GMIT
GALWAY-MAYO INSTITUTE OF TECHNOLOGY
INSTITIÚID TEICNEOLAÍOCHTA NA GAILLIMHE-MAIGH EO

Design, Development and Testing of a CAD Integrated Design for Environment Software Tool

In One Volume

Camelia Chira B.Sc.

June 2002

Submitted for the Degree of
Master of Science

Submitted to	:	Galway Mayo Institute of Technology, Ireland
Research carried out at	:	CIMRU, Ireland
Research Supervisor	:	Dr. Thomas Roche

Declaration

I hereby declare that the work presented in this thesis is my own and that it has not been used to obtain a degree in this university or elsewhere.

Camelia Chira

Dedication

To my parents…

Published work associated with this thesis

Roche T., Man E., Chira C., Browne J., “The DFE Workbench a Case Study”, Second International Symposium on Environmentally Conscious Design and Inverse Manufacturing (EcoDesign 2001), Tokyo Big Sight, Japan, December 2001.

Man E., Diez-Campo J. E., Chira C., Roche T., “Product Life Cycle Design Using The DFE Workbench”, 5th IFIP International Conference on Information Technology for Balanced Automation Systems in Manufacturing and Services (BASYS), Cancun, Mexico, September 2002, In Print.

Chira C., Roche T., "Testing and Validation of a Distributed Design for Environment Tool", 19-th International Manufacturing Conference (IMC-19), Belfast, August 2002, In Print.

Prologue

The research described in this thesis was developed as part of the Information Management for Green Design (IMAGREE) Project. The IMAGREE Project was founded by Enterprise Ireland under a Strategic Research Grant Scheme as a partnership project between Galway Mayo Institute of Technology and CIMRU University College Galway. The project aimed to develop a CAD integrated software tool to support environmental information management for design, particularly for the electronics-manufacturing sector in Ireland.

Acknowledgements

There are many people to whom I would like to thank for their help and contribution to the development of this thesis.

Firstly I would like to thank Tom, my project supervisor, for all the help and guidance, advice and support during the research and development work. Thank you for all the meaningful discussions and for always reminding me to think about my project during my holidays.

I would like to thank Elena for all the help and assistance, for all the suggestions and advices in the design and development of the software and for all the interesting discussions over coffee. Thank you for being such a good friend. Good luck with your PhD – I am sure you will be able to gather only brilliant ideas into a good final work.

Many thanks to Juanen for all your help during the development of the DFE Workbench software. Also for all the stimulating discussions during the writing of this thesis – I will try not to tell you when I've finished the last chapter.

I would like to express my appreciation to all the people in CIMRU and GMIT for their support. To Sharon, for helping me out anytime I needed. To Lavinia, Kati, Dana, Laurentiu, Traian, David, for all the chats we had and for just being here. Many thanks to all the people who participated in the protocol analysis tests – Elena, Juanen, Neil, Maria, Mike, Domhnall, Maggie, Chao, Laurentiu and Niall – thank you for your patience and help.

I would also like to thank all my Romanian friends for just being here to make life easier – to Eli, Arpi, Aurora, Laurentiu, Nora, Dieter.

To my sister Cosmina who made the last three months more beautiful – thank you for all the chats we had over the phone or over coffee here. To my parents for all their support in everything I did.

Thank you to all my friends at home – you managed to help and support me even from distance.

Ovi, I have so many things to thank you for that I wouldn't know where to begin – so you are absolutely right, I will just say *thank you*. Everything was easier because of you. *MESC!*

Table of Contents

Glossary

API	Application Programming Interface
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
DDE	Distributed Design Environment
DFA	Design For Assembly
DFD/EOL	Design For Disassembly / End Of Life
DFE	Design For Environment
DFL	Design For Life
DFM	Design For Manufacture
DFS	Design For Service
DFX	Design For X
DLL	Dynamic Link Library
EIME	Environmental Information and Management Explorer
EM	The Environmental Manual for Power Development
EOL	End Of Life
EOLV	End Of Life Vehicle Directive
EPS	Environmental Profile Screening
ESP	Environmentally Superior Product
GUI	Graphical User Interface
HCI	Human Computer Interaction
IAS	Impact Assessment System
ICT	Information and Communication Technology
IPP	Integrated Product Policy
ISO	International Standards Organisation
JDBC	Java DataBase Connectivity
LAN	Local Area Network
LCA	Life Cycle Analysis
LCAiT	LCA inventory Tool
ODBC	Open DataBase Connectivity
P2-EDGE	Pollution Prevention Environmental Design Guide for Engineers
PA	Protocol Analysis
PDM	Product Data Management

SAM	Structure Assessment Method
SPINE	Sustainable Product Information Network for the Environment
SQL	Structured Query Language
QFD	Quality Function Deployment
TEAM	Tool for Environmental Analysis and Management
WEEE	Waste from Electrical and Electronic Equipment Directive
WISARD	Waste-Integrated Systems Assessment for Recovery and Disposal
WWW	World Wide Web

List of Figures

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

List of Tables

Chapter 4

Chapter 5

Chapter 1

Introduction

1.1 Summary

In the early nineties, the development of environmentally superior products was viewed as being costly and time consuming [Roc01a]. Today's industry must consider the environmental performance of its products concurrently with traditional requirements such as quality, price and functional performance. This is mainly because of the emergent legislation (e.g. Waste from Electrical and Electronic Equipment Directive, End of Life Directive), environmental standards and a shift in consumer opinion toward environmentally superior products [Roc99, Man02a]. Many methodologies and tools have been developed to support the industry in its effort to create environmentally superior products. Design for Environment (DFE) represents an effective strategy for developing environmentally superior products, as it is widely believed that 95% of development costs for a product are determined in the design process [Roc01a]. However, many of the existing tools and methodologies that perform environmental analysis are inadequately integrated in the design process and are not linked with the virtual prototyping environment primarily used by the designer [Roc01a].

This thesis presents the design, development and testing of a CAD[1] integrated DFE Workbench software tool that performs environmental and structural analysis of an emergent virtual prototype by implementing the manual DFE Workbench methodology. The new software application is based on Java (programming language), Oracle (database system) and ColdFusion (web programming language) technologies and has been

[1] Computer Aided Design (a CAD environment is used by designers to create prototypes of products).

developed close to commercialisation. The second part of the thesis focuses on the testing phase of the new developed DFE Workbench software in a distributed design environment using a method based on protocol analysis. This final testing phase starts with a protocol analysis experiment developed to tackle and investigate the human behaviour in a computer based collaborative environment during a problem solving process within a team. The purpose of this preliminary experiment is to study the key technical issues associated with the protocol analysis itself and with the application of protocol analysis techniques in distributed collaborative environments. A method of applying protocol analysis in a distributed environment is developed and applied to test the DFE Workbench software.

1.2 Aims and objectives of the thesis

This thesis aims to develop, test and validate a CAD integrated Enterprise DFE software tool for multiplatform computer systems and web environments. The objectives of the thesis can be summarised as follows:

- To investigate existing DFE tools and methodologies.
- To examine the needs of industry for DFE tools.
- To investigate appropriate programming languages and database systems suitable for multiplatform applications.
- To develop appropriate CAD integrated software tools.
- To develop appropriate testing and validation methodologies.
- To establish conclusions on CAD integrated DFE tools.
- To disseminate research results through published papers.

The *DFE Workbench software* and a *protocol analysis method for distributed environments* have been developed in order to attain these objectives.

1.3 Approach to work

A new software-based version of the DFE Workbench has been developed for three levels i.e. Desktop, Enterprise and Global to address the needs of industry e.g. corporate organizations. The programming language chosen for implementation is Java, an object-oriented language designed for use in distributed applications on corporate networks and the Internet. Therefore, the DFE Workbench software is a portable system that can run on any Java-enabled platform. A robust database system i.e. Oracle has been selected to hold the environmental and structural data used in the analysis, evaluation and improvement of a product. The tool has been integrated in two CAD systems i.e. SolidWorks 2000 and

ProEngineer 2001. The development of the software was based on a specification document that included requirements resulted from a prototype version of the tool and new functional, non-functional and usability requirements.

Industrial partners from the electronic and automotive sector have been heavily involved in the testing of the tool throughout its evolution. New requirements that emerged from this collaboration have also been included in the DFE Workbench software. The development of the DFE Workbench software occurred through a number of iterations and as a result there are many working versions of the DFE Workbench software over a period of eighteen months. This thesis presents the latest version of the software that includes all the requirements identified at the beginning of the research as well as additional features proposed by the industrial partners.

The final part of the research has focused on the design and development of a method for applying the protocol analysis technique in a distributed environment and the final testing phase of the DFE Workbench software in a distributed design environment using the proposed protocol analysis method.

Figure 1.1 summarises the approach to work described above.

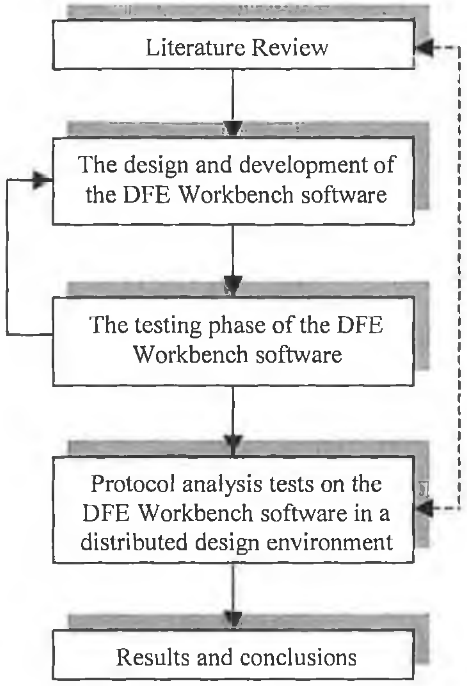

Literature Review
The design and development of the DFE Workbench software
The testing phase of the DFE Workbench software
Protocol analysis tests on the DFE Workbench software in a distributed design environment
Results and conclusions

Figure 1.1 Approach to work

The loop between the design and development phase and the testing phase of the DFE Workbench software suggests the incremental model[2] of the development process. The

[2] See Pressman R.S., "Software engineering: a practitioner's approach", McGraw-Hill Publishing Company, 2000, pp 33-40.

incremental model focuses on the delivery of an operational product with each increment. The core DFE Workbench developed as the first increment was modified throughout the project evolution to better meet the needs of the customer and the delivery of additional features and functionality. Each increment represented a version of the DFE Workbench software that was completely functional.

The final testing phase of the DFE Workbench software started with a literature review on protocol analysis techniques and their use in distributed environments. This final stage of the research included the design and development of a method for applying the protocol analysis technique in a distributed environment based on two protocol analysis experiments that studied and compared the face-to-face collaboration and distributed synchronous collaboration models. The proposed method based on protocol analysis was applied to test the DFE Workbench software in a distributed design environment.

Results and further developments proposed conclude the thesis.

1.4 Thesis structure

The structure of the thesis is presented in figure 1.2.

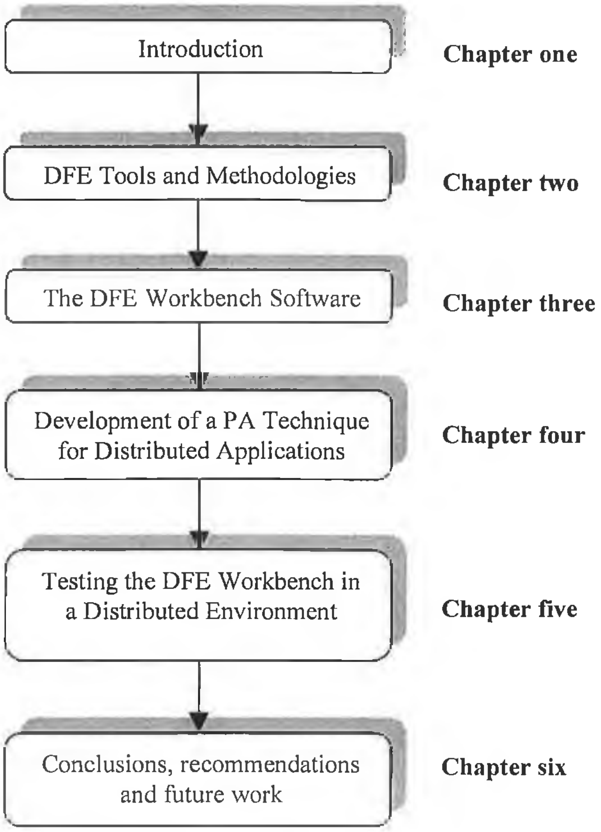

Introduction
Chapter one
DFE Tools and Methodologies
Chapter two
The DFE Workbench Software
Chapter three
Development of a PA Technique for Distributed Applications
Chapter four
Testing the DFE Workbench in a Distributed Environment
Chapter five
Conclusions, recommendations and future work
Chapter six

Figure 1.2 Thesis structure

Chapter one

This chapter presents the thesis motivation, objectives and structure.

Chapter two

The existing DFE tools and methodologies are briefly described by presenting their main features, advantages and disadvantages. The DFE Workbench methodology and the prototype software version are described in more detail. The chapter aims to identify the deficiencies of the first software-based version of the DFE Workbench and to build the specification for a new DFE Workbench software, which includes details about the input data, output data as well as the data held in by the system.

Chapter three

This chapter presents the DFE Workbench software developed after the specification described in Chapter two. The software architecture, database structure, main features of the tool and the CAD integrations of the DFE Workbench are described.

Chapter four

Protocol analysis techniques are reviewed and the concept of distributed environment is presented. This chapter focuses on the design and development of a method for applying the protocol analysis technique in a distributed environment to test the human-to-human interaction via a graphical user interface.

Chapter five

This chapter presents the protocol analysis test on the DFE Workbench software in a distributed environment using the method developed in Chapter four. The data analysis and main results of the distributed protocol analysis test conclude this chapter.

Chapter six

This chapter contains the conclusions of the thesis and presents recommendations for future development of the DFE Workbench software and of distributed protocol analysis techniques.

The thesis structure implies that the evolution of the tool occurred in the following stages: specification creation, implementation and testing. However, the DFE Workbench software has been continuously tested throughout the project evolution and new functionality was added and tested. For example, the DFE Workbench identifies the percentage of hazardous material in an assembly; after one of the testing phases with an industrial partner from the automotive sector, it has been proposed to have quick access to a printable list that contains all the components of an assembly characterised by a hazardous material; the new proposed feature has been implemented in the design and development stage. Also, the final testing phase of the DFE Workbench in a distributed environment is based on a protocol analysis review, which completed the literature review on DFE tools and methodologies already performed. Therefore, it must be noted that the development process of the DFE Workbench software is not completely mirrored by the thesis structure.

Chapter 2

Design for Environment Tools and Methodologies

2.1 Introduction

Current trends and market dynamics combined with the emergence of new environmental legislation and standards force the manufacturers to move towards the development of environmentally superior products (ESP). Design for Environment (DFE) represents an effective strategy for developing ESPs. This chapter introduces the concept of DFE and briefly describes some of the existing DFE tools and methodologies in an attempt to identify their main features and their positive as well as negative aspects. Later on, the chapter introduces a novel DFE methodology and software tool called the DFE Workbench that supports the synthesis, evaluation, analysis, prioritisation and improvement of both environmental impact and structural data associated with an emergent virtual prototype. Both the manual methodology and the software version of the DFE Workbench have been tested in two modes as follows: firstly, involving industrial partners in the development stage and secondly, using protocol analysis techniques on qualified engineers from different engineering backgrounds and employed in different manufacturing sectors. The tests proved the efficiency of the manual methodology in reducing the environmental impacts associated with a product but they also revealed various problems of the DFE Workbench software. This chapter aims to identify the deficiencies of the software-based version of the DFE Workbench and to build a specification document for a new software version of the tool, which would address the identified problems and will consist of new features developed to meet new requirements.

2.2 Design for Environment

Fiksel defines Design for Environment (DFE) as *"the systematic consideration of design performance with respect to environmental health and safety objectives over the full product and process life cycle"* [Fik96]. It is clear from this definition that DFE approaches must take a more holistic view of the life cycle (see figure 2.1) than traditional design methodologies.

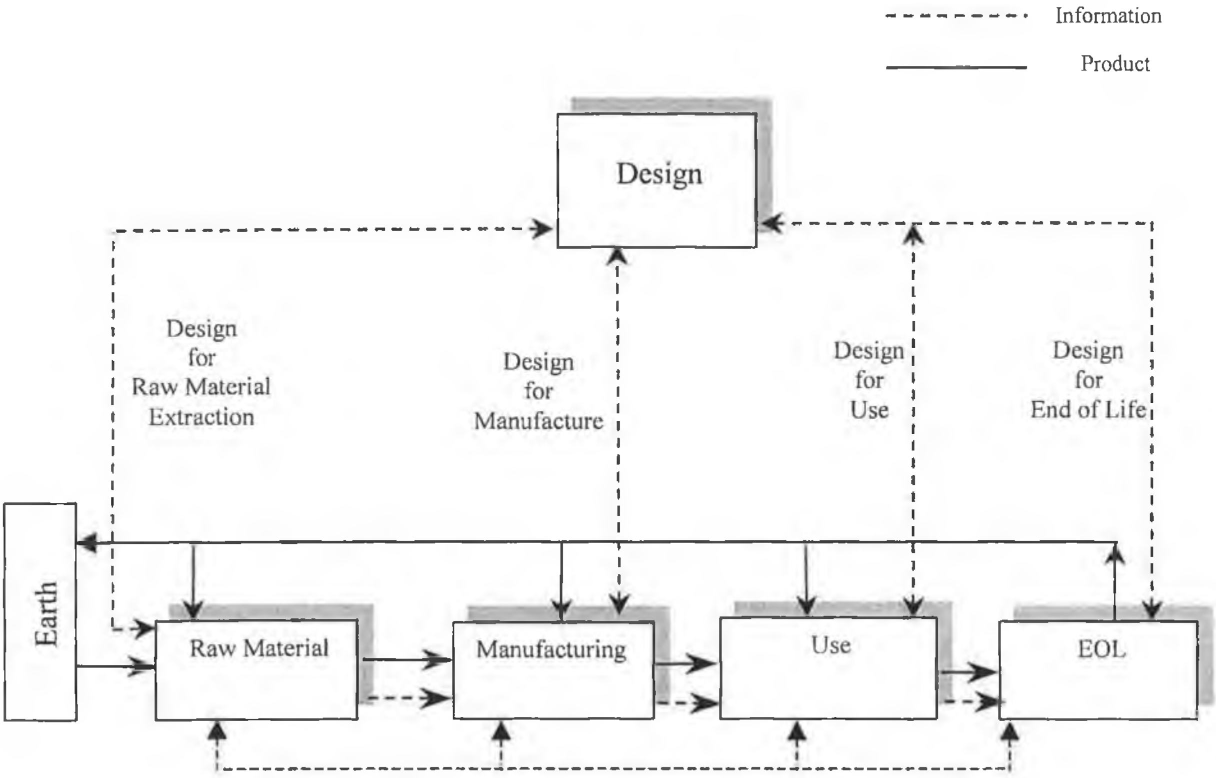


Figure 2.1 Design information loops [Roc99]

In the model shown in figure 2.1 life cycle information is acquired through a set of life cycle design information loops, i.e. design for raw material extraction, design for manufacture, design for use and design for end of life [Roc99, Man02c]. In each of the generic phases showed in figure 2.1 materials and energy are consumed either directly into the product or given off as waste streams. When the product reaches the end of life a decision has to be made to reuse, remanufacture, recycle or dispose of it. Similar decisions have to be made regarding the materials and energies entering the waste stream [Roc01c].

Four generic and interrelated strategies for the development of ESPs can be derived from this model as follows [Roc99, Roc01c, Man02c]:

- Select low impact materials and processes over all life cycle phases.
- Reduce life cycle resource consumption (Materials and Energy).
- Reduce life cycle waste streams (Materials and Energy).
- Resource sustainment by facilitating first life extension and post first life extension, i.e. reuse, remanufacture and recycling

Life Cycle Analysis (LCA) is the only method available to measure the *environmental impact* of products on the environment [Roc99, Roc01c]. The ISO14040 standard defines life cycle analysis as; "*a technique for assessing the environmental aspects and potential impacts associated with a product by: compiling an inventory of relevant inputs and outputs of a system; evaluating the potential environmental impacts associated with those inputs and outputs; interpreting the results of the inventory and impact phases in relation to the objectives of the study*" [ISO97]. Life Cycle Assessment is recognised as one of the most frequently used techniques for systematically evaluating environmental performance of a product throughout its life cycle [Roc01c].

The reduction of life cycle resource consumption and life cycle waste streams requires resource minimisation solutions. Tools need to be developed and integrated into the design process to aid the designer to identify resource wastage directly and indirectly associated with the life cycle of the product. Many of the exiting tools are not appropriately integrated in the design process or indeed across the life cycle of the product. Resource sustainment is an extremely important and effective strategy for the development of ESPs. First life extension may be achieved through designing for serviceability, maintainability, reliability and durability. Post first life extension strategies include policies to reuse, remanufactur
recycle and recover the product at the end of life [Roc01c].

Because of the lifecycle characteristics associated with ESPs the design activity (
2.1) is more likely to occur in a distributed Information and Communicati
(ICT) based environment [Roc01a]. Distributed and collaborative
becoming of crucial importance for the mapping of abstract *lifecycl*
all phases of the design process into detailed design informatio
cycle) [Roc01a].

Distributed Design Environments

The fast development of computing and networking technologies in recent years as well as the rapid advances in collaboration and collaborative technologies made possible the exploitation of heterogeneous, distributed computing platforms [Lau01]. A *distributed environment* represents a computer network-based environment in which many users can collaborate, exchange information and share resources during the process of determining a solution to a given task. More practically, a distributed system consists of a number of computers that can send data to each other via a network (Extra/Intra/Internet). Also, a distributed system should be able to provide multiple users with concurrent, efficient access to multiple system resources [Ihe02]. In design, the distributed environment came as an effective solution to the designer's need to quickly access high quality information that enables him/her to inform the design process. The model of distributed design environment (DDE) represents not only an answer to market demands but also a solution to specific design problems. Any individual or group of individuals involved in a product development process (e.g. designers, product manufacturers, suppliers and design information providers) is a participant to the DDE. The teams of people collaborating in a DDE can have one or more of the following characteristics:

- Geographically dispersed
- Temporally dispersed
- Diverse in their areas of expertise

Computer based systems guarantee the communication and collaboration among distributed users. Hence, human-to-human interaction is realized via a computer user interface. As a result, both human-to-computer and human-to-human interactions are becoming of crucial importance in a distributed and collaborative environment.

Because of the distributed nature of information and people involved in the design process (see figure 2.1) cooperative work teams in a virtual environment have to be supported. The primary elements to distributed and cooperative work teams are as follows [Pen00]:

- *Communication* – refers to the exchange of information, events and activities between participants.
- *Co-location* – focuses on the infrastructure to provide a smooth communication among distributed participants.
- *Coordination* – refers to the management of the workflow and communication process.

- *Collaboration* – describes the process of creation of a shared understanding in a distributed environment.

Although an effective communication is a necessity, it is not a sufficient condition to a meaningful collaboration in a distributed environment. Efficient coordination and collaboration are of significant importance while communication is an integral component in the problem solving process [Pen00]. Due to current trends in the design field toward virtual teams that collaborate over computer networks to achieve global optima in design, there is an increasing need for design teams to establish and maintain a cooperative work through a good communication, co-location, coordination and collaboration. In a distributed virtual environment, these elements are supported by collaborative technologies such as e-mail, video conferencing, chat, shared whiteboards, application sharing, awareness and shared access to databases.

The structure of a distributed architecture is presented in figure 2.2.

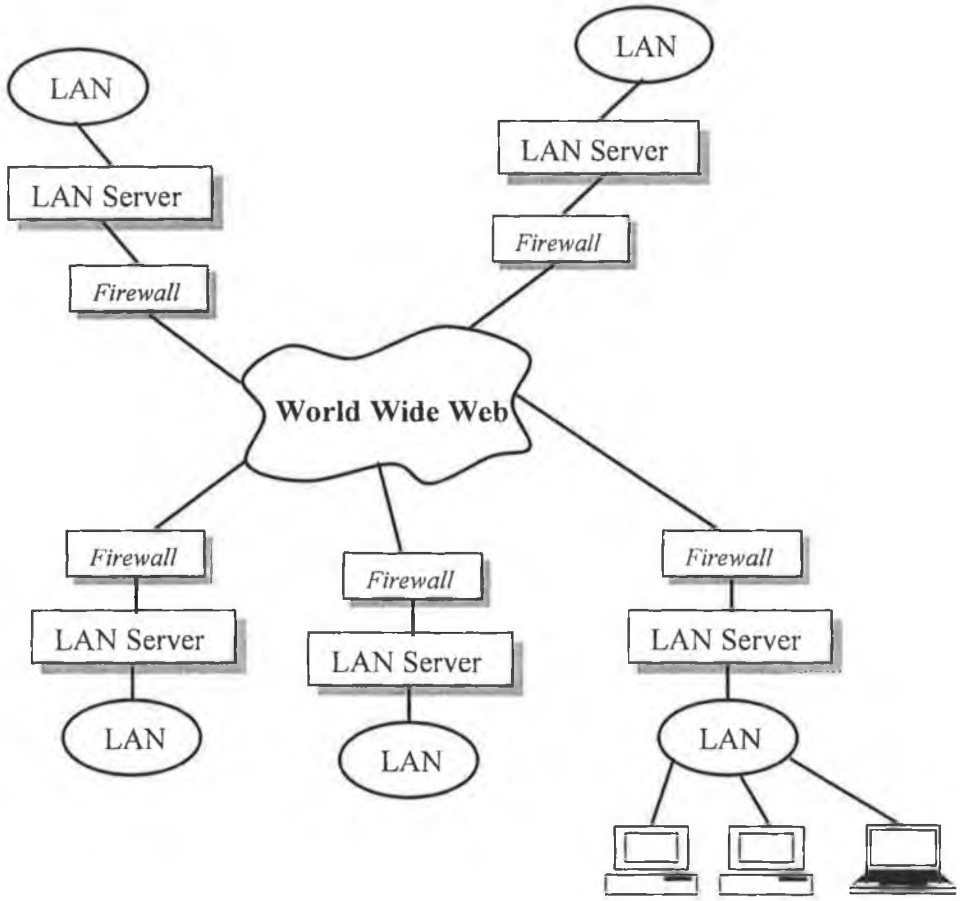


Figure 2.2 A distributed architecture

Every Local Area Network (LAN) in figure 2.2 is administered by a LAN server and can contain any number of users. The junction to the Internet should be secured by firewalls against hostile intruders. The protection of system resources has become extremely important as the number and size of information systems has increased [Ihe02]. A team of individuals located in one LAN can communicate and collaborate with the other teams located in different LANs involved in the DFE problem solving process over the computer network. It should be noticed that a broke down LAN server does not imply the malfunctioning of the whole system.

Whilst DDE represents a necessity for DFE additional benefits include [Pen00, Lau01, Ihe02]:

- Savings in project life-cycle and costs
- Added value to team efforts
- Access to a comprehensive knowledge-based system
- Reliable communication among design teams and members
- Flexible access and retrieval of information
- Timely connectivity with global experts

Research in the area of computer-mediated communication and collaboration technologies has focused on resources allocation problems and optimised design process management [Pen00]. These problems were addressed using artificial intelligence techniques such as neural networks, knowledge-based expert systems and genetic algorithms. Recent studies show that the next generation model for engineering, complex distributed systems consists of agent-based systems, which are autonomous software components embedded in a particular environment that work in conjunction with the user, being able to respond to changes that occur in their environment and to act in anticipation of future goals; agents have the ability to learn, share information and knowledge with each other and with the user and they may have the capability to work without human interaction [Jen00, Anu01, Lee01].

2.3 The need for DFE in industry

In the early nineties the environmental considerations associated with an emergent product were viewed as being costly and time consuming by many manufacturers generating a reactive approach towards environmental issues [Roc01a]. Today's industry has to be more proactive about the environment because of several factors as follows [Man02a]:

- Emerging legislation such as *Waste from Electrical and Electronic Equipment Directive* (WEEE – this Waste directive has been proposed by the Commission of the European Communities on the 13th of June 2000 and is expected to become law by the year 2005) and *End Of Life Vehicle Directive* (EOLV). According to WEEE and EOLV manufacturers are obliged to take responsibility for waste management by implementing re-use, recycle and recovery policies for their products.
- New European policies such as *Integrated Product Policy* (IPP), which has been introduced by the Commission of the European Communities on the 7th of February 2001. The IPP refers to eco-design of products and the creation of information and incentives for an efficient take-up and use of greener products [Roc01a].
- Environmental standards such as *ISO 14000* (Environmental Management System Standard launched in 1996).

Design and development of ESPs by implementing DFE is an effective strategy for complying with the environmental drivers. Companies considering the environmental issues associated with their products are likely to gain significant competitive advantage in the future [Roc01a].

Many methodologies and tools that perform environmental analysis have been developed over the last years. However, many of the existing DFE software tools and methodologies are inadequately integrated in the design process being applicable to the detailed design phase although the support of the early design phases is widely considered to be more important in the development of effective design solutions. The next section briefly describes some of the existing DFE tools. Most of these software tools are based on a manual methodology that supports the development of environmentally superior products.

2.4 Existing DFE tools and methodologies

The existing tools that perform analysis and improvement of the environmental performances of a product were developed based on one or both of the following methodologies [Man00]:

1. Life Cycle Analysis (LCA) – consists of a set of methods used for measuring the environmental impact of a product's life cycle.
2. Design for X (DFX), where X stands for specific design focus such as assembly/disassembly, maintainability, serviceability or recycling – taking in account the relationships among the X techniques may result in the improvement of the product from an environmental perspective.

This section reviews some of the existing DFE software tools under the two categories mentioned i.e. LCA tools and DFX tools in an attempt to identify their characteristics and specific features.

Although there is a large number of LCA tools available, many of them are not commercialised or no further information on them is available [Cal97]. Some of the available LCA tools can be described as follows:

- *LCAdvantage* is one of the most complete packages available for computer based LCA. Developed by Battelle, the tool allows the graphical representation of processes and the connection among them. LCAdvantage is an attractive tool for a wider set of cost-benefit analysis applications because is not restricted to modelling materials and energy [Bad99].
- *Environmental Profile Screening* (EPS) is a Battelle product that combines LCA with a less data intensive, less expensive matrix evaluation procedure. EPS is a question-based system that produces environmental scores associated with individual components of a product, which can be combined to produce an aggregate score for the overall product [Bad99].
- *LCA inventory Tool* (LCAiT) is a software tool developed by CIT Ekologik that calculates a product's environmental loadings throughout the life cycle. The tool is a Windows based application using a relation database to store the LCA data, which is documented according to the SPINE[1] documentation. The results are presented in the form of exportable matrices and charts [Bad99, LCAIT].
- *Environmental Information and Management Explorer* (EIME) is a software package developed by Ecobilan. The tool is an environmental management system specialised in the electronic and electric industry, which incorporates a database with the most commonly used materials and sub-components. Two types of output metrics are available i.e. life cycle indicators and design indicators. The main deficiency of the tool is the lack of interface with CAD/CAM systems and of a complex interface with the product model [Bad99].
- *Tool for Environmental Analysis and Management* (TEAM) is a software tool developed by Ecobilan, which allows calculations of life cycle inventories, potential environmental impacts and associated costs. The tool is based on a database manager but is missing the link with other design software [Bad99, Man00, ECOB].

[1] Sustainable Product Information Network for the Environment (SPINE) enables the efficient handling of the environmental information used in life cycle assessments [LCAIT].

- *Waste-Integrated Systems Assessment for Recovery and Disposal* (WISARD) is a LCA software package that allows users to model all the aspects of a system such as container system, collection, separation, recycling, incineration, composting and landfill and to compare their environmental impacts. The tool is assisted by a waste management database [ECOB].
- *Umberto* is a powerful tool for Life Cycle Assessment, Life Cycle Costing and Corporate Ecobalancing developed by IFU[2] in collaboration with IFEU[3]. The tool can be used to visualize material and energy flow systems. The environmental costs of the system can be displayed and analysed [Bad99, UMB].
- *GaBi* is an LCA tool for designers, consultants and scientists created by IKP[4] in co-operation with PE Europe GmbH. This Windows based application allows weak point analysis of inventories and balances and offers an economic examination of the system on the basis of material/energy costs, personnel costs and machine costs. The tool provides modelling and analysis of complex and data-intensive problems through features such as GaBi 3 Database Manager and project manager [Man00, GABI].
- *REPAQ* is a software tool that performs inventory modelling for products, processes and packaging derived from Franklin Associates' *Resource and Environmental Profile Analysis* studies. The user can update the REPAQ database through the *Custom Materials* feature [Bad99, Man00].
- *SimaPro* 5.0 is a LCA software package developed by PRe Consultants, which collects, analyses and monitors environmental information associated with products and services. The analysis results in Eco-indicator scores associated with each of the life cycle stages of a product [Man00, PRE].
- *ECO-it* is another LCA software from PRe Consultants. This tool calculates the environmental load associated with a product and helps the designer to optimise the environmental performance of products in the design phase. The software is based on a database of over 200 Eco-indicator 99 scores for commonly used materials, production, transport, energy and waste treatment processes. PRe Consultants offer another tool i.e. *ECO-edit* to edit or create databases for Eco-it [Bad99, PRE].
- *EcoScan* is a Windows based application that calculates environmental scores for each of the life cycle stages of a product. It allows the comparison of products in a

[2] Institute for Environmental Informatics Hamburg Ltd.

[3] Institute for Energy and Environmental Research Heidelberg Ltd.

[4] Institute for Polymer Testing and Polymer Science at the University of Stuttgart

single graph, data interchange with other applications or the calculation of transport distances and product mass. The latest version of the tool is delivered with Eco-indicator 95 and 99 databases and EcoScan 97 database and works in conjunction with the IdeMat 2000 materials and processes database [Man00, ECOS].

- *The Boustead Model* is a LCA tool which consists of two parts; firstly, an extensive database that stores data such as fuels and energy use and raw materials requirements; secondly, a software application that enables the user to manipulate the database and to perform data analysis. The software tool supports life cycle inventory modelling [Man00, BOUS].
- *The Significance Wizard* from Entropy International is a Windows based application that guides the designer in the process of implementing an environmental management system. The tool helps the user to evaluate environmental aspects and impacts associated with a process flowchart using a drag-and-drop interface [ENTR].
- *The Environmental Manual for Power Development* (EM) is a software tool that includes environmental and cost data into the decision-making process regarding energy projects [Bad99].
- *Pollution Prevention Environmental Design Guide for Engineers* (P2-EDGE) is a software tool designed to help the user incorporate pollution prevention strategies into the design stage of new products, processes and facilities [Bad99].
- *EcoManager* is a software application designed as an internal, screening and evaluation tool that uses databases with materials, energy, waste and transport [Bad99].

The DFX tools are focused on the product's life cycle and/or methods to improve an aspect of a product. The existing tools can be divided into the following three groups: Design For Assembly (DFA), Design For Life (DFL) and Design for Disassembly / End Of Life (DFD / EOL) [Cal97]. The DFX tools are based on indexing systems that measure the features of a product contributing to the environmental impact of that product [Man00]. Some of the available DFX tools can be described as follows:

- *Green design advisor* is a software application that allows the designer to minimise the environmental impacts associated with manufacturing, use and disposal of electromechanical products based on a ranking system [Man00].

- *ReStar* is a software tool, which optimises the component recovery plan. The application is based on an evaluation methodology for Design for Disassembly. It supports the end of life analysis of electromechanical and electronic products such as automobiles, computers and other consumer electronics [Man00].
- *Design for Assembly* (DFA) was developed by Boothroyd Dewhurst, Inc. The tool assists the user in estimating the cost of manually assembling a product and provides quantitative measures of the suitability of a given product design to assembly. The user selects from the design strategies offered by the tool the one that provides the simplest product structure and minimizes the cost of assembly. Several graph formats or reports are available as output from the tool [Bad99, Man00, DFMA].
- *Design for Manufacture* (DFM) works together with *Design for Assembly* software as the tool was developed by the same company. The DFM tool provides cost estimations for the manufacture of individual parts. Both tools are connected to an internal database from where processes and materials can be selected. The major deficiency of these tools is that a connection with a CAD/CAM system is not available to allow a general tool for design [Bad99, Man00, DFMA].
- *Design for Environment* (DFE) is a Boothroyd Dewhurst tool that allows the optimisation of a product by simulating the disassembly of the product at end of life and by providing estimates of the environmental effects of production and the end of life of a product. The tool performs two main analysis based on the disassembly sequence; firstly, the financial return assessment of disassembly, disposal, reuse or recycling which shows the financial impact at each stage of disassembly; secondly, the environmental impact assessment analysis from the product manufacture to disposal, reuse or recycling. The results of the tool are presented in the End of Life Evaluation graph [Bad99, Man00, DFMA].
- *Design for Service* (DFS) is a DFX tool developed by Boothroyd Dewhurst, Inc. The DFS tool helps the user to evaluate the serviceability of a product in the early stages of design. The tool provides a series of reports that are guiding the designer by prioritising the areas in the service task that must be examined for further improvement [Bad99, Man00, DFMA].
- *DFmA* is a comprehensive generic design model developed by Lucas Engineering and System Ltd. The tool links Quality Function Deployment (QFD) charts with

the use of Design for Assembly and other design tools. The DFmA software focuses on the direct production costs and environmental issues [Bad99].

- *LASeR/Linker* evaluates the serviceability, recyclability and assembly of mechanical designs [Bad99].
- *Reverse Fishbone DisAssembly Tool* is a software tool used to model the disassembly reprocessing sequence of a product at the end of life [Bad99].
- *AMETIDE* is a software tool that provides time disassembly estimation. It offers access for the designer to all possibilities for removing a part using a database with all the disassembly techniques available corresponding to a fastener [AMET].
- *Optimum Disassembly Planning for Environmentally Conscious Manufacturing* is a Windows based software developed to help the designer to analyse product disassemblability. The tool offers optimal disassembly sequences and predicts maximum net profit values [DFDO].
- *Salvage* is a web-based software tool developed to implement the Assembly/Disassembly Optimisation Model, which allows the optimisation of a product lifecycle. The tool incorporates economic and environmental considerations during the virtual prototyping phase of electronic product design. The outputs of the tool are the disassembly cost, primary/secondary assembly cost, quality, waste material, material consumed for the product under design [SALV].

The main disadvantage of the available DFE software tools is the absence of link interfaces with the virtual prototyping environment primarily used by the designer.

The limitations of the existing DFE tools and methodologies reviewed above can be summarised as follows [Man00]:

- *"Most of the LCA tools require environmental expertise.*
- *The results are often difficult to interpret and the use of the tools is laborious and time consuming.*
- *Most of these techniques provide specialised analysis by either addressing specific issues such as manufacturing, disassembly and costing or they are characteristic to a specific stage of the life cycle such as end of life or they address only to specific products such as electronics.*
- *They usually perform the analysis very late, after the product has been designed.*
- *They have data limitations, as it is very difficult to obtain accurate information relating materials, processes, use, transport or end of life."*

These limitations must be eliminated in order to obtain an effective DFE tool.

The DFE Workbench

The DFE Workbench was designed and developed in two forms as follows [Roc99]:

- *The manual DFE Workbench methodology*, which is largely based on using special charts and reference information in a structured manner to evaluate and improve an emergent design.
- *The software based DFE Workbench*, which is a CAD integrated application that effectively automates processes associated with the manual methodology.

The DFE Workbench is based on the principles advanced by Kimura and Tomyama, which predicted two types of environmentally superior products: firstly, products that sustain functional growth and have theoretically unlimited lives and secondly, products with shorter life cycles but with structures that facilitate value creation from manufacturing, recycling and reuse [Roc99]. The methodology supports strategies like life cycle extension and remanufacture by facilitating ease of disassembly, serviceability and the selection of durable materials with low environmental impact. Therefore, it is important to note that the DFE Workbench is not based solely on environmental criterion but rather on an extended product[5] criterion that results in the development of products with environmentally superior characteristics. Roche and Man believe that the DFE Workbench can be applied for the development of extended products [Roc01a, Man02b].

Roche designed and developed the DFE Workbench methodology by considering the design process and associated issues [Roc99]. It is essential for DFE tools and methodologies to support the design process throughout all phases as it is an information transformation process which is effected through a series of recurrent problem solving cycles that are used effectively to evaluate diagnose and improve the design as the design evolves. Roche has identified a set of requirements for the development of methodologies and tools to support the creation of ESPs as follows [Roc99]:

- Methodologies and tools must be integrated as early as possible in the design process without disrupting the design process, as well as being integrated throughout the design process.

