of the project interface. When a user selects an element in the tree view, the element is highlighted, the right-hand panel's layout is changed and the information relevant to that node is displayed in the right-hand panel. The functionality of the tree-view and right-hand panel is explained in more detail in the following paragraphs.

In the top right-hand corner of the design window, an "Open Chat" menu item is provided. The DiDeas II Messenger window, which opens when the user selects this menu item, is described after the discussion of the right-hand panel.



**Figure 35: Project Window**

In the tree-view the entire system structure can be viewed and the various elements in the system structure are displayed as nodes in the tree-structure. The tree structure can be collapsed at each level to hide sub-nodes. As can be seen in Figure 35, each node displays information such as the name of the element (e.g. Stair Lift) and the type of element (e.g. system, sub-system or concept), as well as other relevant information which is determined by the attribute types associated with the specific type of node. The attribute's values can be changed on the "Related Info" tab of the right-hand panel.

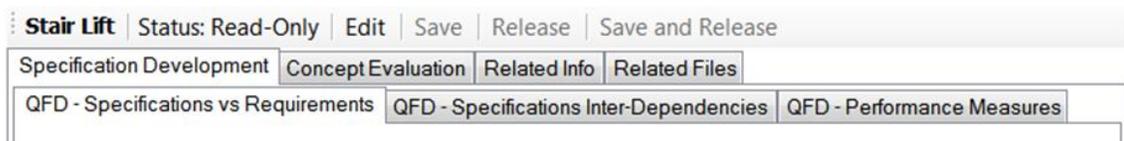
The tree is edited through a right-click menu that is defined in the project template. When the user right-clicks on a selected node, the right-click menu offers the opportunity to create child nodes (of the types allowed for the selected node), delete the selected node or grey-out the specific node. When certain concepts are no longer considered, their nodes can be "greyed-out" to signify that the concept is no longer considered. This allows team members to retain information about all the alternatives that were considered during the design process.

The tree view is a relaxed form of what you see is what I see (WYSIWIS), as described in Section 2.1.2; in that other clients see updates after a short delay. Changes to the tree structure are sent to the server at the instant they are made by the user. When polled by other clients, the changes are then sent to the other clients and each tree-view is quickly updated to provide all users with a shared view of the system structure.

The right-hand panel (RHP) displays information associated with the node that was selected by left-clicking in the tree view. A status bar along the top of the RHP shows the name of the particular node, as well as the access status of the node. The user can choose to enter the edit mode for the node by clicking on the "Edit" button in the status bar. Once in edit mode, the user can save any changes made to the data displayed in the RHP by clicking on the "Save" button (which will allow the user to continue editing the specific node) or by clicking on the "Save and Release" button which will return the node to read-only status. The user can also discard any changes by clicking on the "Release" button.

Changes made to information in the RHP are only sent to the server and shared with other clients once the user chooses to save the information.

The information in the RHP is organized through tab-pages (see Figure 36). The layout of these tab-pages (top-level) and sub-tab pages (second level) are configured in the project template editor as described in Section 3.3.4.3, except that the last two top-level tab pages (Related Info and Related Files) are present in all RHPs.



**Figure 36: Typical Tab-Page Layout**

The data tables shown on the tab and/or sub-tab pages are also determined by the project template and can be customised (see Section 3.3.4.3) to display row headers and column headers according to the desired style of the users (examples are given in Figure 37 and Figure 38; Figure 38 shows a table type that is used to compare a list of items against itself). In edit mode, the user can add or remove columns and rows to input data. The respective names for the column and row headers are specified by the user that created the project template.

Information in tables can be entered in any format and it is the responsibility of the user to ensure that information is in a sensible format.

Stair Lift | Status: Edit Mode | Edit | Save | Release | Save and Release

Specification Development | Concept Evaluation | Related Info | Related Files

QFD - Specifications vs Requirements | QFD - Specifications Inter-Dependencies | QFD - Performance Measures

Add Specification | Remove Specification

	Load Capacity	Physical Size	Speed	Number of modules	Maximum gradient
Safety			3		4
Reconfigurability				3	
Modularity				5	
Installable in existing ...		4			

Add Requirement | Remove Requirement

Figure 37: Data Table Example One

Stair Lift | Status: Edit Mode | Edit | Save | Release | Save and Release

Specification Development | Concept Evaluation | Related Info | Related Files

QFD - Specifications vs Requirements | QFD - Specifications Inter-Dependencies | QFD - Performance Measures

	Load Capacity	Physical Size	Speed	Number of modules	Maximum gradient
Load Capacity		1	4		
Physical Size			3		
Speed					3
Number of modules					
Maximum gradient					

Figure 38: Data Table Example Two

As mentioned above, two tab-pages that are present for each node are the "Related Info" (Figure 39) and the "Related Files" tabs (see Figure 40). A conversation history for a given node is displayed at the top-left of the "Related Info" tab and the text editor is displayed at the bottom-left. This allows users to attach comments to specific nodes to provide context to information and insight into design decisions.

The attributes for the node selected in the tree view can be edited on the right-hand side of the "Related Info" tab page. The attributes are used to display project template defined information in the tree nodes, which quickly provides information relevant to each node to the user.

On the "Related Files" tab (see Figure 40), a file browser allows users to search the windows folder structure for files they wish to attach. Most of the tab page is occupied by a list of all files already attached, along with for each file a short description and the name of the team member that attached the file. From here the user can open or delete files that are attached while in edit mode.

As mentioned above, when the user selects the "Open Chat" menu item (top right-hand corner of the design window), the DiDeas II Messenger window (see Figure 41) is opened. The DiDeas II Messenger is used for instant messaging and is displayed in a separate window, which can be hidden while inactive. The window will reappear each time a new message has been received. When the user wants to send a message, he can select the recipients from a list of the team members. The message history includes a list of all messages that have been sent to the specific team member and each message is displayed along with the date and the name of the sender of the message.

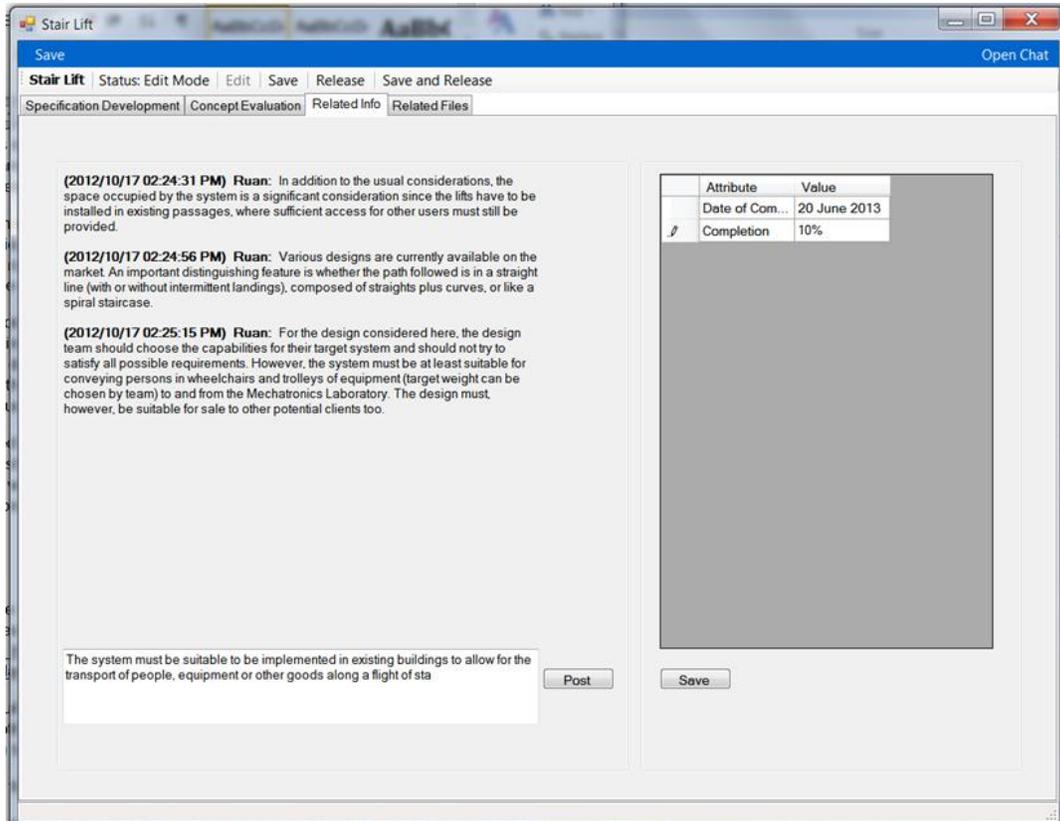


Figure 39: Related Information Tab-Page

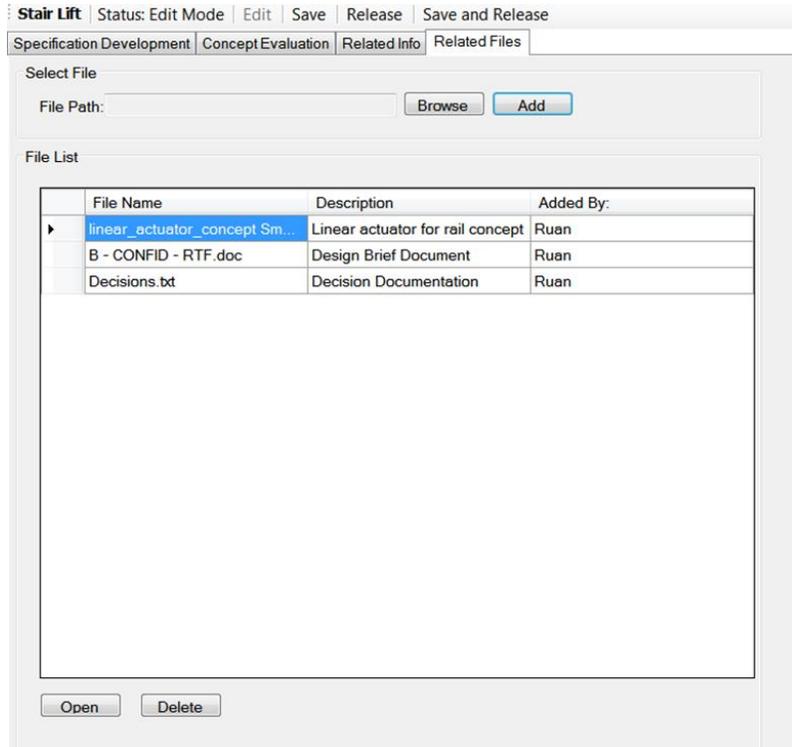


Figure 40: Related Files Tab-Page

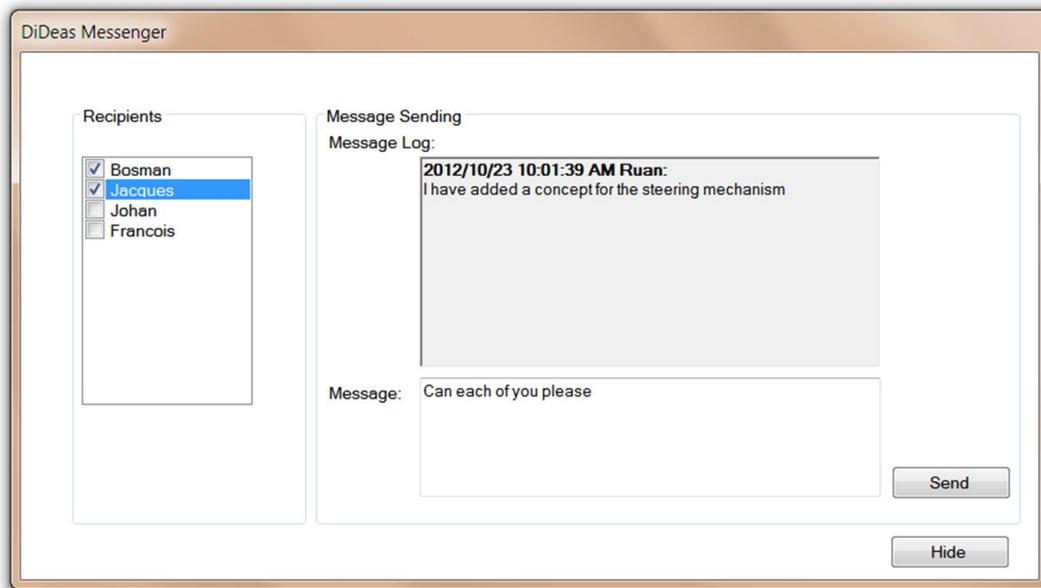


Figure 41: Messenger Window

## 4 Case Studies

The case studies involved two experiments in an academic environment where design tasks had to be accomplished and a case study which included demonstrations of DiDeas II to engineers in industry for whom project templates were created according to the specific industries. Owing to limitations in time and the availability of participants, the case studies were limited in scope and involved eight individuals for the academic case studies and five engineers in industry, each working at a different company.