[5] Thoben and Jagdev have proposed a definition of the Extended Product as a product consisting of three elements: a tangible element representing the physical product and the services associated *"which can be intelligent, highly customised, user friendly and include embedded features like maintenance"*; an intangible element represented by the information and knowledge like services, engineering and software; and finally a collaboration element *that "refers to both material and information flows in order to accelerate the cooperation within the value chain"* [Tho01, Man02b].

- All approaches must take a life cycle view for the development of ESPs as a high degree of coupling can occur between lifecycle product characteristics in the design process.
- LCA is an important tool for the development of ESPs, however quantitative abridged LCA approaches are more likely to cater for the dynamic nature of the design process.
- Approaches must be developed to extend the first life and post first life of products, e.g. design for reuse, remanufacture and recycling.
- Tools and methodologies must support the analysis, synthesis, evaluation and improvement of proposed designs from both structural and impact point of view. As a result, a prioritisation tool combined with an advisor module that guides the designer through his efforts of building environmentally superior products without constraining him in any way, are necessary.
- These tools must be integrated in the Computer Aided Design system used for the development of the product prototype.
- A DFE tool must be easy-to-use, user friendly and must require no expert knowledge in the design for environment area.
- A powerful and updateable database should support DFE tools with information on all environmental issues.
- DFE tools should provide general and detailed reports with charts displaying all the environmental metrics calculated by the tool or required by law.
- Product Data Management (PDM) integrated tools are desirable as they support enhanced communication between design teams.
- It is important to develop web based DFE tools as web applications are powerful communication tools for both Intranet and Internet networks.

The DFE Workbench consists of a set of integrated methodologies to support the execution of the DFE process as follows [Roc99]:

- Impact Assessment System (IAS)
- Structure Assessment Method (SAM)

IAS focuses on the analysis, synthesis, evaluation and improvement of the environmental impacts associated with a product. Roche proposed to integrate an abridged quantitative Life Cycle Analysis tool into the design process, which should be assisted by an advisor agent to help the designer find environmentally superior design options [Roc99]. IAS is based on the Eco-indicator 95 methodology[6].

SAM supports the analysis, synthesis, evaluation and improvement of the structural characteristics associated with a product. The advisor agent helps the designer to improve structural characteristics that enhance the environmental superiority of the product [Roc99]. IAS and SAM are integrated methodologies, as changes in SAM will result in changes in IAS. A set of tables created by the designers of the method supports the data synthesis and the continuous improvement process using both IAS and SAM.

Figure 2.3 presents the structure of the manual DFE Workbench methodology.

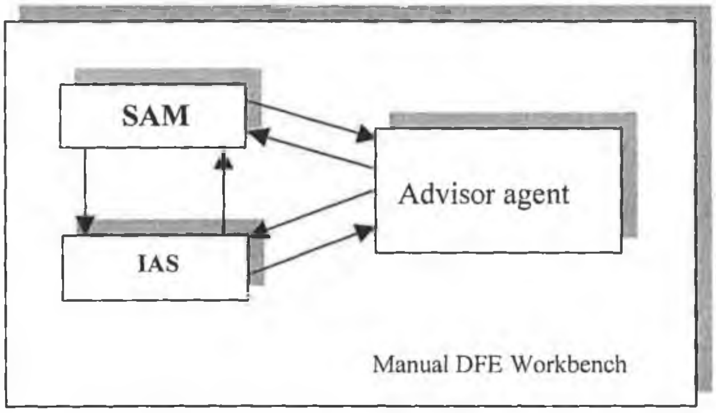


Figure 2.3 The DFE Workbench methodology

The DFE Workbench has been designed to act as *"a platform to facilitate the operation of the methodologies and to manage all the interrelationships between the environmental information in the product"* [Roc99]. Roche proposed the DFE Workbench to be used in four modes i.e. analysis, synthesis, evaluation and improvement of environmental characteristics of the product structure whilst in the design process, as a report for product users, for the evaluation of competitors products from an environmental point of view and to train the designers on how to develop environmentally superior products.

The DFE Workbench software was developed to support the execution of the steps defined by the two methodologies of the DFE Workbench i.e. IAS and SAM [Man00]. This first version of the software-based DFE Workbench was implemented in Visual Basic 5. Microsoft Access was used to store all the data manipulated by the DFE Workbench. This version of the software tool has been integrated into a virtual prototyping environment i.e. SolidWorks98 Plus. The advisor agent developed in the manual methodology was

[6] The Eco-indicator 95 method was developed by Pre Consultants in collaboration with Phillips, Volvo, Oce, Schuurink and the Universities of Amsterdam, Leiden and Delft. The methodology aims to analyse products in order to find the causes of environmental pollution, to improve the product from an environmental perspective and to compare the environmental impacts of two alternative product configurations [Man00].

implemented in the software to automatically identify and prioritise problems in an emergent design and to suggest alternatives to improve specific environmental and structural characteristics of a product [Man00]. This software version of the DFE Workbench further aids the design activity through a knowledge base agent that allows the designer to search information that is not specific to any particular design characteristics. The structure of the DFE Workbench software tool is presented in figure 2.4 [Man00, Roc99].

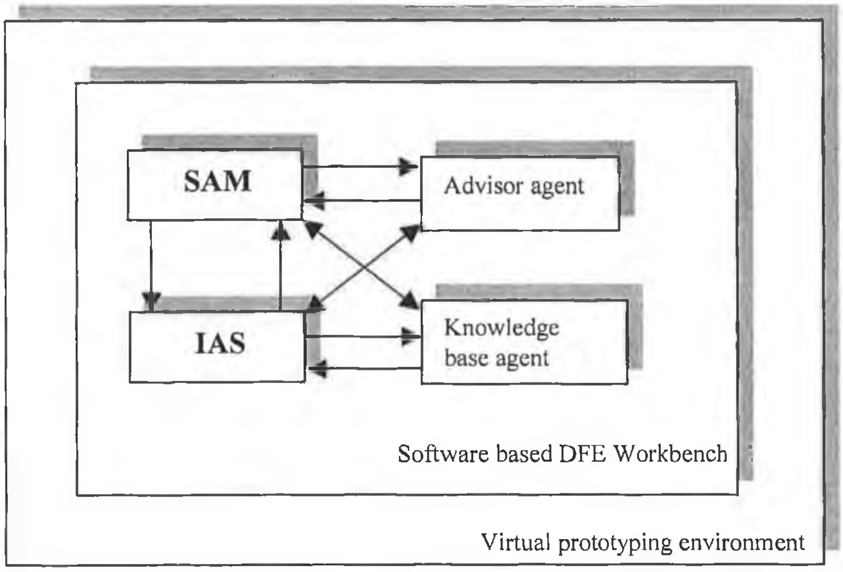


Figure 2.4 The DFE Workbench software

The first version of the DFE Workbench software successfully supported the continuous improvement of an emergent virtual prototype from an environmental perspective [Man00]. However, various problems have been identified in the first version of the software tool from both technical and functional perspectives (see section 2.5).

2.5 Deficiencies of the first version of the DFE Workbench software tool

The DFE Workbench has been tested on individual designers using the protocol analysis technique. The main conclusion of the protocol analysis tests is that the DFE Workbench can be a useful aid for the development of environmentally superior products. The tests showed that it is essential to develop a software application to support the methodology because of the volume of calculations and the manipulation of interdependent relationships [Roc99]. The integration of the software based DFE Workbench in a virtual prototyping environment is also essential because of factors such as the automation of data synthesis activity, the availability of quantitative data directly from the model and the manipulation of this data. The software disadvantages identified after the data analysis phase of the protocols have been completed by deficiencies resulted after the testing phase with industrial partners [Man00].

The problems of the first software version of the DFE Workbench can be summarised as follows:

- The DFE Workbench software tool was designed and developed as a standalone application that can be used only by one designer on a CAD workstation. The data used by the application are not centralized.
- The DFE Workbench software tool does not support distributed design environments (the tool was not developed to meet the needs of corporate organizations).
- The portability of the tool is not supported by neither the programming language used for implementation i.e. Visual Basic nor by the database system used i.e. Microsoft Access.
- Further integration of the tool with different CAD systems is limited to Windows based environments.
- The first software based version of the DFE Workbench is not easy to use e.g. the number of panels the user has to go through in order to perform an environmental analysis for a product is high. The simplicity of the graphical user interface of the tool is also affected by the large number of mouse clicks and keystrokes that are required to accomplish a particular task.
- The designer can perform only one analysis at a time and does not have the option of comparing different environmental evaluations of the product.
- The application supports the analysis and evaluation of simple assemblies only i.e. assemblies containing only components and no subassemblies.

All these problems have to be addressed by designing and developing a new version of the DFE Workbench using a platform-independent programming language and a client-server database system suitable for use in distributed design environments. The new version of the DFE Workbench software tool should address the needs of corporate organizations and should support distributed designers as the design activity is more likely to occur in a distributed Information and Communication Technology based environment because of the lifecycle characteristics associated with ESPs [Roc01a].

2.6 Specification for a new DFE Workbench software tool

The specification for a new software based system that implements the DFE Workbench methodology has to be designed so as to solve all the problems identified in the previous section and implement all the requirements of the first version of the tool. Many definitions have been associated over time with the term specification in the context of computer-based systems (and software). Pressman indicates, *"a specification can be a written document, a graphical model, a formal mathematical model, a collection of usage scenarios, a prototype, or any combination of these"* [Pre00]. However, the system specification has to describe the function and performance of a computer-based system and the information that is input to and output from that system [Pre00].

Some of the functionality of the first version of the DFE Workbench is required in the new version. The requirements derived from the existing DFE Workbench software can be summarised as follows:

- The application has to be integrated in the CAD environment used by the designer and has to communicate with the CAD system to retrieve the necessary information on a product prototype.
- The software has to implement both IAS and SAM methodologies and they have to communicate with each other.
- The advisor agent and the knowledge base agent are required in the new system to help the designer in the process of evaluation and improvement of a product from an environmental point of view.

The rest of the requirements for the new DFE Workbench system can be categorised as follows:

1. Functional requirements
2. Non-functional requirements
3. Usability requirements

It should be noted that these requirements form an initial high-level specification as new requirements evolved with the test and validation of the tool throughout the project evolution.

Functional requirements

Functional requirements describe what is the DFE Workbench system supposed to do (its functionality) by describing the inputs into the system, the outputs expected from the system and the data that must be held in the system.

The inputs from the user to the DFE Workbench system can be summarised as follows:

- Information on components/assemblies is extracted from the CAD system and transferred to the DFE Workbench automatically when the user selects to save a component/assembly using an integrated DFE Workbench menu/toolbar/button from the CAD system.
- The user inputs the following information for the IAS tool at the component level: material, process, finishing, transport, distance, usage and end of life types.
- At the assembly level, the IAS tool requires the parent-child relationships between assemblies and subassemblies as an input (unless this information is available from the CAD system).
- The user inputs the following information for the SAM tool: joint information, fastener type, fastener number, disassembly tool, obstruction information and serviceable components.

The outputs required from the DFE Workbench system can be summarised as follows:

- The assembly structure (the bill of materials) displayed by the IAS tool in a form familiar to the designer (probably similar to the one used by the CAD system).
- Environmental information at both component and assembly level (available for selection from the bill of materials) displayed using tables and charts.
- Percentages and lists of hazardous, recyclable, recycled, biodegradable and sustainable content for an assembly.
- Structural information such as joints list for an assembly, fasteners and disassembly tools tables, component/subassembly removal time and disassembly route grouped in the form of charts and tables.
- Customisable reports generated from both IAS and SAM tools.
- All the information generated by the DFE Workbench reported to the web through Internet/Intranet (to make environmental data available along the supply chain e.g. recyclers need material information and dismantlers need to have quick access to disassembly information).

The data held in by the DFE Workbench can be grouped as follows:

- Predefined environmental and structural data required in the evaluation of a component/assembly.
- Specific environmental and structural data associated with components and assemblies created by the user.

Non-functional requirements

Non-functional requirements *"describe aspects of the system that are concerned with how well it provides the functional requirements"* [Ben99]. The non-functional requirements of the DFE Workbench system include the following:

- Portability (the DFE Workbench system should be operable in a different environment than the one it was implemented in with a minimum of configuration changes).
- Suitable for use in distributed design environments.
- The DFE Workbench system should make use of a robust client server database system.
- The database system should include a secure layer (access to the data held in the system should be allowed only on an username/password basis)
- Database consistency should be assured by the system (database update should be performed by the system using verified data only)
- Concurrent access should be allowed to the DFE Workbench system.

Usability requirements

Over the last ten years, one of the most important developments in the human-computer interaction (HCI) field has been the graphical user interface (GUI) [Edw95, Haz96, Wit93, War00]. The user interface is already considered a critical component of any software package [Gre96, Hua02]. An important aspect of a good user interface is its usability, as nowadays a good program is not only a program that works but is also easy to learn [Tol95, Cor97]. Involving the end-user in the design and evaluation process has been used as a method for developing usable software (user-centred design). The ISO-9241 standard (1998), which is part of the international standards on usability and user-centred design, defines usability as, *"the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use"* [ISO98].

Usability requirements *"are those that will enable us to ensure that there is a good match between the system that is developed and both the users of that system and the tasks that they will undertake when using it"* [Ben99]. The requirements of the users of the DFE Workbench system must be considered in order to reduce errors and maximize the satisfaction of the users with the system [Ben99]. The involvement of a team of designers (of mechanical and industrial engineering background) as well as the continuous testing of the tool with industrial partners throughout the development of the new version of the DFE Workbench software will ensure the delivery of an easy to use final product. However, a set of initial usability requirements can be established, which include the following:

- Dialogues should be consistent and require minimal user input (selecting from a list rather then having to enter a value, using default values, and reusing information that can be generated automatically rather then having to enter or re-enter it).
- Interface elements e.g. menus, toolbars, panels and buttons used for the CAD integration as well as those present in the tool itself should be consistent and easy to understand.
- Error messages should include a section explaining how to recover from the error.
- All frames must be consistent, intuitive (easy to learn), simple, efficient and aesthetically pleasing. The users should be prevented from performing an inappropriate task rather then allowing the task and then providing the user with a message that explains why the action was impossible.

All requirements have to be analysed and mapped into a model of the new version of the DFE Workbench software.

2.7 Conclusions

This chapter has reviewed some of the existing tools and methodologies that support the development of environmentally superior products. The DFE Workbench is one of the proposed DFE tools, which was developed to support the design process throughout its phases. Advantages of the tool include the support for evaluating and improving both environmental and structural data associated with a virtual prototype and the integration of the software based version with a CAD system. The manual DFE Workbench methodology as well as the desktop version of the software-based tool were described. The deficiencies of the first version of the DFE Workbench software have been identified and the specification for a new version of the software tool has been presented by summarising the requirements under four headings as follows:

1. Requirements derived from the existing DFE Workbench system.
2. Functional requirements.
3. Non-functional requirements.
4. Usability requirements.

The design and development of the new software based DFE Workbench will incorporate the following:

- *Event-driven programming* – the user controls the application's tasks via GUI events through entries such as keystrokes and mouse clicks.
- *User-centred design* – the users are involved throughout the development process.

This approach to the development of the DFE Workbench software will ensure that the application developed is user-friendly and simple to use.

The development of the new software-based DFE Workbench is mainly based on the specification document presented in this chapter. However, new requirements resulted from the collaboration with industrial partners from the electronic and automotive sector have been included in time in the DFE Workbench system. The next chapter describes the latest version of the DFE Workbench software from both technical and functional perspectives.

Chapter 3

The DFE Workbench Software

3.1 Introduction

This chapter describes the design and development of a new DFE Workbench software tool that incorporates all the requirements presented in chapter two. The new DFE Workbench is based on the same DFE manual methodology as the first version of the tool. Technical improvements brought to the new tool were completed by new features added as a result of the involvement of industrial partners in the testing and evolution of the tool throughout the project. A new application suitable for distributed design environments was developed from scratch using Java technology and Oracle as a database server. The new tool was integrated in two CAD systems i.e. SolidWorks 2000 and Pro/Engineer 2001.

This chapter starts with a description of the new DFE Workbench tool and continues by presenting the three levels of the DFE Workbench and the relationship among them. The development of the tool occurred in many stages, each of them being influenced by suggestions from industrial partners from the electronic and automotive sector. This chapter presents the latest version of the DFE Workbench software from both technical and functional perspectives.

3.2 The new version of the DFE Workbench tool

The DFE Workbench supports the synthesis, evaluation, prioritisation and improvement of both environmental impact and structural data associated with an emergent virtual product [Roc01a, Roc01b, Man02a, Man02c]. The software tool was built based on a manual DFE methodology, which was designed and developed first. The manual DFE methodology is largely based on using special charts and reference information in a structured manner to evaluate and improve an emergent design. The DFE Workbench software is a Java based software application, which effectively automates processes associated with the manual methodology; the software tool takes advantage of the portability and flexibility of Java to provide a powerful yet easy to use software tool.

The DFE Workbench tool is configured for the following three levels:

- The *DFE Workbench Desktop* is the core application that resides on the designers' CAD workstations and performs synthesis, evaluation, prioritisation and improvement of the environmental and structural data associated with the candidate design through features such as knowledge base agent, prioritisation module, advisor agent and report generator.
- The *DFE Workbench Enterprise* is the application that resides at product system level and does not need a CAD system to be used. It performs evaluation, prioritisation and improvement of the environmental and structural status of the entire product system.
- The *DFE Workbench Global* is a web-based application that allows easy access to the environmental and structural data generated by the DFE Workbench Desktop and Enterprise and allows collaboration among lifecycle actors. A user with the necessary access rights can download up-to-date information from the database and upload reports on the web to make them available to other users.

All three levels of the DFE Workbench are linked directly to an Oracle database server in a distributed environment. The collaboration process between the design team and the system engineer/product manager can take place over a computer network and it is facilitated by instant access to the latest structural and environmental data from an emergent virtual prototype. A web server is connected to the database server to allow access to data for DFE Workbench Global via the Intra/Internet. Figure 3.1 presents the system described. The representation is simplified as the DFE Workbench Enterprise can run on multiple workstations.

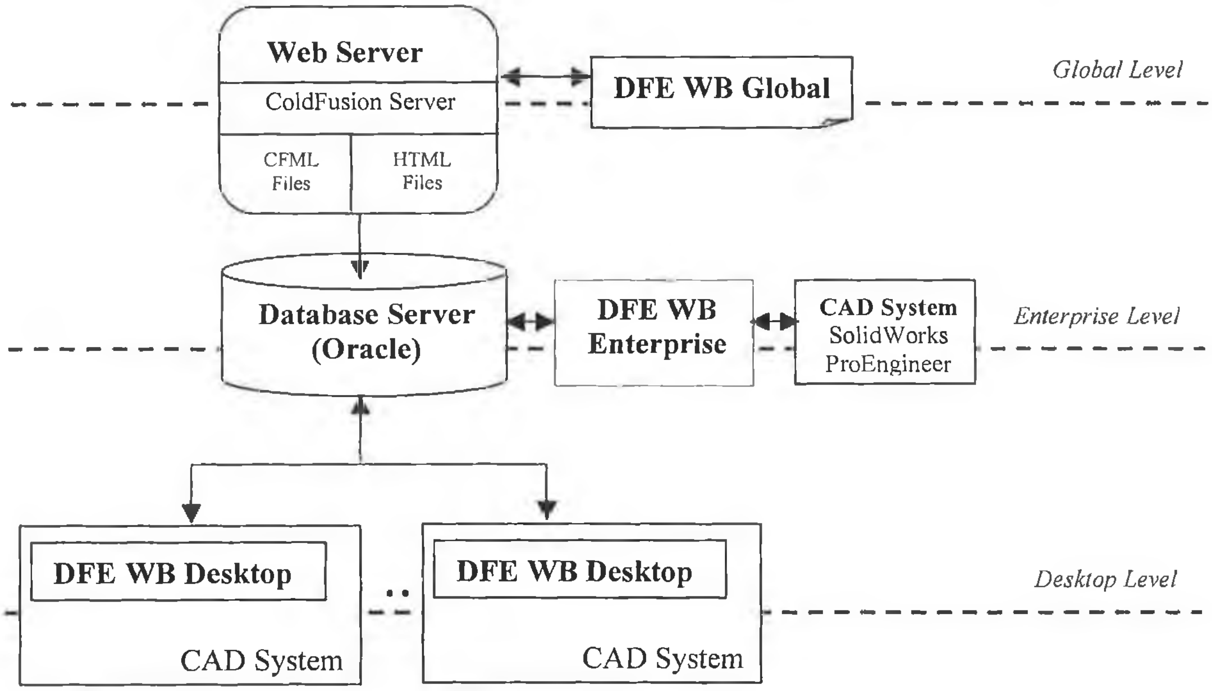


Figure 3.1 The DFE Workbench system configuration

The DFE Workbench Desktop is the level of the DFE Workbench that is completely integrated in the CAD system used by the designer. The main difference between the DFE Workbench Desktop and Enterprise is that the first one can only be accessed from the CAD system used by the designer while the second level of the DFE Workbench can run both inside and outside the CAD system. The DFE Workbench Desktop and Enterprise have the benefits of being multiplatform portable because of the use of Java technology as well as a robust database (Oracle Java Database Connectivity).

3.3 The DFE Workbench software architecture

The software architecture provides a holistic view of the DFE Workbench system by describing the structure of the data and program components [Pre00]. The architecture of any computer-based system indicates the structure and properties of the components that comprise that system and the interrelationships among all architectural components [Pre00]. The DFE Workbench architecture is data centred as the Oracle database resides at the centre of the system and is accessed by other components that update, add, delete or otherwise modify the data. Input data are transformed through a series of components into output data. The program structure is classic: a "main" program invokes a number of program components that in turn may invoke other components (call and return architecture) [Pre00].

3.3.1 Database structure

All the data manipulated by the DFE Workbench are stored in two Oracle *tablespaces* as follows:

- The first one consists of all the *data tables* containing environmental information such as ECO indicators associated with different materials or processes and removal times associated with different disassembly tools.
- The second one consists of all the *working tables* containing environmental and structural data calculated for different prototypes. Information stored in the data tables is used to calculate the figures in the working tables.

The DFE Workbench Desktop and Enterprise can both store new data into the working tables while the DFE Workbench Global is only displaying and reporting information already calculated and stored in the working tables. The Enterprise level of the DFE Workbench has added functionality by providing writing capabilities for the data tables as well as for the working tables. Figure 3.2 presents the relationship between the DFE Workbench software and the Oracle databases.

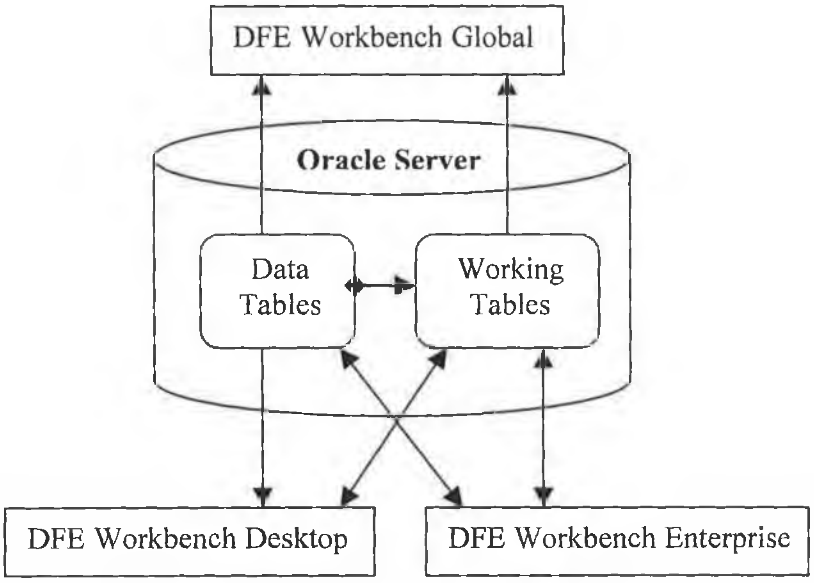


Figure 3.2 The relationship between the Oracle database and the three levels of the DFE Workbench

The data tables hold environmental information for all the life stages i.e. material, process, end of life, use and transportation and structural information such as fasteners, disassembly tools and disassembly times. The working tables hold information about users and passwords and all the data associated with an assembly divided in two main areas:

1. Environmental data such as ECO indicators at assembly/subassembly/component level, mass properties and life stages values.
2. Structural data such as fasteners between various components, disassembly tools and removal times for subassemblies and components.

All the data in the working tables are calculated based on the information stored in the data tables. That is the reason why the relational model was used to build the tables. Figure 3.3 presents the relationship between the data tables and the working tables at the environmental level. Each of the life cycle stages i.e. raw material selection, process, transport, end of life and use is uniquely identified through a key field. The structure of the database at the structural level is similar to the one presented in figure 3.3. The key fields are the identification number of the fastener used for a joint and of the tool used for disassembly.

3.3.2 Database connection

The DFE Workbench is linked to the Oracle database using Java Database Connectivity (JDBC), which is the Java version of Open Database Connectivity (ODBC) [Ben99]. This connection required the use of classes from the *java.sql* package such as (see figure 3.4):

- *Connection* – used to create a connection to the database
- *Statement* – used to execute a SQL (Structured Query Language) statement
- *ResultSet* – used to manipulate the database result set generated by executing a statement that queries the database (the ResultSet object can be iterated to retrieve each row in turn and to extract the values for each column)

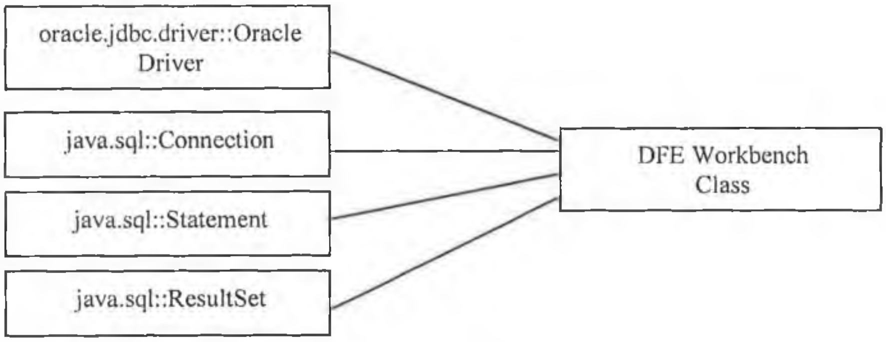
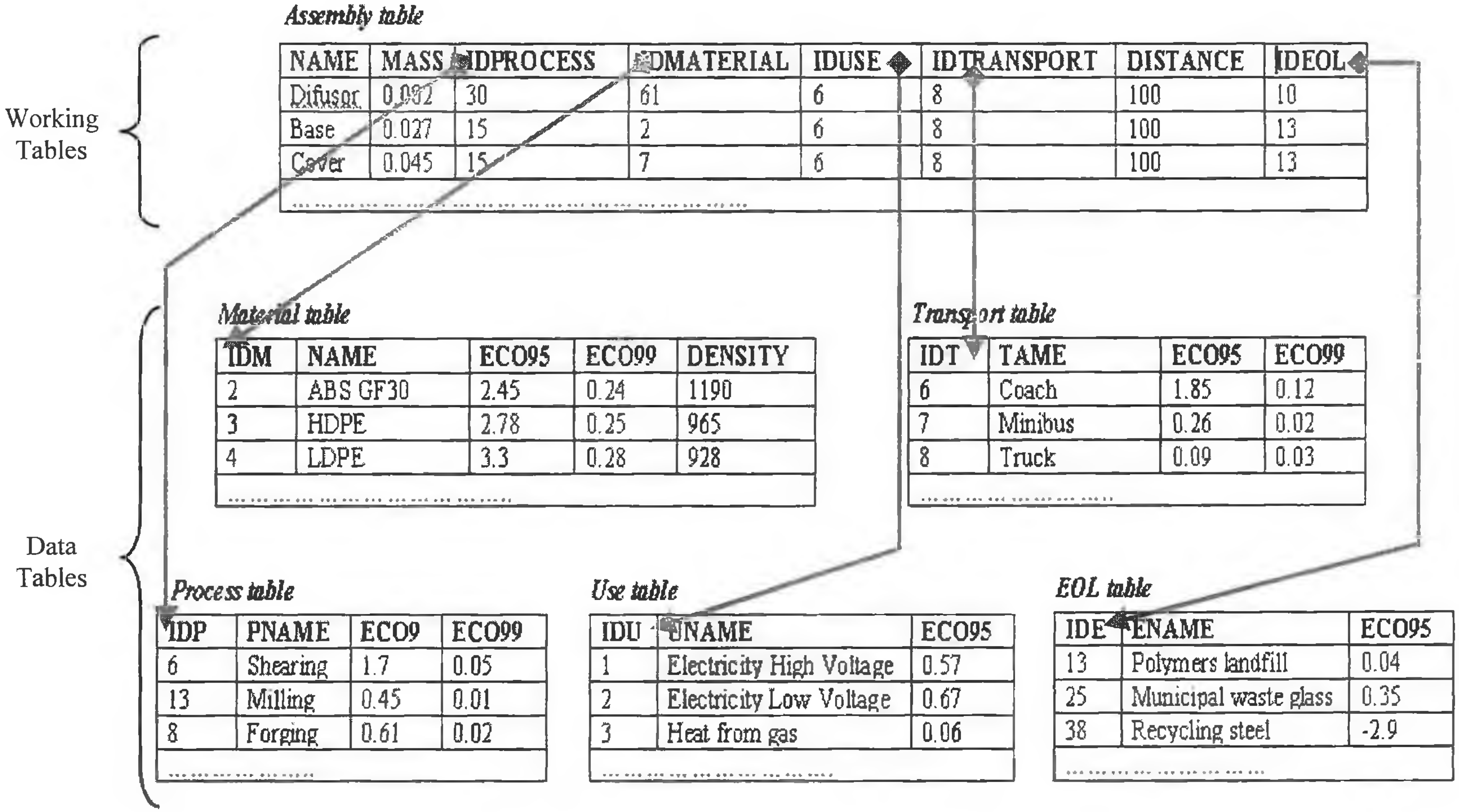


Figure 3.4 A DFE Workbench class and classes from other packages relationship [Ben99]

The *'oracle.jdbc.driver::Oracle'* needs to be loaded to initialise the Java SQL framework. Oracle Thin for Java applets and applications, which is compliant with JDBC version 1.22, has been chosen. SQL statements are used to extract information from the database and to add, delete or update records in the database.



Assembly table

NAME	MASS	IDPROCESS	IDMATERIAL	IDUSE	IDTRANSPORT	DISTANCE	IDEOL
Difusor	0.092	30	61	6	8	100	10
Base	0.027	15	2	6	8	100	13
Cover	0.045	15	7	6	8	100	13

Material table

IDM	NAME	ECO95	ECO99	DENSITY
2	ABS GF30	2.45	0.24	1190
3	HDPE	2.78	0.25	965
4	LDPE	3.3	0.28	928

Transport table

IDT	TAME	ECO95	ECO99
6	Coach	1.85	0.12
7	Minibus	0.26	0.02
8	Truck	0.09	0.03

Process table

IDP	PNAME	ECO9	ECO99
6	Shearing	1.7	0.05
13	Milling	0.45	0.01
8	Forging	0.61	0.02

Use table

IDU	UNAME	ECO95
1	Electricity High Voltage	0.57
2	Electricity Low Voltage	0.67
3	Heat from gas	0.06

EOL table

IDE	ENAME	ECO95
13	Polymers landfill	0.04
25	Municipal waste glass	0.35
38	Recycling steel	-2.9

Figure 3.3 The relationship between the working and data tables at environmental level

3.3.3 Software structure

The DFE Workbench software was built using the object-oriented programming model, approach that improves the maintenance, reusability and modifiability of the software. The programming language chosen is Java, a portable language that can run on any Java-enabled platform e.g. Microsoft Windows, Linux, Unix, Solaris. Java is an object-oriented language designed for use in distributed applications on corporate networks and the Internet. The programming language used to implement the DFE Workbench was selected based on the following positive characteristics of Java:

- Object-oriented
- Platform-independent
- Multi-threaded
- Dynamic general purpose programming environment
- Robust
- Suitable for any complex distributed network

The objects manipulated by the DFE Workbench are described in figure 3.5.

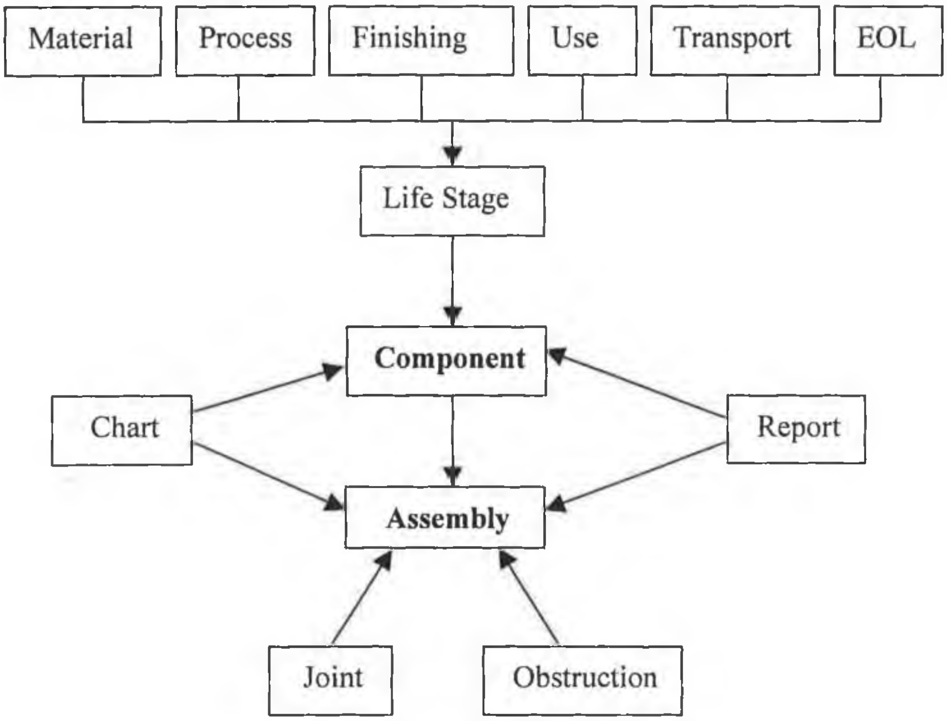


Figure 3.5 The DFE Workbench objects

The DFE Workbench software works on the following two levels:

- Component level – the component object is characterized by life stages such as material, process, finishing, use, transport and End Of Life (EOL)

- Assembly level – the assembly object is formed out of components, joints between various components and obstructions

Charts and reports are used to display information on both components and assemblies. The objects presented in figure 3.5 have generated the class definitions with fields and methods description and inheritance relationships. Annex A presents the Java class hierarchy used to implement the DFE Workbench software.

Figure 3.6 presents the program structure of the DFE Workbench i.e. the organization of modules or program components implying a hierarchy of control. The tree-like diagram notation has been chosen to represent the control hierarchy.

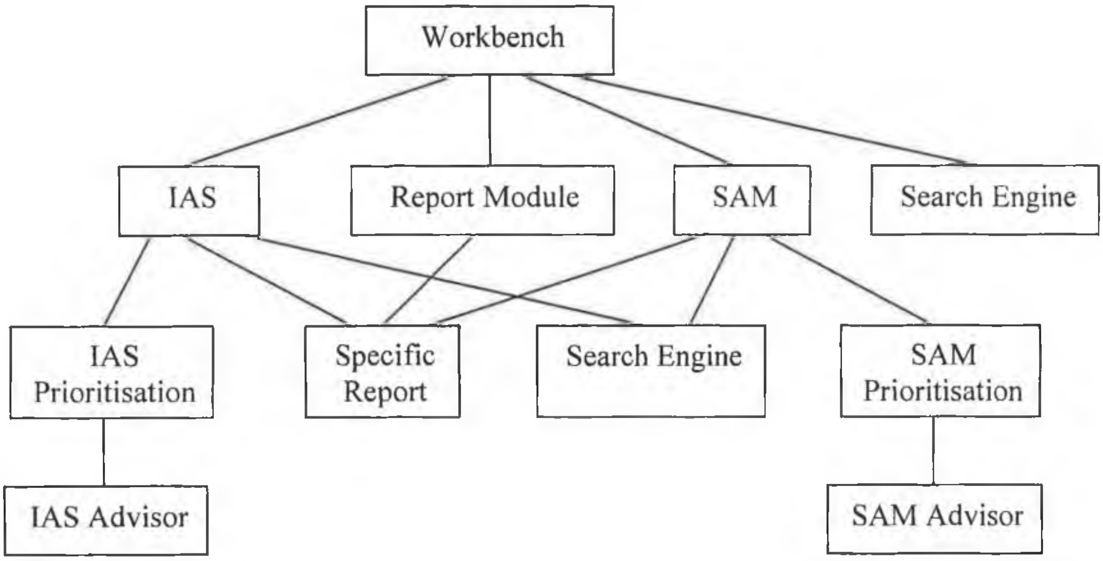


Figure 3.6 The DFE Workbench program structure

The diagram presented in figure 3.6 shows what modules are directly controlled by other modules (fan-out) and how many modules directly control a given module (fan-in). The *Workbench* module is superordinate to modules such as IAS (Impact Assessment System), SAM (Structure Assessment Method), Report Module and Search Engine. The Specific Report module is subordinate to modules such as IAS, Report Module and SAM. The program structure or control hierarchy also shows the visibility and connectivity of the software architecture. Modules that can be accessed by a given component are indicated through visibility. Connectivity indicates the modules or program components that are directly invoked by a given component.

The functional specification for the new DFE Workbench software tool has been summarized into the logic scheme presented in figure 3.7. The sequence of processes as well as the occurrence of decisions or operations is presented.

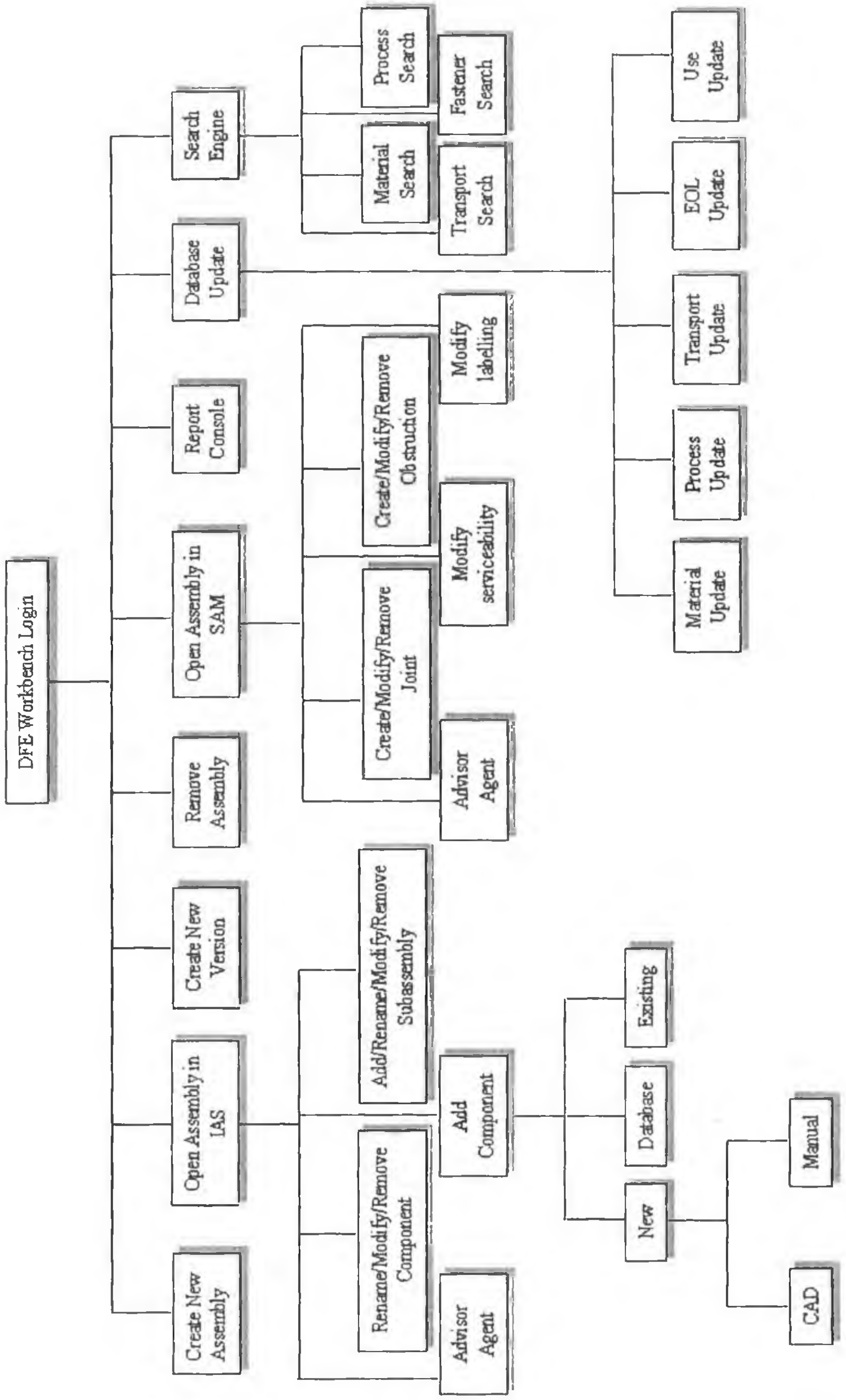


Figure 3.7 The hierarchy of processes in the DFE Workbench

3.4 The DFE Workbench Desktop

The DFE Workbench Desktop resides on the designers' workstation and is completely integrated into the CAD system. Figure 3.8 shows the structure of this level.

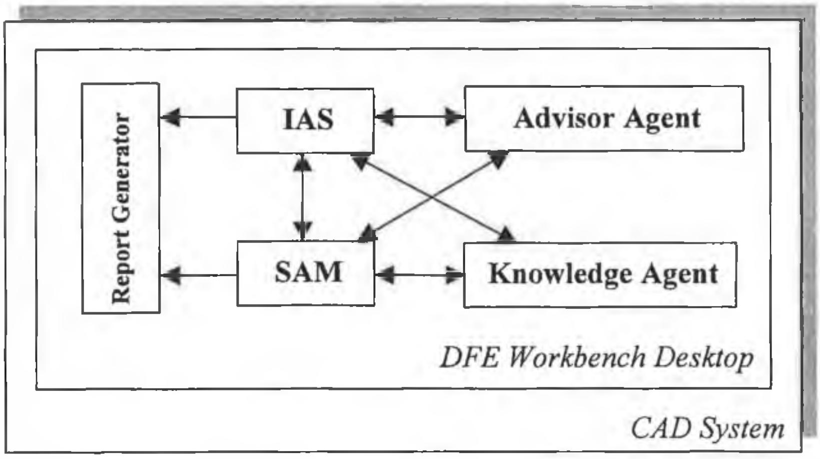


Figure 3.8 The DFE Workbench Desktop Structure

The DFE Workbench Desktop consists of five modules as follows [Roc01a, Man02a]:

- The Impact Assessment System
- The Structure Assessment Method
- The Advisor Agent
- The Knowledge Agent
- The Report Generator

A username and a password are required to log into the DFE Workbench Desktop and have access to any of the modules mentioned above. Figure 3.9 shows the DFE Workbench Login window.

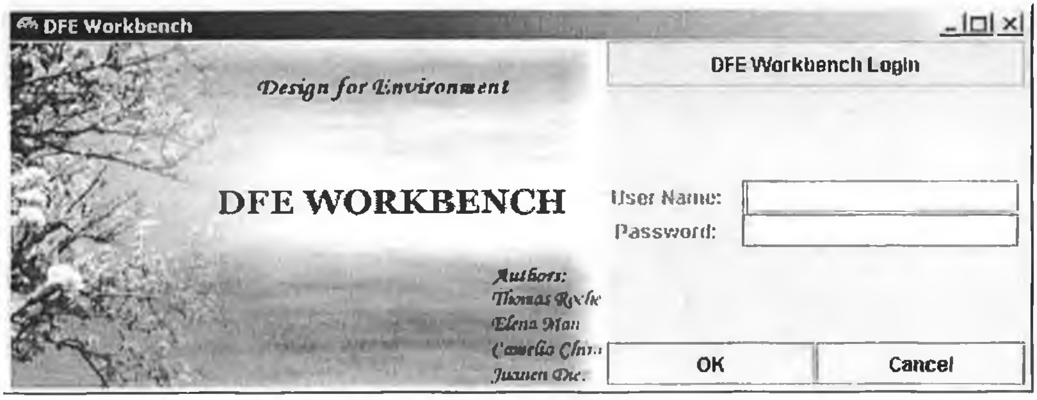


Figure 3.9 The DFE Workbench Login Window

The main frame of the DFE Workbench presented in figure 3.10 was built using a desktop pane, which allows multiple windows to be opened at the same time (see appendix 4). The benefit is the easy comparison that can be performed between different assemblies/subassemblies by having them opened at the same time in one of the two existing modes: Impact Assessment System (environmental data) and Structure Assessment Method (structural data). An icon toolbar is used to offer shortcuts for the most used actions included in the menu such as creating a new assembly/subassembly, opening an assembly/subassembly in one of the two modes mentioned and viewing reports.

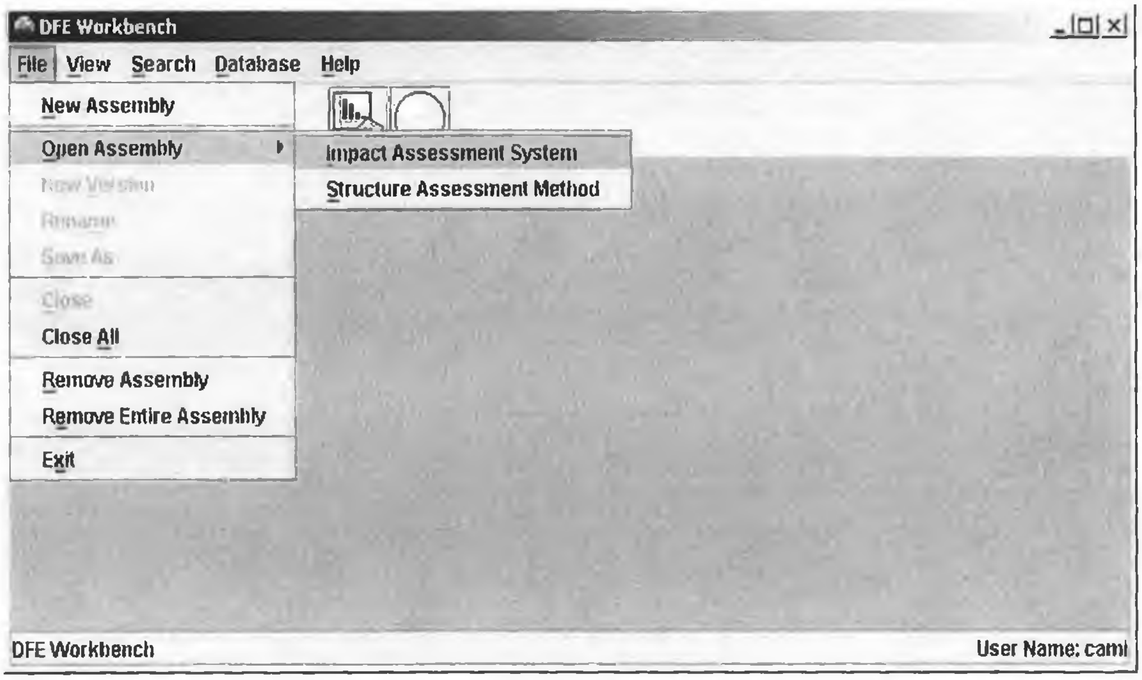
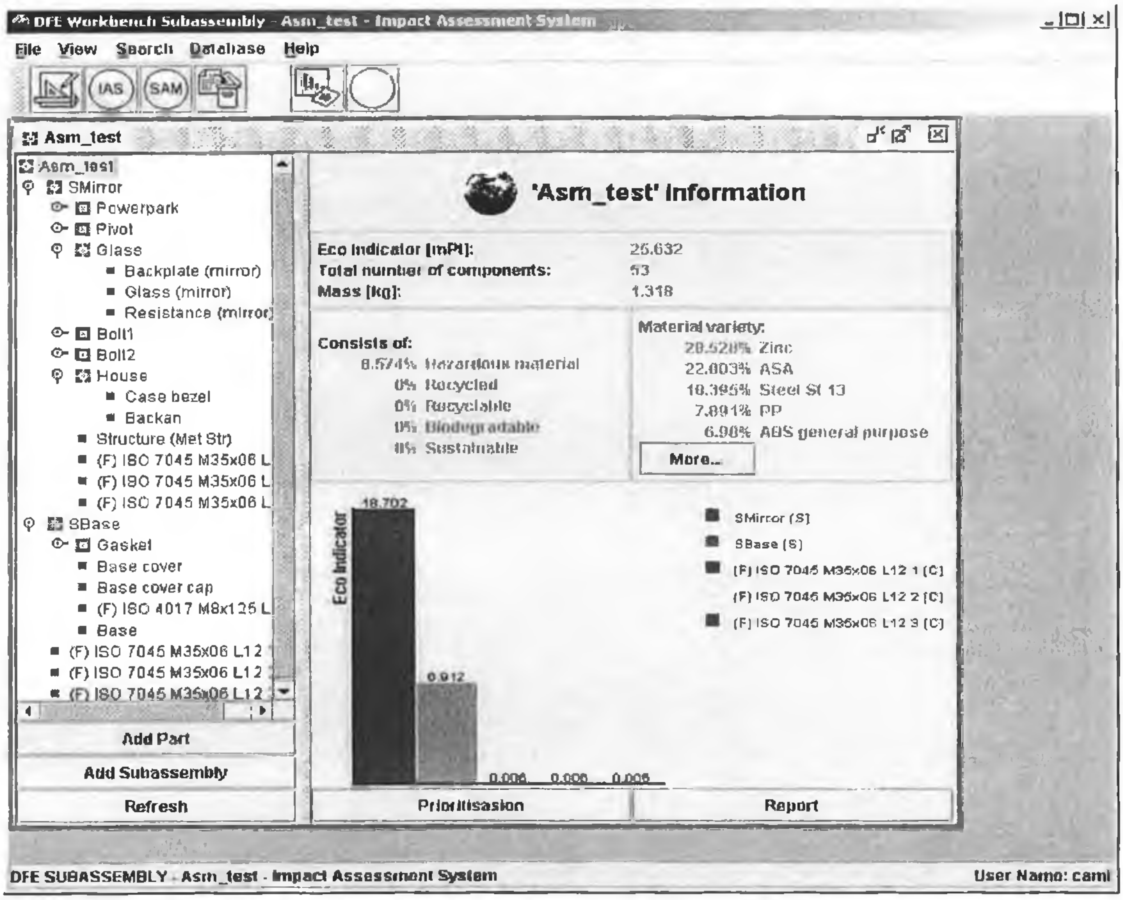


Figure 3.10 The DFE Workbench main window

The *Impact Assessment System* (IAS) is based on an abridged quantitative approach to LCA, performing synthesis, evaluation, prioritisation and improvement of environmental data derived from the virtual prototype within the CAD environment. Impact data can be evaluated and improved for each individual component, for a subassembly or for the entire product system. The IAS calculates, evaluates, improves and reports data such as material type and variety, material intensity of type/s (mass), environmental impact of the candidate design and percentages of recycled, hazardous, recyclable, biodegradable and sustainable material.

Figure 3.11 shows the IAS window for an assembly called 'Asm_test'. All the environmental information displayed for any assembly/subassembly is completed by a bar chart showing the ECO indicator for each of the components and subassemblies starting with the highest ECO indicator value.


DFE Workbench Subassembly - Asm_test - Impact Assessment System
File View Search Database Help
IAS
SAM
Asm_test
Add Part
Add Subassembly
Refresh
More...
Prioritisasion
Report
Eco Indicator
18.702
0.912
0.006
0.006
0.006
■ SMirror (S)
■ SBase (S)
■ (F) ISO 7045 M35x06 L12 1 (C)
(F) ISO 7045 M35x06 L12 2 (C)
■ (F) ISO 7045 M35x06 L12 3 (C)
DFE SUBASSEMBLY - Asm_test - Impact Assessment System
User Name: cami

- Asm_test
  - SMirror
    - Powerpark
    - Pivot
    - Glass
      - Backplate (mirror)
      - Glass (mirror)
      - Resistance (mirror)
    - Bolt1
    - Bolt2
    - House
      - Case bezel
      - Backan
    - Structure (Met Str)
    - (F) ISO 7045 M35x06 L
    - (F) ISO 7045 M35x06 L
    - (F) ISO 7045 M35x06 L
  - SBase
    - Gasket
    - Base cover
    - Base cover cap
    - (F) ISO 4017 M8x125 L
    - Base
  - (F) ISO 7045 M35x06 L12
  - (F) ISO 7045 M35x06 L12
  - (F) ISO 7045 M35x06 L12

'Asm_test' Information

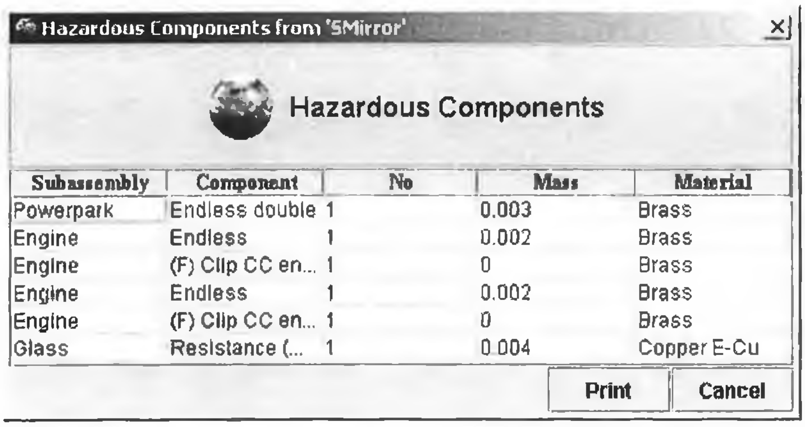
Eco Indicator [mPt]: 25.632
Total number of components: 53
Mass [kg]: 1.318

Consists of:
8.574% Hazardous material
0% Recycled
0% Recyclable
0% Biodegradable
0% Sustainable

Material variety:
28.528% Zinc
22.003% ASA
18.395% Steel St 13
7.891% PP
6.98% ABS general purpose

Figure 3.11 The IAS Window

The IAS window not only shows the percentage of hazardous content for an assembly but also generates a list with the components where hazardous materials were located (see figure 3.12). The same action can be performed for getting the exact recycled, recyclable, biodegradable or sustainable content of an assembly as well as for getting all components made from a specified material. This feature was implemented in the DFE Workbench after suggestions made by an industrial partner from the automotive sector.

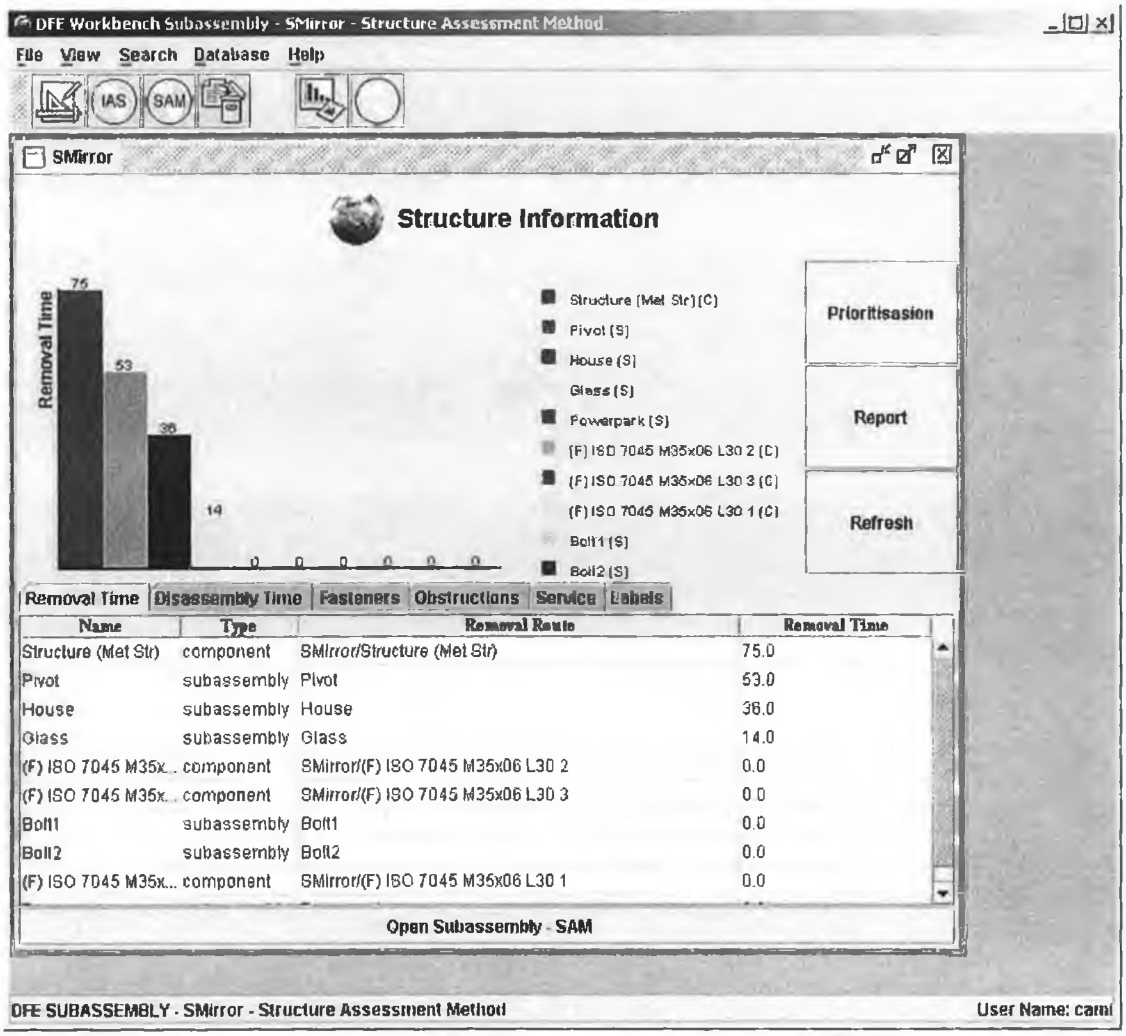


Hazardous Components

Subassembly	Component	No	Mass	Material
Powerpark	Endless double	1	0.003	Brass
Engine	Endless	1	0.002	Brass
Engine	(F) Clip CC en...	1	0	Brass
Engine	Endless	1	0.002	Brass
Engine	(F) Clip CC en...	1	0	Brass
Glass	Resistance (...	1	0.004	Copper E-Cu

Figure 3.12 Hazardous Components List

The *Structure Assessment Method* (SAM) focuses on the structure of the emergent virtual prototype in an attempt to enhance product structural characteristics in the DFE context. SAM is a complex methodology, which quantitatively measures and records data such as number and types of fasteners, number and types of tools required for disassembly, total standard disassembly time, standard part removal time and route, percentage of serviceable components and material compatibility (taking into account fasteners). The coupling between all variables is managed and recorded by the DFE Workbench. Figure 3.13 shows the SAM window for an assembly.

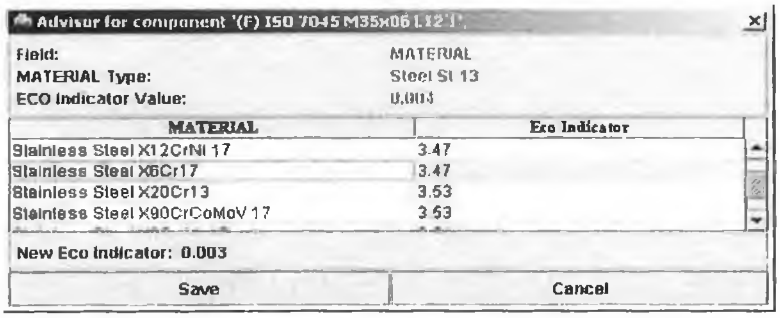


Name	Type	Removal Route	Removal Time
Structure (Met Str)	component	SMirror/Structure (Met Str)	75.0
Pivot	subassembly	Pivot	53.0
House	subassembly	House	36.0
Glass	subassembly	Glass	14.0
(F) ISO 7045 M35x...	component	SMirror/(F) ISO 7045 M35x06 L30 2	0.0
(F) ISO 7045 M35x...	component	SMirror/(F) ISO 7045 M35x06 L30 3	0.0
Bolt1	subassembly	Bolt1	0.0
Bolt2	subassembly	Bolt2	0.0
(F) ISO 7045 M35x...	component	SMirror/(F) ISO 7045 M35x06 L30 1	0.0

Figure 3.13 The SAM Window

The SAM window consists of tables presenting information about the removal time for subassemblies and components, the disassembly time for the assembly, fasteners and disassembly tools. The tables are completed by a bar chart presenting the removal times for all subassemblies and components starting with the highest value.

The *Advisor Agent* has two functions: firstly to prioritise variables generated by the IAS and SAM tools; secondly the Advisor Agent actively gives advice to the designer on alternative solutions to enhance either the environmental impact or structural characterises of the emergent design, taking into account coupling between the variables evaluated for each method. For example the advisor agent may suggest alternative materials or processes to reduce the environmental impact of a product. The advisor agent becomes active after the prioritisation module is called for an assembly/subassembly in either IAS or SAM tool. Figures 3.14 and 3.15 present the Advisor Agent for the IAS and SAM respectively. The IAS Advisor gives the designer compatible alternatives with lower environmental impact for the life stage that has the highest ECO indicator but it also allows the designer to choose alternative values for the remaining life stages. The SAM Advisor aims at reducing the removal time for a component or subassembly by providing the designer with alternative fasteners and disassembly tools for the existing joints. Information such as the current removal time and route is also available.



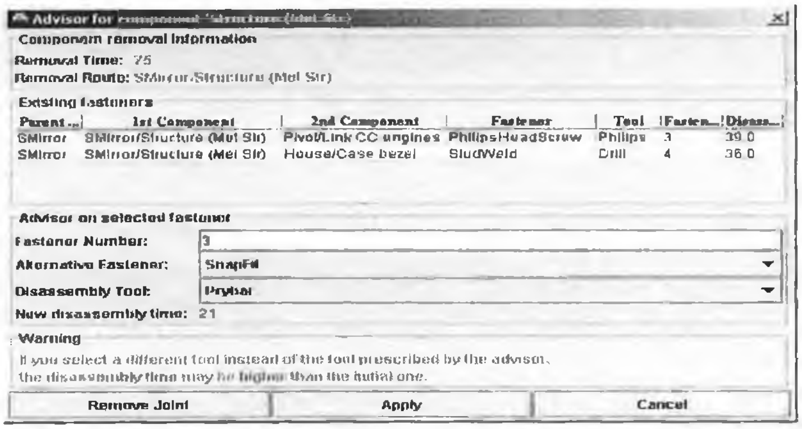
Advisor for component '(F) ISO 7045 M35x06 L12'1'

Field: MATERIAL
MATERIAL Type: Steel St 13
ECO Indicator Value: 0.003

MATERIAL	Eco Indicator
Stainless Steel X12CrNi 17	3.47
Stainless Steel X6Cr17	3.47
Stainless Steel X20Cr13	3.53
Stainless Steel X90CrCoMoV 17	3.53

New Eco Indicator: 0.003

Figure 3.14 The IAS Advisor



Advisor for component 'Structure (Mel Str)'

Component removal information
Removal Time: 75
Removal Route: SMirror/Structure (Mel Str)

Existing fasteners

Parent ...	1st Component	2nd Component	Fastener	Tool	Fasten...	Disass...
SMirror	SMirror/Structure (Mel Str)	Pivot/Link CC engines	PhilipsHeadScrew	Philips	3	39.0
SMirror	SMirror/Structure (Mel Str)	House/Case bezel	StudWeld	Drill	4	36.0

Advisor on selected fastener
Fastener Number: 3
Alternative Fastener: SnapFit
Disassembly Tool: Prybar
New disassembly time: 21

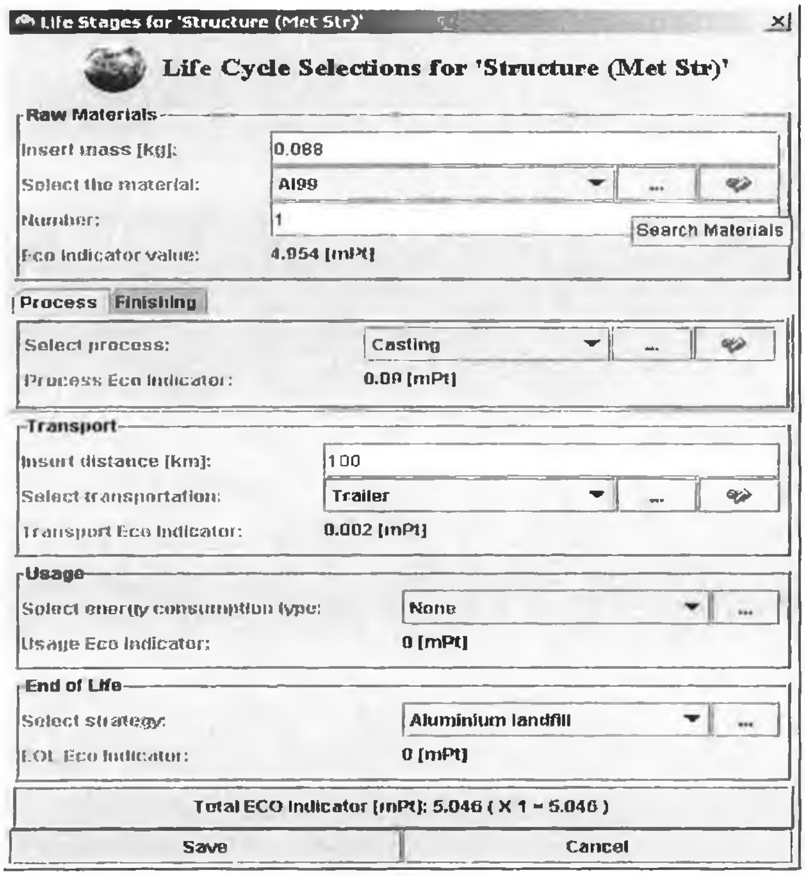
Warning
If you select a different tool instead of the tool prescribed by the advisor,
the disassembly time may be higher than the initial one.

Figure 3.15 The SAM Advisor

Both IAS and SAM Advisor Agents can modify the working tables when the designer chooses to save the new selected values. One important characteristic of the Advisor Agent is that it does not constrain the designer in any way. The designer is free to decide what he/she considers to be the optimal solution for the candidate design. The Advisor Agent communicates with IAS and SAM modules in both ways meaning that data already calculated is extracted from the working tables and new data is stored in the database modifying environmental and structural information when a suggestion given by the Advisor Agent is accepted by the designer.

The *Knowledge Agent* provides advice to the designer in a consultative mode. For example the designer can use the Knowledge Agent to find a material with specified mechanical properties. The Knowledge Agent will provide the designer with a prioritised list of all those materials that match the specified criteria. A search engine is used to find all the information requested. Search criteria can be defined for materials, processes, transportations, fasteners and disassembly tools. Therefore, the Knowledge Agent searches the data tables from the Oracle database for those data that meet all specified conditions. The Knowledge Agent can become active in both IAS and SAM modules as follows:

- IAS: when adding a new component to an assembly/subassembly and a specific material, process or transportation is looked for (see figure 3.16).
- SAM: when creating a new joint between two components and a specific fastener or disassembly tool is looked for (see figure 3.17).


Life Stages for 'Structure (Met Str)'
Life Cycle Selections for 'Structure (Met Str)'
Process
Finishing
...
Search Materials
Save
Cancel

Raw Materials
Insert mass [kg]: 0.088
Select the material: Al99
Number: 1
Eco Indicator value: 4.954 [mPt]

Select process: Casting
Process Eco Indicator: 0.09 [mPt]

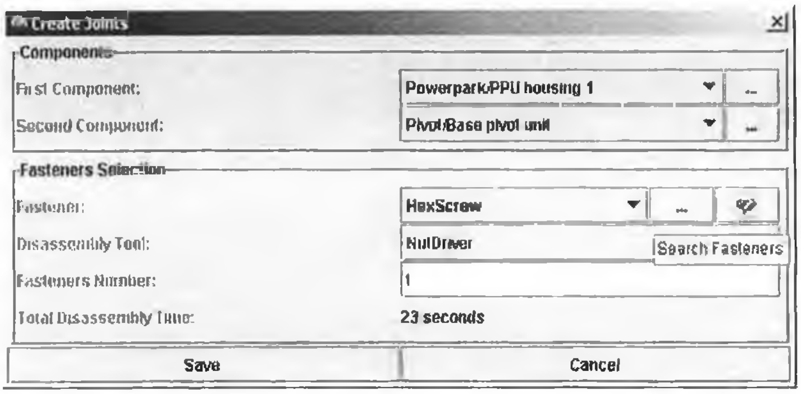
Transport
Insert distance [km]: 100
Select transportation: Trailer
Transport Eco Indicator: 0.002 [mPt]

Usage
Select energy consumption type: None
Usage Eco Indicator: 0 [mPt]

End of Life
Select strategy: Aluminium landfill
EOL Eco Indicator: 0 [mPt]

Total ECO Indicator [mPt]: 5.046 (X 1 = 5.046)

Figure 3.16 The IAS Knowledge Agent Call



Create Joints

Components

First Component: Powerpark/PPU housing 1

Second Component: Pivot/Base pivot unit

Fasteners Selection

Fastener: HexScrew

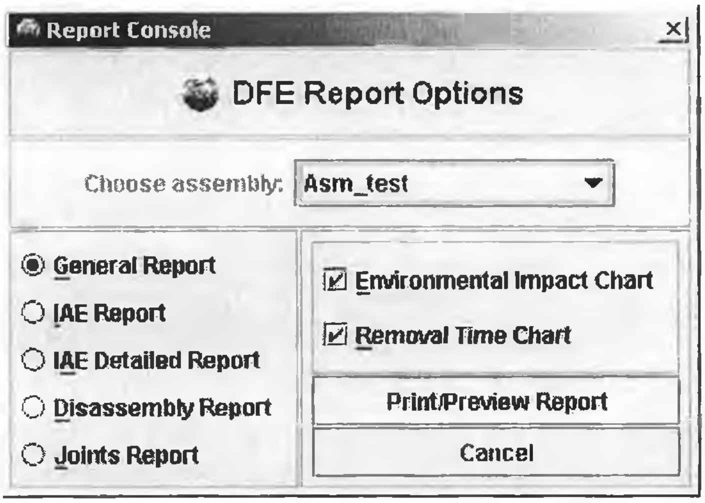
Disassembly Tool: NutDriver

Fasteners Number: 1

Total Disassembly Time: 23 seconds

Figure 3.17 The SAM Knowledge Agent Call

The *Report Generator* automatically generates reports on the product designed by the user. Reports present the environmental and structural data generated by the IAS and SAM tools in the form of charts and tables. All reports are built using a Java based application and a direct connection to the Oracle database. Figure 3.18 shows the Report Console window available from both IAS and SAM tools.



Report Console

DFE Report Options

Choose assembly: Asm_test

● General Report
○ IAE Report
○ IAE Detailed Report
○ Disassembly Report
○ Joints Report

☑ Environmental Impact Chart
☑ Removal Time Chart

Figure 3.18 The DFE Workbench Report Console

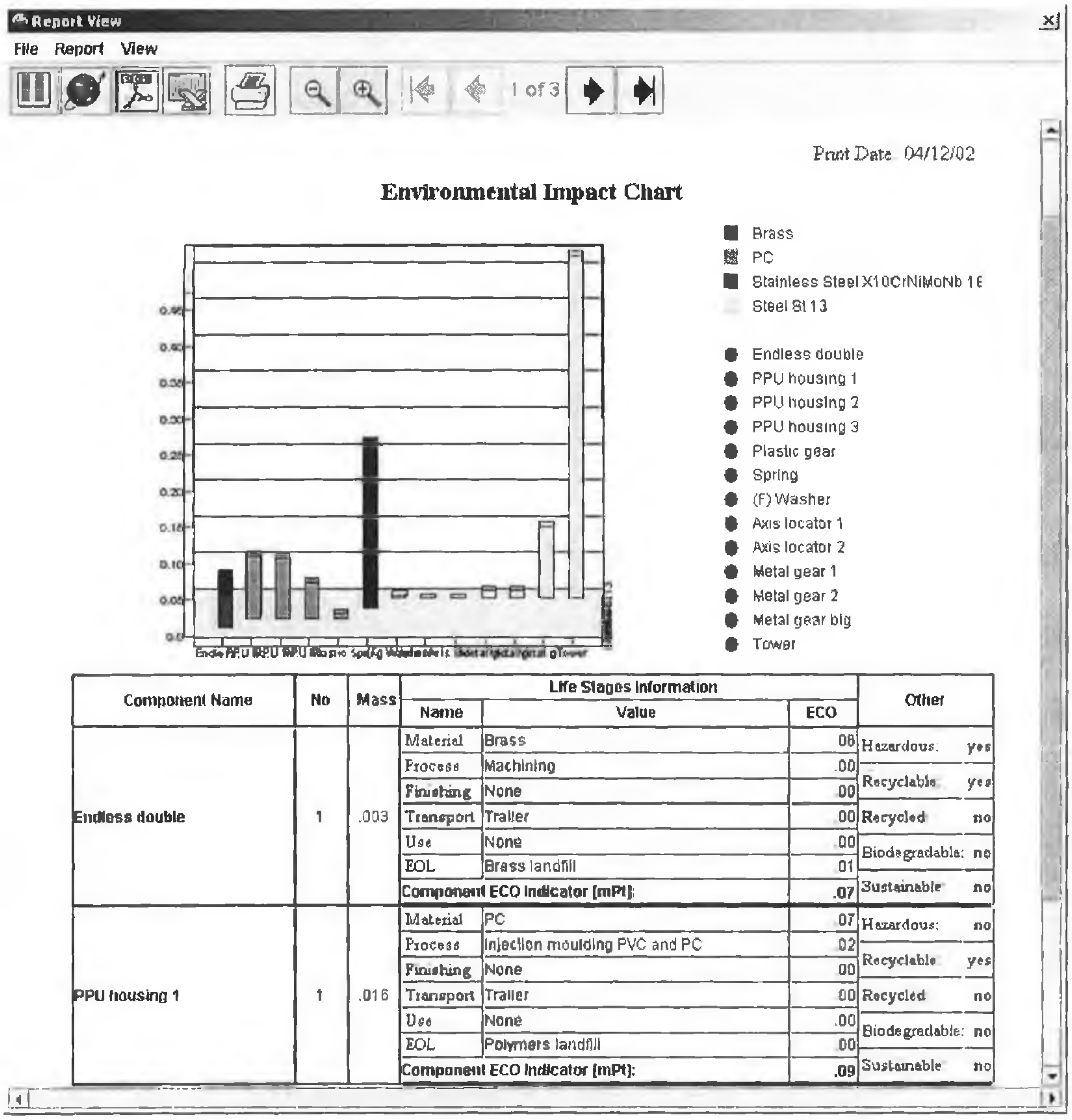
There are three main types of reports that can be previewed or printed using the Report Generator as follows:

- Reports containing general information grouped in tables and charts on both environmental and structural data associated with an assembly/subassembly.
- Reports containing detailed information generated by the IAS module such as the ECO indicator for the whole assembly as well as for contained subassemblies and components, mass properties, life stages values and hazardous components; this

type of report can be generated with or without the environmental impact chart grouped on the material type of the component.

- Reports containing detailed information generated by the SAM module such as joints between different components, fasteners and disassembly tools used, removal times and disassembly routes for subassemblies and components, and the total disassembly time for the entire assembly; this type of report can be generated with or without the removal time chart.

Figure 3.19 presents a detailed report containing IAS information associated with an assembly.



Print Date 04/12/02

Environmental Impact Chart

Component Name	No	Mass	Life Stages Information			Other
			Name	Value	ECO	
Endless double	1	.003	Material	Brass	.08	Hazardous: yes
			Process	Machining	.00	Recyclable yes
			Finishing	None	.00	
			Transport	Trailer	.00	Recycled no
			Use	None	.00	Biodegradable: no
			EOL	Brass landfill	.01	
			Component ECO Indicator [mPt]:		.07	Sustainable no
PPU housing 1	1	.016	Material	PC	.07	Hazardous: no
			Process	Injection moulding PVC and PC	.02	Recyclable yes
			Finishing	None	.00	
			Transport	Trailer	.00	Recycled no
			Use	None	.00	Biodegradable: no
			EOL	Polymers landfill	.00	
			Component ECO Indicator [mPt]:		.09	Sustainable no

Figure 3.19 Detailed IAS Report

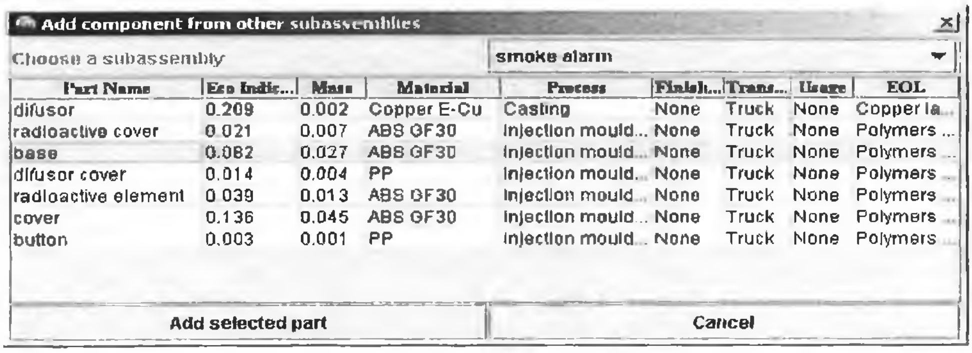
The Java based report viewer presented in figure 3.28 allows easy manipulation of any report through features such as zoom in, zoom out, page by page view and chart rotation. The system reports created by the Report Generator can be printed out or exported to other file formats such as text documents, HTML files, PDF files and Excel files.

The reports generated are made available in two modes, i.e. as system reports that can be printed and viewed locally or as World Wide Web reports that can be made available via an extranet model to people who need product data. For example dismantlers may need to know the location of hazardous materials, the removal time and disassembly route for a specific product type.

Other features of the DFE Workbench Desktop (that can also be found in the Enterprise level) include:

- Create new subassembly/assembly
- Rename a subassembly/assembly
- Remove a subassembly/assembly with or without its entire structure
- Save a subassembly/assembly with a new name
- Create a new version of a subassembly/assembly
- Search engine for materials, processes, transportations and fasteners

A new subassembly/assembly can be created by saving it from the CAD system or by using components or/and subassemblies already evaluated and saved in the working tables. Figure 3.20 shows how an assembly can be created using the second method mentioned above. This feature was added to the DFE Workbench as a result of the direct involvement of partners from the electronic industry in the development of the tool. It has been seen as very efficient to communicate across different projects within the company with the scope of reusing existing drawings of components that have been previously analysed from an environmental and structural point of view and are common to several projects.

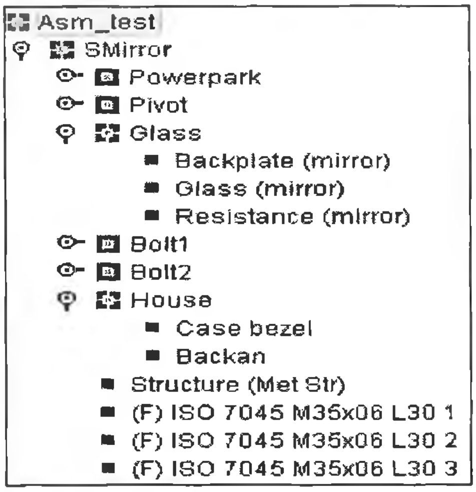


Choose a subassembly: smoke alarm

Part Name	Eco Indic...	Mass	Material	Process	Finish...	Trans...	Usage	EOL
difusor	0.209	0.002	Copper E-Cu	Casting	None	Truck	None	Copper la...
radioactive cover	0.021	0.007	ABS GF30	Injection mould...	None	Truck	None	Polymers ...
base	0.082	0.027	ABS GF30	Injection mould...	None	Truck	None	Polymers ...
difusor cover	0.014	0.004	PP	Injection mould...	None	Truck	None	Polymers ...
radioactive element	0.039	0.013	ABS GF30	Injection mould...	None	Truck	None	Polymers ...
cover	0.136	0.045	ABS GF30	Injection mould...	None	Truck	None	Polymers ...
button	0.003	0.001	PP	Injection mould...	None	Truck	None	Polymers ...