### 4.1 Academic Environment Case Studies

The academic case studies consisted of short design experiments. Two experiments where participants worked synchronously and one where participants worked asynchronously were performed with each experiment involving teams of four members.

#### 4.1.1 Setup of Academic Case Studies

##### 4.1.1.1 Physical Setup and Software

The experiments were carried out with participants located in a single location and participants were instructed to communicate via DiDeas II. A brief introduction to DiDeas II and the design methodology or process that was to be followed was given along with the design brief.

Each participant was provided a PC with internet access and that had the DiDeas II Client application and Autodesk Sketchbook Pro installed. For input of concept sketches into the system, the designers were each provided with either a tablet (Apple iPad or Samsung Galaxy II tablet), a Wacom pen tablet or were required to use the mouse (these devices are shown in Figure 42). Designers used either their finger or a stylus to create free-hand sketches on the tablets which were then transferred to the desktop PC for input into DiDeas II.



Figure 42: Input Devices. a) Apple iPad b) Samsung Galaxy Tab c) Wacom Graphire

The experiments were observed using Microsoft Expression Studio screen-capturing software with the aim of minimising the effect of being observed (e.g. via a video camera) on the behaviour of the participants. A separate PC was used to run the server application.

Users were provided with Autodesk Sketchbook Pro which allows users to create freehand sketches using any of the input methods as it is available on the iPad, the Android operating system used by the Galaxy Tab and on Windows desktop PCs. Figure 43 shows some elements of the Sketchbook Pro user interface.



**Figure 43: Autodesk Sketchbook Pro Interface Elements**

#### *4.1.1.2 Configuration of Ontology (Project Template)*

For the academic case studies using post-graduate engineering students as participants, each team had to complete a design project up to the conceptual design phase and a project template was created with elements of formal design methodologies including QFD for specification development and Pugh's method for concept evaluation as described in Section 2.1.3. For comparability purposes, the initial intention was to use the same template for all experiments; however certain elements were altered to allow for better collaboration after the first synchronous experiment.

The process that team members were to follow included three steps:

1. Capture of user requirements and specification development.
2. Concept generation (including the gathering of information).
3. Concept evaluation.

For the project template, the possible node types included system, sub-system, component and concept nodes in the tree-view and each node in the tree structure could have multiple concept nodes as children.

Data tables were configured in the project template to capture the information related to user requirements and concept evaluation.

The QFD approach (as described in section 2.1.3.1) was implemented for the capture of user requirements and specification development and each team was asked to complete the aspects of the House of Quality (HOQ) diagram that were included in the project template.

In the QFD approach described by Blanchard & Fabrycky (2006), system-level requirements may be developed into more detailed requirements at each stage of the design and development process and are translated to systems, sub-systems, components, manufacturing processes, support infrastructure etc. The requirements (“how’s”) in the House of Quality in a given level of the system structure become the design parameters (“what’s”) on a subsequent level and the inheritance of data among tables as described in Section 3.3.3.2 was used to accomplish this.

The parts of the HOQ diagram employed in the project template include the relations between the requirements and the design parameters, the inter-dependencies among the design parameters and the target values for the design parameters. This was implemented through three different tables each displayed on a separate tab-page namely the “requirements vs. design parameters”, “design parameter inter-relations” and “target value” tables.

At the system level, user requirements are added as the row headers of the “requirements vs. design parameters” table. The user requirements at the lower levels, including the sub-systems and component nodes, are inherited from the design parameters in the corresponding table in the parent node. The design parameters have to be entered manually in each table.

The design parameters in the “requirements vs. design parameters” table are transferred to the “design parameter inter-relation” table which represents the “roof section” of the house of quality.

The design parameters are also transferred to the “target values” table which represents the section that is usually located at the bottom of the HOQ diagram.

Participants then had to develop concepts and attach the relevant information to each node. For the final step, namely concept evaluation, Pugh's method (decision matrices) was to be used where each concept is evaluated against weighted criteria decided upon by the team members. These scores are then summated to provide an overall score for each concept. For this method, the inheritance type as described in Section 3.3.3.2, where the parent node receives values from the various concept nodes, was implemented to allow for the comparison among various concepts.

#### **4.1.2 Setup for Synchronous Case Studies**

For the synchronous case studies, two case studies were performed in which the team members worked simultaneously. These case studies were aimed at assessing the effectiveness of DiDeas II to support real-time communication among

participants working synchronously. The participants were therefore instructed to perform all communication through the functionality provided by DiDeas II.

For the first case study, four post-graduate students from differing backgrounds (differing nationalities and/or undergraduate institutions) participated, while the participants of the second case study were four post-graduate students from the same research office and who completed their undergraduate studies at the same university.

In each case study, the participants were given approximately three hours to complete the task of designing a system up to the concept evaluation phase, with no calculations or detailed design required. The duration of each case study was limited by the availability of participants, which correspondingly limits the degree to which the study could be used to simulate the use of DiDeas II during a real design project. However the case studies still provide insight into users' experience when working with DiDeas II and sheds light on the effectiveness of the user interface.

The design task given to the teams was to design a singulation unit for a re-configurable automated assembly cell (see Appendix B for the design briefing). The singulation unit had to be able to receive parts (with only the approximate size known, but not the shape) in bulk and present them one at a time to a six degree-of-freedom robot.

### **4.1.3 Synchronous Case Studies Observations**

#### *4.1.3.1 Influence of Prior Education and Training*

During the first case study it was evident that the participants were not familiar with the QFD method and were not familiar enough with the user interface of DiDeas II. Before commencing with the second case study, participants were therefore given a detailed introduction to the QFD method and the design process that they were to follow, as well as the DiDeas II user interface and functionality.

During the first case study, where the participants did not receive detailed instruction on the methodology and the procedure that was to be followed for specification development and concept evaluation, participants were unable to follow a systematic approach. User requirements were entered at sub-system nodes instead of at the top level without consulting team members and it was evident that the participants did not understand the distinction between the user needs (“what”) and the design parameters (“how”). Certain users also commenced with design work at the beginning of the project before entering user requirements.

Before the second case study, participants were given a more detailed introduction on the HOQ and how to use the DiDeas II interface. A more systematic approach was followed where user needs were discussed over DiDeas Messenger at the start of the project and entered at the top level in the system hierarchy. Specifications were developed in the same manner and information was entered into the tables

representing the HOQ in the correct format. User needs were, however, not translated to the sub-system level which might have been a result of the time constraint.

#### *4.1.3.2 Allocation of Tasks*

In the second case study, the tables were completed by allocating specific sets of rows and columns to each participant who then filled in the relevant information. After each team member completed a row, he then announced this in the Messenger. The process was quick, but other team members were idle while waiting for their colleagues to finish. The assigning of tasks did, however, quickly provide consensus on how the tables were to be filled in as opposed to the first case study where all tables were filled in individually.

Thereafter the different concepts with their sub-systems and components were created and developed, and finally the different concepts were evaluated as a team effort using Pugh's method.

#### *4.1.3.3 Entering Concepts in the System Hierarchy*

In DiDeas II, once a node with a "concept-of" relation type is added as a child of a specific node, the only option that remains is to add further concepts as children to that node. Conversely, once a sub-system or component is added, concepts cannot be added. This is done for logical reasons as it would not make sense to compare various concepts for a specific element to sub-systems or components thereof.

During the first case study, DiDeas II allowed users to add either a concept or a sub-system/component beneath a certain node in the tree structure. This quickly proved to be problematic as other team members could not add their concept once a sub-system or component node was added. Therefore the ontology was changed for the second case study to always have a concept level beneath a system or sub-system node, to allow different concepts to be developed by different team members. The result was that in the second case study, each team member added a concept for the system which was then developed further by the specific member with sub-systems for which different concepts were also added.

#### *4.1.3.4 Attaching Files*

Each of the participants attached multiple files to the project, yet none of the participants performed any editing on files that were uploaded by other team members during the synchronous case studies.

During the synchronous case studies, image files containing sketches were the only types of files attached to the project. These files were used to communicate concepts to other team members. The discussion tab was hardly ever used to comment on these ideas and the discussion on concepts was done in DiDeas Messenger. The recording of design information was therefore less structured since concept discussions posted in the chat window do not allow the information to be

stored in a structured form as is the case with having discussions attached to specific concepts, where they would be quickly accessible for future reference.

The participants were only able to produce very basic concept sketches via the input methods used. Two participants switched to using a mouse for sketching, which was a method they were more accustomed to. Both produced very neat sketches and one participant produced a very detailed sketch. Figure 44 shows some of the sketches produced during the first case study and Figure 45 shows some of the sketches produced during the second case study.

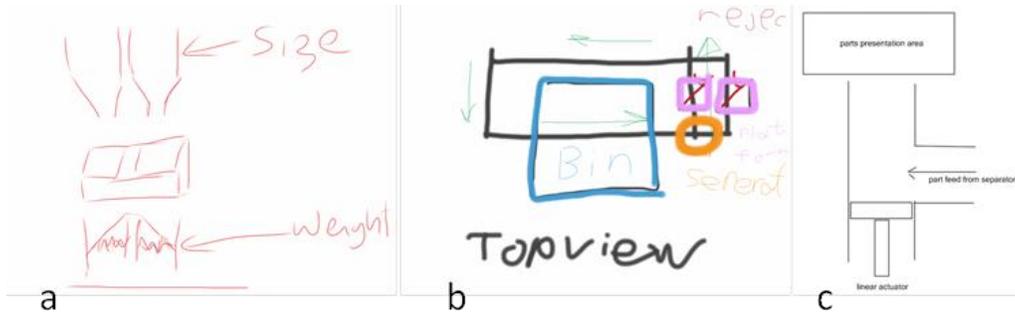


Figure 44: Concept Sketches Produced During First Experiment. Input Methods: a) iPad b) iPad c) Mouse

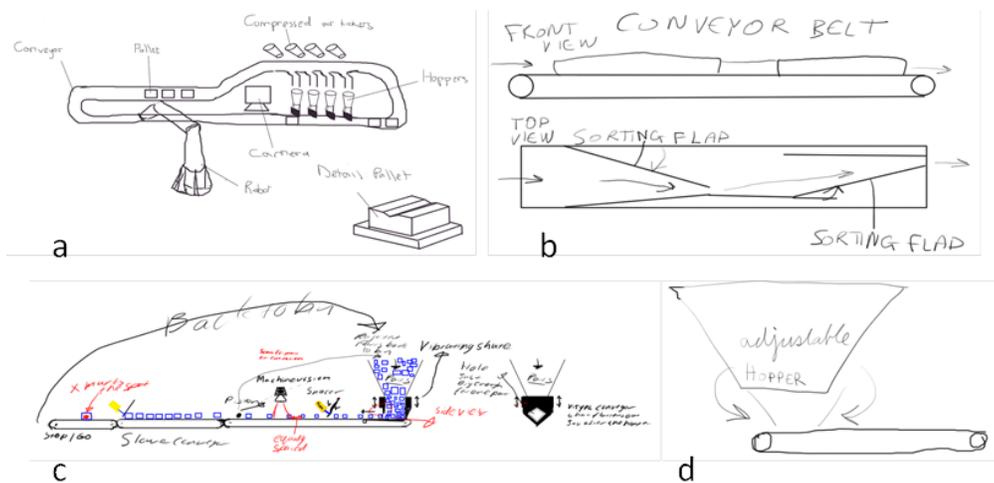


Figure 45: Concept Sketches Produced During Second Experiment. Input Methods: a) Mouse b) Wacom Pen Tablet c) Wacom Pen Tablet d) Wacom Pen Tablet

#### **4.1.4 Synchronous Case Studies Feedback**

Each participant completed a questionnaire to assess their perceptions of the usability and usefulness of DiDeas II. The questionnaire, along with the feedback received, is provided in Appendix D.

##### *4.1.4.1 Overall Assessment*

All the participants of the first case study agreed that DiDeas II was an effective means of communication, while those in the second case study did not think DiDeas II was an effective means of communication and mentioned that the chat functionality is too slow. This might be because the participants from the first case study are not as familiar with each other as the participants from the second case study. The latter might have desired a more direct method such as voice communication.

Out of the eight participants in the synchronous case studies, seven agreed that DiDeas II was an effective means to record design decisions, while one participant stated that it was moderately effective as the user interface made it difficult to record information.

Five participants agreed that DiDeas II was an effective means for design team members to generate concepts. Some comments included that more emphasis should be placed on discussion and analysis, and that using DiDeas II requires someone experienced in design procedures. The other three participants stated that DiDeas II is ineffective in some respects, such as a lacking communication system and that drawings are done in a separate program. One participant stated that it is not effective for the short duration of the case study; however he expects that it will be effective in industry.