Figure 3.20 Add existing component window

The DFE Workbench supports a n-level depth into an assembly meaning that a subassembly/assembly can contain any number of subassemblies and components. Figure 3.21 presents an assembly tree example.



```
Asm_test
  SMirror
    Powerpark
    Pivot
    Glass
        ■ Backplate (mirror)
        ■ Glass (mirror)
        ■ Resistance (mirror)
    Bolt1
    Bolt2
    House
        ■ Case bezel
        ■ Backan
      ■ Structure (Met Str)
      ■ (F) ISO 7045 M35x06 L30 1
      ■ (F) ISO 7045 M35x06 L30 2
      ■ (F) ISO 7045 M35x06 L30 3
```

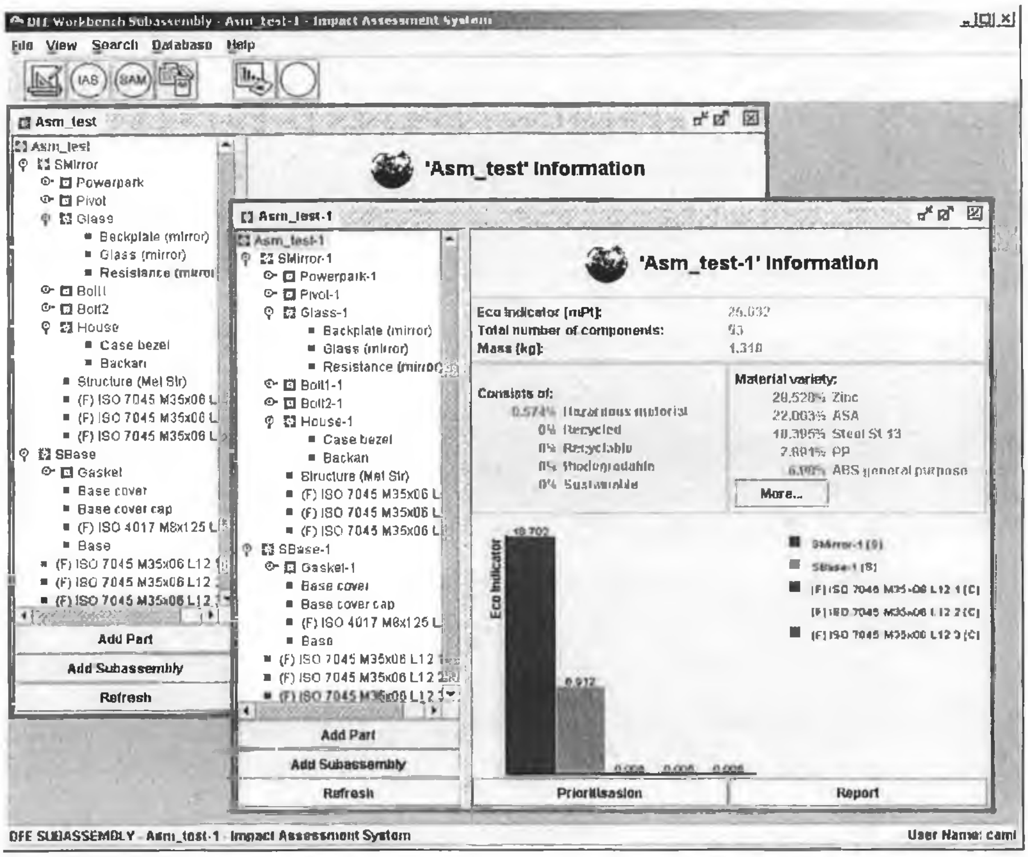
Figure 3.21 Assembly tree example

In the example presented, 'Asm_test' is the name of the assembly that contains subassemblies such as 'Powerpark' or 'Pivot' and components such as '(F) ISO 7045 M35x06 L 30 1'. Any change brought to a subassembly or component from any level will directly affect the data associated with the parent assembly. The tree corresponding to the bill of material of an assembly is built using a recursive method each time an assembly is opened.

Any assembly can be deleted from the database in two ways:

1. Remove the entire structure of an assembly meaning that all containing components and subassemblies from all levels will be removed.
2. Remove the assembly meaning that all components from the first level will be removed but the containing subassemblies will not be removed and will be available for future use.

An assembly/subassembly can be saved with a new name meaning that all the necessary new tables will be created for the new assembly/subassembly and will be populated with the same information that characterized the initial assembly/subassembly. This feature allows the creation of two or more environmental models for the same product prototype for future comparison. The versioning system in place automates this process by keeping track of the versions associated with an assembly/subassembly. Figure 3.22 shows an assembly ('Asm_test') opened in IAS mode compared to its first version created ('Asm_test-1').



'Asm_test-1' Information

Eco indicator [mPt]: 26.032
Total number of components: 93
Mass [kg]: 1.310

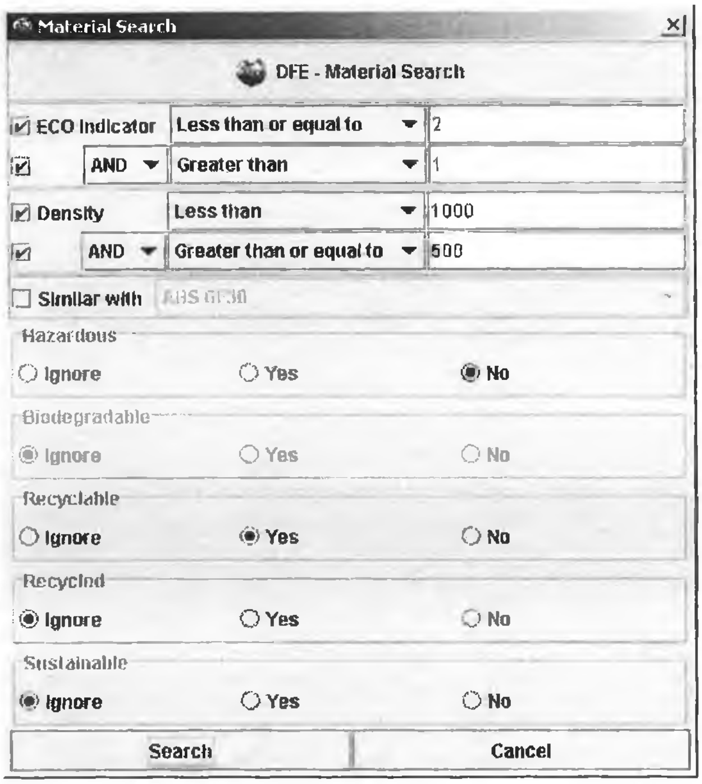
Consists of:
0.574% Hazardous material
0% Recycled
0% Recyclable
0% Biodegradable
0% Sustainable

Material variety:
28.520% Zinc
22.003% ASA
10.395% Steel St 13
7.891% PP
6.90% ABS general purpose

Figure 3.22 The DFE Workbench versioning system

The version number of a new created assembly is zero; this version number is incremented for any new version created. The new version of an assembly inherits all environmental and structural properties associated with the previous version. Once created, the new version of an assembly can be modified and then compared with the original assembly.

The search modules provide lists with materials, processes, transportations or fasteners that match the specified criteria. The data tables are searched for a specified field that has the specified properties. Figure 3.23 shows the search engine for materials. The database can be searched for materials that match at least one or all of the conditions associated with properties such as the ECO indicator, density, hazardous, biodegradable, recyclable, recycled and sustainable. Process and transportation searches accept conditions on the ECO indicator value only. The fastener database can be searched based on the disassembly time and tool associated.


Material Search

DFE - Material Search

☑ ECO Indicator	Less than or equal to	2
☑ AND	Greater than	1
☑ Density	Less than	1000
☑ AND	Greater than or equal to	500
□ Similar with	ABS GF30	

Hazardous: ○ Ignore ○ Yes ● No

Biodegradable: ● Ignore ○ Yes ○ No

Recyclable: ○ Ignore ● Yes ○ No

Recycled: ● Ignore ○ Yes ○ No

Sustainable: ● Ignore ○ Yes ○ No

Figure 3.23 Material search window

3.5 The DFE Workbench Enterprise

The DFE Workbench Enterprise has all the functionality of the DFE Workbench Desktop but works at the product system level and therefore offers a holistic view over the environmental and structural data associated with the entire product. Figure 3.24 shows the structure of the DFE Workbench Enterprise.

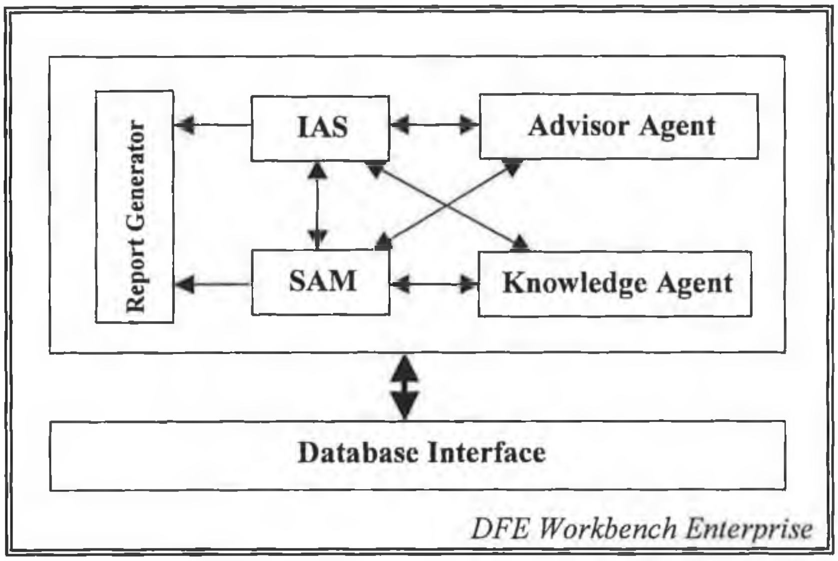


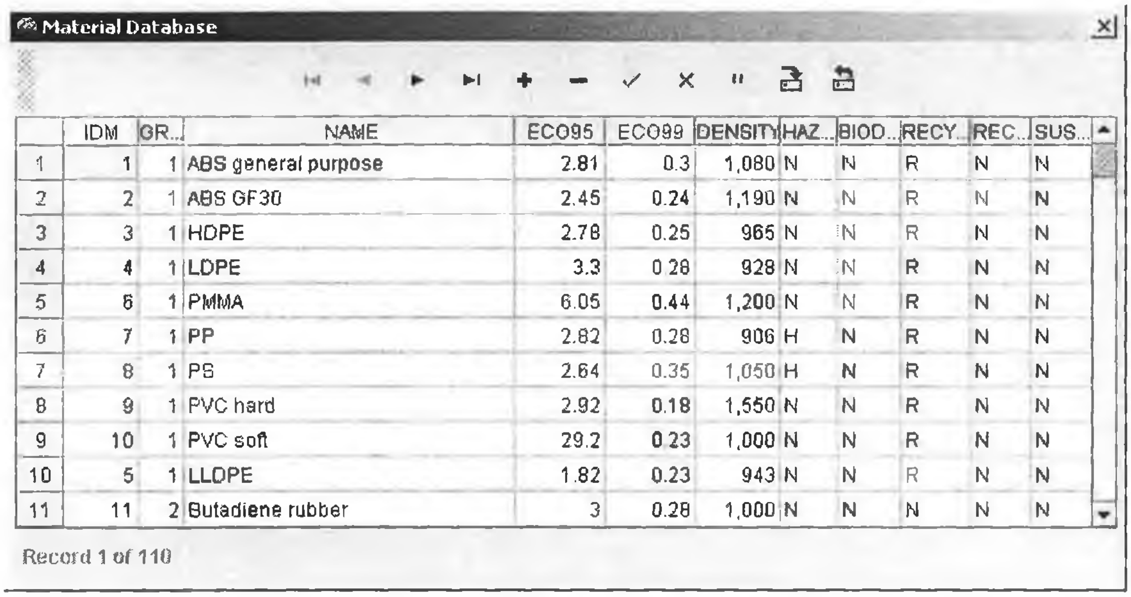
Figure 3.24 The DFE Workbench Enterprise Structure

It can run both inside and outside the CAD environment. Inside the CAD system new components, subassemblies or assemblies can be saved using the DFE Workbench while outside the CAD system assemblies already saved by the designers can be evaluated. Any problem identified during a prioritisation with a component or subassembly can be solved at this level or can be send back to the designer who created it using the DFE Workbench Desktop. This level of the DFE Workbench has added functionality by offering access to the database for the different functional departments; for example, a new material can be made available to designers by adding it to the database through the database interface in DFE Workbench Enterprise.

The DFE Workbench Enterprise supports database update for both working and data tables. The database interface allows read and write access to the following data tables:

- Material
- Process
- Transport
- Usage
- EOL
- Fasteners
- Components

Figure 3.25 presents the sixth module of the DFE Workbench Enterprise i.e. the database interface for the material table. All existing materials can be viewed and new materials can be added.



	IDM	GR...	NAME	ECO95	ECO99	DENSITY	HAZ...	BIOD...	RECY...	REC...	SUS...
1	1	1	ABS general purpose	2.81	0.3	1,080	N	N	R	N	N
2	2	1	ABS GF30	2.45	0.24	1,190	N	N	R	N	N
3	3	1	HDPE	2.78	0.25	965	N	N	R	N	N
4	4	1	LDPE	3.3	0.28	928	N	N	R	N	N
5	6	1	PMMA	6.05	0.44	1,200	N	N	R	N	N
6	7	1	PP	2.82	0.28	906	H	N	R	N	N
7	8	1	PS	2.64	0.35	1,050	H	N	R	N	N
8	9	1	PVC hard	2.92	0.18	1,550	N	N	R	N	N
9	10	1	PVC soft	29.2	0.23	1,000	N	N	R	N	N
10	5	1	LLDPE	1.82	0.23	943	N	N	R	N	N
11	11	2	Butadiene rubber	3	0.28	1,000	N	N	N	N	N

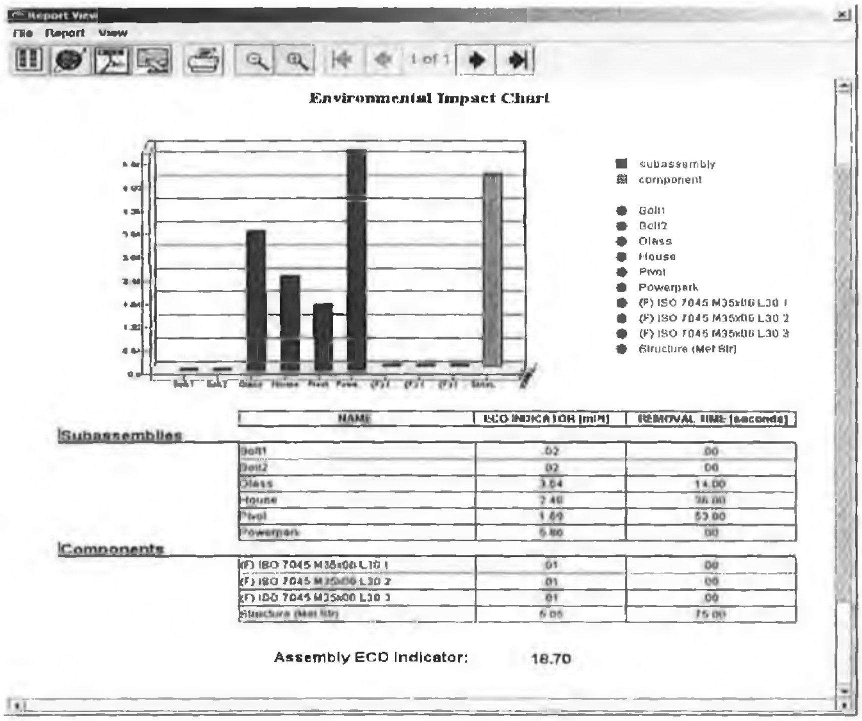
Record 1 of 110

Figure 3.25 The material database interface

This new module of the DFE Workbench Enterprise was viewed as a very important part of the tool by the functional departments of an industrial partner in the automotive industry who tested and reviewed the DFE Workbench. For example, the material department can easily log into the DFE Workbench Enterprise and add a new material, which will be made available instantly to all designers.

The table of database components is accessible through the database interface module. This table was added to the Oracle database following a suggestion made by one of the industrial partners from the electronic sector. Through this feature, the designer has access to standard components[1] for which the environmental performances are already calculated. The database already includes components such as integrated circuits, PCB and batteries. The database interface makes possible the addition of new components already evaluated from an environmental point of view based on the access rights of the user logged into the DFE Workbench Enterprise.

The first five modules of the DFE Workbench Enterprise i.e. IAS, SAM, Advisor Agent, Knowledge Agent and Report Generator are similar to the ones described for the DFE Workbench Desktop except that information at the assembly level rather than subassembly or component level can be viewed, evaluated, modified and reported. Figure 3.26 shows an example of report at this level.


Report View
File Report View
1 of 1
■ subassembly
▨ component
● Bolt1
● Bolt2
● Glass
● House
● Pivot
● Powerpack
● (F) ISO 7045 M35x06 L30 1
● (F) ISO 7045 M35x06 L30 2
● (F) ISO 7045 M35x06 L30 3
● Structure (Met Str)
Bolt1 Bolt2 Glass House Pivot Powe.. (F)I (F)I (F)I Stru..

Environmental Impact Chart

Subassemblies

NAME	ECO INDICATOR [mPt]	REMOVAL TIME [seconds]
Bolt1	.02	.00
Bolt2	.02	.00
Glass	3.04	14.00
House	2.48	36.00
Pivot	1.69	53.00
Powerpack	5.80	.00

Components

NAME	ECO INDICATOR [mPt]	REMOVAL TIME [seconds]
(F) ISO 7045 M35x06 L30 1	.01	.00
(F) ISO 7045 M35x06 L30 2	.01	.00
(F) ISO 7045 M35x06 L30 3	.01	.00
Structure (Met Str)	6.05	75.00

Assembly ECO Indicator: 18.70

Figure 3.26 Report example for assembly level

[1] Standard components are the components that are usually met in a particular sector of industry, have standardised values and are used in almost all products of that particular industry sector.

The information available on an assembly is presented grouped on subassemblies and components in the form of charts and tables. Other types of reports at this level can show disassembly information for an assembly such as fastener types between components in different subassemblies or removal times for subassemblies and components. The system reports are fully customisable as all the information saved in the database can be reported in almost any format.

3.6 CAD Integration of the DFE Workbench tool

The DFE Workbench tool has been integrated in two CAD systems as follows:

- SolidWorks 2000
- Pro/Engineer 2001

The DFE Workbench application extracts the necessary data characteristic to a product prototype from the CAD system using the CAD systems' Application Programming Interface (API) as shown in figure 3.27.

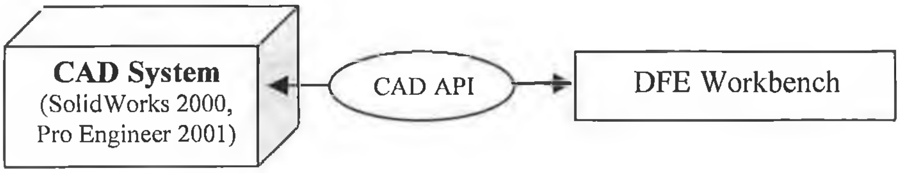
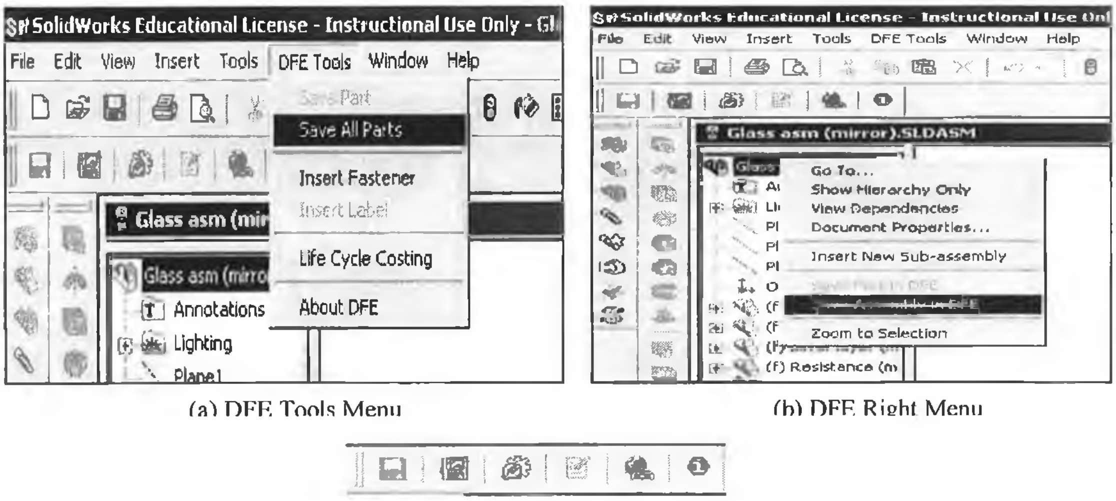


Figure 3.27 The relationship between the CAD system and the DFE Workbench

The integration with SolidWorks 2000 has been done using Microsoft Visual C++. A DLL (Dynamic Link Library) file was built to activate the DFE Tools menus and toolbar each time SolidWorks is started (figure 3.28).



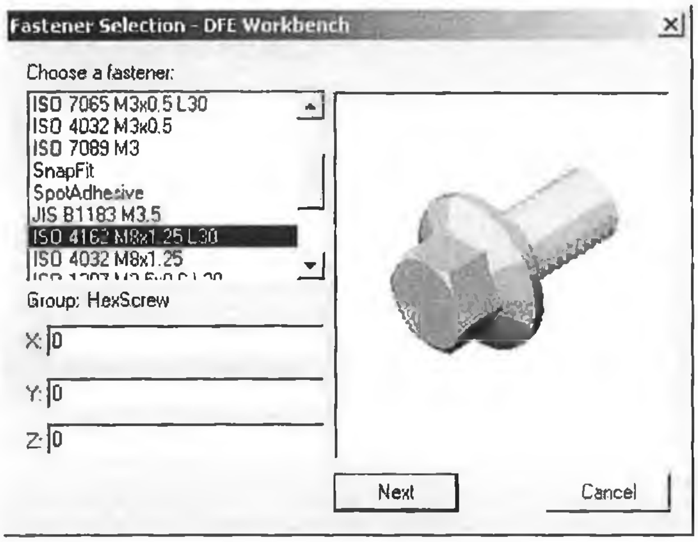
(a) DFE Tools Menu

(b) DFE Right Menu

(c) DFE Toolbar

Figure 3.28 The DFE Workbench menus and toolbar in SolidWorks 2000

The designer has the option of saving one part or an entire assembly from SolidWorks in the DFE Workbench. All the necessary data is extracted from the CAD model using the API and the needed Java classes are loaded. The information is then stored in the working tables from the Oracle database. Structural changes can be performed at the assembly level from SolidWorks 2000 by inserting a fastener. This option brings up the fastener selection window showed in figure 3.29.


Fastener Selection - DFE Workbench

Choose a fastener:

ISO 7065 M3x0.5 L30
ISO 4032 M3x0.5
ISO 7089 M3
SnapFit
SpotAdhesive
JIS B1183 M3.5
ISO 4162 M8x1.25 L30
ISO 4032 M8x1.25

Group: HexScrew

X: 0

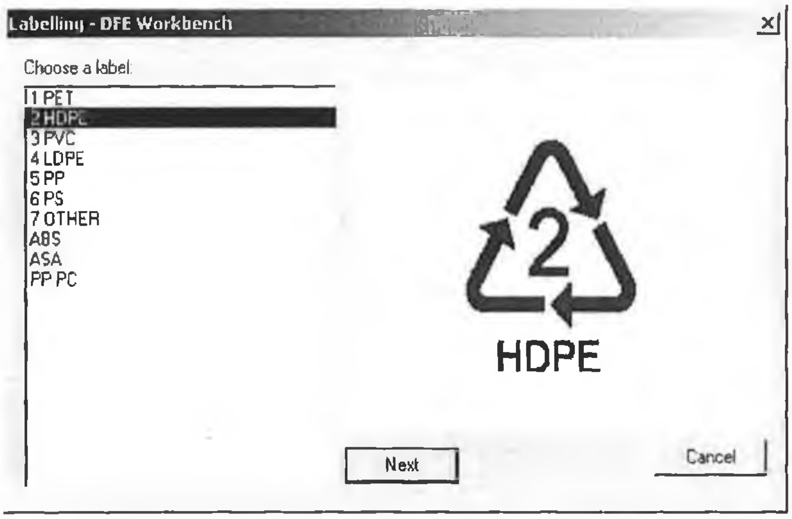
Y: 0

Z: 0

Figure 3.29 The DFE Workbench fastener selection window in SolidWorks 2000

The selected fastener is inserted in the CAD model as a new SolidWorks part and in the DFE Workbench working tables as a new fastener between two components. The fastener insertion feature is only available when an assembly (SLDASM file) is active.

The SolidWorks integration also allows the insertion of a label for a component as shown in figure 3.30.


Labelling - DFE Workbench

Choose a label

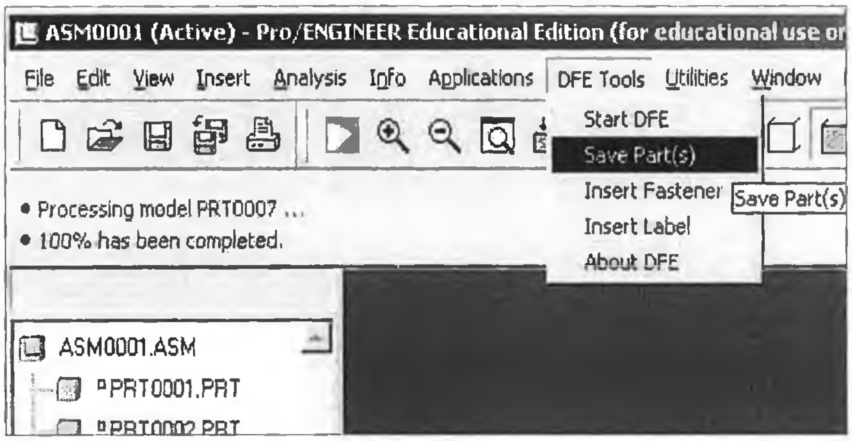
1 PET
2 HDPE
3 PVC
4 LDPE
5 PP
6 PS
7 OTHER
ABS
ASA
PP PC

Figure 3.30 The DFE Workbench label selection window in SolidWorks 2000

After a label is selected, the designer is asked to select the point in the CAD model of the component where the label should be inserted. The component is then labelled in SolidWorks and this new information is saved in the working tables of the DFE Workbench.

The label insertion feature is only available when a component (SLDPRT file) is active.

The integration with Pro/Engineer 2001 has been realized using a Java language toolkit for Pro/Engineer called J-Link. A new Java class was created and added to the existing DFE Workbench package that can access the internal components of a Pro/Engineer session using J-Link. This made possible the transfer of the entire tree structure of an assembly to the Oracle database of the DFE Workbench. The designer has the option of saving just one component at a time or the entire assembly using the DFE Tools menu from Pro/Engineer (figure 3.31).



- Processing model PRT0007 ...
- 100% has been completed.

Figure 3.31 The DFE Tools menu in Pro/Engineer 2001

An important aspect of both CAD integrations of the DFE Workbench is that the designer is not constricted in any way by the tool. He/she can choose the moment of saving a component or an assembly using the DFE Workbench. The tool can also be accessed for getting any environmental or structural data at any point from the CAD system by the means of menus and toolbar buttons.

3.7 The DFE Workbench Global

The DFE Workbench Global brings all the information generated by the first two levels of the DFE Workbench to the web through Intra/Internet. A user with the necessary access rights (username/password) can view or download specific environmental and structural metrics calculated, evaluated and improved by the DFE Workbench Desktop and

Enterprise. Reports are made available to the web through the DFE Workbench Global. A collaboration module is available to communicate with other DFE Workbench Global users over the Intra/Internet. Figure 3.32 presents the DFE Workbench Global structure.

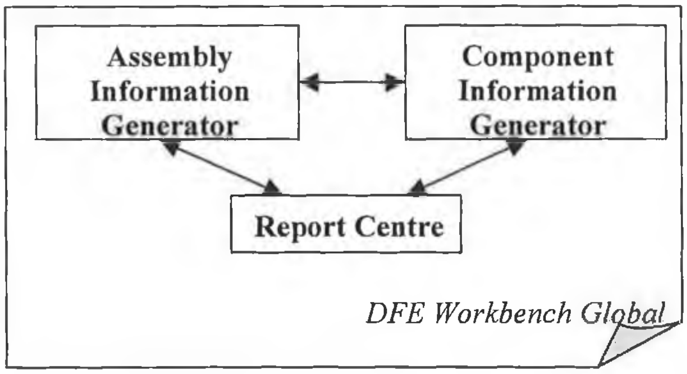
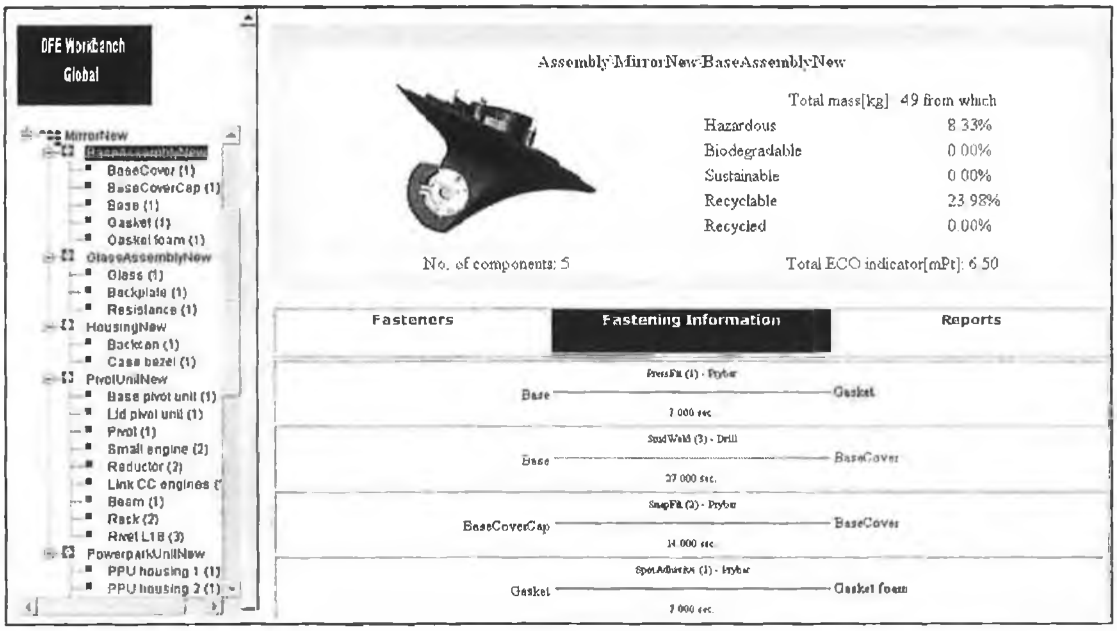

Assembly Information Generator
Component Information Generator
Report Centre
DFE Workbench Global

Figure 3.32 The DFE Workbench Global structure

The DFE Workbench Global consists of three modules as follows:

- Assembly Information Generator
- Component Information Generator
- Report Centre

The *Assembly Information Generator* reports to the web all the information saved in the database for a specific assembly (see figure 3.33). The data associated with an assembly is grouped on the web page in a table and the following sub pages: Fasteners, Fastening Information and Reports.



Assembly MirrorNew BaseAssemblyNew

Total mass[kg]	49 from which
Hazardous	8.33%
Biodegradable	0.00%
Sustainable	0.00%
Recyclable	23.98%
Recycled	0.00%

No. of components: 5

Total ECO indicator[mPt]: 6.50

PressFit (1) - Prybar
Base —— Gasket
7.000 sec.

StudWeld (3) - Drill
Base —— BaseCover
27.000 sec.

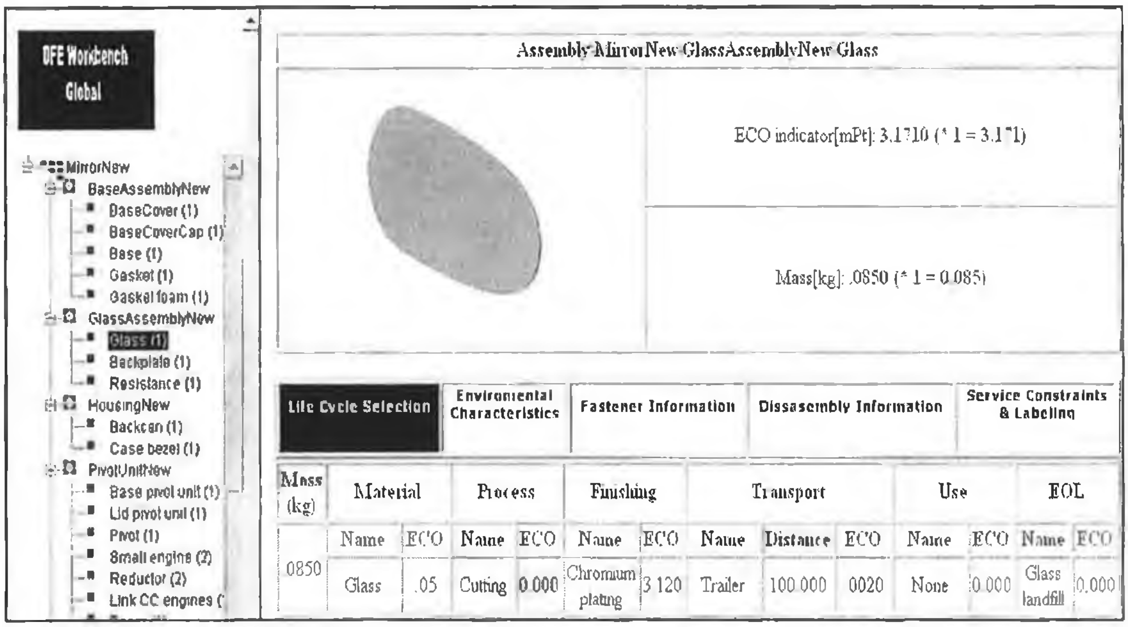
SnapFit (2) - Prybar
BaseCoverCap —— BaseCover
14.000 sec.

SpotAdhesive (1) - Prybar
Gasket —— Gasket foam
7.000 sec.

Figure 3.33 The DFE Workbench Global – Assembly Information

The Assembly Information Generator reports general type of information which is instantly calculated based on the data available in the working tables such as the number of subassemblies, the number of components, total mass, total ECO indicator, percentages of hazardous, biodegradable, recyclable, recycled and sustainable content, the list of fasteners and disassembly tools used

The *Component Information Generator* reports to the web all the information saved in the database for a specific component selected from an assembly tree. Some information reported such as mass, material type, process type, transportation, usage and EOL value is extracted directly from the working tables. Other information such as the ECO indicator value, the removal time and the disassembly route is calculated based on the information available in the database. Figure 3.34 presents the DFE Workbench Global web page that reports information on a component. The environmental and structural information available for a component is grouped in sub pages as follows: Life Cycle Selections, Environmental Characteristics, Fasteners Information, Disassembly Information and Service Constraints & Serviceability.



Assembly MirrorNew GlassAssemblyNew Glass

ECO indicator[mPt]: 3.1710 (* 1 = 3.171)

Mass[kg]: .0850 (* 1 = 0.085)

Mass (kg)	Material		Process		Finishing		Transport			Use		EOL	
	Name	ECO	Name	ECO	Name	ECO	Name	Distance	ECO	Name	ECO	Name	ECO
.0850	Glass	.05	Cutting	0.000	Chromium plating	3.120	Trailer	100.000	0020	None	0.000	Glass landfill	0.000

Figure 3.34 The DFE Workbench Global – Component Information

The *Report Centre* facilitates the downloading and uploading of different types of reports at assembly or component level. All reports are created by the Report Generator module available in the DFE Workbench Desktop and Enterprise. They are made available to the web in PDF format.

The DFE Workbench Global reports to a web site up-to-date information of both IAS and SAM type because this web application is connected directly to the Oracle database server through Microsoft ODBC for Oracle. The web application server used for implementation is Cold Fusion Server that has been selected for security reasons.

3.9 Conclusions

This chapter presents the DFE Workbench software by describing the system configuration, the software architecture and the database structure. All three levels of the tool i.e. Desktop, Enterprise and Global as well as the DFE Workbench CAD integrations have been described in detail. The DFE Workbench Desktop and Enterprise are Java based applications designed to help the user in the process of the improvement of a product prototype from an environmental point of view. The Desktop level of the tool is integrated in the CAD system used by the designer i.e. SolidWorks 2000 and Pro/Engineer 2001. The user can interact with the DFE Workbench Desktop at any point to evaluate and improve the environmental impact of an emerging product. The Enterprise level of the tool offers an integral view over the environmental and structural information associated with an assembly. The DFE Workbench Enterprise is designed to be used outside the CAD system in a distributed environment but it can also be accessed from the CAD environment if necessary. The Global level of the DFE Workbench is a web-based application that reports all the environmental information generated by the first two levels of the tool using the Internet/Intranet. All three levels of the DFE Workbench are connected with an Oracle database server. The benefits of the DFE Workbench can be summarized as follows:

- Flexible and platform-independent application
- Suitable for use in distributed environments
- Supported by a powerful and updateable database
- CAD integrated
- System and web reports with general as well as detailed information in the form of charts and tables generated
- Environmental information is made available on the web

The DFE Workbench software can be easily upgraded and reused due to the object-oriented model used in implementation.

Chapter 4

Development of a PA Technique for Distributed Applications

4.1 Introduction

In the early stages of development of the DFE Workbench, protocol analysis techniques were successfully used to test and validate the desktop version of the software [Roc99, Man00]. However, as the tool evolved, the limitations of the protocol analysis technique as a test and validation methodology became apparent. These limitations were centred around the geographical dispersion of the subjects and the applications. As described in chapter three, the DFE Workbench software is a distributed application that can be used by distributed designers in a virtual collaborative environment. Hence, a new approach to distributed protocol analysis needed to be developed for the study of distributed and collaborative decision making processes in a virtual environment.

This chapter begins with a review of the protocol analysis technique by presenting its definition, typology, the data analysis stage, positive and negative aspects of the method and related work in the human-computer interaction and engineering fields.

This chapter focuses on the design and development of a technique that can be used to apply protocol analysis to evaluate the DFE Workbench software in a distributed design environment. The new distributed protocol analysis template should facilitate the analysis and evaluation of both human-to-computer and human-to-human interaction in distributed environments. The design of the new protocol analysis template for distributed environments is based on the results of two protocol analysis experiments that studied the behaviour of team members during a problem solving process in two situations as follows:

1. Team members located in the same room.
2. Team members distributed over a computer network.

This chapter presents these two protocol analysis experiments and their results concluding with a specification for a protocol analysis template suitable for testing and evaluating distributed applications.

4.2 Protocol Analysis

The major techniques used in the evaluation of computer-based system include [Gre96]:

- Observational usability methods such as protocol analysis or 'think aloud' method, constructive interaction and post-session interviews.
- Controlled experimentation methods such as experimental design, statistical testing and interpretation.

Protocol analysis, logged data, questionnaires and interviews have been used as the prominent basic evaluation methods, sometimes being combined in an attempt to achieve better and more complete results [Hen95]. The protocol analysis technique has been widely used in the information technology field, not only for usability studies of computer-based systems or interfaces but also for systems development tasks and model formulation for decision support systems [Ben01]. Based on the examination of verbal protocols or user verbalizations, protocol analysis is the most systematic and valid technique from all observational methods [Eri99, Ben01].

4.2.1 Definition

Protocol Analysis (PA) is a qualitative evaluation method for human cognitive processes. It consists of collecting verbal data reports and systematically analysing them [Eri99, Cha00, Ben01]. In particular, protocol analyses of interfaces rely on the direct observation of a real interaction between the user and the computer-based system. In a protocol analysis session, the subject is asked to complete a set of predetermined tasks and is observed by the evaluator who typically records users' actions using video and audio techniques. The users are asked to think aloud during or after performing the tasks describing what they believe is happening, what they are attempting to do, why they take a specific action and other task-related thoughts. The process of verbalization reveals the assumptions, misconceptions, inferences and problems that users face while performing tasks or solving problems [Eri99, Cha00, Ben01, Ger01]. The verbal reports are based on a subset of information held in the short-term and long-term memory. Ericsson and Simon hypothesised that human cognition is information processing meaning that "*a cognitive process can be seen as a sequence of internal states successively transformed by a series of information processes*" [Eri99]. Furthermore, they stated that information stored in the short-term memory is the information that was recently acquired and is directly accessible for further processing such as producing verbal reports while information from the long-term memory must be first retrieved, transferred to the short-term memory and then it can be reported. Hence, verbal reports tap this short-term memory. Ericsson and Simon used this conclusion to validate and promote think aloud reports and verbalizations.

First studies of protocol analysis were documented in the 1920s when Watson used the think-aloud protocol to illustrate some general characteristics of cognitive process in problem solving. However, in this early work, the evaluator's method was based on taking notes while the subject was thinking aloud and then examine these notes and draw a conclusion from them without the possibility of going back and reinterpret or re-evaluate some records. It wasn't until 1945 that tape recorders became available making the investigators job much easier, being possible now to analyse the verbalizations without any real time constraints. By the 1970s, video recording enabled a new level of detail that could be used in the evaluation process, making it possible for the evaluator to observe the subjects' gesture and mimics [Eri99, Roc99].

4.2.2 Typology

In general, there are three main types of techniques for performing PA as follows:

1. Concurrent
2. Retrospective
3. Introspective

In the *concurrent protocol analysis technique*, subjects are asked to perform a task whilst simultaneously verbalizing their thoughts. This protocol uses information stored in the short-term memory of the subject which is just encoded orally, the implication being that verbalization will not interfere with ongoing processes. In their research on protocol analysis, Ericsson and Simon demonstrated that "*the concurrent report reveals the sequence of information heeded by the subject without altering the cognitive process*" [Eri99]. The concurrent protocol analysis technique can be used to understand how users form their cognitive model of the system under evaluation and investigate causes of errors, mistakes and misinterpretations. Concurrent protocols capture how users approach a specific task and why problems occur during the user interaction with a process. The user verbalization can be recorded using video or audio techniques and screen capture routines. Time-stamp video of using the system is very useful as it offers much more information than the users verbalization alone [Ben01].

In the *retrospective protocol analysis technique*, subjects are asked first to perform a task or set of tasks and then to report information about the completion of specific tasks. The subjects are asked about cognitive processes that occurred at an earlier point in time. The retrospective protocol uses information preserved partially in short-term memory and partially stored in long-term memory. While the information in short-term memory can be accessed directly, the information in long-term memory has to be retrieved first and then verbalized. This process may generate errors or incompleteness in the results. Ericsson and Simon have showed that "*the information that is heeded during performance of a task, is the information that is reportable; and the information that is reported is information that is heeded*", concluding that both concurrent and retrospective protocols are direct verbalizations of specific cognitive processes [Eri99]. Ideally, the user is asked for the retrospective report on the system interaction immediately after the completion of the tasks while much information is still in the short-term memory. Videotape of the user undertaking the predefined tasks can be used in retrospection to assist in the recall of the interaction with the system under study.

The *introspective protocol analysis technique* can be viewed as another type of retrospective report. After the subject is asked to perform a set of tasks, the evaluator collects retrospective answers to questions about prior behaviour. In an introspective protocol session, users may report information that they have inferred or otherwise generated instead of recalling related information. Methods used include tape recording interviews on exploring the cognitive processes in problem solving behaviour [Cha00, Eri99]. After a review of the introspective report, Ericsson and Simon conclude, "*even if introspective information was not necessarily incorrect and uninformative, it was unnecessary and could be replaced by appropriate behavioural measures*" [Eri99]. Therefore, the introspective protocol is discredited for its value of verification.

The first two forms of protocol analysis, the concurrent and the retrospective protocols, are the most powerful means for gaining detailed information about specific cognitive activities during problem-solving processes [Eri99, Cha00]. In a comparison of concurrent and retrospective reports, Ericsson and Simon showed that the two protocols are very similar, but the retrospective reports often have details omitted due to the decay of the long-term memory [Eri99].

For the evaluation of the usability of a computer-based system, the most popular technique used is the concurrent protocol analysis technique as a powerful method for obtaining detailed information about user thinking and understanding why problems occur during the process of user interaction with a system [Hen95, Gre96, Roc99, Bra00, Ben01]. Also in the field of engineering design, the main method for protocol analysis used is the concurrent reporting [Gol95, Roc99, Atm99, Cro95, Llo95, Pur98, Cha00, Kav01, Ste02]. Both concurrent and retrospective protocol analysis techniques can be applied to obtain more detailed information about the problems that occur during the problem-solving process [Bra00, Ger01].

4.2.3 The analysis phase of the data gathered during the protocol study

The analysis of the verbalization report aims to identify the heeded information produced during a cognitive process. The analysis phase begins at the end of the protocol session after a full transcript of the session was built. When applying concurrent protocol analysis to evaluate the usability of a computer based system, transcripts may include exact verbalizations made by the subject, observer's reminders, records of the user's actions and screens used during the execution of the predefined tasks. The first step is to identify verbalization units called segments that correspond to units of heeded information. This encoding process is not usually difficult, the resulting segments being identifiable in fact by a statement made by the subject during the think aloud protocol [Eri99, Roc99]. A study carried out in the human-computer interaction field shows that *"the principal type of data generated by think aloud research consists of a catalogue of episodes, critical as well as typical, exemplifying theoretical points"* [Ben01]. Other studies have showed that the analysis of the protocol is a three-stage process [Roc99] as represented in figure 4.1.

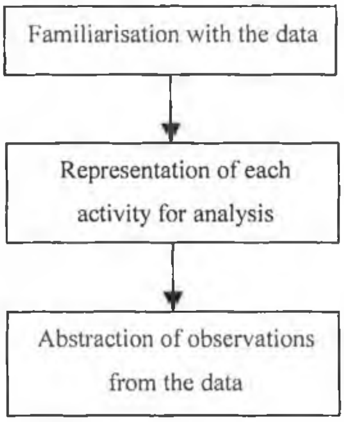

Familiarisation with the data
Representation of each activity for analysis
Abstraction of observations from the data

Figure 4.1 The analysis phase of the protocol as a three-stage process

The second stage can be identified with the protocol segmentation phase already presented. The third stage of the analysis process is conducted by observing patterns of activity, by collecting instances of that activity and by comparing the collected instances [Roc99].

The development of an *a priori* coding scheme by which the protocols can be split into small parts can be very useful, ensuring that the findings of the analysis are not data driven [Ben01].

Protocols are generally scanned for anecdotal information and the frequency of key items is captured using scoring techniques. In the field of human-computer interaction, any combination of scanning and scoring as an analysis technique can be very useful to identify usability problems [Ben01].

4.2.4 Related Work

The protocol analysis method is an excellent choice for qualitative researchers interested in a reach source of data. A lot of studies in the human-computer interaction field proved the efficiency of this method in revealing important usability problems associated with computer-based systems [Hen95, Gre96, Roc99, Bra00, Ben01]. In the field of engineering, the protocol analysis technique has been used as the main method to study the cognitive activity of the designer whilst in the design process [Cro95, Roc99, Cha00, Ger01].

Henderson et al. [Hen95] examined four basic evaluation methods: verbal protocol analysis, logged data, questionnaires and interviews. These four user-based methods of software evaluation were used to evaluate three different software types (i.e. spreadsheet, word processor and database) using one hundred and forty-eight participants. The verbal protocol procedure used was an aided subsequent verbal protocol: videotapes of the subjects undertaking the software evaluation task were played back while the subjects were asked to report their actions (retrospective protocol). This procedure was considered advantageous for purely methodological cross comparison reasons and also for having more validity than the subsequent unaided protocol analysis (as the procedure chosen produced data patterns similar to the more usual concurrent verbal protocol analysis). *"On the positive side, the method transformed the user's role in the evaluation from a passive subject to a more active participant, which may have resulted in increased commitment and more usable data"* [Hen95]. As a negative aspect, the research shows that the verbal protocol analysis is time-consuming. The process that consisted of videotaping individuals using the system, conducting a post-session verbal protocol and analysing the information obtained is long and tedious. The study concludes that the verbal protocol analysis method is the most robust one proving to be very efficient in highlighting the usability problems associated with the system under evaluation. However, the study suggests that a multimethod strategy is probably the most powerful approach to evaluate the usability of software systems.

In a recent study, the usability of a commercial web site is evaluated using the protocol analysis technique [Ben01]. The research analysed verbal protocols of eight users interacting with a greeting card web site using a combination of modelling, scanning and scoring techniques. Protocol analysis proved to be one of the best techniques to examine the interaction among the following three constructs: objective usability, direct experience and perceived ease of use. The research concludes, *"protocol analysis offers a wealth of information that is generally not available through other methods and due to the richness of the data collected, protocol analyses do not need to be conducted over a large sample of users"* [Ben01].

A research on investigating the information-seeking processes of early adolescents explores the use of both concurrent and retrospective verbal protocols – Think Alouds and Think Afters [Bra00]. Five subjects were observed while interacting with a CD-ROM encyclopaedia (Microsoft Encarta 98) trying to access specific information. The study shows that the amount of data in the concurrent protocol was larger than the one in the retrospective protocol, the Think Aloud method providing much more detail about the affective nature of the information-seeking processes. In fact, the concurrent protocol provided the most complete and detailed description of the information-seeking processes. If the decision points were clear in the concurrent protocol, the reasons behind these decisions were often explained during the retrospective protocol. The results of this research are consistent with the work of Ericsson and Simon [Eri99], who explained the difficulties of information retrieval during the retrospective verbal protocols. To increase the amount and detail of data in the retrospective protocol, the study suggests that other methods to record data than tape recording are necessary such as transaction logs, screen captures and video recording. The researcher and the subject can interact with the recorded information after the task is completed and discuss the cognitive, affective and behavioural processes involved. The conclusions of the research include that *"the data generated from Think Alouds and Think Afters is quite different. For researchers interested in looking at a phenomenon as it happens, Think Alouds provide rich data. Some participants, however, may find it difficult to generate Think Alouds while carrying out a new task or a task that involves a lot of cognitive processing. Think Afters are better for gathering rich data in these kinds of situations. Yet, Think Afters may be influenced by forgetting and* *fabrication"* [Bra00]. However, this study shows that both techniques of protocol analysis i.e. concurrent and retrospective are very efficient in providing data about the behavioural, cognitive and affective processes.

A recent study on the differences between retrospective and concurrent protocols [Ger01] indicates that both types of protocol produce very similar outcomes in terms of exploring the process-oriented aspects of designing and that there is no associated interference with the ongoing design process when using concurrent protocols. Gero and Tang have showed that concurrent protocols appear to reveal more information in the beginning of the design process whilst in retrospective protocols subjects could not recall these early processes, even with the help of the video. However, the retrospective protocol can reveal information about pauses during the speech made by the subject in the concurrent protocol because subjects can recall sometimes the thinking process [Ger01]. This result is similar with the argument of protocols proposed by Ericsson and Simon [Eri99]. The results of the experiments conducted by Gero and Tang indicate that the concurrent protocol reveals more information related to the functional aspect of the design process when the problem is formulated whilst the retrospective protocol reveals more information in producing solutions and evaluation. The research points out that *"in terms of the process-oriented aspects of designing, concurrent protocols have the same abilities as retrospective protocols. They are still the most efficient and applicable methods in exploring this aspect of the design process."* [Ger01]. The study concludes that the contents revealed by concurrent and retrospective protocols are similar (result possible also because of the visual information generated during the design process and used during the retrospective protocol) in level of abstraction, function-behaviour structure, macro strategies, and in the analysis-synthesis-evaluation model.

In a design study, Goldschmidt [Gol95] has used protocol analysis to compare the behaviour and performance of an individual and a team of designers. She conducted a linkographic analysis of the protocols, identifying moves in the design process as well as links among a given move and previous moves (backlinks) or subsequent moves (forelinks). The data analysed by Goldschmidt consisted of one think aloud protocol and one protocol of conversation. Therefore, the compatibility of the two modes of thought verbalization i.e. think aloud and conversation was discussed. The statement that the two protocols are equal windows into the cognitive processes involved in design thinking is based on the relation between thought and speech proposed by Vygotsky. There are two planes of speech: the inner and the external. Inner speech refers to the semantic aspect of speech, abbreviated speech and is not an aspect of external speech, being a function on itself. Both the inner and external planes of speech are more than representations of thought (*"Thoughts are not merely expressed in words; it comes into existence through them."* [Gol95]). Together the two planes of speech i.e. inner and external form a unity of speech. Goldschmidt concludes, *"thinking aloud can be seen as being close to inner speech, whereas a conversation is certainly a sample of external speech"* [Gol95]. Based on the comparison between critical moves rich in forelinks and the ones rich in backlinks and a careful examination of the link index (the number of links reported to the number of moves) as an indicator of the 'strength' of the design process, Goldschmidt concludes that *"there are almost no differences between the individual and the team in the way they bring their work to fruition"* [Gol95].

Cross and Clayburn Cross [Cro95] present the results of the analyses of a teamwork experiment of the Delft Protocols Workshop. Protocol analysis was used to observe the following aspects of teamwork in design: roles and relationships of members within the team, planning of the design process and acting of the team according to that plan, information gathering and sharing, problem analysing and understanding, concept developing and adopting, and avoiding and resolving conflicts. The study concludes that teamwork is a social process and that the design process is nowadays an integration of a technical process, a cognitive process and a social process.

Gero and McNeil [Ger97] used protocol analysis to investigate the process of designing. They extended the think aloud protocol through the use of a domain-dependent coding scheme, which *"brings structure to the unstructured data of the protocols without detracting from the richness of the data"* [Ger97]. The methodology developed and applied by Gero and McNeil demonstrates different aspects of the behaviour of individual designers, providing a basis for a better understanding of the design process.

4.2.5 Advantages and disadvantages of PA in the testing of computer-based systems

The verbal protocol analysis method is the most efficient in terms of its ability to highlight usability problems, the information obtained being highly relevant to the software developer [Hen95]. Another positive characteristic of this technique is that it doesn't require large sample sizes. Due to the richness of data obtained via protocol analyses, a small number of users representative of the target population can yield important results [Ben01]. Advantages of the protocol analysis technique applied in the human-computer interaction process include:

- It pinpoints important usability problems.

- It determines why problems occur when users interact with computer based systems.
- It locates the negative aspects concerning the user acceptance of the system.
- It offers an understanding of how users form their cognitive model of the system.
- It captures the user attitude toward the computer-based system (how users approach a task).
- It represents a robust and efficient method for investigating causes of errors, mistakes and misinterpretations.
- It does not require a large number of users (subjects).
- It represents a reality check for user interface designers.

Some critiques to the verbal protocol analysis method include the lack of realism due to the presence of the observer. Also, the need for concurrent verbalization may alter users behaviour. However, these facts should not be considered a disturbance to the thought process since the talking can be executed almost automatically [Ben01]. Ericsson and Simon have demonstrated that the process of verbalization during the protocol does not alter the cognitive process [Eri99]. A consequence is that users spend more time for completing a task because of the concurrent verbalization. Also in the case of the retrospective protocol, the user verbalizations are collected after the tasks are completed and therefore it is time consuming. In a recent study on the differences between concurrent and retrospective protocols [Ger01], it has been concluded that both protocols have the same duration but that the number of segments in the retrospective protocol is greater than in the concurrent.

The subjects chosen for the protocol analysis tests have to be representative for the target population and they have to be good communicators. Not all subjects are equally suited for the think aloud method because people do not know what is going on in their head or just find it difficult to verbalize their every thought [Roc99, Ben01].

Another problem identified is that it requires intense work to analyse and evaluate the data gathered during protocols [Hen95]. The direct consequence of this aspect of protocol analysis techniques is higher cost.

Whilst PA techniques have been used to study cognitive behaviour interactions for co-located teams, very little work has been done on the use of PA techniques for geographical dispersed design teams collaborating in a virtual environment.

To summarise, disadvantages of the protocol analysis technique include [Hen95, Roc99, Ben01, Ger01]:

- *Realism* – the observer and the think aloud method alters users behaviour.
- *Cost* – the analysis of the protocol is work-intensive.
- *Accuracy* – subjects may report irrelevant accounts and reports of actual activity may be incomplete.
- *Time* – talking affects performance time.
- *One to one evaluation* – PA was originally designed for one to one evaluations; some research is necessary to apply the PA technique in a distributed collaborative environment and study interlacing distributed protocols.

Studies on user computer interaction based on the analysis of verbal protocols can highlight specific usability problems, identify features that draw out negative opinions or user dissatisfaction and show how objective usability factors affect perceived ease of use [Hen95, Gre96, Roc99, Bra00, Ben01]. However, very little work has been done in the study of distributed computer-based systems using PA techniques and on collective cognitive processes.

4.3 Design and development of a PA template for distributed environments

The proposed approach to test the DFE Workbench software in a distributed environment was to develop some preliminary tests to provide us with an understanding of the key technical issues with the PA technique itself and with the application of PA techniques in distributed collaborative environments. The design of a template for the test of a distributed application using the PA technique was based on two considerations as follows [Chi02]:

A. The testing and validation methodology could not be developed based on the tool that is evaluated i.e. the DFE Workbench.

B. It is crucial to separate the cognitive behaviour of the distributed team from the problem that was being solved.

In order to design an appropriate test to evaluate the cognitive behaviour of distributed teams, we first benchmarked the cognitive behaviour with well defined and constrained problems that can be used in either co-located or distributed environments. The problem had to be simple and sufficiently constrained so as not to distort the problem solving process (e.g. complex learning of the tool) and to be able to identify clearly the characteristics of the co-located and distributed cognitive decision making processes.

Two PA experiments were developed as follows:

1. Study of *face-to-face collaboration*
2. Study of *distributed synchronous collaboration*

These experiments are based on the Same Time – Same Place and Same Time – Different Places models out of the existing four collaboration models shown in the space-time communication matrix in figure 4.2 [Anu01].

	Same Time	Different Times
Same Place	**Face-to-Face Collaboration**	**Asynchronous Collaboration**
Different Places	**Distributed Synchronous Collaboration**	**Distributed Asynchronous Collaboration**

Figure 4.2 Collaboration models

Team behaviour that occurred when the subjects were co-located was studied and compared with that of teams in a computer based collaborative environment.

4.3.1 Description of the protocol analysis experiments

Five teams of two people were observed while they were collaborating in order to solve a puzzle game called Tangram[1]. The goal of this ancient Chinese puzzle is to form a given shape using seven pieces i.e. five triangles of different sizes, a square and a rhomboid.

The test was divided in two stages that took place in different days as follows:

1. The co-located test
2. The distributed test

Selecting these two tests, the behaviour of co-located and distributed team members during a problem solving process could be compared.

During the *co-located test,* teams of two people located in the same room were observed and videotaped while they were collaborating during the problem solving process (see Photograph 4.1).

[1] http://www.tangram.i-p.com/

Photograph 4.1 Snapshot from the co-located test

The two members of each team were asked to collaborate with each other in solving the problem, communicating all task related thoughts to the other member of their team. An observer was present in the room during the co-located test with the role of reminding the participants to think aloud in case they forget to do so and to take notes on the collaboration between subjects during the problem solving process. The co-located test was divided in three phases as follows:

- Participant induction and training
- Performing the task
- Questionnaire

Each participant received an induction sheet in which the purpose of the test was explained and some instructions for the test were given (see table 4.1). Each team was allowed approximately fifteen minutes for learning the game. In the second phase of the test, participants were given two shapes to assemble using the shapes provided. The average time needed for this phase of the test was thirty minutes.

The co-located test was concluded with a short questionnaire in which participants were asked to rate the collaboration process and to list some positive and negative aspects of it.

The *distributed test* was conducted after the co-located test using the same subjects and in the same format. The two members of each team were distributed over a computer network and were collaborating with each other through a virtual communication environment i.e. Lotus Sametime to solve an electronic version of the puzzle. Each participant was connected to the Sametime Server through a web browser.

Participant Introduction and Instructions
Co-located teams

Dear Participant,

Thank you for giving the time to this distributed protocol analysis study.

This activity is intended to evaluate the collaboration process between co-located team members working together in a problem solving process.

You are part of a two-member team that is assigned to solve a problem. Your role is to perform a task (described below) and in the same time give a running comment about what are you attempting to do. You have to collaborate with the second member of your team in completing the task by communicating all your thoughts.

The time is divided into three parts as follows:

1. Introduction and training in using the game
2. Task performing
3. Questionnaire

Task

Adjust 7 geometric shapes to fit into one big shape. You have to use all 7 pieces to solve the puzzle.

Operation of the test

The test is based on a research method called protocol analysis. You will be videotaped while performing the task for later analysis. Your actions as well as verbalisations will be recorded, so it is extremely important to remember to think aloud while solving the problem. For example verbalisation may be *"...probably we can place this shape here because..."*.

During the session, a researcher will be present with you in the room having the role to observe your actions, to record your words and to remind you to speak in case you forget that. You may ask the observer questions, otherwise please try to ignore his presence in the room.

After the test is complete a short review will be held in which you will be asked to complete a short questionnaire.

Thank you for your time.

Table 4.1. Participant Introduction and Instructions

Figure 4.3 presents the environment used for the distributed test.

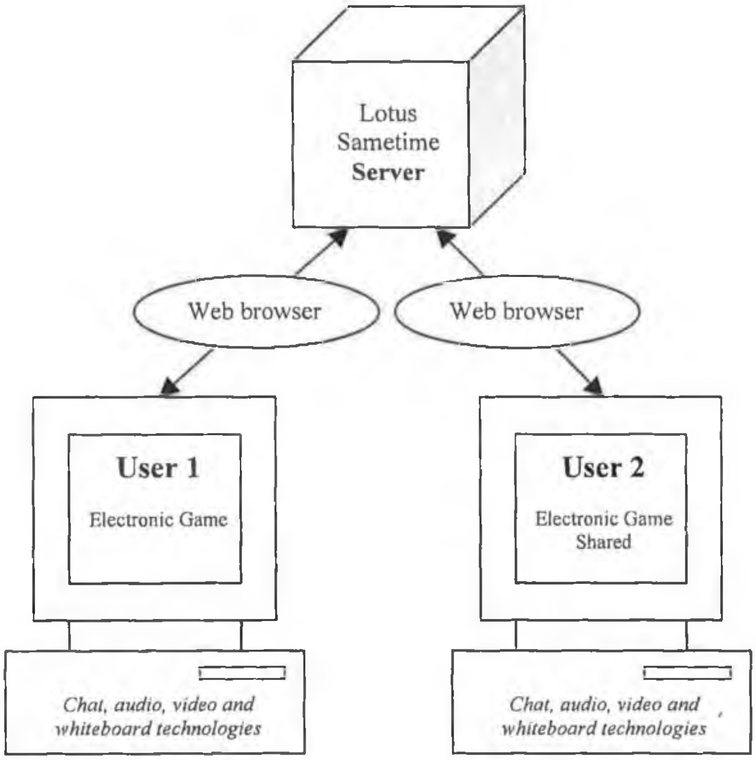

Lotus
Sametime
Server
Web browser
Web browser
User 1
Electronic Game
User 2
Electronic Game
Shared
Chat, audio, video and whiteboard technologies
Chat, audio, video and whiteboard technologies

Figure 4.3 The environment of the distributed test

The collaboration process took place in a virtual meeting room (created for the test), which allows users to transmit real time audio and video, share applications (e.g. the game) or a whiteboard, and send or receive instant messages. The electronic version of the puzzle game was running on one of the users' machine and was shared with the second member of the team so as he/she could have access and control to the same instance of the game. Besides the test environment (distributed vs. co-located), another difference between the two PA experiments was that subjects had alternative access to the game in the distributed environment whilst they could engage at any point in time with the physical game in the co-located environment.

The distributed test was designed using the same phases described for the co-located test and was performed in the same way. Each participant was given an induction sheet different to the one for the co-located test only in the description of the task. Photograph 4.2 presents a snapshot of the distributed users during the problem solving process over a computer network.

Photograph 4.2 Snapshots from the distributed test

Each member of the team was videotaped while performing the task. In all cases, the video camera was positioned behind the user so as to record both the screen and the hand movements of the user.

4.3.2 Data analysis

Interpersonal communication was examined as a set of codes i.e. verbal codes and nonverbal codes such as facial expression, gaze, gestures and other bodily movements, bodily posture, orientation, territorial behaviour and nonverbal aspects of speech [Har93]. The collaboration process was observed using the twelve behaviour categories in Interaction Process Analysis [Har93] i.e. shows solidarity, shows tension release, agrees, gives suggestion, gives opinion, gives orientation, asks for suggestion, asks for opinion, asks for orientation, disagrees, shows tension, shows antagonism.

During each episode, the following seven categories were measured for each participant:

1. Gives suggestion/opinion/orientation
2. Asks for suggestion/opinion/orientation
3. Agrees
4. Disagrees
5. Shows solidarity
6. Shows tension
7. Shows tension release

These seven behaviour categories were identified through verbal or nonverbal codes. Nonverbal communication was normally used to show tension or tension release. The rest of the behaviour categories were observed mainly through verbal codes. The difference between agreements and solidarity was as follows; a member of a team *agrees* with his/her partner when a suggestion/opinion/orientation given is completely understood and accepted by the other member; *shows solidarity* behaviour was identified when a suggestion/opinion/orientation is encouraged without necessarily being fully understood by the team partner. Table 4.2 presents an illustrative example for each of these behaviour categories.

No	Behaviour category	Example
1	Gives suggestion/opinion/orientation	*I think the two big triangles should form a square*
2	Asks for suggestion/opinion/orientation	*Where do you think the rhomboid goes?*
3	Agrees	*Yes, you are right*
4	Disagrees	*No, I don't think so...*
5	Shows solidarity	*Ok, go on...*
6	Shows tension	Nonverbal codes
7	Shows tension release	Relaxing, Laughing

Table 4.2 Examples for each behaviour category

Teamwork was observed in both co-located and distributed environments based on the primary elements to cooperative work teams i.e. communication, co-location, coordination and collaboration described in chapter two [Pen00].

The approach taken was to observe co-location and coordination through codes such as territorial behaviour and gestures and to category se communication and collaboration according to the behaviour categories in the Interaction Process Analysis as shown in table 4.3.

Element	Behaviour category
Communication	Gives suggestion/opinion/orientation Agrees Disagrees Shows tension Shows tension release
Collaboration	Gives suggestion/opinion/orientation Asks for suggestion/opinion/orientation Shows solidarity

Table 4.3. Association table

The behaviour category *Gives suggestion/opinion/orientation* was used to measure both communication and collaboration because it was assumed in this model that this behaviour category is present in both.

The analysis phase began by making a transcript of each session completed by the observer's notes. The protocols of each session were divided into episodes delineated by a suggestion, opinion or orientation given by one of the team members that changed the direction in the problem solving process. The segmentation of protocols for the co-located test was significantly easier than the one for the distributed test because there was only one tape to be analysed. However, the distributed test consisted of two protocols (one for each member of the team), which were segmented into the same episodes based on the same suggestion/opinion/orientation by interlacing one video over the other.

Another aspect of the analysis compared the interpersonal communication skills and how they are sustained in the co-located and in the distributed environments. These skills include nonverbal communication, reinforcement, questioning, reflecting, explanation and listening [Har93].

The transcripts of each session included exact verbalization made by the subject, observer's reminders, and a measure for each behaviour category (see appendix 3). An example is presented in table 4.4.

Observer notes:

- Problems in thinking aloud
- One of the users tries solutions without any explanation to the other member of his team, making or letting the other (unconsciously) to think more and giving the other the time to come up with new strategies in solving the problem
- Both participants found the game difficult especially because of the scale
- Team work was difficult because of one dominant user
- The user who is not in control of pieces has time to think about new solutions making suggestions all the time (define roles of team members)

No	Time Start	Duration	Give suggestion/ opinion/ orientation		Ask for suggestion/ opinion/ orientation		Agree		Disagree		Shows solidarity		Shows tension		Shows tension release	
			U1	U2	U1	U2	U1	U2	U1	U2	U1	U2	U1	U2	U1	U2
1	30:37	0:50	2	2			2		1							
2	31:27	1:03	3	1			1									
3	32:30	0:45	2	1										1		
4	33:15	1:30	4	3		1	1									
5	34:45	0:37	2	2			1				1					
6	35:22	1:21	2	1	1		1		1		1					
7	36:43	1:01	2	1			1							1		
8	37:44	0:10	2	1												
9	37:54	1:21	3													
10	39:15	0:40	2	1											1	
11	39:55	1:07	1	2					2		1					
12	41:02	0:59	2												1	

Observer at 33:25

Observer at 39:20

Task accomplished - Duration: 11 minutes and 24 seconds – 12 episodes

Table 4.4 Transcript example of a protocol session

The transcript of each session was completed by line charts built for each of the behaviour category measured. An example for the Give suggestion/opinion/orientation category corresponding to the above table is presented in figure 4.4.

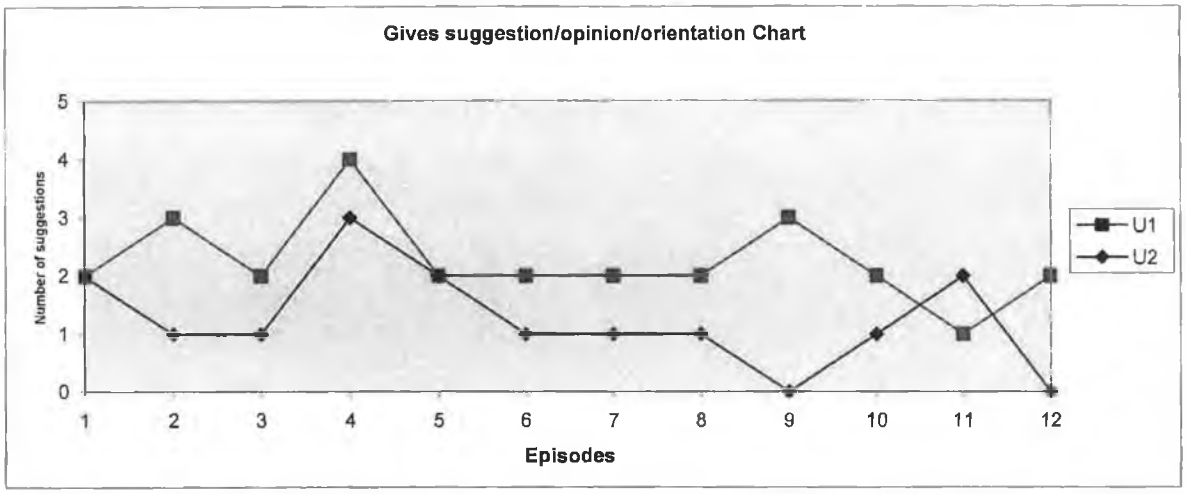


Figure 4.4 Example of Give suggestion/opinion/orientation line chart

4.3.3 Protocol analysis results

For the co-located test both verbal and nonverbal communication were used to solve the problem. The user attention was focused on the game on the table rather than on the other person. It was concluded that nonverbal communication was important from three perspectives as follows:

- Gestures
- Territorial behaviour
- Nonverbal aspects of speech

Other nonverbal codes such as facial expression, gaze, bodily posture and orientation played an insignificant role because of the problem nature (the focus of attention was on the game).

The distributed environment forced people to verbally communicate more since nonverbal communication had to be almost entirely replaced by verbal communication and mouse pointing (instead of finger pointing). Gestures normally used in face-to-face communication were still used in the distributed test but were replaced in time by mouse point and verbal codes. Because the team members had to share the game in the distributed environment, territorial behaviour was eliminated creating room for nondominant people to better participate in the problem solving process. Audio technology was successfully used during the distributed test allowing the use of nonverbal aspects of speech (e.g. “ums”, “ahs”, giggles, pauses, silence, hesitation).

In general, both environments supported defined roles of participants: one member of the team prefers to try out ideas without much explanation or to control the mouse while the second member of the team takes the thinker role and gives suggestions, opinions and orientations. The difference between the two types of environments is that in the distributed one any idea is well explained first and executed only if the team gives it a chance of success whilst in the co-located environment parallel work was favoured.

For each team, the number of communication codes presented in table 4.3 was summed up and measured against the time took to solve the task and against the number of episodes. It should be noticed that each team received the same problem to solve (the same shape to assemble) in the protocol analysis experiments. Therefore, two types of communication charts resulted as follows:

1. *Communication / Time Chart*, which contains values calculated after the following formula for each team: the sum of Gives suggestion/opinion/orientation, Agrees, Disagrees, Shows tension and Shows tension release times showed by both members of the team divided to the total time of the session.
2. *Communication / Episodes Chart*, which contains values calculated after the following formula for each team: the sum of Gives suggestion/opinion/orientation, Agrees, Disagrees, Shows tension and Shows tension release times showed by both members of the team divided to the number of episodes identified in the segmentation process.

Figure 4.5 presents the communication charts for the co-located test compared to the distributed one for each team. The communication charts suggest that more verbalization has been used per episode in the distributed environment. During the co-located test, participants communicated very well with each other and tried out more solutions. In the distributed environment, the subjects preferred to explain any idea before changing the direction in finding the solution. This resulted in a higher number of episodes for the co-located test and a better average communication per episode in the distributed environment. The co-located environment provides a better communication environment by supporting both verbal and nonverbal codes.

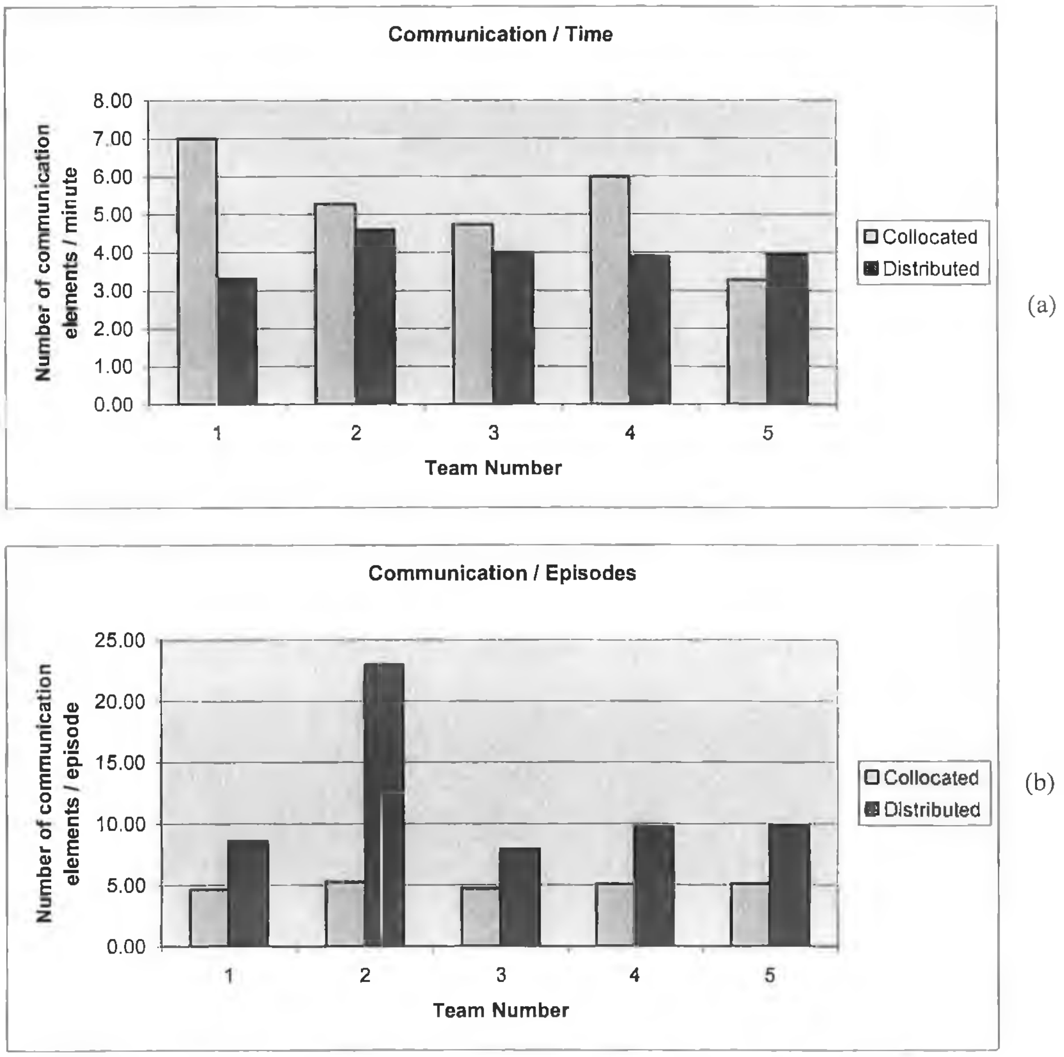


Figure 4.5 (a) Communication/Time Chart (b) Communication/Episodes Chart

In general, subjects got to a solution quicker in the distributed environment because they were forced to think before they communicate. Therefore, the co-located environment favoured learning by doing (situatedness) whilst the distributed environment favoured learning by communication, commonality, interdependency and infrastructure (Vygotskian principles of learning [Bal01]).

The collaboration element was also measured based on the association with the behaviour categories presented in table 4.3. Figure 4.6 compares the collaboration charts created for the two protocol analysis experiments (co-located vs. distributed).

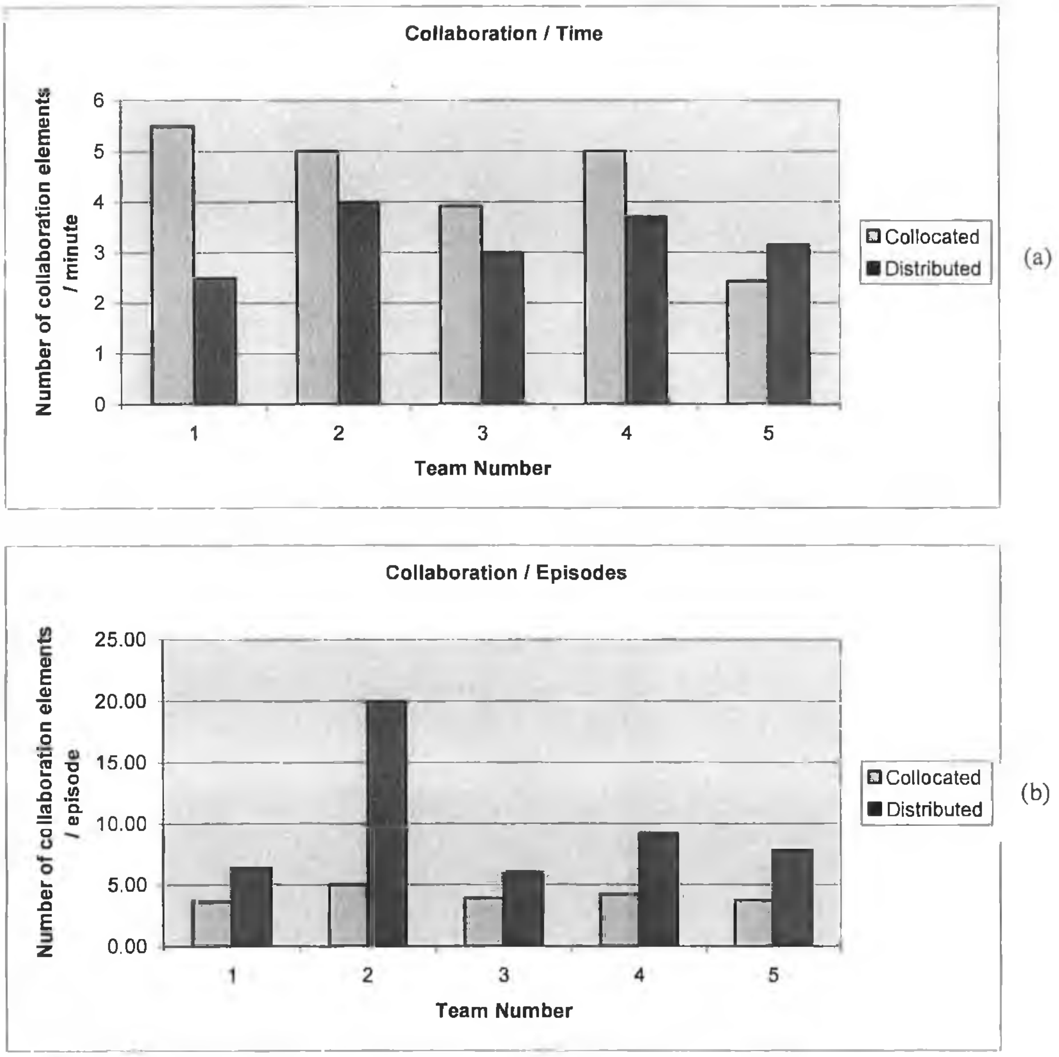


Figure 4.6 (a) Collaboration/Time Chart (b) Collaboration/Episodes Chart

The collaboration charts presented in figure 4.6 were created similar to the communication charts:

1. *Collaboration / Time Chart*, which contains values calculated after the following formula for each team: the sum of Gives suggestion/opinion/orientation, Asks for suggestion/opinion/orientation and Shows solidarity times showed by both members of the team divided to the total time of the session.
2. *Collaboration / Episodes Chart*, which contains values calculated after the following formula for each team: the sum of Gives suggestion/opinion/orientation, Asks for suggestion/opinion/orientation and Shows solidarity times showed by both members of the team divided to the number of episodes identified in the segmentation process.