Five participants also thought that DiDeas II was an effective means for design team members to make design decisions. Comments included that it provided flexibility by allowing team members to view the work of other members, that the discussions and the ability to view other members' notes were helpful, and that the concept evaluation functionality works well. Among the other three participants, mention was again made that DiDeas II was not effective for the short time-span of the case study and that the decision making process is slow, yet appropriate for distributed teams. One participant stated that he disliked formal design processes.

##### *4.1.4.2 Communication Functionality*

During the synchronous case study, the discussion tab was only used once and participants preferred to post discussions in DiDeas Messenger. Some participants mentioned that they had used the discussion tab which they confused with the Messenger indicating that not everyone was aware of the discussion tab's purpose.

Almost all of the participants found DiDeas Messenger to be an effective means of communication; however some participants in the second case study did perceive it

to be slow and some participants found the appearance of the DiDeas Messenger window during design work, to be disruptive.

When asked if they would prefer other communication methods in addition to the functionality in DiDeas II, some participants stated that they would like to use e-mail in addition to the messenger, while three participants mentioned voice communication and one participant mentioned that he prefers the familiar interface of Skype. One participant mentioned that he finds it useful to include photos and equations in e-mails and another participant mentioned that he prefers the Discussion tab to e-mail as it structures the information.

Participants mentioned that they were constrained for time which is why they would prefer voice communication for things such as meetings and brainstorming, otherwise text communication would be sufficient. One participant mentioned that he would like to be able to send longer messages with the Messenger.

#### *4.1.4.3 Sketch Input*

The users that were provided with tablets (iPad or Galaxy) for sketching reported that they quickly became accustomed to the device, while the users provided with Wacom tablets, reported longer periods to become accustomed, with one user opting for the mouse instead.

The participants provided with tablets (iPad or Galaxy) all agreed that it was an effective means for sketch input, while the other participants found that the methods they employed were less effective and would prefer other methods.

The majority of participants stated that they were able to convey the same information as with pencil and paper, while some mentioned that they might need more practice.

#### *4.1.4.4 Ease of Use*

The participants agreed that the tree structure clarified the system structure and also provides structure to the information. The participants agreed that the tree nodes provided the appropriate information (they were not provided with the functionality to edit the information displayed in a specific tree node).

All the participants agreed that the tab pages were helpful when searching for information that they were looking for.

Most of the participants responded that the tables allowed them to find information that they were looking for, while some mentioned that it was confusing at the beginning when there was no information displayed in the tables.

Some of the participants of the first case study mentioned that some aspects of the QFD and concept comparison processes were confusing, while the participants of

the second case study who received more detailed instruction on the processes to be followed, stated that these aspects were clear to them.

When asked what information the users were unable to record, three users replied that they found it difficult to describe their concepts. One person mentioned a conferencing ability that would allow users to discuss why certain values were chosen for design factors. One person mentioned conversations in chat between individuals.

During the first case study, where participants worked more independently, users seldom tried to access locked nodes. However, during the second case study where tables were completed as a team effort, users tried to access locked nodes more often. None of the participants stated that they tried to locate nodes that had been deleted by others.

During the first case study, participants had information overwritten by other users. This was identified as a bug in the code and rectified before the second case study, with the result that for the second case study, none of the participants had information overwritten by other team members.

When asked what aspects of the software severely hindered design work, the aspects mentioned were slow execution, multiple window interface, uploaded files not appearing at the client-side, trying to access locked nodes and only being able to open files attached to a specific node while in “edit” mode.

#### *4.1.4.5 Ideas for Future Development*

When asked for suggestions for future development of DiDeas II, the participants provided the following ideas:

- Real-time collaboration tools such as conferencing through VoIP and a collaborative sketching tool (whiteboard) was mentioned, as well as changing to a web-based interface that allows team mates to view editing in real-time. The ability to share screenshots of the desktop was also mentioned.
- Integration of more commonly used tools such as Skype and Dropbox to make a “sandbox” style design tool. Integration with Siemens NX or Inventor, with parts tracking.
- An update stream that allows users to quickly see what changes have been made to the project.
- A tutorial or wizard that guides the designer through the process of design and concept generation.
- Sharing of images in the Messenger.
- The ability to edit the names of nodes as well as row and column headers.
- Incorporation of a non-technical user interface for non-technical people where information for system requirements can be input at user level.
- A description of the system, as well as each sub-system.
- Strong systems engineering components will also be welcome.

- A read-only option for nodes where sketches can be viewed from locked nodes.
- The reasons behind weighting of criteria needs to be displayed to allow for better interpretation.

#### **4.1.5 Synchronous Case Studies Conclusions**

The DiDeas II Messenger was not highly effective and users mentioned that they would prefer more real-time communication through a medium such as Voice over Internet Protocol (VoIP). Skype was specifically mentioned, which is a software package that allows various forms of communication including text-based, voice and video communication. The purpose of employing chat functionality directly in DiDeas II was to capture implicit design decisions. When using an external communication medium, a functionality to capture meeting minutes quickly in DiDeas II might provide an effective means to capture design decisions. The DiDeas II Messenger was, however, well received by the participants of the first case study who were not very familiar with each other.

With adequate instruction, the users were able to complete a design task while following a specified design process, even though it was their first time using DiDeas II. Although certain aspects of the user interface needs refinement, the overall structure of the interface was well received.

#### **4.1.6 Asynchronous Case Study Set-up**

One case study was performed where two participants from each of the synchronous case studies participated. At the start of this case study, the participants had a face-to-face meeting for one hour, after which each team member completed four design sessions with a total of sixteen sessions among the participants. These sessions were performed asynchronously. A final half-hour session in which the participants performed a design evaluation for the generated concepts, was completed at the end of the case study.

The differentiating factor of the asynchronous case study is the ability to test the effectiveness of the DiDeas II interface in allowing users to record their work in such a way that other team members can understand the context of the work. The ability to convey context can be regarded as a measure of the transparency with which information captured in DiDeas II is displayed. Due to the limited availability of participants, time periods of up to a week passed in-between sessions, thus providing an opportunity to test the ability of DiDeas II to convey the context of design information.

For the asynchronous design case study, the participants were asked to design a configurable stair-lift (see Appendix C for the design brief) capable of transporting passengers and goods along a staircase.

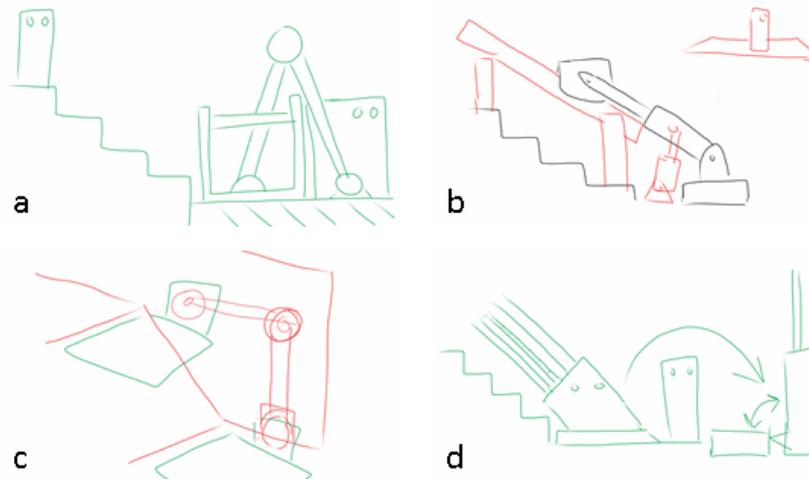
#### 4.1.7 Asynchronous Case Study Observations

Team members used the discussion tab within each concept node's right-hand view to explain and discuss the various concepts frequently and text files containing explanations on a certain concept were occasionally attached along with a sketch.

Each user entered a description of the work he had done into the discussion tab of the top-level or system node and when each user started a session, they viewed this discussion tab to obtain information on what work had been completed since their last session. This discussion tab was also used for a description of the concept choice after a concept evaluation had been performed.

The user requirements were translated to the sub-systems level and the various tables representing the HOQ diagrams were completed for each sub-system with information entered in a correct and consistent format. A scheme for the values of the relations between the requirements and design parameters was decided upon during the first session and all team-members adhered to this while users adhered to the format used by the participant in the first session for the "target values" table.

Twelve sketches were produced with one of these sketches annotated by another team member (see Figure 46). The sketches were also described in the discussion tabs of the relevant concept nodes.



**Figure 46: Selected Sketches from Asynchronous Case Study. a) Simple Side-view b) Annotated Sketch c) Isometric View d) Representation of Movement**

#### 4.1.8 Asynchronous Case Study Feedback

The same questionnaire as in the synchronous case study was used for this case study, with participants being prompted to only complete the relevant sections (See Appendix D).

#### *4.1.8.1 Overall Assessment*

When asked if DiDeas II was an effective means of communication, all the participants in this case study agreed that it was and mentioned that the discussion tab is particularly useful (which is relevant to the asynchronous case study). Two of these participants did not think that DiDeas II was an effective means of communication during the synchronous case study. This difference of opinion can also be influenced by the participants being more familiar with the DiDeas II interface and specifically the discussion tab for communicating design ideas.

All the participants in this case study thought that DiDeas II was an effective means to record design decisions with one person adding that the structuring of the data is useful and another person mentioned as a reason that any file type can easily be viewed and saved.

All participants agreed that DiDeas II was an effective means to generate concepts as it was simple to create and share sketches along with various other file formats.

When asked if DiDeas II was an effective means for team members to make design decisions, three participants answered yes, with one mentioning that it provided well-structured evaluation processes. The fourth said that real-time conversation is necessary, while the tables and stored files might be of help.

#### *4.1.8.2 Communication Functionality*

When the participants were asked about the value of the discussion tabs they responded that it coordinated their work and gave them “insight into decisions”. All the participants responded that they read the discussion related to a specific node every time before commencing work on that node.

#### *4.1.8.3 Ease of Use*

For the asynchronous case study, when asked about the various aspects of the HOQ and Pugh’s matrix implemented in DiDeas II, all participants stated that these aspects were clear to them.

When asked to what extent the tab pages helped in finding information, two participants thought they were helpful and easy to use. However, the other participants thought that it was difficult to find the tab pages and to know when new information was available.

When asked to what extent the tab pages helped in deciding what to do during the design process, one person mentioned that the process can be followed by following the tab pages and another mentioned that the tabs reminded the user what needed to be done in the design process. One participant mentioned that he needed an explanation on the different tables before he knew what to put in them.

When asked what information the participants wished to record but could not, the only mention was that of information in sketches.

#### *4.1.8.4 Ideas for Future Development*

When asked what additional features were desired in DiDeas II, a system to alert the user to new information and updates was mentioned by two participants. An automatic weighting and totalling function for table data was also mentioned.

When asked what suggestions the participants have for future developments of DiDeas II, tutorials to explain the working of DiDeas II as well as step-by-step guidance during projects were mentioned. A feature that provides notifications on what new information is available or what updates have been made was also mentioned.

#### **4.1.9 Asynchronous Case Study Conclusions**

The limited scope and short duration of the design task limited the usefulness of certain features, such as the information stored in the tables.

The Discussion tab was used very often and well received during the asynchronous case study. Users used it to see if new information related to the specific node was available. It might, however, be useful to add functionality that alerts the user when new information in a specific node is available to make information more transparent. After long periods of inactivity on the project, designers were able to read through the discussion tab and navigate the system structure and quickly continue work on the project. The conclusion is therefore that DiDeas II is able to record design information and display it to a degree of transparency that allows designers to view the work of others in context and continue work on the project.

Many problems in the user interface were identified, but still the overall structure of the user interface allowed users to quickly navigate the system structure to find relevant information.

## **4.2 Industry Expert Case Study**

The industry case study was primarily aimed at testing the tailorability of DiDeas II. The case study started with and ended with interviews, here referred to, respectively, as the initial and final interviews. During the initial interviews, DiDeas II was demonstrated to an engineer in each of three companies and enquiries were made about the companies' design procedures, along with any requirements that the engineers have for a design support tool. DiDeas II was then tailored according to each company's design style, using the process described in Section 3.3.4.3. Thereafter, during the final interviews, DiDeas II was again demonstrated and the participants were asked to complete a questionnaire. Only two of the engineers involved in the initial interviews took part in the final interviews, since the third engineer was not available and engineers from two other industries were also interviewed.

#### 4.2.1 Participant Company Profiles

Three different companies with varying design styles were approached for the initial interviews referred to above. Unfortunately, permission to mention the name of the specific company could only be obtained for one of these companies. Two of these companies and two more companies took part in the final interviews. The profile of each company is described here.

The first company, Reutech Radar Systems, was involved in both the initial and final interviews. It supplies systems to the military and can be characterised by its use of a highly formalised systems engineering approach, even though the approach was described as generic by the person interviewed. The importance of capturing all the specifications was stressed and a base-line is used to trace the specifications at different levels (which are referred to as allocated specifications) back to the user requirements. Design reviews are held at various stages of the project.

The second company, that was involved in only the initial interview, is routinely involved in once-off projects that are built specifically to the needs of clients from various industries and can be characterised as a custom design and manufacturing company. The process typically followed by the company is as follows: Firstly the requirements and/or specifications (functionality and cost) for a specific project are received. A concept is then generated and the specifications are developed further which are ultimately driven by costs. This concept is then shown to the customer and discussed along with the cost of the project before further work is commissioned. During the design, if the various alternatives are unable to meet the requirements, they are compared according to various criteria and the various options are shown to the client who then makes a selection. Clients have in the past requested that the information be displayed in a decision matrix. Alternatives or concepts are developed on all levels of the system structure, as well as for aspects such as access and maintenance.

The design process is well documented and the information for each project is stored on the internal network in a folder structure and different file formats including drawings, images, videos, etc., to which everyone has access to, are stored. However, only "bigger" design decisions are recorded.

The third company, involved in both the initial and final interview, can be characterised as a product development company. The company asked not to be named and will therefore only be referred to as the "product development company". It is involved in projects where ideas are developed into prototypes or products and are either involved in the entire development process or only certain stages thereof. After a client has approached the company, a feasibility study is performed in terms of the available technologies and the market potential. To capture the information on a given project, Microsoft OneNote is used which has a less structured interface and allows information to be attached in various file formats.

The fourth company, HG Molenaar, only participated in the exit interviews. This company produces production equipment for the food processing industry. Their design activities are focussed on their own products which include large-scale machinery for food and beverage manufacturers across the globe.

The fifth company, EMSS, also only participated in the exit interviews. EMSS's main focus is on electro-magnetic and antenna analysis services as well as the development of analysis software, highly specialised RF equipment, antenna systems and related sub-components. They make extensive use of a formal systems engineering approach during development.

#### **4.2.2 Observations, Participant Feedback and Interpretation**

A questionnaire was used to obtain the participants' feedback and was aimed at obtaining their perceptions of the usability and usefulness of DiDeas II. The questionnaire along with the feedback received can be viewed in Appendix E. It should be noted that the participants did not actually use DiDeas II themselves, due to their limited availability, but completed the questionnaire after only viewing a short demonstration of DiDeas lasting approximately half an hour. Ontologies were created for the different participants to address most of their recommendations during the initial interviews.

It was observed during the case study that the participant working at the company involved in systems engineering, was more interested than the other participants in the tables that make up the sections of the house of quality diagram, while all participants were interested in the concept evaluation tables.

In the following paragraphs, the participant's responses to the questionnaire are summarised:

When asked if DiDeas II was an effective means of communication with team members that are working elsewhere, two participants agreed, while one person said that it is moderately so and another said that he does not see it as a tool for real-time communication, but for communication of design intent. The participants, however, did not have the opportunity to evaluate the real-time communication functionality of DiDeas II.

When asked if DiDeas II was an effective means for design team members to communicate while working at different times, three participants said that it is effective, with one participant mentioning that by capturing information at a certain time and making it available to other team members at a later time, it is an effective means of communication. One participant said it is reasonable; however follow-up interviews would be required to ensure that information is not misinterpreted.

When asked how effective DiDeas II is as a means to record their design work, all four participants agreed that it is effective, but one also added that it will require provision for configuration management.

When asked how effective DiDeas II was in enabling team members to develop design requirements, three participants thought that it was effective, and one participant thought that it was adequate for academic case studies, but should be extended to allow for the development of user requirements into design specifications.

When asked how effective a means DiDeas II provides for design team members to generate concepts, two participants agreed that it was effective, while one participant replied that it is moderately so. Another participant replied that concept generation is a creative process that cannot be formalised; however software can aid by defining and visually displaying the “design space” in which designers operate.

When asked if DiDeas II provides an effective means to evaluate concepts, the participants mentioned that more functionality is required, including the ability to calculate the total weighted scores for each concept.

When asked if DiDeas II was an effective means for design team members to record design decisions, three participants agreed while one participant was concerned that it will be inconvenient to enter all design decisions into the system and that only larger design decisions should be handled rigorously by the system.

When asked to what extent DiDeas II could be tailored to the company’s specific design style, one participant involved in systems engineering answered that it can be tailored well. One person stated that more pre-configuration is required; however a project template was not created specifically for this company. Two participants mentioned that it is difficult to comment with one mentioning that he did not have enough time to evaluate the software and the other mentioning that his company’s design style requires quick and spontaneous decision making.

When asked what functionality was lacking in DiDeas II, one participant mentioned the ability to organise related files into folders and sub-folders within each node. Another participant mentioned the totalling of concept scores during evaluation (which was also mentioned by other participants during the demo of DiDeas II). One participant mentioned the ability to link to project scheduling software and export project progress.

When asked if the tree view in the left hand pane provides information in a logical and transparent manner, all participants agreed.

When asked if the tree view structure clarifies the system structure, three participants agreed with one mentioning that complex, nested trees might however be difficult to present clearly. A fourth participant stated that other methods of displaying the system structure might be explored, such as a database (tabular structure) rather than a tree structure.

When asked if the tree nodes provide the appropriate information, all participants agreed with two mentioning that it is tailorable and can thus display the appropriate information as needed. One participant mentioned the fact that it can display an updated status.

When asked to what extent the tabs in the right-hand table will assist in finding information, a mention was made of using the colour of tab headings to indicate that new information is available. Two people said that it will be effective and another mentioned that aptly named tabs are useful in general.

When asked to what degree the tabs will assist in deciding what to do during the design process, only one person thought it would be effective.

When asked to what extent the tables in the right-hand panel will assist in finding information, a mention was made that the information should be linked to information used in making decisions (in the case of concept comparison). One person said that well-presented tables are generally useful and one person said that it will depend on the user and the configuration of the system.

When asked to what extent the tables will help to communicate design work, another mention was made of linking the information in the tables to information on why decisions were made. Two participants agreed that it will be effective.

When asked about the overall impression of DiDeas II, the following responses were given:

- The participant from the custom design and manufacturing company thought that it had a very logical layout and provides good record keeping of decisions made in the past. This person also thought that it is a good idea to have designers follow a specific procedure. The person was, however, concerned that it will be time consuming if the application is not kept simple and might be time consuming if applied to evaluation of alternatives at all levels during design.
- The participant involved in the design and manufacture of large machines in the food industry mentioned that success will depend on user friendliness and convenience and the ability to structure the design process where beneficial. A project template that is relevant to the company is important (i.e. tailorability).
- One participant with a systems engineering background thought that it was effective without being complicated and can handle large system structures with the various concepts. The person liked the idea that engineers monitor the process in real-time while working from different locations.

- Another participant involved in systems engineering believes it currently presents a good framework, illustrating the concept and intention of the software, and it would already be a good tool to capture design information, but would require a measure of configuration management to be useable in industry.

The following suggestions for future development of DiDeas II were made:

- More focus should be placed on the design criteria concept evaluation table and the software should enhance the selection process more as the selection of the best concept is the ultimate goal of the process.
- Project or context information should be displayed in a separate window on screen to identify which project is open.
- Linking DiDeas II to project scheduling software, and getting progress information back from it, may be very useful. During the case study, the engineer at the product development company identified the need to have information such as the level of completion of a sub-system or the supplier details for a component displayed in the tree-structure to allow quick access to information.
- Concerns were raised over the access rights to data. In addition to the locking of data in use, the desire to allocate access rights to users was mentioned. One participant mentioned that the lack of configuration management limits the real-time collaboration suitability. One participant mentioned during the interview that the company had trouble in the past with accessing information that was locked by a user after the person had left the company and a mechanism should be implemented to avoid this.

Certain shortcomings were mentioned in the questionnaire answers by the two participants that did not view DiDeas II before the demonstration for evaluation. Mention was made of the lack of configuration management (aside from locking nodes that are currently being edited). The system does, however, allow for the ability to develop various alternatives for each element in the system structure, providing the ability to record all the system configurations that were investigated. A mention was also made that a template relevant to the specific company would be important. This might indicate that the project template used for demonstration was not tailored correctly to the company's needs or that the participant did not understand the degree to which DiDeas II can be tailored.

The participants work in different industries with design teams that are structured differently. They all have differing needs for a design aid, which was also evident in the different functionality and aspects of DiDeas II on which each participant focussed during the evaluation.

During the interviews (including the initial interviews as described in Section 2.4) it was noted that users are sceptical about the use of formal design methods. The feedback from participants that a more formal or rigorous concept evaluation procedure is desired is therefore surprising, yet can be interpreted that users might be willing to use such functionality if it is available to them.

The concerns display the need in industry to have data access schemes for different team members. These access schemes are also to be customisable according to the needs of each team.

## 5 Conclusions

DiDeas II is a software tool that is focussed on supporting the early stages of mechanical engineering design (up to and including the concept evaluation phase) and facilitating communication among team members that are geographically dispersed and working asynchronously. The objective of this thesis was to improve DiDeas II and to evaluate the ability of the improved version to support design teams.

The user needs for the design support tool were identified via a survey of literature, a web survey and interviews to better understand the design and communication habits of engineers in industry, and through a survey of existing software solutions. In response to the results of the needs analysis, inter-team member communication functionality was added to DiDeas II and the customisability of the user interface and data structure was expanded. After considering different networking architectures, the client-server architecture was retained; however it was re-implemented during this research to allow for the desired functionality including instant messaging and WYSIWIS (see Section 2.1.2) elements within the user interface.

DiDeas II is not a commercial product, but a research platform. It was therefore developed to a level where it could be tested in academic case studies and demonstrated to engineers in industry. DiDeas II was tested in an academic environment in two case studies, a synchronous design study and an asynchronous design study. Its overall functionality and tailorability was also demonstrated to engineers in industry.

The academic case studies showed that DiDeas II is effective in capturing “soft” information during the early part of the design process, while providing context for the information that was captured and also providing some insight into the reasoning behind design decisions for future reference. The asynchronous communication elements of DiDeas II were more successful at facilitating communication among users than the synchronous elements. DiDeas II's user interfaces proved to be fairly intuitive to designers. Users who used the software for a second case study were already familiar with the interface and were able to use the system more efficiently.

DiDeas II could be tailored to meet most requirements from engineers in industry, but they were somewhat sceptical about using design support tools that might restrict them during the design process, introduce extra work or be time consuming to use. Many engineers in industry do not use formal design methods and the usefulness of DiDeas II for such individuals would lie more in file sharing and communication, and providing a means to record design information, including design decisions, in a single place for future reference. Engineers in industry also expected automation of some features, such as totalling weighting concept

evaluation scores, but DiDeas II does not provide for that functionality in its current form.

Since DiDeas II is primarily a research tool, there are numerous improvements that can be considered. Ideally, DiDeas II should be tested more extensively by engineers in industry, since the feedback above was based only on brief demonstrations that focussed mainly on the usefulness of DiDeas II. Testing in a real-life design environment and feedback from engineers after lengthier exposure to the system would provide more insights. This includes allowing participants to create project templates and testing how quickly new users can learn this process. To develop DiDeas II to the level where it can be used in industry, will require considerable effort. For example, the self-implemented data storage will have to be replaced with more efficient databases, e.g. MySQL, for increased flexibility and search efficiency. Database software will also aid in providing role-based access control. The data access control implemented during this research followed the basic *Lock-modify-unlock* model, whereby other users are denied access to a part of a file whilst it is being edited. To provide for greater efficiency during collaboration, the implementation of version control which follows the *Copy-modify-merge* model should be investigated.

By integrating DiDeas II with software that users are more familiar with, such as commercial packages, might improve their willingness to use DiDeas II. This might allow for the capture of information from software packages that perform functions which DiDeas II does not provide for as well as functions provided for by DiDeas II. One such example is the integration of project management software such as Microsoft Project for scheduling (which was mentioned by one of the participants in the industrial case studies).

Industry users would also expect commonly available user interface features, such as "drag and drop" functionality, that is not presently available in DiDeas II. Considerable development effort will be required to provide an acceptable user interface.

There are also further research-oriented developments that can be considered. For example, making use of web services (interfaces that are platform independent and available from anywhere) will allow users to access project data from anywhere and provide greater mobility. The HTML interface of DiDeas I provided limited functionality, but the currently available technology for web interface development, however, provides rich content that extend the capabilities of the web interface.

Since functional modelling is a widely used methodology, DiDeas II should be expanded to provide appropriate support for those activities. Also, users wished for a sketching tool within DiDeas II or for the ability to display sketches in the chat window. Attaching sketches as files to the project provides an asynchronous communication method. A collaborative sketching facility might provide an additional means for synchronous communication.