The collaboration charts show that subjects asked more for suggestions or opinions from the other member of their team in the distributed environment during an episode. This result is in alignment with the findings on examining the communication charts. However, nonverbal communication aided the collaboration process in the co-located environment while the technologies available did not support it in the distributed environment. Reinforcement (behaviour that encourages the other member of the team to carry on whatever he/she is doing) and reflecting (behaviour that forces the other member of the team to detail an idea) had to be verbalized during the distributed test.

As implied in the communication and collaboration charts, there were more episodes in the co-located environment than in the distributed one. This is because the subjects were less inclined to try out possible solutions in the distributed environment. Any idea was well thought and discussed before it was applied (learning by communication and not by doing).

The PA experiments on co-located and distributed environments can be concluded as follows:

- Teamwork can be analysed by measuring and studying communication, collaboration, co-location and coordination using the twelve behaviour categories in Interaction Process Analysis (see tables 4.2 and 4.3).
- During a distributed protocol analysis session, multiple interlaced protocols produced by distributed subjects collaborating in a virtual environment have to be analysed.
- Three protocols resulted after the distributed test i.e. think aloud, communication and mouse tracking protocols.

4.3.4 Questionnaire results

The questionnaire for both co-located and distributed tests included the following topics (see appendix 1):

- Rate the collaboration process between team members during the problem solving process on a 1 to 7 scale (where 1 represents very poor and 7 represents very good).
- Rate the importance of knowing the other member of the team.
- List some positive and negative aspects of the teamwork that occurred.
- Compare the co-located and distributed environment.
- Any further comments.

The questionnaire used for the distributed test also required some comments on the communication environment used to collaborate over a computer network, which reflected the existing restrictions if any and the importance of technologies available to communicate i.e. video, audio, instant messaging and whiteboard.

The questionnaire showed that all subjects have an engineering background, are using a computer on a regular basis and are working in a team based environment. All participants stated that they did not feel constrained by the video camera. The questionnaire results can be summarised as follows:

1. The collaboration process was highly rated as participants felt that a second opinion on the problem speeds up the process of finding a solution.
2. After the co-located test, subjects felt that the collaboration would probably be worse in a distributed environment but changed their opinions after the distributed test.
3. The dominance of some team members was seen as a negative aspect of the collaboration process in both tests.
4. The majority of subjects did not feel restricted by the communication technology in the distributed environment. Audio was the main technology used to communicate in a closed relationship with video technology while chat and whiteboard played an insignificant role.
5. Some participants felt that a better collaboration occurred in the distributed environment rather than the co-located one because it forces people to act in turn and to better explain their ideas through verbal codes.

4.3.5 The protocol analysis template for distributed environments

Any distributed application should support and maintain a cooperative work through efficient communication, co-location, coordination and collaboration. The results of the co-located and distributed tests suggest that the testing of a distributed application using the protocol analysis technique should include the analysis of the *communication* and *collaboration* process among distributed team members based on the seven behaviour categories and the association presented in table 4.3. *Co-location* and *coordination* can be studied by examining the screen capture of the distributed application as used by the subject. Therefore, video and audio recording and screen capture routines are recommended to be used to record user verbalization and to track user actions. A think aloud protocol will result from the videotape/audiotape of each user verbalizing their task-related thoughts while interacting with the application. A mouse tracking protocol will be the result of the screen capture video, with the possibility of associating each mouse action with a timestamp. Because of the distributed and collaborative nature of the environment where the concurrent protocol analysis is applied, a third type of protocol will be recorded: the communication protocol.

From the preliminary tests described in the previous section it can be concluded the data obtained via a distributed and concurrent protocol analysis will consist of the following three protocols [Chi02]:

- Think aloud protocol
- Mouse tracking protocol
- Communication protocol

Communication will be examined through both think-aloud and communication protocols. Collaboration data will be extracted from the communication protocol while co-location and coordination will be studied using the mouse tracking protocol.

The arguments for accepting verbalization reports resulted from think aloud exercises as a reflection of the cognitive activity have already been well established [Ben01, Bra00, Cha00, Eri99, Ger01, Gol95, Hen95, Roc99]. Also, the compatibility of the two modes of thought verbalization i.e. think aloud and communication has been discussed in the third section of this chapter. Thinking aloud and communicating with other users are considered similar reflections of the cognitive processes analysed [Gol95] and therefore, information provided by both of them will be used in the process of evaluation.

While the think aloud protocol analysis will highlight results and data regarding the human-to-computer interaction, the communication protocol will result in data on human-to-human interaction. The actions of each user on the distributed network will be recorded simultaneously. User verbalizations (think aloud and communication protocols) will be captured and the user's actions will be tracked into a timestamp video (mouse tracking protocol).

The analysis phase of the protocol will have to take into account that resulting data from all distributed stations are interrelated. During a protocol session, the three protocols mentioned will be recorded according to a timing scheme similar to the one presented in figure 4.7. The mouse tracking protocol is captured during the entire protocol session. The think aloud protocol and the communication protocol are recorded alternatively. Between any two moments in time there is either the think aloud protocol or the communication protocol taking place while the mouse tracking protocol is a continuous one.

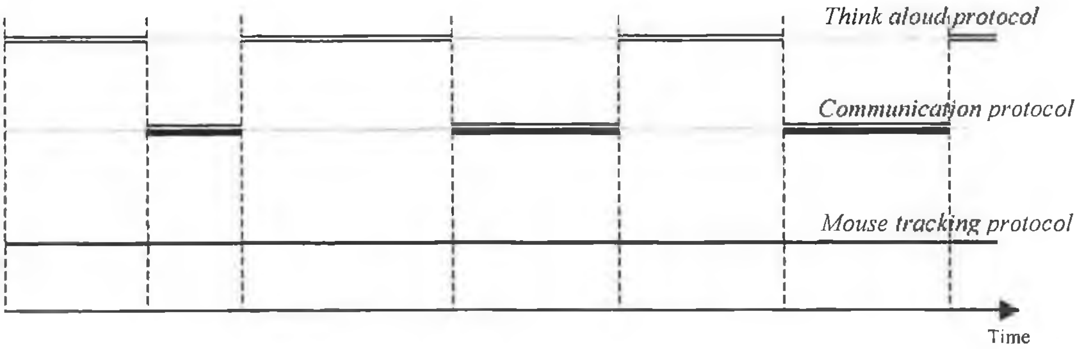


Figure 4.7 Timing scheme of recording the three types of protocol: the think aloud protocol, the communication protocol and the mouse tracking protocol

A positive aspect of applying protocol analysis in distributed environments is the reduced time of collecting the users verbal reports. This implies however that the method is more expensive as the number of technical equipments necessary to record user verbalization and actions increases in the same time with the number of subjects that simultaneously undertake the specified tasks.

The proposed approach of testing the DFE Workbench in a distributed environment consists of applying the proposed distributed protocol analysis template. Teamwork will be observed through communication, collaboration, co-location and coordination. The protocol analysis results can be combined with questionnaire data to achieve more complete results.

4.4 Conclusions

This chapter has reviewed the concept of distributed collaborative environment and the protocol analysis technique. Different types of protocols, the analysis process, advantages and disadvantages of the method as well as different case studies of protocol analysis in the human-computer interaction and engineering fields have been presented. It has been showed that protocol analysis is a robust and systematic technique and the most efficient one used in the evaluation of computer-based systems.

The necessity of implementing a technique to apply protocol analysis in a distributed environment for evaluating a computer-based system has been emphasized. The design and development of this technique was based on the results of a protocol analysis experiment that studied the collaboration process between team members during a problem solving problem in two cases as follows: firstly, when subjects were located in the same room and secondly, when subjects were distributed over a computer network and used a virtual communication environment to collaborate.

The proposed distributed protocol analysis template will be used to evaluate the DFE Workbench software in a distributed design environment. Users interaction with the computer and with each other via the graphical user interface of the DFE Workbench will be observed and recorded using video techniques and screen capture routines. Three protocols will be analysed as follows: the think aloud protocol, the mouse tracking protocol and the communication protocol. Teamwork will be studied through communication, co-location, coordination and collaboration.

Protocol analysis results will be combined with questionnaire results in an attempt to achieve more detailed information about user thinking and problems that occur during the process of user interaction with an interface in a distributed environment.

Hence, the proposed approach to test the DFE Workbench software in a distributed design environment includes:

- Concurrent protocol analysis will be applied in a distributed environment using the proposed distributed protocol analysis template.
- The analysis process will consist of modelling, scanning and scoring techniques.
- The behaviour categories described in section 4.4 as well as teamwork elements such as communication, collaboration, co-location and coordination will be studied.
- The protocol analysis results will be combined with questionnaire data.

The next chapter presents the data analysis and the results of the distributed protocol analysis test on the DFE Workbench software.

Chapter 5

Testing the DFE Workbench Software in a Distributed Environment

5.1 Introduction

The DFE Workbench software has been tested in a distributed design environment using the distributed protocol analysis template presented in the previous chapter. The analysis phase of the interlaced distributed protocols was based on the segmentation of the protocols according to the behaviour categories identified in chapter four. Teamwork elements such as communication, collaboration, co-location and coordination were studied using the three protocols recorded i.e. the think aloud protocol, the mouse tracking protocol and the communication protocol.

This chapter commences with a short description of the distributed protocol analysis test on the DFE Workbench software. It continues by describing the data analysis phase of the distributed protocols including the segmentation process and the creation of a process flow model for the different DFE Workbench screens. The results of the distributed protocol analysis test and of the questionnaire on the DFE Workbench software are presented.

The conclusions of the test presented at the end of the chapter are focused on the functionality of both the DFE Workbench software and the distributed protocol analysis template developed as part of this research.

5.2 Test description

The distributed protocol analysis template developed in chapter four was used to test the DFE Workbench software tool in a distributed design environment. Protocol analysis results were combined with questionnaire data in an attempt to achieve more complete results at the end of the test.

A team of two engineers (referred to as Subject 1 and Subject 2) distributed over a computer network were asked to complete a set of tasks using the DFE Workbench software. Both subjects were working on computers connected to the Oracle database server with SolidWorks 2000 installed (figure 5.1). The DFE Workbench Enterprise was running for Subject 1 while Subject 2 used the DFE Workbench Desktop. Both workstations were connected to the Lotus Sametime Server as described for the distributed test in chapter four.

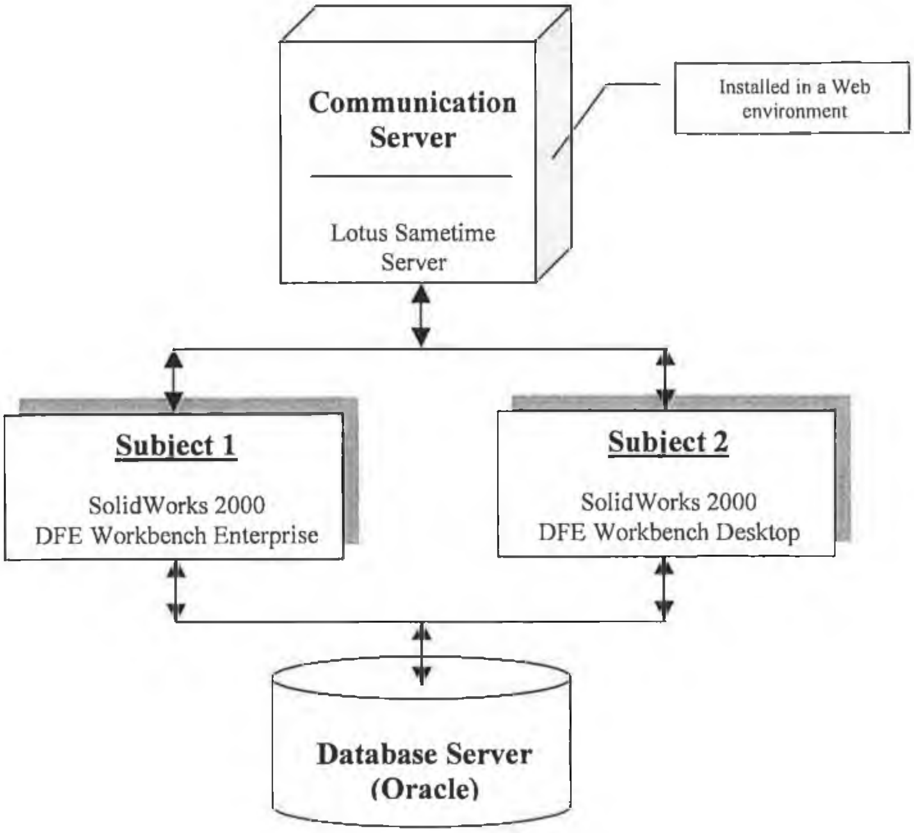
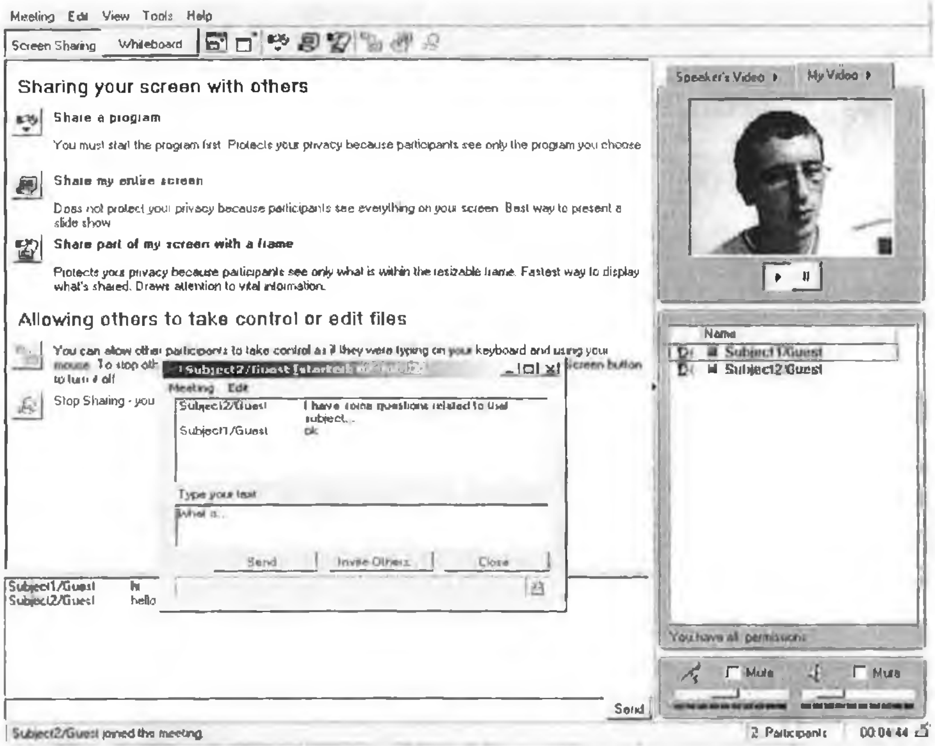


Figure 5.1 The environment for the DFE Workbench test

The collaboration process was facilitated by a virtual meeting room (i.e. Lotus Sametime) that was enabled with communication technologies such as audio, video, instant messaging and whiteboard (see figure 5.2).



Sharing your screen with others

Share a program

You must start the program first. Protects your privacy because participants see only the program you choose

Share my entire screen

Does not protect your privacy because participants see everything on your screen. Best way to present a slide show

Share part of my screen with a frame

Protects your privacy because participants see only what is within the resizable frame. Fastest way to display what's shared. Draws attention to vital information.

Allowing others to take control or edit files

You can allow other participants to take control as if they were typing on your keyboard and using your mouse. To stop oth[illegible] screen button to turn it off

Stop Sharing - you

Subject2/Guest [started [illegible]]

Meeting Edit

Subject2/Guest I have some questions related to that subject...
Subject1/Guest ok

Type your text

What is...

Subject1/Guest hi
Subject2/Guest hello

Subject2/Guest joined the meeting

You have all permissions

Figure 5.2 Lotus Sametime collaboration environment

An observer was present with each of the users to monitor the subject's actions and behaviour and to remind him/her to talk aloud when necessary. The test was divided in three parts as follows:

- *User Introduction* – the context of the test was explained to participants and the environment of the test was described (see table 5.1).
- *The DFE Workbench Tasks* – this part consisted of the actual performing of the tasks assigned to each user. The set of tasks assigned to each of the users was designed in such a way that the users would need each other results in order to complete some of the tasks.
- *Questionnaire* – a short review was held at the end of the test in which participants were asked to rate the ease of learning and use of the DFE Workbench, to comment on the collaboration process and to list any negative or positive aspects of the interaction that took place over a computer network through the DFE Workbench (see appendix 2).

Participant Introduction and Instructions
The DFE Workbench Test

Dear Participant,

Thank you for giving the time to this distributed protocol analysis study. This activity is intended to evaluate the collaboration process among team members distributed over a computer network and working together in a problem solving process.You are part of a two-member team that is assigned to solve a problem using the DFE Workbench software tool. Your role is to perform a set of tasks and in the same time give a running comment about what are you attempting to do. In other words, you are being asked to "think aloud". Also, you have to collaborate with the second member of your team in completing the task using the communication technology available. Feel free to use any of the tools included to communicate: chat, audio or videoconference, whiteboard.

Tasks for Subject 1

1. Open the assembly '*testasm*' in SolidWorks 2000 and save it using the DFE Workbench.
2. Having the assembly '*testasm*' opened in IAS mode, add the subassembly created by Subject 2.
3. Create a new version of the assembly '*testasm*'.
4. Improve the environmental impact of the new version created ('*testasm-1*') using the **Prioritisation Module** and the **Advisor Agent,** involving Subject 2 where necessary.
5. Print out an IAE Detailed report on the improved version of '*testasm*'.

Tasks for Subject 2

1. Open the assembly '*testsub*' in SolidWorks 2000 and save it using the DFE Workbench.
2. Add a database component ('*Baterrie Alkaline*') to the assembly '*testsub*'.
3. Generate an IAE Detailed report on '*testsub*' and save it as a PDF file.
4. Improve the environmental impact of the '*testsub*' assembly using the **Prioritisation Module** and the **Advisor Agent** collaborating with Subject 1where necessary.
5. Print out an IAE report on the improved version of '*testsub*'.

Operation of the test

The test is based on a research method called protocol analysis. You will be videotaped while performing the tasks. Your actions as well as verbalisations will be recorded, so it is extremely important to remember to think aloud while solving the problem. During the session, a researcher will be present with you in the room having the role to observe your actions, to record your words and to remind you to speak in case you forget that. You may ask the observer questions, otherwise please try to ignore his presence in the room. After the test is complete a short review will be held in which you will be asked to complete a short questionnaire.

Thank you for your time.

Table 5.1 Participant Introduction and Instructions for the DFE Workbench test

One of the tasks of the test was to save an assembly from SolidWorks 2000. The assembly selected (figure 5.3) consisted of four components and a subassembly. Subject 1 had to save the components of the assembly i.e. radioactive cover, difusor, difusor cover and button while the corresponding task assigned to Subject 2 was to save the subassembly (that consisted of three components i.e. base, battery alkaline and PCB).

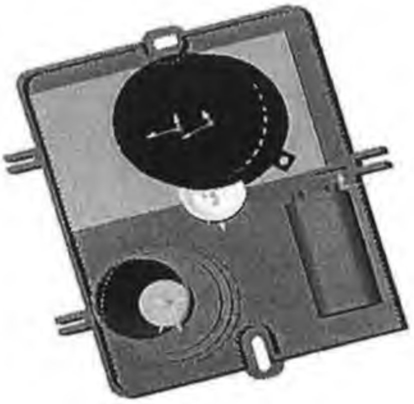


Figure 5.3 The CAD model of the assembly selected for the test

Both subjects received the information presented in table 5.2, which characterises all the components of the assembly selected for the test.

Component Name	**Material**	**Process**	**Finishing**	**Transport**	**Distance [km]**	**Use**	**EOL**
Base	ABS GF30	Injection moulding	None	Truck	100	None	Polymers landfill
Cover	ABS GF30	Injection moulding	None	Truck	100	None	Polymers landfill
Radioactive cover	ABS GF30	Injection moulding	None	Truck	100	None	Polymers landfill
Radioactive element	ABS GF30	Injection moulding	None	Truck	100	None	Polymers landfill
Difusor	Copper E-Cu	Casting	None	Truck	100	None	Copper landfill
Batterie alkaline	Not available	Not available	Not available	Truck	100	Electricity Low Voltage	Not available
Button	PP	Injection moulding	None	Truck	100	None	Polymers landfill
Difusor cover	PP	Injection moulding	None	Truck	100	None	Polymers landfill

Table 5.2 The components table

However, the data provided to subjects has changed after the environmental improvements performed at subassembly and component level within the assembly. Also, the drawings of all components and subassemblies were available in SolidWorks 2000 for each subject.

The set of tasks assigned to each of the subjects was designed from the following perspectives:

- To capture the approach used to solve a problem using the DFE Workbench software in a distributed environment.
- To allow the identification of the problems faced by the users when collaborating over a computer network through the graphical user interface of the DFE Workbench software (the second and fourth tasks of the test were created so as to require the collaboration between subjects e.g. the improvement of the total environmental impact of the assembly).
- To monitor the co-location supported by the DFE Workbench tool in a distributed environment.
- To determine the importance of communication tools (e.g. audio, video, chat) during a distributed problem solving process using the DFE Workbench.

Exact verbalizations made by users while performing the tasks were registered using a video camera positioned behind the right shoulder of the subject.

5.3 Data analysis

The transcript of the DFE Workbench protocol analysis session was designed to support the capture and analysis of the following:

- The subject's exact verbalizations.
- The observer's notes.
- The records of the user's actions.

Three protocols were recorded as follows: think aloud, communication and mouse tracking protocols. The transfer between talk aloud and communication protocols did not represent a difficulty for the subjects.

Figure 5.4 shows the process flow for the protocol recording activity. After Subject 1 completed task 1.1, input from Subject 2 was required in order to perform task 1.2. A high level of communication between the subjects was required at this point, hence a communication protocol was recorded. After each subject completed the first three tasks assigned, collaboration was necessary in order to complete the fourth task. Therefore, the think aloud and communication protocols were recorded alternatively. The mouse tracking protocol was recorded during the whole session.

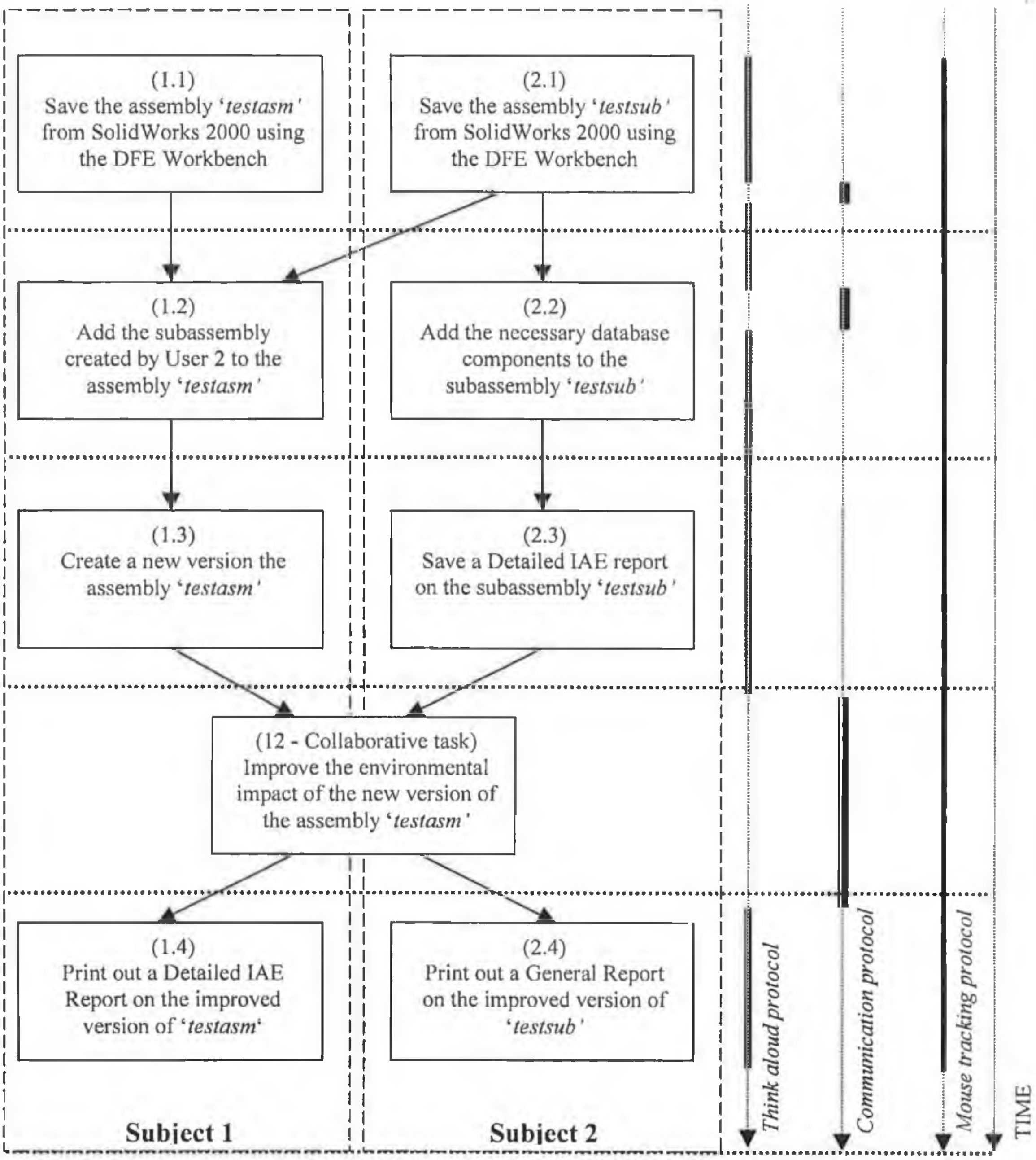

(1.1)
Save the assembly '*testasm*' from SolidWorks 2000 using the DFE Workbench
(2.1)
Save the assembly '*testsub*' from SolidWorks 2000 using the DFE Workbench
(1.2)
Add the subassembly created by User 2 to the assembly '*testasm*'
(2.2)
Add the necessary database components to the subassembly '*testsub*'
(1.3)
Create a new version the assembly '*testasm*'
(2.3)
Save a Detailed IAE report on the subassembly '*testsub*'
(12 - Collaborative task)
Improve the environmental impact of the new version of the assembly '*testasm*'
(1.4)
Print out a Detailed IAE Report on the improved version of '*testasm*'
(2.4)
Print out a General Report on the improved version of '*testsub*'
Subject 1
Subject 2
Think aloud protocol
Communication protocol
Mouse tracking protocol
TIME

Figure 5.4 The think aloud, communication and mouse tracking protocols occurrence during the tasks

A process flow model for the screens necessary to complete the DFE Workbench tasks was created (see table 5.3). Each screen was assigned a screen code, which includes a letter and a number representing the task number from figure 5.4 in which the screen was needed (e.g. screen code A(1.1,1.2) represents screen A required to support tasks 1.1 and 1.2).

Screen Code	Screen Name	Steps
A(1.1,1.2)	DFE Workbench in SolidWorks2000	1. Save a part (menu, toolbar, right click menu) 2. Choose *Save Part in DFE*
B(1.1,1.2)	Choose assembly	1. Select the desired option (new or existing) 2. Select the assembly name 3. Press the *Save Part(s)* button
C(1.1,1.2)	Life Stages Selection	1. Select the material, process, finishing, transport, usage and EOL values using the combo box, the *List All* button or the *Search Engine* 2. Input the distance value 3. Press the *Save* button
D(1.1,1.2)	Input component name	1. Input the name 2. Press the *OK* button
E(1.2)	IAS Window	Not defined
F(2.2)	Add component	1. Select the type of component to add (New, Database, Existing) 2. Pres the *OK* button
G(2.2)	Add database component	1. Select the component type 2. Select the usage and transport values using the combo box, the *List All* button or the *Search Engine* 3. Input the distance value 4. Press the *Save* button
H(1.2)	Add subassembly	1. Select the subassemblies to add 2. Press the *Add subassembly* button 3. Press the *OK* button
I(1.3)	Create new version	1. Select *File/New* version 2. Press the *OK* button
J(12)	Prioritisation module	1. Press the *Advisor* button to get the Advisor module for the current component 2. Press the *Ignore component* button to ignore the current component and get to the next one
K(12)	Advisor	1. Select an alternative from the list 2. Press the *Save* button to save changes or *Cancel* to cancel changes
L(2.3,1.4,2.4)	Report Console	1. Choose assembly (optional) 2. Choose the type of report (General, IAE, IAE Detailed, Disassembly, Joints) 3. Select the charts for the report 4. Press the *Print/Preview Report* button
M(2.3,1.4,2.4)	Report Viewer	1. Select the *Print* button to print the report or *Export to PDF* to save the report in PDF format (from the menu or from the toolbar) 2. Close the window (click the close button or select *File/Close*)
N(1.3, 2.3,1.4,2.4)	DFE Workbench main window	Not defined

Table 5.3 Screens necessary to complete the set of tasks using the DFE Workbench

The segmentation of the transcripts of the think aloud protocol was done according to the screens and steps that subjects used (the mouse tracking protocol). During each episode, the behaviour categories described in chapter four i.e. gives suggestion/opinion/orientation, asks for suggestion/opinion/orientation, agrees, disagrees, shows solidarity, shows tension and shows tension release were measured. Table 5.4 presents the transcript of the protocol analysis session, which includes the observer's notes, the segmentation of the episodes and the measurement of the seven behaviour categories for each subject.

Observer's notes:

- No problem with verbalization.
- Only four minutes were necessary to save the components needed from the CAD system
- The transfer between the DFE Workbench screens and Lotus Sametime is tedious.
- Efficient collaboration during the process of improvement of the final assembly.
- The '*Refresh*' button was quickly noticed and understood.
- The Report Console was not very easy to access.
- Agrees, Shows solidarity, Gives suggestion/opinion/orientation, Asks for suggestion/opinion/orientation behaviour categories present during the collaboration process.

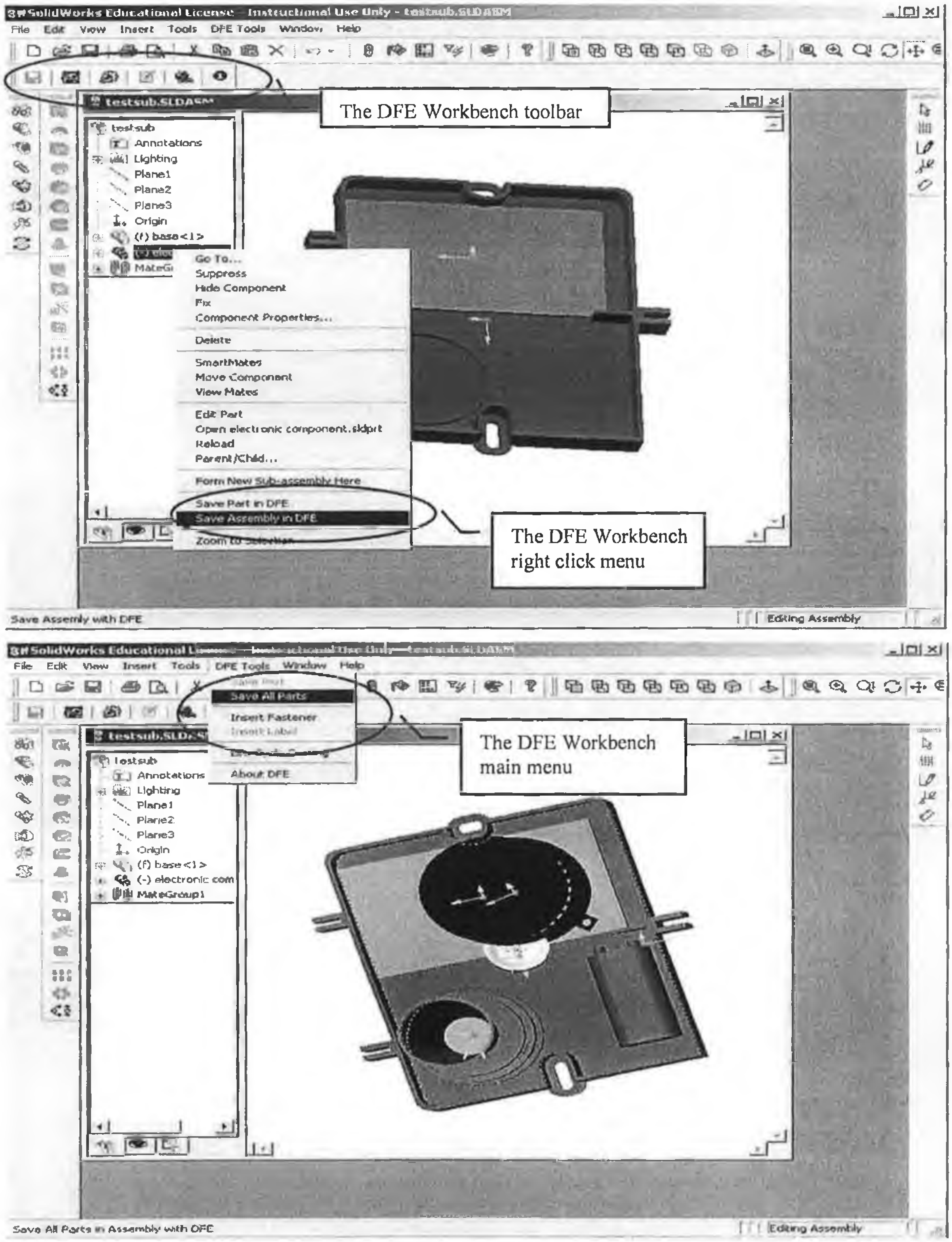
No	Time Start	Time End	Give suggestion/ opinion/ orientation		Ask for suggesti on/ opinion/ orientati on		Agree		Disagre e		Shows solidarity		Shows tension		Shows tension release	
			U1	U2	U1	U2	U1	U2	U1	U2	U1	U2	U1	U2	U1	U2
1	6:50	7:20		1	1		1									
2	11:35	13:12	1	3	1		1	1								
3	16:52	22:16	8	2		4	2	3			1				1	1
4	22:17	26:00	10	3	1	3	2	3							1	1
Duration: 28 minutes – 4 episodes																

Table 5.4 Transcript of the DFE Workbench protocol analysis session

The communication protocol was analysed by looking at the seven behaviour categories described in the previous chapter (communication and collaboration analysis). Co-location and coordination were studied by evaluating the screens used during the communication protocol.

5.4 Protocol analysis results

None of the users had any problems accessing the DFE Workbench application from SolidWorks 2000 (figure 5.5). Both the *'DFE Tools'* menu and the right click menu corresponding to a component have been successfully used.



The DFE Workbench toolbar

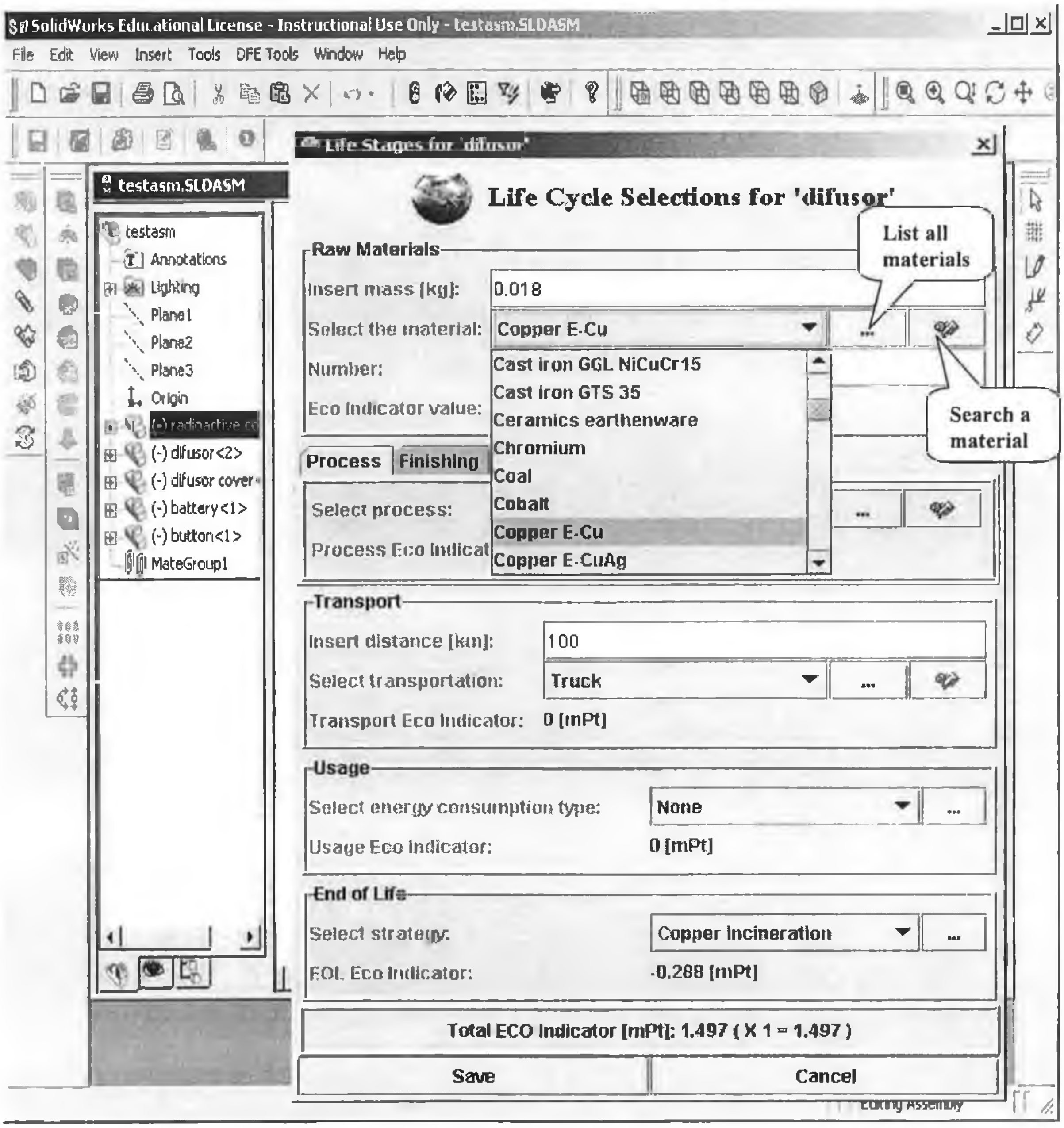
The DFE Workbench right click menu

The DFE Workbench main menu

Figure 5.5 The DFE Workbench menus, toolbar and buttons in SolidWorks 2000

During the life stages selection (screen C, step 1), both users preferred to scroll down the combo box to select the appropriate material instead of using features such as *'list all materials'* with their properties or *'search a material'* with specified properties (see figure 5.6).

Subject 1 finished the first task quicker than Subject 2 and starts the first communication episode. Waiting for the necessary input from Subject 2 to perform the second task, Subject 1 starts to make a prioritisation at the assembly level and tries out some features of the DFE Workbench such as listing hazardous components and renaming an assembly.



Life Cycle Selections for 'difusor'

Raw Materials

Insert mass [kg]: 0.018

Select the material: Copper E-Cu

Number:

Eco Indicator value:

Cast iron GGL NiCuCr15
Cast iron GTS 35
Ceramics earthenware
Chromium
Coal
Cobalt
Copper E-Cu
Copper E-CuAg

Process | Finishing

Select process:

Process Eco Indicat

Transport

Insert distance [km]: 100

Select transportation: Truck

Transport Eco Indicator: 0 [mPt]

Usage

Select energy consumption type: None

Usage Eco Indicator: 0 [mPt]

End of Life

Select strategy: Copper Incineration

EOL Eco Indicator: -0.288 [mPt]

Total ECO Indicator [mPt]: 1.497 (X 1 = 1.497)

Save | Cancel

Figure 5.6 Life Cycle Selection window

The second communication episode is initiated by Subject 2 who provides the name of the subassembly created. Subject 1 found the screen H (*'Add subassembly'*) easy to use. The subassembly created by Subject 2 was immediately identified and added to the entire assembly (see figure 5.7).

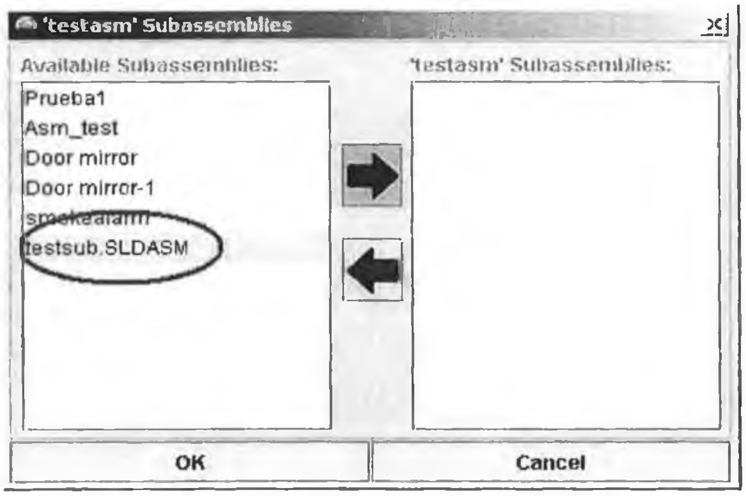
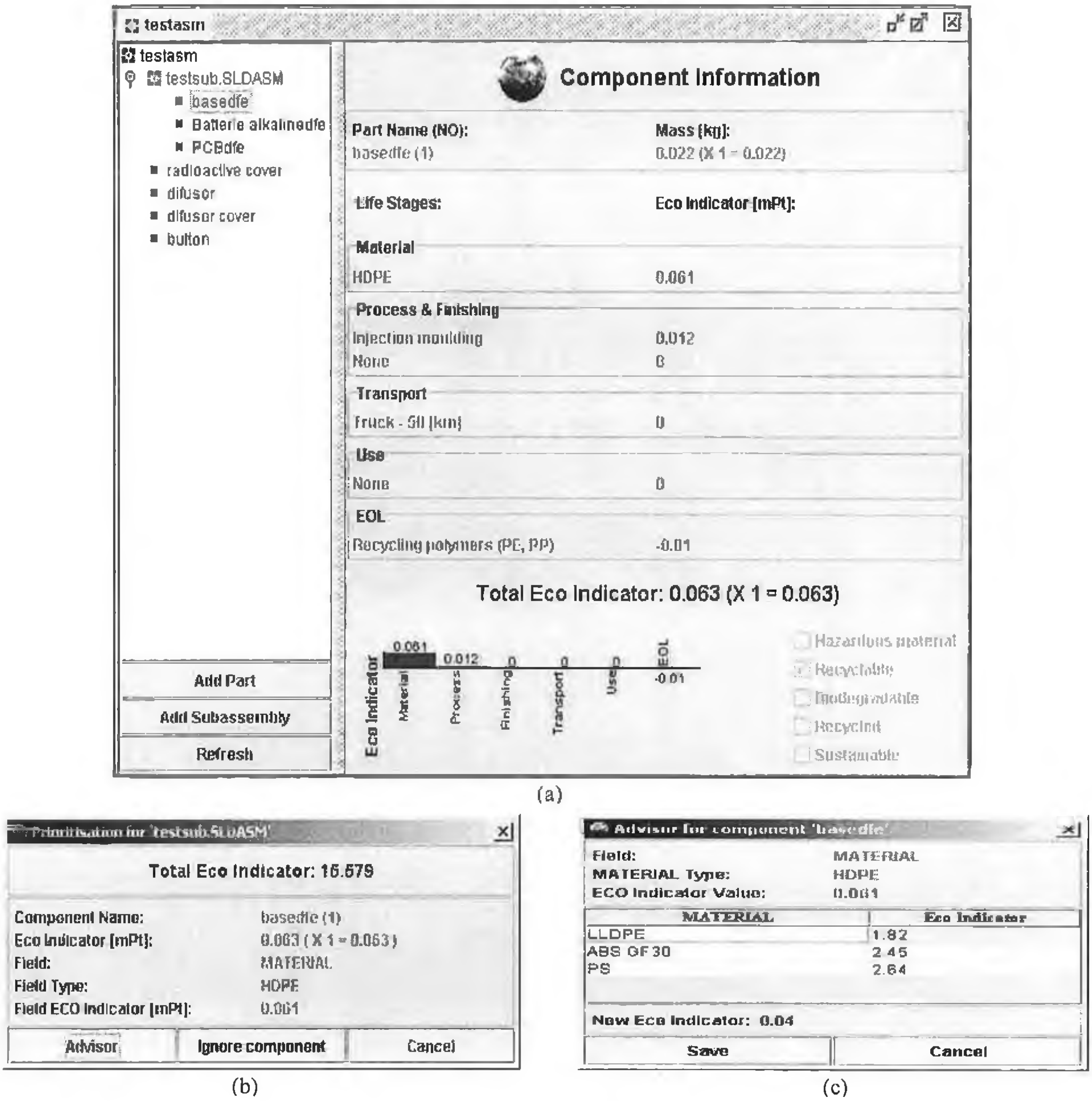


Figure 5.7 Add subassembly screen (screen H)

Efficient coordination completed by the co-location offered by the DFE Workbench software was observed at this point of the session from both communication and mouse tracking protocols.

The DFE Workbench software provides distributed users with the infrastructure necessary to collaborate in the process of improvement of an assembly from an environmental point of view through screens such as N, E, J and K (see table 5.3). The mouse tracking and communication protocols showed that the DFE Workbench software supported the coordination of activities.

A third communication episode was recorded during the fourth task, which required efficient collaboration between subjects in order to improve the environmental impact of the entire assembly. For example, Subject 1 communicates his/her opinion on a problem *("Maybe you can modify something on the base component...I used the material LLDPE for my components...")* after consulting the information displayed by the IAS module (screen E) for the subassembly created by Subject 2 (see figure 5.8a). As the same data is displayed through the distributed screens, Subject 2 transmits his/her opinion on the matter and asks for more direction from Subject 1 (Ask for suggestion/opinion/orientation behaviour category). Based on the advice from Subject 1 and the information provided by the Prioritisation Module and the Advisor Agent, Subject 2 further improves the environmental impact of the product by changing the material for a component from HDPE to LLDPE (see figure 5.8).



testasm

testasm
testsub.SLDASM
basedfe
Batterie alkalinedfe
PCBdfe
radioactive cover
difusor
difusor cover
button

Add Part
Add Subassembly
Refresh

Component Information

Part Name (NO):	Mass [kg]:
basedfe (1)	0.022 (X 1 = 0.022)
Life Stages:	Eco Indicator [mPt]:
Material	
HDPE	0.061
Process & Finishing	
Injection moulding	0.012
None	0
Transport	
Truck - 50 [km]	0
Use	
None	0
EOL	
Recycling polymers (PE, PP)	-0.01

Total Eco Indicator: 0.063 (X 1 = 0.063)

□ Hazardous material
☑ Recyclable
□ Biodegradable
□ Recycled
□ Sustainable

(a)

Prioritisation for 'testsub.SLDASM'

Total Eco Indicator: 15.579

Component Name:	basedfe (1)
Eco Indicator [mPt]:	0.063 (X 1 = 0.063)
Field:	MATERIAL
Field Type:	HDPE
Field ECO Indicator [mPt]:	0.061

Advisor | Ignore component | Cancel

(b)

Advisor for component 'basedfe'

Field:	MATERIAL
MATERIAL Type:	HDPE
ECO Indicator Value:	0.061

MATERIAL	Eco Indicator
LLDPE	1.82
ABS GF30	2.45
PS	2.64

New Eco Indicator: 0.04

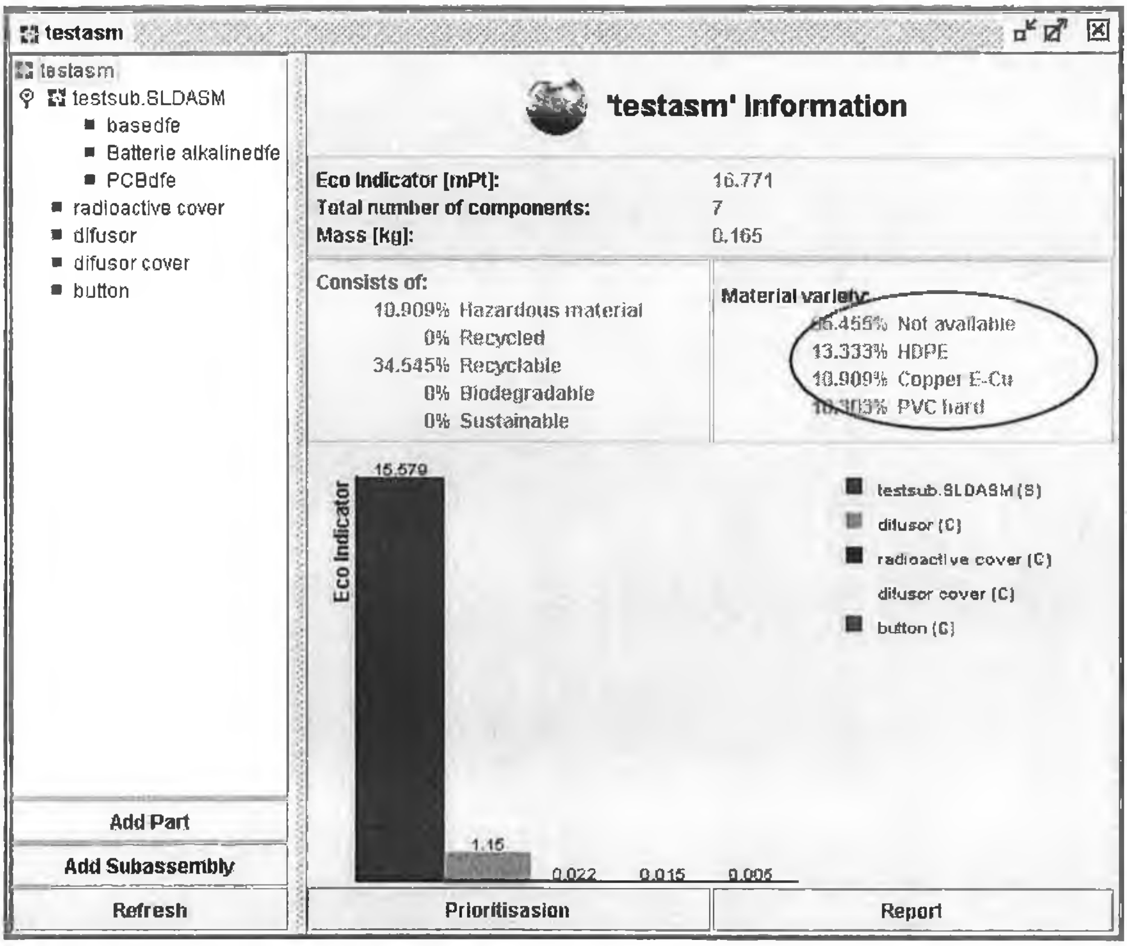
Save | Cancel

(c)

Figure 5.8 (a) The IAS window at Subject 1 workstation (b,c) The Prioritisation Module and the Advisor Agent at Subject 2 workstation

There were numerous instances in the protocol analysis session where bidirectional collaboration was observed.

The last communication episode is initiated by Subject 1 who proposed a comparison between the initial assembly and its improved version with a view to identifying potential for further improvements. For example, Subject 2 was able to make a suggestion of improvement for one of the components created by Subject 1 based on the material variety displayed in the IAS window – screen E (see figure 5.9).



Eco Indicator [mPt]: 16.771
Total number of components: 7
Mass [kg]: 0.165

Consists of:
10.909% Hazardous material
0% Recycled
34.545% Recyclable
0% Biodegradable
0% Sustainable

Material variety:
65.455% Not available
13.333% HDPE
10.909% Copper E-Cu
10.303% PVC hard

Figure 5.9 The IAS window (screen E) for the entire assembly

Whilst instant messaging, video and whiteboard technologies were available, the subjects mainly used the audio technology to communicate.

The number of suggestions/opinions/orientations given or asked for, agreements/disagreements, times when solidarity, tension or tension release was expressed were counted and summed up for each episode as explained in section two. The result is the communication and collaboration chart of the session (figure 5.10), which shows that both communication and collaboration between participants were higher during the latter episodes of the session (the improvement of the environmental impact associated with the entire assembly created).

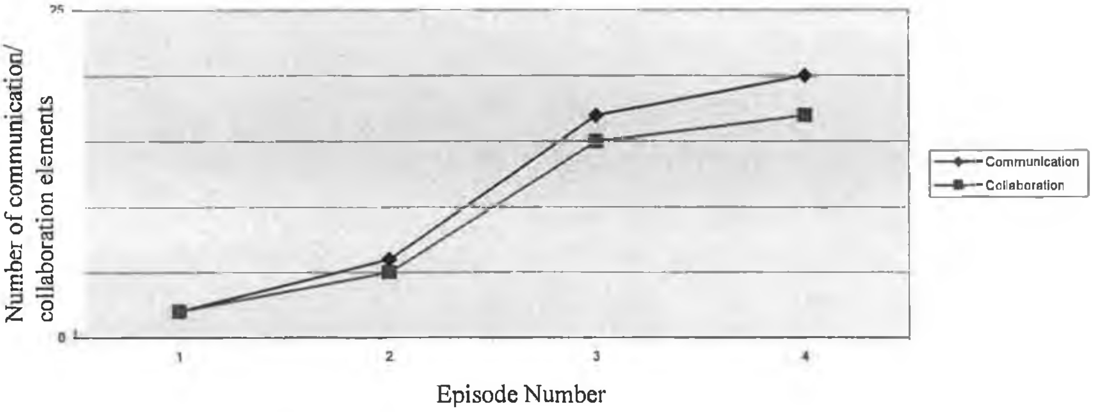
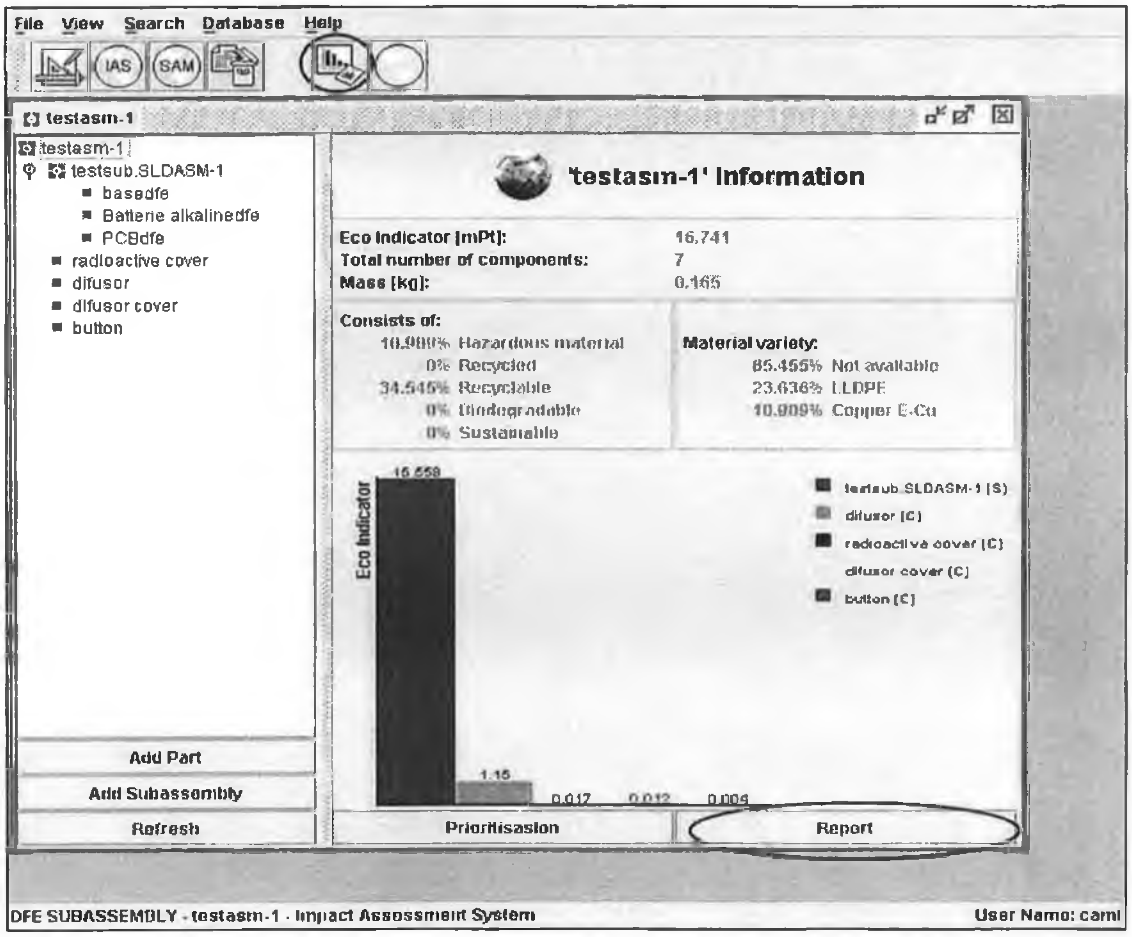


Figure 5.10 The communication and collaboration chart for each episode of the communication protocol

The think aloud and mouse tracking protocols also revealed some usability problems of the DFE Workbench software. For example, the *Report* button in screen E is easy to be missed out because of its position within the frame (figure 5.11).



'testasm-1' Information

Eco Indicator [mPt]: 16.741
Total number of components: 7
Mass [kg]: 0.165

Consists of:
10.909% Hazardous material
0% Recycled
34.545% Recyclable
0% Biodegradable
0% Sustainable

Material variety:
65.455% Not available
23.636% LLDPE
10.909% Copper E-Cu

DFE SUBASSEMBLY - testasm-1 - Impact Assessment System User Name: caml

Figure 5.11 The *Report* button placements within the DFE Workbench software

Subjects used the general report button (present in the DFE Workbench toolbar) that requires the selection of the assembly or subassembly to be reported instead of choosing to use the *Report* button from the IAS window, which allows direct access to reports specific to the assembly opened.

However, these problems need further research, as the focus of this research was not the testing of the graphical user interface of the DFE Workbench.

The results of the distributed protocol analysis test of the DFE Workbench software can be summarised as follows:

- None of the subjects had any problems nor to verbalize their thoughts neither to communicate with the other member of their team during the test.
- The transfer between thinking aloud and communication did not represent a difficulty for subjects.
- The collaboration between users resulted in the improvement of the total environmental impact through changes made at assembly level as well as subassembly level.
- Audio technology and instant messaging were the main communication technologies used to collaborate during the problem solving process.
- The communication process was aided by the DFE Workbench software through the screens presented in table 5.3.
- Coordination and co-location teamwork elements were both supported by the DFE Workbench software.
- The three protocols recorded i.e. think aloud, mouse tracking and communication protocols showed that the DFE Workbench supports distributed users in the decision making process by offering the infrastructure necessary to collaborate in a distributed design environment.

5.5 Questionnaire results

The questionnaire for the DFE Workbench software included the following topics (see appendix 2):

- Rate the collaboration process between team members during the problem solving process on a 1 to 7 scale (where 1 represents very poor and 7 represents very good).
- Rate the ease of learning of the DFE Workbench software on a 1 to 7 scale.
- Rate the ease of use of the DFE Workbench software on a 1 to 7 scale.

- List some positive and negative aspects of the DFE Workbench software in a distributed design environment.
- List any restrictions of the communication technology used.
- Rate the importance of the different technologies available to communicate in a virtual environment e.g. video, audio, instant messaging and whiteboard.
- Any further comments.

The questionnaire showed that none of the participants felt constrained by the video camera behind them. Both subjects have an engineering background, use the computer on a regular basis and were already familiar with the DFE Workbench software. However, this was the first time that they used the DFE Workbench software in a distributed design environment. The questionnaire results can be summarised as follows:

1. The DFE Workbench software is considered to be easy to learn and to use (rated as an average of 1.5 on a 1 to 7 scale, where 1 represents "very easy" and 7 represents "very difficult").
2. The collaboration process was highly rated by both subjects.
3. Access to the analysis of the shared assembly in real time was considered a positive aspect.
4. Even if the communication technology was relatively easy to use, using the Lotus Sametime and the DFE Workbench software alternatively was considered a negative aspect of the collaboration process.
5. Audio and whiteboard communication technologies were highly rated whilst video and chat were rated as less significant.

5.6 Protocol analysis conclusions

5.6.1 Conclusions on the DFE Workbench software

The analysis of the distributed protocol analysis session showed that the DFE Workbench software is suitable for use in distributed design environments offering the infrastructure required for the collaboration and communication among distributed participants. The mouse tracking protocol combined with the results of the communication protocol showed that the graphical user interface of the DFE Workbench helped the interaction between distributed users. However, the DFE Workbench does not support the exchange of information between participants. Therefore, a communication technology i.e. Lotus Sametime had to be used to provide the communication environment to distributed users.

Using the DFE Workbench and Lotus Sametime alternatively proved to be difficult: the users had to change the focus of attention from the DFE Workbench to the Lotus Sametime window. However, after the communication was established audio technology facilitated efficient collaboration allowing the transfer of the focus back to the DFE Workbench window. The distributed protocol analysis test of the DFE Workbench software resulted on specific observations at a detailed level as follows:

- A communication process has to be established to find out the name of a subassembly to get from the *Add subassembly* list, process that can become difficult when a large amount of data is handled.
- The information displayed by the IAS module of the DFE Workbench has to be refreshed to get the data updated by another user in the distributed environment, process that can be automated.
- The IAS module (through features such as life stages information at component level, material variety and percentages/list of hazardous content) helps designers to offer suggestions to distributed users for further improvement of the environmental impact of the assembly.
- The Prioritisation Module and the Advisor Agent used at both assembly and subassembly level in the DFE Workbench Enterprise offer the user a better understanding of the problem and new ideas for improvement from an environmental point of view.
- The Report Generator can be accessed from any workstation in the distributed environment to report the last available environmental information associated with an assembly/subassembly.
- Audio technology supports the communication process among distributed users in the best way. Instant messaging and whiteboard are also important.
- The switch between the DFE Workbench and Lotus Sametime for the communication process is tedious. Since it was showed that audio and chat are important communication technologies, it can be concluded that an integrated communication tool should be developed.

General observations on the DFE Workbench software when used in a distributed design environment can be summarised as follows:

- The DFE Workbench supports the collaboration process among distributed users through the design of its graphical user interface.

- The DFE Workbench offers the infrastructure necessary for efficient communication among distributed users.
- The DFE Workbench does not support the management of the communication processes in a distributed environment.
- The DFE Workbench helps distributed users to create a shared understanding of the problem (see the system configuration presented in figure 3.1).

The main conclusion of the distributed protocol analysis test on the DFE Workbench software is that even though the application supports the collaboration process among distributed users, further improvement can be achieved to refine collaboration support in a distributed design environment.

5.6.2 Conclusions on the distributed protocol analysis template

The protocol analysis template developed as part of this research proved to be efficient in highlighting the problems of the DFE Workbench software when used in a distributed environment. Communication and collaboration measured against the behaviour categories in the Interaction Process Analysis offered a way to analyse the episodes identified in the think aloud and communication protocols. Co-location and coordination were studied using the mouse tracking and communication protocols, which indicated the suitability of a software application in distributed environments. However, the protocol analyses proved to be time consuming. The activity of counting the number of suggestions/opinions/orientations given by a subject when performed by humans is error prone. The time needed for the segmentation of the protocols and the analysis phase was high because of the distributed nature of the test: one session resulted in two protocols to analyse. The time as well as the cost of applying the method increases when the protocol analysis technique is applied on more than two distributed users.

The positive aspects of applying the distributed protocol analysis template to evaluate a distributed application can be summarised as follows:

- The three protocols recorded i.e. the think aloud protocol, the communication protocol and the mouse tracking protocol offer an understanding of the human-to-human interaction in the distributed environment via the application under evaluation.
- The measurement of communication and collaboration processes indicates the level of team cooperation during the problem solving process using the distributed application.

- Co-location and coordination observed from the communication and mouse tracking protocols analysis indicates the support offered by the application for the collaboration among distributed users.
- The technique can be used for usability analysis.

However, the following negative aspects were observed:

- The distributed protocol analysis method is data intensive (multiple interlaced protocols have to be analysed).
- The method can be expensive when applied to a large number of distributed users.
- The human element can cause errors in the communication and collaboration measurement.

In summary, the protocol analysis template designed to test a distributed collaborative process is capable of evaluating distributed problem solving processes. However, some improvements need to be made specifically the automation of the data synthesis and evaluation activities.

5.7 Conclusions

This chapter presented the testing phase of the DFE Workbench software in a distributed collaborative environment. The method of the test was the distributed protocol analysis template designed and developed in chapter four. The data analysis focused on three areas as follows:

1. The analysis of distributed interlaced protocols through the think aloud, communication and mouse tracking protocols recorded.
2. The measurement of the following behaviour categories (described in chapter four): gives suggestion/opinion/orientation, asks for suggestion/opinion/orientation, agrees, disagrees, shows solidarity, shows tension and shows tension release.
3. The study of communication, collaboration, co-location and coordination teamwork elements.

The results of the test showed that the DFE Workbench software supports distributed users during a problem solving process in a virtual environment and offers the infrastructure necessary for an efficient collaboration over a computer network.

The conclusions of the distributed protocol analysis test presented at the end of the chapter are focused on two areas:

A. The functionality and usability of the DFE Workbench software in distributed environments.

B. The functionality of the distributed testing methodology developed as part of the research.

The chapter concluded that the DFE Workbench software is suitable for use in distributed design environments by supporting the coordination of activities among distributed users and helping participants to the decision making process to create a shared understanding of the problem. However, it has been showed that the DFE Workbench software does not support the direct communication among distributed users. The transfer of the focus between the application and the environment used to communicate proved to be tedious. Since audio and instant messaging communication technologies are important for the process of collaboration, it has been concluded that a DFE Workbench integrated communication tool should be designed and implemented to aid distributed users during a problem solving process.

The conclusions of the distributed protocol analysis template include positive aspects such as the analysis of the think aloud, communication and mouse tracking protocols, the measurement of the communication and collaboration elements and the study of co-location and coordination teamwork elements. However, negative aspects of the methodology were also identified e.g. work intensive method, errors in the measurement of communication and collaboration elements, expensive method.

The results of the distributed protocol analysis template developed in chapter four and applied in this chapter on the DFE Workbench software form the basis of the specification for further development of both the application tested and the distributed protocol analysis template developed.

Chapter 6

Conclusions, Recommendations and Future Work

6.1 Thesis summary

This thesis presents the research carried out over a period of two years in the following areas (see figure 6.1):

- *Design for Environment* – requirements for the design of tools and methodologies to support the development of environmentally superior products.
- *Distributed Design Environments* – area of integration of DFE methodologies into distributed design environments (due to current trends in the design field toward virtual teams that collaborate over computer networks to achieve global optima in design).
- *CAD Systems* – area of integration of the DFE methodologies into virtual prototyping environments through the use of Application Programming Interfaces.
- *Protocol Analysis* – the application of the protocol analysis technique in the evaluation of distributed software applications.
- *Programming languages* – the selection of appropriate software tools for the development of CAD integrated and distributed application.
- *Database systems and models* – the selection of an appropriate system model for distributed environments.
- *Programming paradigms* – the investigation of programming models for the implementation of a distributed software tool.

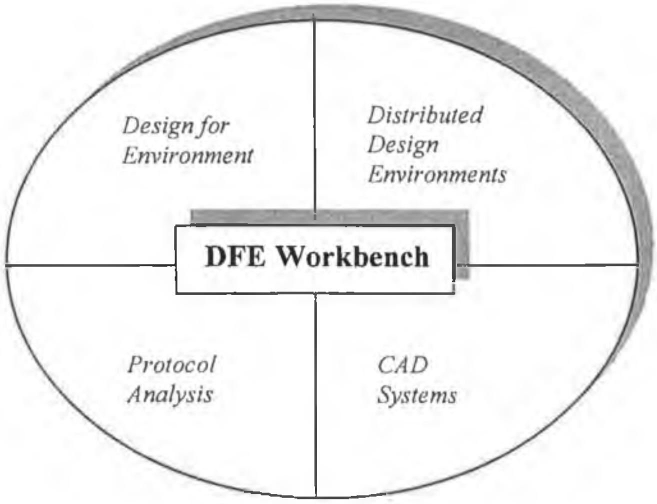


Figure 6.1 Current thesis research areas

The research areas described above form the basis of the development and evolution of the DFE Workbench software because of the following considerations:

- The DFE Workbench tool has to incorporate all the requirements of *Design for Environment* tools and methodologies identified in chapter two.
- The DFE Workbench tool has to be functional in a *Distributed Design Environment.*
- The DFE Workbench tool has to be integrated in the prototyping environment primarily used by the designer (*CAD system*).
- The testing phase of the DFE Workbench tool is based on distributed *protocol analysis* techniques.

The work involved in the development of the current thesis was focused on two main areas as follows:

A. The design and development of a CAD integrated Design for Environment software tool i.e. the DFE Workbench.
B. The testing phase of the DFE Workbench software in a distributed design environment using a methodology based on protocol analysis.

Firstly the thesis presents a review of the existing DFE tools and methodologies concluded with a specification document for a new DFE tool i.e. the DFE Workbench that is CAD integrated, platform independent and suitable for use in distributed design environments.

Next the DFE Workbench software tool is presented by describing the system configuration, the software architecture, the database structure and the integration of the tool with two CAD systems i.e. SolidWorks 2000 and Pro Engineer 2001. The collaboration with industrial partners from the automotive and electronic sectors facilitated the continuous testing and improvement of the DFE Workbench software throughout the project evolution.

The second part of the thesis presents the final testing phase of the DFE Workbench software in a distributed design environment using protocol analysis techniques. A method for applying the protocol analysis technique in a virtual collaborative environment has been designed and developed based on the results of some preliminary protocol analysis experiments, which studied the behaviour of team members during a problem solving process in both co-located and distributed environments. The new developed protocol analysis template is based on the study of communication, collaboration, co-location and coordination teamwork elements from distributed interlaced protocols. The description and results of the test are presented in chapter five.

The current chapter presents the conclusions of the thesis and proposes further development of both the DFE Workbench software and the distributed protocol analysis template.

6.2 Conclusions

The review of DFE tools and methodologies presented in chapter two is concluded with a specification document for a new software based system that implements the DFE Workbench methodology. The new specification is based on requirements derived from the standalone version of the software as well as new functional, non-functional and usability requirements. It has been concluded that the new version of the DFE Workbench software must meet the following requirements:

- The application has to be integrated in the CAD environment used by the designer and has to communicate with the CAD system to retrieve the necessary information on a product prototype.
- The software has to implement both Impact Assessment System (IAS) and Structure Assessment Method (SAM) methodologies and they have to communicate with each other. The advisor agent and the knowledge base agent are required in the new system to help the designer in the process of evaluation and improvement of a product from an environmental point of view.
- The new system should be operable in a different environment than the one it was implemented in with a minimum of configuration changes (portability).

- The application has to be suitable for use in distributed design environments. Concurrent access should be allowed to the DFE Workbench system.
- The new system should make use of a robust client server database system. The database system should include a secure layer (access to the data held in the system should be allowed only on an username/password basis). Database consistency should be assured by the system (database update should be performed by the system using verified data only).
- The application should be supported by a report generator module, which allows access to different file formats containing reports and graphical displays of all the environmental scores and the other metrics calculated by the tool.

The testing phase of the new developed DFE Workbench software commenced with a review of the protocol analysis technique. The conclusions of the research in the area of protocol analysis are as follows:

- The protocol analysis technique represents a robust and efficient evaluation method for investigating causes of errors, mistakes and misinterpretations during a problem solving process.
- The concurrent protocol analysis technique is a powerful method for obtaining detailed information about user thinking and understanding why problems occur during the process of user interaction with a computer based system.
- The protocol analysis technique was originally designed for one to one evaluations; some research is necessary to apply the protocol analysis technique in a distributed collaborative environment and study interlacing distributed protocols.

The protocol analysis experiments that studied face-to-face collaboration and distributed synchronous collaboration (developed to design an appropriate test to evaluate the cognitive behaviour of distributed teams that collaborate through a graphical user interface) are concluded as follows:

- Three protocols have to be analysed as follows: the think aloud protocol, the mouse tracking protocol and the communication protocol.
- Teamwork can be studied through communication, co-location, coordination and collaboration elements.
- Distributed interlaced protocols can be analysed based on the twelve behaviour categories in Interaction Process Analysis.

The application of the new developed protocol analysis template in a distributed design environment resulted in a set of conclusions for both the DFE Workbench software and the testing methodology used. Based on a limited set of tests[1], the conclusions for the DFE Workbench software can be summarised as follows:

- The DFE Workbench software supports the collaboration process among distributed users through modules such as Impact Assessment System Module, Prioritisation Module and Advisor Agent.
- The DFE Workbench software offers the infrastructure necessary for efficient communication among distributed users.
- The DFE Workbench helps distributed users to create a shared understanding of the problem.
- Audio technology supports the communication process among distributed users in the best way. Instant messaging and whiteboard are also important.
- The DFE Workbench does not support the management of the communication processes in a distributed environment. The switch between the DFE Workbench and the communication technology is tedious. Since it was showed that audio and chat are important communication technologies, it can be concluded that an integrated communication tool should be developed.

The conclusions for the distributed protocol analysis template are as follows:

- The three protocols recorded i.e. think aloud, communication and mouse tracking protocols offer an understanding of the human-to-human interaction in the distributed environment via the application under evaluation. However, the method is data intensive (multiple interlaced protocols have to be analysed) and can be expensive when applied to a large number of distributed users.
- The measurement of communication and collaboration processes indicates the level of team cooperation during the problem solving process using the distributed application. However, the human element can cause errors in the communication and collaboration measurement.
- Co-location and coordination studied from the communication and mouse tracking protocols analysis indicates the support offered by the application for the collaboration among distributed users.

[1] Many protocol analysis studies showed that a small number of users representative of the target population can yield important results [Eri99, Ben01].

6.3 Recommendations for future work

The conclusions presented in the previous section provide the basis for future work that can be performed for the evaluated software application and for the distributed protocol analysis template.

The suggestions for the improvement of the DFE Workbench software to further aid teamwork in a distributed environment can be summarised as follows:

1. Assembly information should include data such as the author, the creation date, the date when the assembly was last modified. This information would help the user to quicker find subassemblies created by other users in the network.
2. Co-location should be improved by adding a Repository Control Module, which would keep track of all assemblies and components by operations such as Check In, Check Out, Release and Obsolete. This module can replace or upgrade the current versioning system of the DFE Workbench.
3. The security module of the DFE Workbench should be improved by creating users and groups of users with access rights such as Create Assembly, Create New Version, Add Component, Rename Assembly/Component, Modify Assembly/Component, Remove Assembly/Component, Access to the Advisor Agent, Create Joint/Obstruction, Modify Joint/Obstruction, Remove Joint/Obstruction and Generate Reports.
4. The DFE Workbench software should have an integrated communication package, which allows the exchange of information, activities and events among distributed users through technologies such as audio, video, whiteboard and message board.

It should be noted that all the improvements suggested above can be implemented in the current version of the DFE Workbench by reusing the entire Java package of the software.

It is proposed to improve the distributed protocol analysis template by implementing an agent-based system, which automatically generates all the information required for the analysis phase. Agents have been identified as the next generation model for engineering complex distributed systems [Woo95, Nwa96, Anu01, Jen00, Lee01]. An agent can be defined as a software and/or hardware component that works in conjunction with people or represent people and act in their behalf [Anu01, Lee01]. The ideal and primary attributes of an agent are as follows (this may not be a necessary or sufficient set) [Nwa96]:

- *Ability to learn* – agents must have the ability to learn as they react and /or interact with their external environment.
- *Ability to co-operate* – agents must have the ability to share information and knowledge with each other and the user.

- *Autonomy* – refers to the ability of the agent to work without human interaction.

The agent-based distributed protocol analysis system should support the synthesis and analysis of interlaced protocols in distributed environments. Figure 6.2 presents the user-agent-server relationship in a distributed environment that can be used for applying the agent based distributed protocol analysis method.

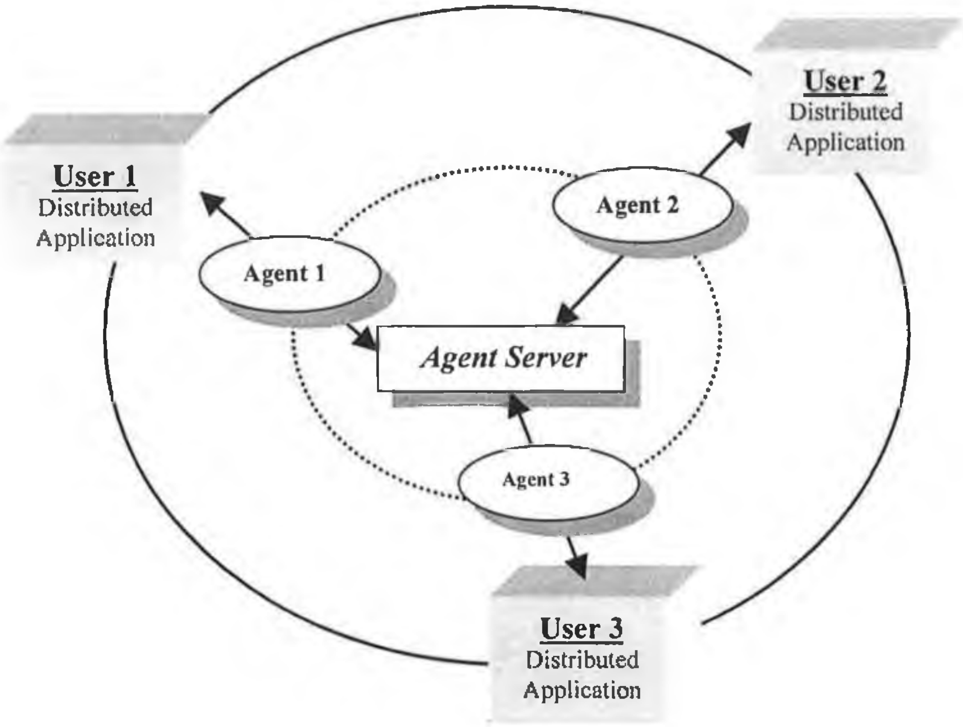


Figure 6.2 Agent-user relationships

The actions of each user on the distributed network will be recorded simultaneously by designating an agent to each distributed user station. This agent will act as a supervisor for other agents assigned for specific tasks such as:

- Capture user verbalization (think aloud and communication protocols).
- Record user actions into a timestamp video (mouse tracking protocol).

The information captured is forwarded to an Agent Server where the protocols from all agents are centralized. The analysis phase of the protocol will have to take into account that resulting data from all agents are interrelated. The segmentation is done at the Agent Server level, but the interpretation of the results will probably still have to be performed by humans.

Further research is necessary to find a method to better measure and evaluate the co-location and coordination offered by a distributed application.

References

[AMET] http://greenmfg.me.berkeley.edu/green/cad/ametide/

[Anu01] Anumba C.J., Ugwu O.O., Newnham L., Thorpe A., “A multi-agent system for distributed collaborative design”, MCB University Press, 2001.

[Atm99] Atman C.J., Chimka J.R., Bursic K.M., Nachtmann H.L., “A comparison of freshman and senior engineering design processes”, Design Studies, Elsevier Science, 1999.

[Bad99] Badescu M., “Development of a Design for Environment Software Tool for the Optimization of the Product Life Cycle”, B.Tech.IV Thesis, 1999.

[Bal01] Bal J., Yvette J.G., "Learning Style Preferences of Engineers in Automotive Design", Journal of Workplace Learning, Vol. 13, No. 6, 2001.

[Ben99] Bennett S., McRobb S., Farmer R. “Object-Oriented Systems Analysis and Design using UML”, McGraw-Hill Publishing Company, 1999.

[BOUS] http://www.boustead-consulting.co.uk/products.htm

[Bra00] Branch J. L., “Investigating the Information-Seeking Processes of Adolescents: The Value of Using Think Alouds and Think Afters”, Library & Information Science Research, 2000.

[Ben01] Benbunan-Fich R., “Using protocol analysis to evaluate the usability of a commercial web site”, Information & Management, 2001.

[Cal97] Caluwe N., “Ecotools manual - A comprehensive review of Design for Environment tools”, Design for the Environment Research Group, Manchester Metropolitan University, 1997.

[Cha00] Chan C. S., “An examination of the forces that generate a style”, Design Studies, Elsevier Science, 2000.

[Chi02] Chira C., Roche T., "Testing and Validation of a Distributed Design for Environment Tool", 19-th International Manufacturing Conference (IMC-19), Belfast, August 2002, In Print.