## 6 References

- Oracle Security Overview 10g Release 1 (10.1). (2003, December). Retrieved December 2, 2012, from docs.oracle.com: [http://docs.oracle.com/cd/B12037\\_01/network.101/b10777.pdf](http://docs.oracle.com/cd/B12037_01/network.101/b10777.pdf)
- "groupware". (2012). Retrieved 11 5, 2012, from merriam-webster.com: <http://www.merriam-webster.com/dictionary/groupware>
- Typical Collaborative Software Applications. (2012). Retrieved November 29, 2012, from Usability First: <http://www.usabilityfirst.com/about-usability/web-application-design/collaborative-software-groupware/typical-collaborative-software-applications/>
- Altshuller, G. (2002). *40 Principles: Triz Keys to Technical Innovation*. Worcester: Technical Innovation Center, Inc.
- Baecker, R. M. (1993). *Readings in Groupware and Computer-Supported Cooperative Work: Assisting Human-Human Collaboration*. San Francisco: Morgan Kaufman Publishers.
- Baset, S. A., & Schulzrinne, H. (2004, September 15). *An Analysis of the Skype Peer-to-Peer Internet Telephony*. New York: Columbia University.
- Basson, A. H., Bonnema, G. M., & Liu, Y. (2004). A Flexible Electro-Mechanical Design Information System. *Proceedings of the Fifth International Symposium on Tools and Methods of Competitive Engineering TMCE 2004*, (pp. 879-900). Lausanne, Switzerland.
- Blanchard, W. J., & Fabrycky, B. S. (2006). *Systems Engineering and Analysis, Fourth Edition*. New Jersey: Pearson Prentice Hall.
- Blessing, L. T., & Chakrabarti, A. (2009). *DRM, A Design Research Methodology*. London: Springer.
- Chandrasekaran, B., Josephson, J. R., & Benjamins, V. R. (1999, January/February). What Are Ontologies and Why do We Need Them? *IEEE Intelligent Systems*, pp. 20-26.
- Chang, X., Sahin, A., & Terpenney, J. (2008). An Ontology-based Support for Product Conceptual Design. *Robotics and Computer-Integrated Manufacturing*, 755–762.
- Cross, N. (2000). *Engineering Design Methods: Strategies for Product Design*. West Sussex: John Wiley & Sons.
- Cross, N. (2006). *Engineering Design Methods, Strategies for Product Design, Third Edition*. West Sussex: John Wiley & Sons, Ltd.
- Ferraiolo, D., Cugini, J., & Kuhn, D. R. (1995). Role-based Access Control (RBAC): Features and Motivations. *Proceedings of the 11th Annual Computer Security Application Conference*, (pp. 241-248).
- Greenberg, S. (1991). Computer supported cooperative work and groupware: An introduction to the special edition. *International Journal of Man Machine Studies*, 34(2), pp. 133-143.
- Heer, J., Viégas, F. B., & Wattenberg, M. (2007). Voyagers and Voyeurs: Supporting Asynchronous Collaborative Information Visualization. *CHI*.

- Johansen, R. (1988). *Groupware: Computer Support for Business Teams*. Free Press.
- Kamrami, A. K., & Nasr, E. A. (2010). *Collaborative Engineering Theory and Practice*. New York: Springer.
- Liu, Y. (2007). *A Flexible Distributed Design Assistance*. Stellenbosch: Stellenbosch University.
- Liu, Y., & Basson, A. H. (2007). An Ontology Based Approach to a Flexible Aid For Mechanical Conceptual Design. *ICED07: 16th International Conference of Engineering Design*, (pp. 565 - 566). Paris, France.
- Louridas, P. (2006). Version Control. *IEEE Software* 23.1, 104-107.
- Microsoft Corporation. (2000, October). *Usability in Software Design*. Retrieved 11 10, 2012, from msdn.microsoft.com: <http://msdn.microsoft.com/en-us/library/ms997577.aspx>
- Miltchev, S., Smith, J. M., Prevelakis, V., Keromytis, A., & Ioannidis, S. (2008). Decentralized Access Control in Networked File Systems. *ACM Computing Surveys (CSUR)* 40.3, 1-30.
- Mørch, A. I. (1997). Three Levels of End-User Tailoring: Customization, Integration, and Extension. In *Computers and Design in Context* (pp. 51-76). Cambridge: MIT Press.
- Neches, R., Fikes, R., Tim Finin, Gruber, T., Patil, R., Senator, T., et al. (1991). Enabling Technology for Knowledge Sharing. *AI Magazine* 12.3, 36-56.
- Nielsen, J. (2012, January 4). *Usability 101: Introduction to Usability*. Retrieved 11 10, 2012, from Jakob Nielsen's Alertbox: <http://www.useit.com/alertbox/20030825.html>
- Pahl, G., & Beitz, W. (2007). *Engineering Design: A Systematic Approach*. Springer.
- Priebe, T., Wolfgang, D., & Kamprath, N. (2006). Supporting attribute-based access control with ontologies. *The First International Conference on Availability, Reliability and Security*. IEEE.
- Sandhu, R. S., & Samarati, P. (1994, September). Access Control: Principles and Practice. *IEEE Communications Magazine* 32.9, 40-48.
- Schlichter, J. H., & Borghoff, U. M. (2000). *Computer-Supported Cooperative Work: Introduction to Distributed Applications*. Springer.
- Schueller, A. (2002). *Aspects of Distributed Conceptual Design Support*. Stellenbosch: Stellenbosch University.
- Slagter, R., Biermans, M., & ter Hofte, H. (2001). Evolution in Use of Groupware: Facilitating Tailoring to the Extreme. *Proceedings of the Seventh International Workshop on Groupware* (pp. 68-73). Washington D.C.: IEEE.
- Sowa, J. F. (2008). Conceptual Graphs. In F. van Harmelen, V. Lifschitz, & B. Porter, *Handbook of Knowledge Representation* (pp. 213-237). Amsterdam: Elsevier.
- Spinellis, D. (2005). *Version Control Talk Demystified*. Retrieved 12 02, 2012, from Diomidis D. Spinellis home page: <http://www.spinellis.gr/tot/vcs-glossary.html>
- Stiemerling, O., Kahler, H., & Wulf, V. (1997). How to Make Software Softer - Designing Tailorable Applications. *DIS '97 Proceedings of the 2nd conference on Designing interactive systems: processes, practices, methods, and techniques* (pp. 365-376). New York: ACM.

- Suh, N. P. (2001). *Axiomatic Design: Advances and Principles*. New York: Oxford University Press.
- Tolone, W., Ahn, G.-J., Pai, T., & Hong, S.-P. (2005). Access control in collaborative systems. *ACM Computing Surveys (CSUR)* 37.1, 29-41.
- Ullman, D. G. (2003). *The Mechanical Design Process, Third Edition*. New York: McGraw-Hill.
- Verma, D. C. (2004). *Legitimate Applications of Peer-to-Peer Networks*. John Wiley & Sons, Inc.
- Walthall, C., Sauter, C., Deigendesch, T., Devanathan, S., Albers, A., & Ramani, K. (2009). Survey of Wikis as a Design Support Tool. *International Conference on Engineering Design*, (pp. 6:211-222). Stanford.
- Wang, L., Shen, W., Xie, H., Neelamkavil, J., & Pardasani, A. (2002). Collaborative Conceptual Design - State of the Art and Future Trends. *Computer-Aided Design*, 981-996.
- Wilson, P. (1991). *Computer Supported Collaborative Work: An Introduction*. Springer.
- Yang, B., & Garcia-Molina, H. (2001). Comparing Hybrid Peer-to-Peer Systems. *Proceedings of the 27th VLDB Conference*. Rome.

## Appendix A: Web-Survey

This web-based survey was used to gather information on the design and communication habits of engineers in industry. The survey was created via the Stellenbosch University survey service (<https://surveys.sun.ac.za/>) which implements the Checkbox® v4.7 tool. The survey is presented here in its original form and consists of four sections namely Business Context, Communication, Documentation and Design, each of which is displayed on a separate web-page.

---

**Thank you for taking part in this survey. It is part of Ruan van der Merwe's thesis research in the MScEng Mechanical programme at the Department of Mechanical and Mechatronic Engineering of Stellenbosch University, under supervision of Prof Anton Basson. His work contributes to research into tools to support communication in design teams.**

**This questionnaire focuses on the designers of mechanical/mechatronic systems, particularly their preferred communication patterns, design methods and documentation methods.**

**We are specifically interested in the early phases of design, i.e. up to and including concept selection and layout design, but before detailed modelling and manufacturing drawings.**

**Please answer the questions from your experience in the past six months.**

**The questionnaire is divided into four sections namely:**

- **Business Context**
- **Communication**
- **Documentation**
- **Design**

**Each of these sections are displayed on a separate page. After answering all questions on a given page, click on the next button to proceed. All questions are to be answered before you are able to proceed to the next section.**

**Please Note: All responses will be gathered anonymously.**

**Click next to proceed.**

Page 1 of 4

**Please note: All questions are to be completed before commencing to the next section. Certain questions are to be answered in a desired format before commencing to the next page.**

### **Business Context**

\*1. How many people are employed at your company?

\*2. How many years of experience as a design engineer do you possess?

\*3. What position in the company do you hold (your job title)?

\*4. Please list your post-high school qualifications, the year in which each was obtained and the institution from which each was obtained:

Qualification Subject(e.g. Mechanical Engineering) Year Institution

1

2

3

4

5

\*6. How many design projects are you typically involved with at any one instant?

\*7. How many design projects do you work on in a typical day?

\*8. What percentage of your work is focused on technical aspects of design/development, as opposed to management-related work and work not related to product development?

\*9. How much of your design time (in your current position) do you typically spend on each of the following forms of design (7: most of the time; 0:never):

	0	1	2	3	4	5	6	7
Original design (no similar product exists)	<input type="radio"/>							
Designs are new to your section/company	<input type="radio"/>							
Re-design/improvement of your company's own products	<input type="radio"/>							
Re-design/improvement of another company's products	<input type="radio"/>							
Configuration design (primarily putting together previously existing modules/components)	<input type="radio"/>							
Design of subsystems/components for systems being developed by other engineers	<input type="radio"/>							
Designs where your client is the end-user or retailer of the product	<input type="radio"/>							
Design of equipment to be used in your company	<input type="radio"/>							

**For the following questions, please consider your work in the early phases of design/development projects, specifically technical aspects that precede the detailed design phase, e.g. formulating/changing the requirements/specifications, considering feasibility, and formulating/evaluating/selecting concepts or alternatives (i.e. before detailed drawings and CAD models of each component).**

**Communication**

\*10. How frequently do you use the following to communicate about the early phases of design/development projects?

	*E-mail	*Fax	*Phone	*Memoranda and Reports	*Previously arranged face to face meetings	*Ad hoc or impromptu face to face meetings	*Skype, Video Conferencing	Other (please specify)	Other (If specified)
With design team members within your section of the company									
With other teams outside your section/company									
With clients									

\*12. During a design project at your company, how are user requirements typically communicated to various members of the design team (list the methods used e.g. team meetings, formal documents etc.)

\*13. How frequently is progress on the design project communicated to people outside of the design team that are involved in the project?

\*14. How is progress on the design project communicated to people outside of the design team that are involved in the project?

\*15. How frequently is progress communicated among team members?

\*16. How is progress communicated among team members?

\*17. What means are there within your company whereby team members can view project progress on demand (e.g. none, repository etc)?

Page 3 of 4

### Documentation

\*18. How do you store the following forms of communication?  
(E.g not stored, kept in personal design journal, kept in a physical project file, scanned and stored electronically with/without attaching metadata such as project information, etc.)

Free-hand sketches	<input type="text"/>
Notes	<input type="text"/>
Phone Conversations	<input type="text"/>
E-mails	<input type="text"/>
Documentation of design decisions (e.g. specification changes, reasons for selecting a particular concept, etc.)	<input type="text"/>

\*19. How often is research information (e.g. information from catalogues, web-sites, articles etc.) that was gathered during preceding projects, re-used in future projects?

\*20. How is research information (e.g. information from catalogues, web-sites, articles etc.) that was gathered during projects, archived for future use?

- In a design/project file (on paper)
- Electronically
- Both electronically and in a design/project file
- Other, please specify

\*21. Do you use software for the documentation of projects (e.g. database management systems etc.)?

- Yes
- No

Page 4 of 4

## Design Methods

\*22. Which CAD software do you use for layout design (note: before detailed part design)?

- None
- Catia
- Pro-Engineer
- SolidEdge
- SolidWorks
- Inventor
- AutoCAD
- Unigraphics/ Siemens NX
- Other, please specify

\*23. How often are the following sources of information used during a typical design project?

- In-house catalogue library
- Personal catalogue library
- Web searches
- Websites of component/subsystem supplier companies
- Visits to/by supplier company representatives
- Text books
- In-house design procedure documents
- Other

\*24. If other selected in the above list, please specify:

\*25. Do you make use of project planning software during the early phases of design (e.g. Microsoft Project)?

- Yes
- No

\*26. If so, which of the following software features do you use?

- Checklist
- Gantt chart
- Work breakdown structure
- Resource Allocation
- Other, please specify

\*27. Do you use any formal methods of concept generation (e.g. brainstorming meetings, patent searches, TRIZ, axiomatic design, systems engineering, etc)?

- Yes
- No

\*28. If so, name or describe the methods used:

\*29. Do you use any formal methods of concept evaluation and elimination (e.g. Pugh's method, comparison matrices, etc.)?

- Yes
- No

\*30. If so, name or describe the methods used:

## **Appendix B: Synchronous Case Study Design Brief**

This design brief was provided to the participants of the synchronous case studies.

### **Design Brief**

#### **Conceptual design for a singulation unit for a reconfigurable assembly system**

#### **Background**

A singulation unit in an assembly system has to receive parts in bulk (typically "dumped" in a bin by an operator), and deliver the parts one by one to a pick-and-place robot. Ideally, the robot should never have to wait for a part that can be picked up. In practice this means that the time between having two consecutive parts ready for collection should be of the order of 1 to 2 sec. Vibratory bowl feeders are often used in industry for this purpose, but they have the drawback that each feeder is designed to feed one specific part shape.

In reconfigurable assembly systems (RASs), the objective is to design the system to be able to handle a range of products, rather than just one product. A singulation unit for an RAS should therefore also be able to handle a range of part shapes, and not only one. Vibratory bowl feeders are therefore not suitable.

We require a singulation unit that will present singulated parts on a platform to be picked up by a 6DOF articulated arm robot (Fig 2 and Fig 3). A suitable gripper (that will often be part-specific) will be attached to the robot (at axis 6 in Fig 2). A camera will be used to determine the pose of the part on the platform and the exact position of the part. The control system will then instruct the robot to approach the part from the appropriate direction and pick up the part. The robot can handle any approach direction in the upper hemisphere (relative to the presentation platform), but there must be no obstructions in the approach direction.

Since a camera and 6 DOF robot is used, the singulator does not have to orientate the part (in contrast to what vibratory bowl feeders do). However, not all poses presented to the robot will be collectable. If a pose is not collectable, the singulation unit has to take back the part and try again (with the same or a next part). If any approach direction for the robot is blocked by the singulation unit, it reduces the likelihood that the part will be in a collectable pose.

A singulation unit will be loaded with a specific part type at a given time. However, one must be able to manually empty the singulation unit and to use the same singulation unit for another part at another time.

## Brief

Produce a concept design of a reconfigurable singulation unit, including the mechanical parts layout and the actuator type selection. No detailed calculations are required, but in the assessment of concepts, take into account throughput rate (how many parts per minute will be singulated, taking into account the likelihood that the part will be in a collectable pose), cost, complexity of control (number of sensors and actuators; level of control required, e.g. open loop, bang-bang, PID), reliability, maintainability, and ability to work with a range of part of sizes between 15 and 75 mm.

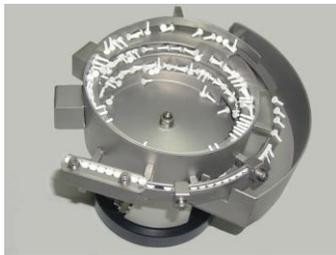


Fig 1: Vibratory bowl feeder

[\[http://en.wikipedia.org/wiki/File:-Bowl\\_Feeder.jpg\]](http://en.wikipedia.org/wiki/File:-Bowl_Feeder.jpg)

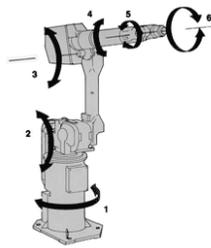


Fig 2: 6 DOF articulated arm robot

[\[http://www.robots.com/faq/28/what-are-six-axis-robots\]](http://www.robots.com/faq/28/what-are-six-axis-robots)

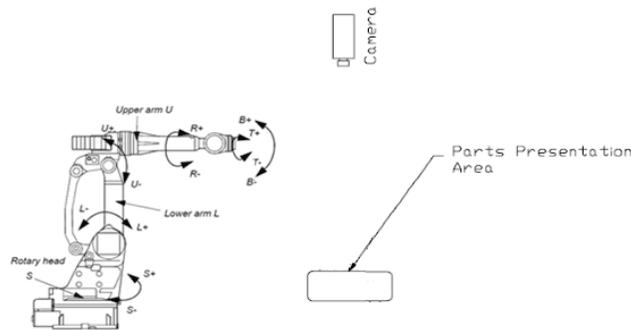


Fig 3: Surrounding subsystems

## Appendix C: Asynchronous Case Study Design Brief

This design brief was provided to the participants of the asynchronous case study.

### Design Brief

## Conceptual Design of the Product Architecture for a Modular Self-Actuated Transport System for Differing Flights of Stairs

### Background

Two scenarios are relevant here:

1. Wheelchair access: many older buildings at, for example, the University, were designed without wheelchair access in mind. The Dept Mechanical and Mechatronic Engineering's laboratory area is a good example, where there is no wheel-chair access to the Mechatronics and Material Science laboratories. There is therefore a need to fit "wheel chair lifts" in various places.
2. The emphasis on health and safety of workers has increased dramatically in the past decade. Where it was previously acceptable to let workers carry heavy loads up and down stairs (with the corresponding risks of injury or falling), employers now have to provide safer alternatives. A "trolley lift" could be useful in these situations.

In addition to the usual considerations, the space occupied by the system is a significant consideration since the lifts have to be installed in existing passages, where sufficient access for other users must still be provided. Various designs are currently available on the market. An important distinguishing feature is whether the path followed is in a straight line (with or without intermittent landings), composed of straights plus curves, or like a spiral staircase.

For the design considered here, the design team should choose the capabilities for their target system and should not try to satisfy all possible requirements. However, the system must be at least suitable for conveying persons in wheelchairs and trolleys of equipment (target weight can be chosen by team) to and from the Mechatronics Laboratory. The design must, however, be suitable for sale to other potential clients too.

### Brief

Develop the architecture for a suitable transportation system. This includes identifying the main subsystems and principles of operation, as well as the requirements that each subsystem must meet. It is more important that the team formulates a complete architecture within the allocated time, rather than considering a very large number of alternatives.

The system must be suitable to be implemented in existing buildings to allow for the transport of people, equipment or other goods along a flight of stairs. The system will be sold in different configurations according to the client's needs and must therefore be modular.

The design should allow the system to be easily configured and installed with minimal re-design.

As the system will be carrying passengers or operating in an environment where it poses a risk to people, the necessary safety precautions should be implemented in the system.

The load to be carried by the system will vary and the team can select a suitable upper limit. Additional considerations regarding the load to be carried are also to be taken into consideration for example the comfort of a passenger.

The system should be able to wheelchair passenger to a safe position in the event of a power failure.

The system should be controlled by either a passenger or from the edge of the staircase.



**Figure 1: Domestic Stairlift**

[\[http://stairliftstop.com/concord-stairlift/index.html\]](http://stairliftstop.com/concord-stairlift/index.html)



**Figure 2: Public Stairlift Platform**

[\[http://www.stairliftsbuyerguide.com/stair-chair-lift-7\]](http://www.stairliftsbuyerguide.com/stair-chair-lift-7)

## Appendix D: Student Questionnaire

Each participant completed this questionnaire at the end of each of the academic case studies. Participants 1-4 took part in the first synchronous case study, while participants 5-8 took part in the second case study. Two participants from each of these case studies took part in the asynchronous case study and their answers are indicated by numbers 9-12.

### DiDeas II Design Evaluation Questionnaire

Name: \_\_\_\_\_

E-Mail: \_\_\_\_\_

Please answer all the questions in the questionnaire.

Background Information (personal information will be kept confidential)

#### Of which nationality are you?

Case Study 1:

1. Zambia
2. South Africa
3. Libya
4. South Africa

Case Study 2:

5. RSA
6. RSA
7. RSA
8. RSA

#### Where and when did you (or do expect to) complete your undergraduate studies?

1. Copperbelt University, 2008
2. University of Pretoria
3. Al Fateh University
4. Stellenbosch

5. Stellenbosch University
6. Stellenbosch, 2007
7. Stellenbosch
8. Stellenbosch

#### In which discipline (e.g. mechanical/mechatronics) was/are your undergraduate studies?

1. Electro-Mechanical

- 2. Mechanical
- 3. Mechanical
- 4. Mechanical
- 5. Mechanical
- 6. Mechatronics
- 7. Mechanical
- 8. Mechanical

**How many years or months of design experience do you have?**

- 1. two years
- 2. two years
- 3. three years
- 4. blank
  
- 5. one year in total
- 6. six years
- 7. four weeks
- 8. four years

**Which of the following design procedures are you familiar with?**

	<i>Unknown to me</i>	<i>Heard about it</i>	<i>Had training in it</i>	<i>Used it</i>
<i>Ullman's procedure</i>	2,3,4	7	6	5,8
<i>Blanchard and Fabrycky's systems engineering</i>	5,6,8	3	4	2,7
<i>Cross' procedure</i>	2,4,6,8	3,5		7
<i>Pahl and Beitz's procedure</i>	2,4,5,6,7,8	3		
<i>Other</i>	5			
<i>None</i>				

**How much have you used the following design methodologies in the past?**

	<i>Never</i>	<i>Seldom</i>	<i>Occasionally</i>	<i>Regularly</i>
<i>Ullman</i>	2,4,6,7	5 8		
<i>Blanchard and Fabrycky</i>	5,6,8		2,4,7	

<i>Cross</i>	2,4,5,6,8			7
<i>Pahl and Beitz</i>	2,4,5,6,7,8			
<i>Other</i>	2,5	1		
<i>None</i>	3			
<i>If other, please specify: 1. QFD</i>				

### Overall assessment

**Was DiDeas II an effective means for design team members to communicate with each other?**

1. Yes
2. Yes, because of communication on multiple levels.
3. Yes, very effective
4. Yes, the file sharing, design tree and discussions are excellent.
5. The chat was good, but a little bit slow. Something like Skype would probably work better. However I did notice how easy it was to organize oneself and everyone had their say.
6. No
7. The right tool are there, but the messenger is too slow for real-time communication.
8. No
9. Yes, during the conceptual design process the discussions were particularly helpful.
10. Yes, following other members' comments was simple.
11. Yes, for the communication of ideas, but the chat/discussion communication are time consuming.
12. Yes, using the discussion tab.

**Was DiDeas II an effective means for design team members to record their design work?**

1. Yes
2. Yes
3. Yes, very effective
4. Yes
5. It was effective. The file upload function and tree on the left hand side is a good idea.
6. Moderately so. In concept it works, but due to the difficult user interface, it's difficult to do so.
7. Yes, it provides the structure for conceptual design.
8. Yes

9. Yes
10. Yes, any type of file is easily saved and viewed.
11. Yes, structuring data is useful.
12. Yes

**Was DiDeas II an effective means for design team members to generate concepts?**

1. Yes. However, it requires someone who has experience in design procedures.
2. Yes
3. Yes, with the contribution of the members.
4. Yes. However more emphasis could be placed on discussion and analysis.
  
5. In 3 hours no, but maybe for an extended period of time (typical to industry), yes.
6. Due to the lacking communication system, no.
7. To a lesser extent since drawings must be done in a separate program.
8. Yes
  
9. Yes, quite so, since one can add various file formats to include drawings.
10. Yes, creating sketches and sharing them is simple and useful.
11. Yes, concepts are generated because problems are highlighted.
12. Yes

**Was DiDeas II an effective means for design team members to make design decisions?**

1. Yes. It gave flexibility in terms of looking at what others were doing.
2. Its slow, but might be the best way for long distance decision making.
3. Yes, and to get other team members' notes.
4. Not in a short time span.
  
5. Yes. Especially because it could be discussed.
6. I am not a big fan of the QFD or house of qualities process (Ullman, etc), therefore my answer is biased, no.
7. Yes. The concept evaluation works well.
8. Yes
  
9. Yes, well structured evaluation processes.
10. I believe that decisions are better made in real time conversation, but the tables and stored files help with this.
11. Yes, but the communication is hard to get to an idea.
12. Yes

## **Meta Information Context and Information Transfer**

### **File transfer**

**How many times during the design process did you upload files that you created to the server?**

1. Three times
2. Six times
3. Two files
4. Two
  
5. Two times
6. Two times
7. Four or five times
8. Two times
  
9. Five or so
10. Many
11. Six
12. Six to ten

**How many times during the design process did you edit files that had been uploaded to the server by others?**

1. None
2. None
3. None
4. Zero
  
5. None, but it might have happened with more time
6. Zero
7. Never
8. Zero
  
9. One time
10. Whenever someone uploaded a new file.
11. Zero
12. Never

**Discussions associated with nodes in the left-hand tree (sub-system, part, etc.)**

**How often did you read the discussions of a specific node?**

1. Five times
2. None (there was none).
3. We only discussed using chat window.
4. As often as I was alerted of an update or was analysing a concept.
  
5. There was none.
6. Often
7. Never, but we discussed in the messenger.
8. Zero
  
9. Every time I worked at a node I read the discussion.

10. Every time before commencing work.
11. Always
12. Always

**What value did the discussion posts have for you?**

1. Yes, I benefitted from the discussion and the popping up of the chat made it easy to check others' comments during the discussion.
2. None to me, but I did use it.
- 3.
4. Other team members did not use them. However I used them and believe they are invaluable.
5. /
6. Quite a bit
7. It would have been helpful, had we used it.
8. Zero
9. It coordinated the work done between the team members, and brought one up to speed at the start of each working period.
10. It helped me follow what others were doing.
11. Good to get everyone on the same page.
12. Insight into decision making.

**Node locking (locking prevents more than one user from editing the same node)**

**How often did you try to access a node that was locked?**

1. Once
2. 30% of the time
3. Seldom
4. Seldom
5. Twice
6. All the time
7. Quite often, but that is due to the short time frame.
8. Five
9. No - asynchronous test case.
10. Never
11. Zero
12. Never

**How often did you wish to locate a node for viewing or editing after it had been deleted by another team member?**

1. None
2. None
3. None
4. Never

- 5. None
- 6. None
- 7. Never
- 8. Zero

- 9. Never
- 10. Never
- 11. Zero
- 12. Never

**Did it occur that information that you saved, was later overwritten by another team member?**

- 1. No
- 2. Yes
- 3. No
- 4. Discussions seem to get overwritten.

- 5. No
- 6. No
- 7. No
- 8. Yes, but it was intentional.
- 9. No
- 10. No, an updated copy of the sketch was uploaded.
- 11. Zero
- 12. Never

**If so, please specify all relevant types of information:**

Information in an attached file: 12: (“by asynch se *pictures* so nvt”)

Related files that were missing from the list: 10

Table headers that were missing or changed: 11

Table data that was changed or missing:

A post in the discussion of a node that was missing or overwritten: 10

Other: \_\_\_\_\_

Comments:

6: I think the process will go much easier if team mates can see edits made in real-time. A web based program might work well.

9: The “save and release” system worked perfectly in the end.

## **Messaging**

**Did you ever during the design process find the appearance of the DiDeas Messenger window to be disruptive?**

- 1. No, it was easy for me to check what has come up.
- 2. Yes
- 3. No

4. No, I would prefer it to appear when receiving a new message.
5. Not really, but it would be neat if it was docked.
6. Yes
7. Yes, but the disruption was not excessive.
8. No, I did not.
  
9. No
10. blank
11. blank
12. Never

**Was the DiDeas Messenger window effective to facilitate communication?**

1. Yes, it was.
2. Yes
3. It was very quick and useful, especially when all users are online.
4. Yes
  
5. Yes
6. No
7. Yes, but it was too slow for real-time communication.
8. Yes, but slow.
  
9. We didn't use it at all.
10. blank
11. blank
12. Never used it.

**Would you prefer to use other communication methods such as e-mail?**

1. No
2. No
3. Using the Discussion tab of DiDeas is much better than e-mails; it keeps all the info organized well.
4. I think e-mail could be used in addition to the messenger.
  
5. Maybe voice communication.
6. Skype
7. E-mail has its place, but the messenger has value.
8. Skype (voice call).

**If so, please explain why:**

1. Nice
2. I often find it useful to include photos and equations in e-mails.
3. blank
4. blank

5. I think the process would go faster. However, once again the project was crammed into 3 hours. Given more time, the text would most probably be sufficient.
6. Skype is familiar, and I believe it's responsiveness is much better. It makes sound, distinguish users, messages are instant, and it is possible to instantly share files. If this was combined with dropbox, it could be awesome.
7. One isn't able to post long messages on the messenger, or messages with text attached.
8. When pressed for time, it would be better to "voice" communicate. The chat works better when enquiring, not for meetings or brainstorming.

## **Sketch input devices**

**Which input method for sketching was assigned to you?**

**Wacom graphics tablet / iPad / Galaxy Tablet**

1. iPad
2. iPad
3. Wacom
4. Wacom
  
5. Mouse and Sketchbook Pro
6. Wacom graphics tablet
7. Wacom
8. Galaxy Tablet
9. Galaxy Tablet
10. Galaxy Tablet
11. Galaxy Tablet
12. Galaxy Tablet

**How long did it take to become accustomed to this method?**

1. It took a short time. But the quality of my sketches was not good.
2. It was easy.
3. Around 30 minutes.
4. I gave up on using it.
  
5. Short (15 minutes)
6. Instant
7. One hour
8. ± 30 minutes
  
9. 30 minutes or so
10. Not long
11. One minute
12. Quickly (5 minutes)

**How effective did you find this input method to record your own sketches?**

1. It was very effective in terms of the sketch quality but was convenient when sending the sketches.

2. Highly effective.
3. It was sufficient enough for sketches.
4. I preferred using the mouse.
  
5. Not effective. Stylus would be better. Pad even more.
6. I find Inventor way better, but only because I have no drawing talent at all. A mouse is still better.
7. Less effective than a mouse.
8. 70%
  
9. It is fairly intuitive, but it is difficult to make fine detail drawings and write annotations.
10. It was very effective in producing relatively simple sketches.
11. Good
12. Difficult to draw and write with finger.

**Could you convey the same information as with a sketch using pencil and paper?**

1. I think the information can be conveyed. However, the quality of sketch may depend on how long one has used the iPad.
2. It might be
3. Yes
4. Yes
  
5. No, but with practice one might.
6. No, pencil is 10x better.
7. Not quite - the scaling from the pad to the screen was too un-intuitive.
8. Yes, struggled with text.
  
9. See above comment
10. No, I can add more detail more easily with a pencil and paper.
11. Yes
12. No

**How many times did you edit/annotate a sketch uploaded by another team member?**

1. None
2. Never
3. Once
4. Never
  
5. None
6. None
7. Never
8. Zero

9. Once
10. Never
11. Zero
12. Never

**How effective did you find this input method to edit/annotate other team members' sketches?**

1. I didn't edit a team members' sketch because I was putting up my sketches.
2. /
3. It was enough to do the required job.
4. Not used

5. /
6. N/a
7. /
8. /

9. It's just too difficult to make fine detail sketches
10. N/a
11. /
12. /

## **Understanding**

### **Tree view in the left-hand pane**

#### **What value did the tree view have for you during the design process?**

1. It showed how many sub-systems made up a system and what concepts have been developed for a sub-system.
2. High value
- 3.
4. Separating the system meaningfully.
  
5. It was good to have the layout neatly. Good function.
6. Separating design aspects.
7. It worked well.
8. Brought structure to environment.
  
9. It kept every piece of info organized and available.
10. Gave a good overview of the system and sub-systems.
11. Good
12. Showed structure of project.

#### **Did the tree view clarify the system structure?**

1. Yes, it did.
2. Yes
3. Yes
4. Yes, definitely.

5. Yes
6. Only to a personal extent.
7. Yes
8. Yes

9. Yes
10. Yes
11. Yes
12. Yes

**Did the tree nodes provide appropriate information? (please elaborate)**

1. Yes. A system was made up of sub-systems and sub-systems were designed on different concepts.
2. Yes
3. blank
4. Yes
  
5. Yes. The name of designer and unique no. is good for referencing.
6. Yes. The information is constantly carried through.
7. Yes, all helpful info.
8. /
9. Yes
10. Yes, however I don't quite see the use in displaying the unique number of each node.
11. Yes, good visual summary.
12. Yes (showed sub-systems).

**Tabs**

**To what extent did the tab pages in the right-hand panel help you to find the information you were looking for?**

1. \*The words used were self-explanatory and familiar.
2. It is clear and simple, good.
3. Tabs were very helpful and info easy to find.
4. Each tab did exactly what I expected and didn't find anything lacking.
  
5. The structure was neat and effective.
6. All the time (there is not much confusion).
7. Once I had figured out how the structure works, they were quite helpful.
8. It took a little getting used to.
  
9. It worked very well.
10. They were very easy to use.
11. It can be hard to find a tab.
12. Hard, didn't know which tabs contained new information.

**To what extent did the tab pages in the right-hand panel help you to decide what to do during the design process?**

1. They were marked. However, short forms like specs vs. reqs should have been written in full.
2. I did not know all the functionality but what I did know was made easy.
3. Very much
  
4. I used them to see what still needed doing or checking.
5. Easy access helped to obtain required info quickly.
6. None
7. They provided a guide as to what to do next.
8. The order of the tabs helped.
  
9. It reminds one of what still needs to be done in the design process.
10. The QFD process can be followed and completed in logical steps by following the tabs from left to right.
11. Fair
12. Needed an explanation on what the tabs were before I knew what to put in them.

**Tables**

**To what extent did the tables on the pages in the right-hand panel help you to find the information you were looking for?**

1. They were very visible and coloured and properly aligned.
2. It's a little bit confusing at first, but gets useful later.
3. Very helpful.
4. I didn't understand the requirements vs. specs tables.
  
5. Similar to previous answer.
6. It was difficult since there was nothing until I created it.
7. They worked reasonably well, except for not being able to scroll while someone else is editing.
8. It was easy, except for scrolling.
  
9. They worked fairly well - It's just a pity one can't see the whole table at once.
10. It is an easy way to locate design parameters such as power usage etc.
11. Not much.
12. Acceptable, scrolling while not in edit mode was an issue though.

**To what extent did the tables on the pages in the right-hand panel help you to communicate your design work during the design process?**

1. They were all on one page which made it easy to find what I wanted.
2. It will help a lot when we know what it means.
3. blank
4. I found the concept comparison table very useful.

5. Giving numbers to weight the different design “factors” is a bit difficult to interpret since it could not be linked to the person’s thoughts who placed the value.
6. None
7. They worked really well.
8. It was difficult as we were rushed.
  
9. The related files table helped for communicating design work, but not really the QFD tables
10. Filling in the tables are easy and understandable.
11. Fair
12. Well, when combined with a short discussion.

**Was it clear or confusing when the system requirements were transferred to subsystem level?**

1. It made sense. This gave me an understanding that from a system requirement, the sub-systems can be made. Also that bottom-up design is possible.
2. Clear
3. Little confusing
4. Confusing
  
5. Clear
6. Clear
7. I don’t think we reached that part.
8. Clear
  
9. Very clear
10. Clear
11. Clear
12. Clear (after a discussion on what they represented).

**Was it clear or confusing when concept properties were collated for concept comparison?**

1. Yes it was. I didn’t get it.
2. Clear
3. Clear
4. Clear
  
- 5 Clear
- 6 Clear
- 7 Very clear - work very well.
- 8 Clear
  
9. Clear
10. An indication of the weights and an automatic totalling and weighing function would be helpful. However, the tables were clear and easy to use.
11. Did not use it

12. Clear

**Was it clear or confusing when the subsystem specifications were displayed with both the higher level requirements and the target values?**

1. No. It kept the overall system requirement in perspective.
2. Clear
3. Clear
4. Confusing

5. Clear

6. Clear

7. /

8. Clear

9. Clear

10. Clear

11. Clear

12. Clear

**What information did you want to record, but could not or was difficult to record?**

1. blank
2. Concepts of the system.
3. Describing the concept.
4. None

5. Once again. Thoughts around the values in design factors would help make a better decision (Need for conferencing ability).

6. It is very hard to communicate concepts in this way. I feel a general sandbox approach with multiple levels of input could work.

7. /

8. When I am speaking to a specific person in chat.

9. Writing in drawings.

10. Better sketches. But these could easily be scanned and uploaded if given those facilities.

11. /

12. Nothing

## **Features**

**What features were lacking that you would have liked to use?**

1. Editing. If you make a mistake, you just have to remove. Inserting a column and rows need to be added.

2. To view sketches or files from a locked node.

3. /

4. I would like an update stream somewhere, telling me when people make edits to nodes.

Also I would like to be able to right-click and edit a node.

5. Conferencing ability. Being able to view tables while someone else is editing (especially when the table cells become too small for the amount of characters. "Enter" sends the message in "chat".

6. A general sketch pad next to the chat that you can draw pictures everyone can see, while you can talk via VoIP.

7. VoIP/ teleconference

8. VoIP

9. A "MAKE COFFEE" button :-)

10. I would like something that alerts me to updates made by other members, such as:

"Johan has added a sketch under...."

11. /

12. Markers showing "unread" information.

## **Ease of Use and Usability**

**Did you experience significant waiting periods after saving data to the server or downloading files from the server?**

1. No. It was fast.

2. Yes

3. Yes, slow computer.

4. No

5. None

6. Yes

7. The waiting was cumbersome, but only because the project time-scale was too short, otherwise acceptable.

8. Yes, but they are acceptable. Not for chat though.

9. No, it worked smoothly enough.

10. Yes, but those were rectified and were not a problem.

11. Yes

12. No

**Did you experience any aspects of the software that significantly hindered your design work? If so, please describe:**

1. No

2. The multiple windows.

3. Yes, locking of nodes.

4. Files uploaded to the server would often not appear in the client.

5. None
6. Yes. Slow execution made it hard to feel in control of the creative process. I have done this before using Skype and Dropbox.
- 7.
8. The fact that you could only open a file when in edit mode.
  
9. No
10. No
11. Hard to find the tabs.
12. No, maybe having to page through all the tabs to see if any new info had been entered.

## **Additional Comments**

### **What was your overall impression of DiDeas II?**

1. It is a good software. It can make design processes more relaxed and interactive. Also because it is distributed, people from different parts of the globe can engage in relaxed and interactive design processes.
2. It was surprising how fast concepts could be generated. I think it's a good tool.
3. Impressed with the concept. Very good information exchange and design contribution platform.
4. Very good. I can see it being very useful for a real project over a longer period of time.
  
5. I think it is a good idea. The graphic user interface could maybe be "modernized" aesthetically, but it is a good way of sharing ideas.
6. I would never use it. I feel the whole design approach in this formalized way is completely devoid of creativity, spontaneity and allows users to propagate clearly stupid ideas.
7. It is a good tool for what it is meant, but the user interface needs a bit of work to make it work more smoothly.
8. At first I was not clear on its purpose or objective. I have since realised the use. With enough time, teams could make good use of the software.
  
9. Works well, and all together the right tool for the job of conceptual design.
10. It was a good way of sharing thoughts and ideas without having to physically have meetings.
11. Its useful, and can be used to get to a conclusion of concept.
12. Good method of documenting work done.

### **Do you have any suggestions for future development of DiDeas II?**

1. I suggest that it also incorporates a friendly non-technical interface for non-technical people so that the information for a system requirement at user level can be input. This is a case of QFD.
2. A way (like a wizard) to guide you in the way of designing and concepts generation.
3. - Renaming the nodes, sub-systems was not possible.

- It was not possible to edit a concept then save it as new concept.
- Description/introduction to the system as well as to each sub-system will be helpful.

4. Inclusion of an “in-client” file viewer for images or PDF’s. Also a way to share images in chat.

5. - Conference ability.

- Live sketching “others see as you sketch” almost like a whiteboard.

6. I would make a “sandbox” style design tool. I would integrate “Skype” and shared “drop box” folders. A sketch pad along the Skype panel allows for fast sharing of ideas, while the software records all activity.

The Dropbox allows for instant sharing of literature, media and creative material. A new specification sheet is needed. Users must be able to see edits in real time via the web.

It must also be able to save screen shots of your desktop via the web.

Strong systems engineering components will also be welcome. Integration with Siemens NX or Inventor, with parts tracking.

7. - Enable longer messages (multi-line) in the messenger box.

- And the ENTER key.
- Enable side scrolling on nodes that are not open for editing.
- Have a read-only option for nodes.

8. VOIP?

Default templates

Tutorials

Longer, softer testing times.

9. Integrated coffee machine.

10. Update field as previously mentioned. ->Notifications.

11. /

12. Work on GUI

“Unread” markers.

Tab scrolling/table cell sizes “fit to text”.

Tutorials or videos to explain workings.

## Appendix E: Industry Questionnaire

This questionnaire was completed by each of the industry experts at the conclusion of each final interview.

### DiDeas II Design Evaluation Questionnaire

Name: \_\_\_\_\_

E-Mail: \_\_\_\_\_

1. Ivan Marais
2. Tom Thring HG – Molenaar
3. Carel Kriek – Reutech Radar Systems
4. Louwrens Marais – EMSS

### Permission to disclose in the thesis and papers reporting the results of this research

**May your name be disclosed?**

1. Yes
2. Yes
3. Yes
4. Yes

**May the name of your company/employer be disclosed?**

1. No
2. Yes
3. Yes
4. Yes

### Professional background

**In which discipline of engineering do you work most of the time?**

1. Mechanical
2. Mechanical Engineering
3. Mechanical
4. Mechanical/Systems Engineering

**How many years of experience do you have in each of the following design areas?**

**High technology systems (e.g. military systems):**

1. 5
2. 0
3. 29
4. 9

**Major subsystems for manufacturing or process plants:**

- 1. 5
- 2. 2
- 3. 25
- 4. /

**Minor subsystems for manufacturing or process plants:**

- 1. 5
- 2. 20
- 3. 25
- 4. /

**Consumer products:**

- 1. 0
- 2. 1
- 3. /
- 4. /

**Other (please specify):**

Missiles, Radar Systems, Automotive

**Which of the following design procedures are you familiar with?**

	<i>Unknown to me</i>	<i>Heard about it</i>	<i>Had training in it</i>	<i>Used it</i>
<i>Ullman's procedure</i>	2	3	1,4	4
<i>Blanchard and Fabrycky's systems engineering</i>	1,2	3	4	4
<i>Cross' procedure</i>	1,2 4	3		
<i>Pahl and Beitz's procedure</i>	1,2,4	3		
<i>Other</i>				
<i>If other, please specify:</i>	4. Had training with Project Performance International in Systems Engineering and using those techniques.			

**How much have you used the following design methodologies in the past?**

	<i>Never</i>	<i>Seldom</i>	<i>Occasionally</i>	<i>Regularly</i>
<i>Ullman</i>	2	1,3		4
<i>Blanchard and Fabrycky</i>	1,2	3	4	
<i>Cross</i>	1,2	3		
<i>Pahl and Beitz</i>	1,2	3		
<i>Other</i>				
<i>None</i>				
<i>If other, please specify:</i>	4. PPI as noted above			

**Overall assessment: How effective do you consider DiDeas II to be as a means for design team members to Communicate with other design team members that are working elsewhere?**

1. Good
2. Moderately effective
3. Good
4. See Remark 1

**Communicate with other design team members that are working on the project at other times?**

1. Reasonable, but would require a follow up conversation to discuss the information in context. Could be interpreted wrongly.
2. Effective
3. Good
4. See Remark 2

**Keep a record of their design work?**

1. Good
2. Effective
3. Very Good
4. See Remark 2

**Develop design requirements?**

1. Good
2. Yes + communicate and structure them.
3. Very Good
4. See Remark 3

**Generate concepts?**

1. Good. Forces you to document it.
2. Sometimes
3. Very Good
4. See Remark 4

**Evaluate concepts?**

1. Requires more detailed evaluation tables with weighted values and also has the ability to calculate.
2. Decision matrix looks useful.
3. Good. (Add facility to add totals).
4. See Remark 5.

**Make and record design decisions?**

1. Good
2. Not sure of practicalities (convenience). Maybe method of identifying important decisions for (more) rigorous handling with this system.
3. Good
4. See Remark 6.

**Tailorability and functionality**

**To what extent is DiDeas II able to be tailored to your company's general design style?**

1. Difficult to say, due to our Development type of work, most of the decisions are made quickly and spontaneously.
2. More Pre-configuration/Template desirable.
3. Well
4. As time constraints only allowed a short demo, I cannot sufficiently comment.

**Please describe any elements of your design process that cannot be provided for with DiDeas II**

1. Time saving: Many design decisions have to be made on a daily basis, it would be very time consuming to use this program continuously.
2. Six thinking hats (everyone looks at the problem differently).
3. /
4. As time constraints only allowed a short demo, I cannot sufficiently comment.

**What functionality was lacking that you would have liked to use?**

1. Add sub-folders/divisions for filing.
2. "
3. Just totalling of comparative matrix.
4. See Remark 7.

## Understanding

### Tree view in the left-hand pane

Does the tree view present information to you in a logical and transparent manner?

1. Yes
2. Yes
3. Yes!
4. Yes

Does the tree view clarify the system structure?

1. Yes
2. Maybe database rather than tree. (Matrix that compares e.g. parts for different projects).
3. Yes!
4. Yes, to a point. Complex, nested trees may prove difficult to present clearly.

Do the tree nodes provide appropriate information (please elaborate)?

1. Yes
2. Configurable! (Can choose appropriate information).
3. Yes, I like that it can show updated status.
4. The node information seems highly customizable so one should be able to tailor it correctly.

## Tabs

To what extent will the tab pages in the right-hand panel help you to find the information you were looking for?

1. Good, training will support this.
2. Add green/red text in tabs will help.
3. Quite well.
4. \*

To what extent will the tab pages in the right-hand panel help you to decide what to do during the design process?

1. Not much
2. A little
3. Well
4. \*

## Tables

To what extent will the tables on the pages in the right-hand panel help you to find the information you were looking for?

1. The information in the tables has to be linked to information that was used to make the decisions.
2. Depends on user and configuration of system.
3. Well

4. \*

**To what extent will the tables on the pages in the right-hand panel help you to communicate your design work during the design process?**

1. Same
2. Good
3. Well
4. \*

\* As there was not time to work with it myself, I can only say that correctly labelled tabs are in general very useful. Similar for well-presented tables.

### **Additional Comments**

**What was your overall impression of DiDeas II?**

1. - Very logical layout
  - Good idea to force designers to follow a specific procedure
  - Could be very time consuming if the application in company is not kept simple.
  - Could have a negative time consuming effect on our company's specific design evaluation process.
  - Good record keeping of decisions made in the past.
2. Key would be to provide a system that allows structure to the design process where it would be beneficial and to be convenient and user friendly. Template relevant to our environment would be important.
3. - Effective without being complicated. Can handle a large system i.t.o. System, Sub-system, concepts etc.
  - Like the idea that Engineers can work from different locations and monitor the process in real-time.
4. As it exists today I believe it presents a good framework, illustrating the concept and intention of the software. It would already make a good tool to capture design information, but would require a measure of configuration management to make that useable in industry.

**Do you have any suggestions for future development of DiDeas II?**

1. Focus more on the Design Criteria Concept Evaluation Table.  
All the information gathered and comments of all the users have one goal: Select the best concept.  
The software should enhance this selection process more.
2. –
3. Data must be protected so that only authorized users have access, permission to change/edit.
  - Project or Context information in separate window on screen to identify which project is open.
4. Suggestions have been incorporated in the attached remarks.

(See the following page for the attached remarks as mentioned by participant 4).

## **DiDeas II Evaluation Summary**

Louwrens Marais  
EMSS Antennas

### **Remark 1**

I don't see it being used primarily as a "direct communication" tool, but rather as a method of capturing design intent and design information. Because it currently lacks a form of configuration management (aside from locking nodes that are currently being edited) its suitability as a real-time tool is (currently) limited.

### **Remark 2**

The ability to capture design information is very useful, and I can actually see it being used as a tool to organise and access a system's datapacks (from customer requirements, specifications, and design documentation at system and subsystem levels, down to detail drawings at component level). Again, this will need some sort of configuration management to build it. By capturing this information at a certain time, it is effective in making that information available at a later time to other team members.

### **Remark 3**

The TQM / House of Quality approach currently incorporated is, for the purpose of the academic study, sufficient to demonstrate the intention of the requirements development portion. In my opinion however, in practise it does not provide sufficient guidelines on how to develop (often vague) customer requirements into engineering specifications. As a next step in the development of the software, a tool to assist with performing a thorough Functional Analysis of the required system may be incorporated. A template, prompting the user to at least consider the major "-ilities" (non-functional requirement) that may be applicable to the system would also be very useful. Design constraints (very importantly, including those of scheduling, manpower and budget) need to be captured. From a more formal Systems Engineering viewpoint, it may even prove useful to incorporate one of the standard Specification Documentation formats as a guideline.

### **Remark 4**

Conceptual design remains a creative process that cannot (again, in my opinion) really be formalized or programmed. However, referring to remark 3, if the software can effectively present to the user(s) the results of a thorough functional analysis, together with the non-functional requirements of the system and the design constraints, it will aid greatly in defining, and helping them visualise the "design space" within which they will operate.

### **Remark 5**

TQM does help in making concept evaluation less subjective. Its success in practise is however dependent on the establishment of an effective value system against which concepts can be judged. Often such a value system is itself heavily reliant on subjective measures (e.g. good / acceptable / bad) rather than objective ones ("how

much” of a parameter). Techniques exist to aid in translating subjective measures to objective measures (generally cost), but these rely on statistical analysis and evidently beyond the scope of the current project. Incorporating such a process into this software may be a very useful follow-up project.

**Remark 6**

By presenting the results of concept evaluations, the software will aid in the making of design decisions and maintaining traceability of those decisions. The ability to add additional information and files to each node is very useful.

**Remark 7**

I'd like to see a tool that captures each specification, and links it to the requirement from which it originates, in order to generate a requirements traceability matrix. Also, I'd like to see a verification matrix, which allows the users to allocate in which manner each specification is going to be verified (e.g. D/I/A/T = by Design / Inspection / Analysis / Testing) during each of the different development phases.

Also, as a future development, a way to link into project scheduling software, and get progress information back from it, may be very useful.