[Cor97] Cortes L., “Designing a Graphical user Interface”, Medical Computing Today, Elsevier Science, 1997.

[Cro95] Cross N., Clayburn Cross A., “Observations of teamwork and social processes in design”, Design Studies, Elsevier Science, 1995.

[DFDO] http://www.ie.eng.fsu.edu/research/decm/odp_details.html

[DFMA] http://www.dfma.com

[ECOB] http://www.ecobalance.com

[ECOS] http://www.environmental-center.com/software/ecoscan/ecoscan.htm

[Edw95] Edwards A.D.N., "The rise of the graphical user interface", University of York, Department of Computer Science, 1995.

[ENTR] http://www.entropyinternational.com

[Eri84] Ericsson K.A., Simon H.A., "Protocol Analysis: Verbal Reports as Data", The MIT Press, Cambridge, 1984.

[Fik96] Fiksel J., "Design For Environment Creating Eco-Efficient Products and Processes", McGraw Hill, 1996.

[GABI] http://www.gabi-software.com

[Ger97] Gero J.S., McNeil T., "An approach to the analysis of design protocols", Design Studies, Elsevier Science, 1997.

[Ger01] Gero J.S., Tang H.H., "The differences between retrospective and concurrent protocols in revealing the process-oriented aspects of the design process", Key Centre of Design Computing and Cognition, Elsevier Science, 2001.

[Gol95] Goldschmidt G., "The designer as a team of one", Design Studies, Elsevier Science, 1995.

[Gre96] Greenberg S., "Teaching Human Computer Interaction to Programmers". ACM Interactions, 3(4), pp. 62-76, ACM Press, 1996.

[Har93] Hartley P., "Interpersonal Communication", Routledge, 1993.

[Haz96] Hazemi R., Macaulay L., "Requirements for graphical user interface development environments for groupware", Interacting with computers, Elsevier Science, 1996.

[Hen95] Henderson R., Podd J., Smith M., Varela-Alvarez H., "An examination of four user-based software evaluation methods", Interacting with computers, 1995.

[Hua02] Huang S.M., Shieh K.K., Chi C.F., "Factors affecting the design of computer icons", International Journal of Industrial Ergonomics", Elsevier Science, 2002.

[Ihe02] Iheagwara C., Blyth A., "Evaluation of the performance of ID systems in a switched and distributed environment the RealSecure case study", Elsevier Science, 2002.

[ISO97] International Standards Organisation, "Environmental Management - Life Cycle Assessment - Principles and Framework", ISO14040, 1st edition, 1997.

[ISO98] International Standards Organisation, "Ergonomic requirements for office work with visual display terminals (VDTs): Guidance on usability", ISO-9241, 1st edition, 1998.

[Jen00] Jennings N., "On Agent Based Software Engineering", Journal of Artificial Intelligence, V117, PP277-296, Elsevier Science, 2000.

[Kav01] Kavakli M., Gero J.S., "The structure of concurrent cognitive actions: a case study on novice and expert designers", Design Studies, Elsevier Science, 2001.

[Lau01] Laure E., "OpusJava: A Java framework for distributed high performance computing", Elsevier Science, 2001.

[Lee01] Lees B., Branki C., Aird I., "A framework for distributed agent-based engineering design support", Automation in construction, Elsevier Science, 2001.

[Llo95] Lloyd P., Lawson B., Scott P., "Can concurrent verbalization reveal design cognition?", Design Studies, Elsevier Science, 1995.

[LCAIT] http://www.lcait.com

[Man00] Man E., "Development of a Design for the Environment Workbench Software Tool", M.Eng. Thesis, 2000.

[Man02a] Man E., Díez-Campo J.E., Roche T., "DFE Workbench: a CAD integrated DFE tool", in Environmentally Conscious Manufacturing II, Surendra M. Gupta, Editor, Proceedings of SPIE Vol. 4569, page 93-99, 2002.

[Man02b] Man E., Roche T., Browne J., "Development of a CAD integrated DFE Workbench for Design of Extended Products", 2002, In Print.

[Man02c] Man E., Díez-Campo J.E., Chira C., Roche T., "Product Life Cycle Design Using The DFE Workbench", 5th IFIP International Conference on Information Technology for Balanced Automation Systems in Manufacturing and Services (BASYS), Cancun, Mexico, September 2002, In Print.

[Nwa96] Nwana H.S., "Software Agents: An Overview", Advanced Applications & Technology Department, BT Laboratories, Knowledge Engineering Review, Vol. 11, No 3, pp.1-40, Sept 1996, © Cambridge University Press, 1996.

[Orr02] Orr M., "The FCE Specking test: using reports to help interpret test scores", System, Elsevier Science, 2002.

[Pen00] Pena-Mora F., Hussein K., Vadhavkar S., Benjamin K., "CAIRO: a Concurrent Engineering Meeting Environment for Virtual Design Teams", Artificial Intelligence in Engineering, pp 203-219, Elsevier Science, 2000.

[Pre00] Pressman R.S., "Software engineering: a practitioner's approach", McGraw-Hill Publishing Company, 2000.

[PRE] http://www.pre.nl

[Pur98] Purcell A.T., Gero J.S., "Drawings and the design process", Design Studies, Elsevier Science, 1998.

[Roc99] Roche T., "The Development of a DFE Workbench", Ph.D. Thesis, September 1999.

[Roc01a] Roche T., Man E., Browne J., "The Development of a CAD Integrated DFE Workbench Software Tool", Proceedings EcoDesign Conference, Tokyo, pp 223-226, 2001.

[Roc01b] Roche T., Man E., Chira C., Browne J., "The DFE Workbench a Case Study", Second International Symposium on Environmentally Conscious Design and Inverse Manufacturing (EcoDesign 2001), Tokyo Big Sight, Japan, December 2001.

[Roc01c] Roche T., Man E., Browne J., "Development of a CAD Integrated DFE Workbench Tool", Proceedings of International Symposium on Electronics and the Environment, IEEE, Denver, May 2001.

[SALV] http://www.glue.umd.edu/~sandborn/research/salvage.html

[Ste02] Stempfle J., Badke-Schaub P., "Thinking in design teams – an analysis of team communication", Design Studies, Elsevier Science, 2002.

[Tho01] Thoben K.D., Eschenbacher J., Jagdev H., "Extended Products: Evolving Traditional Product Concepts" 7th International Conference on Concurrent Enterprising, pp 429-439, 2001.

[Tol95] Tollmar K., Sundblad Y., "The design and building of the graphic user interface for the collaborative desktop", Computer & Graphics, Elsevier Science, 1995.

[UMB] http://www.umberto.de

[War00] Ward K., Heeman P.A., "Acknowledgments in Human-Computer Interaction", Oregon Graduate Institute of Science & Technology, 2000.

[Wit93] Witten I.H., Greenberg S., "User Interfaces", In A. Ralston and E.D. Reilley (Eds.), Encyclopedia of Computer Science, pp. 1411-1414, Van Nostrand Reinhold, 1993.

[Woo95] Wooldrige M.J., Jennings N.R., "Agent Theories, Architectures and Language: A Survey", 1995.

Bibliography

Arunachalam V., Sasso W., "Cognitive Processes in Program Comprehension: An Empirical Analysis in the Context of Software Reengineering", Journal of Systems and Software, Elsevier Science, 1996.

Baran P., "On Distributed Communications: Introduction to Distributed Network", 1964.

Bartres R., Lu M.L., Naka Y., "An Agent-based Environment for Operational Design", Computers & Chemical Engineering, Elsevier Science, 1997.

Benyon D., "The new HCI? Navigation of information space", Knowledge-Based Systems, Elsevier Science, 2001.

Borkowski A., Branki C., Grabska E., Palacz W., "Towards collaborative creative design", Automation in Construction, Elsevier Science, 2001.

Burgoon J.K., Bonito J.A., Bengtsson B., Cederberg C., Lundeberg M., Allspach L., "Interactivity in human-computer interaction: a study of credibility, understanding, and influence", Computer in Human Behaviour, Elsevier Science, 2000.

Carroll J.M., "Five reasons for scenario-based design", Interacting with Computers, Elsevier Science, 2000.

Chen K.H., Chen S.J., Lin L., Changchien S.W., "An integrated graphical user interface (GUI) for concurrent engineering design of mechanical parts", Computer Integrated Manufacturing Systems, Elsevier Science, 1998.

Davies S.P., "Effects of concurrent verbalization on design problem solving", Design Studies, Elsevier Science, 1995.

Dix A., Finlay J., Abowd G., Beale R., "Human-Computer Interaction", Hillsdale, http://www.hcibook.com/, 1993

Ferber J., "Multi-Agent Systems: An Introduction to Distributed Artificial Inteligence", Addison-Wesley, 1999

Forta B., Weiss N., King A., Dinowitz M., Libertelli G., Bellack J., “Advanced ColdFusion 4.0 Application Development”, Que, 1999.

Frank A., Mitschang B., “A customizable shared information space to support concurrent design”, Computers in Industry, Elsevier Science, 2002.

Franklin S., Graesser A., "Is It an Agent or just a Program? A Taxonomy for Autonomous Agents", Proceedings of the Third International Workshop on Agent Theories, Architectures, and Language, New-York:Springer-Verlag, 1996

Grabowski H., Rude S., Grein G., “Universal Design Theory”, Proceedings of the Workshop Universal Design Theory, Karlsruhe, Germany, 1998.

Greenberg S., “Collaborative Interfaces for the Web”, In C. Forsythe, E. Grose and J. Ratner (Eds), Human Factors and Web Development, Chapter 18, p241-254, LEA Press, 1997.

Greenberg S., “Supporting Casual Interaction between Intimate Collaborators”, In M.R. Little and L. Nigay (Eds) Engineering for Human-Computer Interaction (8th IFIP International Conference, EHCI 2001, Toronto, Canada, May), Lecture Notes in Computer Science Vol 2254, p3, Springer-Verlag, 2001.

Han S., Kim Y., Lee T., Yoon T., “A framework of concurrent process engineering with agent-based collaborative design strategies and is application on plant layout problem”, Computers & Chemical Engineering, Elsevier Science, 2000.

Hartson H.R., “Human-computer interaction: Interdisciplinary roots and trends”, Jpurnal of Systems and Software, Elsevier Science, 1998.

Jinfonet Software, JReport Documentation and Manuals,
http://www.jinfonet.com/docs.htm

Klein J., Moon Y., Picard R.W., "This computer responds to user frustration: Theory, design, and results", Interacting with Computers, Elsevier Science, 2002.

Lefrançois P., Cloutier L., Montreuil B., "An agent-driven approach to design factory information systems", Computers in Industry, Elsevier Science, 1996.

Liang W.L., Huang C.C., "The agent-based collaboration information system of product development", International Journal of Information Management, Elsevier Science, 2002.

Lin Y.B., Geigel J., "A Graphical User Interface Design for Network Simulation", Journal of Systems and Software, Elsevier Science, 1997.

Loia V., Gisolfi A.,"A Distributed Approach for Multiple Model Diagnosis of Physical Systems", Information Sciences, Elsevier Science, 1997.

Lotus IBM, "Getting Started with Sametime Audio/Video",
http://www-10.lotus.com/ldd/products.nsf/products/sametime

Lotus IBM, "Lotus Sametime Application Development Guide",
http://www-10.lotus.com/ldd/products.nsf/products/sametime

Lotus IBM, "Lotus Sametime Real Time Collaboration That's Fit for Business",
http://www-10.lotus.com/ldd/products.nsf/products/sametime

Lotus IBM, "Sametime 2.0 Java Toolkit Tutorial",
http://www-10.lotus.com/ldd/products.nsf/products/sametime

Naressi A., Couturier C., Devos J.M., Janssen M., Mangeat C., Beer R., Graveron-Demilly D., "Java-based graphical user interface for the MRUI quantitation package", Magnetic Resonance Materials in Biology, Physics, and Medicine, Elsevier Science, 2001.

Novick D.G., Maupin K., "Testing User Interfaces",
http://www.cse.ogi.edu/Interactive/usability/

Ong Y.S., Gooi H.B., Lee S.F., "Java-based applications for accessing power system data via intranet, extranet and internet", International Journal of Electrical Power & Energy Systems, Elsevie Science, 2001.

Pelczarski M., "Graphical User Interface Design", http://graphicsmagician.com/vbcourse/11design/gui.htm

Petzold C., "Programare in Windows 95", Teora, 1999.

Reed P., Holdaway K., Isensee S., Buie E., Fox J., Williams J., Lund A., "User interface guidelines and standards: progress, issues, and prospects", Interacting with Computers, Elsevier Science, 1999.

Roche T., Asensio P., Man E., "Environmental Advisor Tool to Support Green Design", ICPR Manufacturing for a Global Market, 1999.

Roche T., "Life Cycle Design of an Environmentally Superior Office Chair", Improving your Business through the Environment Conference, Enterprise Ireland, 2001.

Roche T., Man E., Browne J., "The DFE Workbench", Proceedings of Intelligent Systems and Advanced Manufacturing Conference, Boston, 2001.

Sander V., Erwin D., Huber V., "High-performance management based on Java", Future Generation Computer Systems, Elsevier Science, 1998.

Sen A., Winsor J., Gay R., "Computer Integrated Manufacturing", Volume 1, World Scientific Publishing & Global Publications Services, 1993.

Serco Ltd., "Cost effective User Centred Design", http://www.usability.serco.com/trump/index.htm

Spolsky J., "User Interface Design for Programmers", http://www.joelonsoftware.com/

Sun Microsystems, "Creating a GUI with JFC/Swing",
http://java.sun.com/docs/books/tutorial/uiswing/

Sun Microsystems, "JDBC(TM) Database Access",
http://java.sun.com/docs/books/tutorial/jdbc/

Suwa M., Tversky B., "What do architects and students perceive in their design sketches? A protocol analysis", Design Studies, Elsevier Science, 1997.

Szykman S., Racz J., Sriram R.D., Bochenek C., "Web-based System for Design Artifact Modelling", Journal of Design Studies, 2000.

Talin, "A Summary of Principles for User-Interface Design",
http://www.sylvantech.com/~talin/projects/ui_design.html

Tanenbaum A.S., "Retele de Calculatoare", Computer Press Agora, 1997.

Tang C., Xu L.D., Feng S., "An agent-based geographical information system", Knowledge-Based Systems, Elsevier Science, 2001.

Tzafestas S.G., "Knowledge Based Systems; Advanced Concepts Techniques & Applications", World Scientific Publishing, 1997.

Wu D.J., "Software agents for knowledge management: coordination in multi-agent supply chains and auctions", Expert Systems with Applications, Elsevier Science, 2001.

Appendix 1

The questionnaires for the distributed and co-located tests

Questionnaire – Co-located teams

A. Background ____________________________________

Rate some of your skills on the following scale:

B. Computer usage Novice Medium Advanced
1 ----- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7

C. Team working environments Novice Medium Advanced
1 ----- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7

D. Distributed environments Novice Medium Advanced
1 ----- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7

Questionnaire:

1. Did you feel constrained in any way by the video camera?

□ No

□ Yes

If yes, please explain:

____

____

____

2. How important do you think it is to know the people you are collaborating with?

not important very important
1 ----- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7

□ Don't know

□ Don't understand

3. Rate the collaboration process between you and the other member of your team on the following scale.

very poor   very good
1 ----- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7

□ Don't know

□ Don't understand

4. Do you think you would have done a better job on your own rather then within a team?

yes   no
1 ----- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7

□ Don't know

□ Don't understand

Why?

____

____

____

____

5. How would you rate the collaboration process between you and the other member of your team if you were collaborating in a distributed environment i.e. each member of the team working on a computer and collaborating with the others via a computer network using video/audio, chat, etc. capabilities?

worse   the same   better
1 ----- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7

□ Don't know

□ Don't understand

Why?

____

____

____

____

6. List some of the positive aspect(s) of the collaboration process between you and the other member of your team:

____

____

____

____

7. List some of the negative aspect(s) of the collaboration process between you and the other member of your team:

____

____

____

____

8. Any further comments:

____

____

____

____

____

____

____

____

____

____

Questionnaire – Distributed teams

1. Did you feel constrained in any way by the video camera?

   □ No

   □ Yes

   If yes, please explain:

   ____

   ____

   ____

2. How important do you think it is to know the people you are collaborating with over a computer network?

   not important   very important
   1 ----- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7

   □ Don't know

   □ Don't understand

3. Where do you feel it was more important to know the person you are collaborating with?

   □ Co-located environment

   □ Distributed environment

   □ Equally important in both co-located and distributed environments

   Why?

   ____

   ____

   ____

   ____

   ____

4. Rate the collaboration process between you and the other member of your team on the following scale.

very poor very good
1 ----- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7

□ Don't know

□ Don't understand

5. Do you think you would have done a better job on your own rather then within a team?

yes no
1 ----- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7

□ Don't know

□ Don't understand

Why?

____

____

____

____

6. List some of the positive aspect(s) of the collaboration process between you and the other member of your team:

____

____

____

____

7. List some of the negative aspect(s) of the collaboration process between you and the other member of your team:

____

____

____

____

8. Did you feel restricted in any way by the communication technology used?

□ No

□ Yes

If yes, please explain:

9. Rate the importance of the following technologies available to communicate in a distributed environment:

not important   very important

Video 1 ----- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7

Audio 1 ----- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7

Chat 1 ----- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7

Whiteboard 1 ----- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7

10. Which environment assured a better collaboration between you and the other member of your team?

□ Collocated

□ Distributed

□ Both collocated and distributed

Please explain:

11. Do you think the other member of your team missed some of your communication signals?

□ No

□ Yes

If yes, please explain:

____

____

____

____

12. Any further comments:

____

____

____

____

____

____

____

____

____

____

____

____

Appendix 2

The DFE Workbench questionnaire

Questionnaire – The DFE Workbench

1. Did you feel constrained in any way by the video camera?

   □ No

   □ Yes

   If yes, please explain:

   ____

   ____

   ____

2. Rate the collaboration process between you and the other member of your team on the following scale.

   very poor very good
   1 ----- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7

   □ Don't know

   □ Don't understand

3. Rate the ease of learning of the DFE Workbench on the following scale.

   very easy very difficult
   1 ----- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7

   □ Don't know

   □ Don't understand

4. Rate the ease of use of the DFE Workbench on the following scale.

   very easy very difficult
   1 ----- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7

   □ Don't know

   □ Don't understand

5. List some of the positive aspect(s) of the DFE Workbench tool:

____

____

____

____

6. List some of the negative aspect(s) of the DFE Workbench tool:

____

____

____

____

7. Did you feel restricted in any way by the communication technology used?

□ No

□ Yes

If yes, please explain:

____

____

____

____

8. Rate the importance of the following technologies available to communicate in a distributed environment:

not important   very important

Video 1 ----- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7

Audio 1 ----- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7

Chat 1 ----- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7

Whiteboard 1 ----- 2 ---- 3 ---- 4 ---- 5 ---- 6 ---- 7

9. Do you think the other member of your team missed some of your communication signals?

□ No

□ Yes

If yes, please explain:

10. Any further comments or suggestions for improvement:

Appendix 3

Protocol transcripts for the co-located and distributed tests

Observer's notes and questionnaire observations for the co-located test

Session	**PA observations**	
	User 1 - questionnaire	**User 2 - questionnaire**
Session 1 22.02.02 15:25 – 17:00	• Hard for both users to verbalize their thoughts • Both participants found the game difficult • Good team work	
	- Very good collaboration process - Less time to find a solution within a team - Same collaboration in DE - No dominant participants	- Good collaboration process - Less time to find a solution within a team - Probably takes longer to get to a solution in a DE -People couldn't share different knowledge
Session 2 26.02.02 9:10 – 9:50	• Users collaborated very well communicating all thoughts to each other • The learning element is an issue • Both participants felt that competition is there (even if users are informed that not their intelligence is tested but the collaboration process between them) • The game was too easy (especially after a few minutes of training – the learning element) • Very good parallel work	
	- Important to know the other participant - Good collaboration process - Team member offered ideas/solutions from a different point of view - Probably the collaboration process would be worse in a DE	- Important to know the other participant - Medium collaboration process - A second opinion might speed up the process of finding a solution (different perspectives) - Probably worse in DE because of the dominant people

<table>
<tr>
<td rowspan="2"><b><u>Session 3</u></b>
26.02.02
10:20 – 11:00</td>
<td colspan="2">• Problems in thinking aloud
• One of the users tries solutions without any explanation to the other member of his team, making or letting the other (unconsciously) to think more and giving the other the time to come up with new strategies in solving the problem
• Both participants found the game difficult especially because of the scale
• Team work was difficult because of one dominant user
• The user who is not in control of pieces has time to think about new solutions and makes suggestions all the times – defined roles</td>
</tr>
<tr>
<td>- Important to know the other participant
- Medium collaboration process
- Probably would have done a better job on his own because the possibility that his train of thoughts might be interrupted by the other participant
- Probably worse in DE because the tools for DE are unresponsive compared to human-to-human interaction
- Positive aspect = defined roles</td>
<td>- Important to know the other participant
- Medium collaboration process
- Definitely would have done a better job on his own
- Probably worse in DE because it is easier to communicate face to face
- Positive aspect = second opinion on the problem
- Negative aspect = interruption</td>
</tr>
<tr>
<td rowspan="2"><b><u>Session 4</u></b>
26.02.02
11:10 – 11:55</td>
<td colspan="2">• Good collaboration process (what one says gave the other new ideas)
• User2 had some problems to express his thoughts
• Language might have been an issue in this session
• Good parallel work
• Decisions were agreed by both participants even though user1 seemed more dominant</td>
</tr>
<tr>
<td>- Not very important to know the other participant
- Good collaboration process
- Definitely would not have done a better job alone because of the other participant understanding of the problem and skill input
- Probably worse in DE because of frustration (has to communicate every move explicitly)
- Language to some extent caused a barrier</td>
<td>- Important to know the other participant
- Medium collaboration process
- Time delay because of the need to explain everything to the other participant
- Same towards worse in DE
- Positive aspect = more ideas</td>
</tr>
</table>

Session 5 26.02.02 12:05 – 13:00	• Good collaboration process even though the team did not agree in continuing or not the process of finding a solution (user1 wants to try another shape while user2 does not agree to give up on the current one) • One dominant participant (User1) who also found it very difficult to think aloud • Both participants found the game very challenging • When all team ideas fail, the collaboration process stops and each participant tries to find the solution on his own	
	- Probably not important to know the other participant (3) - Very good collaboration process - Same result alone - Same in DE - Positive aspects = more solutions and time to think reduced	- Important to know the other participant - Medium collaboration process - Probably would not have done a better job alone - Positive aspect = different perspective - Negative aspect = the dominance of the other participant (taking the pieces to his side of the table)

Table 1. Observer's notes and questionnaire observations for the co-located test

Segmentation tables for the co-located test

Session 1

No	Time Start	Duration	Give suggestion/ opinion/ orientation		Ask for suggestion/ opinion/ orientation		Agree		Disagree		Shows solidarity		Shows tension/ tension release	
			U1	U2	U1	U2	U1	U2	U1	U2	U1	U2	U1	U2
1	22:35	0:50	2	2				1				1		
2	23:25	0:34	1	2			1	1						
3	23:59	0:31	1	1							1		0/1	0/1
Task accomplished - Duration: 1 minute and 55 seconds – 3 episodes														
1	24:50	1:44	3	6	1		3	1		1	1	1		
2	26:34	0:43	1	2		1		1			1			
3	27:17	0:34		1									0/1	0/1
4	27:51	0:25	2					1	1	1		1		
5	28:16	0:38	2	1		1	1							
6	28:54	1:06	2	2			2							
7	30:00	1:40	2	3			1	1	1		1	1		
8	31:40	1:52	3	3			2				1	1		
9	33:32	0:44	2	3		1	3	1						
10	34:16	1:19	2	2			1							
11	35:35	2:16	3	5			1	1	1		1		0/1	0/1
12	37:51	0:34	1	2			1				1			
13	38:25	1:01	2	2	2	1	1	1						
14	39:26	0:56	3	2		1					1			
15	40:22	1:18	2	3								1		
16	41:40	1:45	4	2	1	1	1	2						
17	43:25	0:30	1	1			1	1			1		0/1	0/1
Duration: 19 minutes and 5 seconds – 17 episodes														

Table 2. Segmentation table for Session 1

Session 2

No	Time Start	Duration	Give suggestion/ opinion/ orientation		Ask for suggestion/ opinion/ orientation		Agree		Disagree		Shows solidarity		Shows tension/ tension release	
			U1	U2	U1	U2	U1	U2	U1	U2	U1	U2	U1	U2
1	00:01	0:46	3	1			1	1						
2	00:47	1:11	4	3		1	1							
3	01:58	1:03	3	3		1	1							
4	03:01	0:22	1	1					1			1		
5	03:23	0:33	1	2	1									
6	03:56	0:58	2	2		1					1			
7	04:54	1:49	2	2	1			1			1	1		
8	06:43	0:52	1	1		1		1			1			
9	07:35	1:00	3	1			1	1						
10	08:35	0:51	1	2			1							
11	09:26	0:57	2	2			1	2			1		0/1	0/1
Task accomplished - Duration: 10 minutes and 23 seconds – 11 episodes														
1	10:56	0:17	1	1			1							
2	11:13	1:04	2	2			1	1	1			1		
3	12:17	1:03	1	1	1		1							
Task accomplished - Duration: 2 minutes and 24 seconds – 3 episodes														

Table 3. Segmentation table for Session 2

Session 3

No	Time Start	Duration	Give suggestion/ opinion/ orientation		Ask for suggestion/ opinion/ orientation		Agree		Disagree		Shows solidarity		Shows tension/ tension release	
			U1	U2	U1	U2	U1	U2	U1	U2	U1	U2	U1	U2
1	30:37	0:50	2	2			2		1					
2	31:27	1:03	3	1			1					Observer at 33:25		
3	32:30	0:45	2	1										1/0
4	33:15	1:30	4	3		1	1							
5	34:45	0:37	2	2			1				1			
6	35:22	1:21	2	1	1		1		1		1			
7	36:43	1:01	2	1			1							1/0
8	37:44	0:10	2	1					Observer at 39:20					
9	37:54	1:21	3											
10	39:15	0:40	2	1									0/1	
11	39:55	1:07	1	2					2		1			
12	41:02	0:59	2										0/1	
Task accomplished - Duration: 11 minutes and 24 seconds – 12 episodes														
1	42:05	0:46	2	2				1						
2	42:51	1:08	4	3			2		1					
3	43:59	0:59	3	2			1				1			1/0
4	44:58	0:56	2	1		1								
5	45:54	0:32	1	1		1					1			1/0
6	46:26	0:30	1	1			1							
7	46:56	0:37	2	2	1				1					
8	47:33	0:29	1	1			1				1			
9	48:02	1:32	3	3			1		1		1			
10	49:34	1:29	3		1						Observer at 51:36			
11	51:03	1:01	2	1						1				
12	52:04	0:40	2	1	1									
13	52:44	0:47	1											
14	53:31	0:39	1								1		1/0	
15	54:10	1:20	3	1	1				1				0/1	
16	55:30	0:55	2	1	1					1				
17	56:25	0:32	1										1/0	
18	56:57	0:46		1			1						1/0	1/0
Duration: 14 minutes and 52 seconds – 18 episodes														

Table 4. Segmentation table for Session 3

Session 4

No	Time Start	Duration	Give suggestion/ opinion/ orientation		Ask for suggestion/ opinion/ orientation		Agree		Disagree		Shows solidarity		Shows tension/ tension release	
			U1	U2	U1	U2	U1	U2	U1	U2	U1	U2	U1	U2
1	00:02	0:36	2	2	1		2				1			
2	00:38	0:37	3	2	1	1	1	1						
3	01:15	0:33	1	2			1						0/1	
4	01:48	1:04	3	3	1		1	2					0/1	0/1
5	02:52	0:34	2	1					1		1			
6	03:26	1:22	3	2	1		1				1			
7	04:48	0:32	2	1					1					
8	05:20	1:39	3	2			1				1		0/1	
9	06:59	0:30	1	2			1							
10	07:29	0:55	1	1			1							
11	08:24	0:26	1										0/1	
12	08:50	0:45	1	2			1				1			
13	09:35	0:55	2	1	1			1	1	1	1			
14	10:30	1:12	3	1	1	1	1							
15	11:42	0:22	1	1			1		1					
16	12:04	0:34	2	1	1									
17	12:38	0:36	1	1			1				1			
18	13:22	0:37	2	1			1		1					
19	14:15	1:18	4	2	1		1		1					
20	15:43	1:04		2			1						0/1	0/1
Task accomplished - Duration: 16 minutes and 45 seconds – 20 episodes														
1	16:28	2:17	5	4	1	1	1		1					
2	18:45	1:30	3	2	1		3				2			
3	20:15	0:49	2	3	1		1	1	2					
4	21:04	0:56	1	2			1				1			
5	22:00	1:19	3	3	1		2	1			1			
6	23:19	0:35	1	2					1	1				
7	23:54	0:24	2	1					1	1			0/1	0/1
8	24:18	0:42	2	2			2				1			
9	25:00	0:21		1		1								
10	25:21	1:23	2	1	1		1				1			
11	26:44	2:31	4	3	1		1		1		1		0/1	0/1
12	29:15	2:25	4	2	1		1		1					
Duration: 15 minutes and 12 seconds – 12 episodes														

Table 5. Segmentation table for Session 4

Session 5

No	Time Start	Duration	Give suggestion/ opinion/ orientation		Ask for suggestion/ opinion/ orientation		Agree		Disagree		Shows solidarity		Shows tension/ tension release	
			U1	U2	U1	U2	U1	U2	U1	U2	U1	U2	U1	U2
1	32:00	2:12	3	2	1	1		1					0/1	0/1
2	34:12	0:53	1	3	1		1				1			
3	35:05	2:35	2	2				2	1			1		0/1
4	37:40	0:33		1										
5	38:13	1:20	1	2			1				1		0/1	
6	39:33	2:22	2	3		1	1		2	1			1/1	
7	41:55	0:41	1	2		1		1						
8	42:36	1:05	1	1			1		1	1			1/0	
9	43:41	1:47	2	1								1	0/1	0/1
10	45:28	1:52	1	2	1					1		1		
11	47:20	0:50		1					1					
12	48:10	1:37	1	3			2	1						
13	49:47	1:50	2	2			1			1	1		0/1	
14	51:37	0:19	1	1			1							
15	51:56	1:58	1											
16	53:54	1:08	1	3		1						1		
17	55:02	1:53	2	1			1							
18	56:55	2:47	1	1			1		1	1				
Duration: 27 minutes and 42 seconds – 18 episodes														

Observer at 35:42, 36:50

Observer at 39:26

Observer at 44:31

Table 6. Segmentation table for Session 5

Observer's notes and questionnaire observations for the distributed test

Session	PA observations	
	User 1 - questionnaire	User 2 - questionnaire
Session 6 28.02.02 11:25 – 12:15	• DE forced users to talk more • Time lost by establishing a better communication (technical stuff) • Constrained by the communication technology because of unreliability • Good collaboration process, no dominant participant • Dominance of a participant is minimized • The joking element is more important (to relax both participants)	
	- 4 important to know the other participant - Good collaboration process - Shared control - Restricted by the communication technology - Both environments assured a good collaboration	- Not very important to know the other participant - Very good collaboration process - Collaboration was easy (no problems in explaining ideas and drag the mouse in the same time) - Same conditions for both users (equality) - Slower process compared to CE - You can't transmit emotions in a virtual meeting => CE is better - Audio was the most important mean to communicate
Session 7 1.03.02 16:10 – 17:00	• Users collaborated very well and they talked more ("Talk to me!") • The game was harder to manipulate • The process of collaboration is different in the sense that control of the game is shared now (when User1 has control, User2 assists User1 with suggestions, but when the suggestion is too complicated to be expressed in words User2 asks for control) • The control is much more easily shared after a while – learning element • Body language is replaced by talking, so that the participants are using their intellectual capabilities to a greater extent (even though users still point out with the finger at the screen) • The joking element	
	- More important to know the other participant in DE - Medium collaboration process - Both participants have the same knowledge to the problem => the benefit of the collaboration was average - Positive aspect = There was always some activity towards a solution - Negative aspect = shared control - Restricted by CT because the game was slow???? - Audio and whiteboard = 7 - Better collaboration in CE because of the nature of the problem - The mouse couldn't replace the hand in pointing out	- More important to know the other participant in DE - Good collaboration process - Suggestions from the other participant are very helpful (good team work) - Positive aspect = shared use of the mouse, shared responsibility to move the process forward - Negative aspect = the game was slow, only one person could move the mouse at any given time - Not restricted by CT - Audio = 7, the other three less important - Better collaboration in DE because it forces people to act in turn (more give & take)

<table>
<tr><td rowspan="2"><b>Session 8</b>
5.03.02
10:10 – 10:50</td><td colspan="2">• The game is harder to control
• User1 gives indications and User2 executes (Defined Roles – User1 is taking the “thinker” role and doesn’t try that much to be in control of the mouse)
• User1 is also backing up User2 in decisions
• The user in control of the mouse talks less and imprecisely
• The team communication was much greater than in CE</td></tr>
<tr><td>- Equally important to know the other participant in both CE and DE
- Good collaboration process
- The other participant could see the problem differently and propose alternative ideas
- Positive aspect = shared control
- Negative aspect = slow game, chat couldn’t be used in the same window with the game
- Restricted by the CT because user interface difficulties
- Audio and Chat = 6
- Better collaboration in CE</td><td>- Equally important to know the other participant in both CE and DE
- Good towards medium collaboration process
- Positive aspect = more input from User1
- Negative aspect = communication is hard to maintain
- Not restricted by the CT even though the network was slow???
- Audio = 5, Chat = 4, the other not important
- Both CE and DE assured a good collaboration
- Audio connection was not good</td></tr>
<tr><td rowspan="2"><b>Session 9</b>
5.03.02
11:15 – 12:10</td><td colspan="2">• More talking than in CE
• Language affected communication
• Good collaboration process
• Sharing control was difficult at beginning but became easier after a while (the learning element)
• Users used the finger to point out at screen and in the same time verbalized their thoughts; after a while however the mouse pointer replaced the finger (gestures normally used in face-to-face communication are still used in DE)
• The participant in control of the mouse uses only short sentences
• The focus of attention translates to the computer screen in DE</td></tr>
<tr><td>- Equally important to know the other participant in both CE and DE
- Very good collaboration process
- Interaction within a team opens the way for using other methods to solve the problem
- Positive aspects = equal involvement in finding the solution, very good communication, rewarded each other for shared efforts
- Negative aspects = language but as a negligible barrier
- Not restricted by the CT
- Audio = 7, Video =5
- Better collaboration in DE (the communication in DE was more pronounced and therefore more effective)</td><td>- More important to know the other participant in CE
- Very good collaboration process
- Positive aspects = more talking
- Negative aspects = slow computer system, language
- Not restricted by the CT
- Audio = 7, Chat & Whiteboard = 3
- Better collaboration in DE (in DE you have to keep talking)
- It would be better if both participants could use the mouse at the same time</td></tr>
</table>

<table>
<tr>
<td rowspan="2"><b>Session10</b>
5.03.02
12:30 – 13:20</td>
<td colspan="2">• Not a very good collaboration process
• When a user gets an idea just wants to try it out without explaining it (after they User1 was reminded to communicate his thoughts => a slight improvement)
• When User1 just moved the shapes without any explanation, User2 takes a position: “Can you explain your movements, ?” (User1 feels that he looses time if he starts explaining everything that he’s doing)
• The game was difficult to manipulate
• The use of hand gestures (Co-located type gestures)
• User1 gave more indications rather then suggestions when User2 was in control of the mouse
• When one user is in control of the mouse, the other one gives suggestions and shows solidarity (Defined Roles)
• The control exchange process is fluent and quite dynamic
• The collaboration process improved after a success was registered</td>
</tr>
<tr>
<td>- Equally important to know the other participant in both CE and DE
- Very good collaboration process
- Positive aspects = shared control, shared power, virtual democracy
- Negative aspects = the game was slow, the video image of the other participant
- Not restricted by the CT
Video, Audio, Chat = 7
- Better collaboration in DE (maybe also because of learning)</td>
<td>- More important to know the other participant in CE
- Very good collaboration process
- Positive aspect = communication was more important in DE
- Negative aspect = communication was difficult at the beginning, slow game
- Not restricted by the CT
- Audio = 7, Video = 5
- Better collaboration in DE because the participants are forced to collaborate</td>
</tr>
</table>

Table 7. Observer’s notes and questionnaire observations for the distributed test

Legend:

CE = Co-located Environment
DE = Distributed Environment
CT = Communication Technology

Segmentation tables for the distributed test

Session 6

No	Time Start	Duration	Give suggestion/ opinion/ orientation		Ask for suggestion/ opinion/ orientation		Agree		Disagree		Shows solidarity		Shows tension/ tension release	
			U1	U2	U1	U2	U1	U2	U1	U2	U1	U2	U1	U2
1	00:00	2:06		3	2	1	3				1			
2	02:06	3:12	5	4		1	2	2				2		
3	05:18	1:03		3			2							
4	06:21	3:44	4	5	1	2	2	2			1	2		
5	10:05	2:05	2	2			1	2		1	1	2		
6	12:10	1:54	1	2	1		1	1				1	0/1	
7	14:04	2:31	1	2	1	1		1				1	0/1	
8	16:35	1:35	3	3			2	1			1	1		
9	18:10	2:35	2	1	2		2	2			1		0/1	0/1
10	20:45	1:30	1	2	1								0/1	0/1
11	22:15	1:40	3	3		1	1	1			1	1	0/1	0/1
12	23:55	2:40	2	4		1	2	1						
13	26:35	3:16	2	2			1	2			1	1	0/1	0/1
14	29:51	3:00	3	2	1			2			1	2		
15	32:51	1:08	1	2			1							
16	33:59	5:18	5	4	1	2	3	2	1	1				
17	39:17	4:34	4	4	1			2				2	0/1	0/1
Task accomplished – Duration: 43 minutes and 51 seconds – 17 episodes														

Table 8. Segmentation table for Session 6

Session 7

No	Time Start	Duration	Give suggestion/ opinion/ orientation		Ask for suggestion/ opinion/ orientation		Agree		Disagree		Shows solidarity		Shows tension/ tension release	
			U1	U2	U1	U2	U1	U2	U1	U2	U1	U2	U1	U2
1	45:35	4:54	5	6		2	5	4	1		3	4	0/1	0/1
Task accomplished – Duration: 4 minutes and 54 seconds – 1 episode														
1	50:45	1:55	2	3	1	3	3				1			
2	52:40	1:10	2	2				1				1		
3	53:50	0:44	2	2		1		1						
4	54:34	1:14	3	2		2	1	2				1		
5	55:48	4:54	7	6	1	4	2	2	1		2	3	0/1	0/1
6	60:42	2:48	3	4	1	2	2	2				1		
7	63:30	2:50	4	3		4	3	1	1		2	2		
8	66:20	4:15	5	6	2	3	3	3	1		3	1	0/1	0/1
9	70:35	4:42	4	5	2	1	4	3			2	1	0/1	0/1
Task accomplished – Duration: 24 minutes and 32 seconds – 9 episodes														

Table 9. Segmentation table for Session 7

Session 8

No	Time Start	Duration	Give suggestion/ opinion/ orientation		Ask for suggestion/ opinion/ orientation		Agree		Disagree		Shows solidarity		Shows tension/ tension release	
			U1	**U2**	**U1**	**U2**	**U1**	**U2**	**U1**	**U2**	**U1**	**U2**	**U1**	**U2**
1	00:00	1:50	5	1			1	1						
Task accomplished – Duration: 1 minute and 50 seconds – 1 episode														
1	02:10	1:01	2	2		1	1		1	1				
2	03:14	1:47	2	2			2	1			1			1/1
Task accomplished – Duration: 2 minutes and 48 seconds – 2 episodes														
1	05:33	2:04	4	3			1	2			1			
3	07:37	3:23	6	3			1	2			2	1		
4	11:00	2:30	3	4	1			1	1		1			1/0
5	13:30	1:35	2	2		1				1	2			1/0
Duration: 9 minutes and 32 seconds – 5 episodes														

Table 10. Segmentation table for Session 8

Session 9

No	Time Start	Duration	Give suggestion/ opinion/ orientation		Ask for suggestion/ opinion/ orientation		Agree		Disagree		Shows solidarity		Shows tension/ tension release	
			U1	U2	U1	U2	U1	U2	U1	U2	U1	U2	U1	U2
1	18:41	2:04	2	2			3	2			1	1		
2	20:45	2:13	4	2	1		2	2			1	1		
3	22:58	1:57	3	2	2		2				1			
4	24:55	3:19	4	2	1		3	1	1		4	3	0/1	0/1
Task accomplished – Duration: 9 minutes and 33 seconds – 4 episodes														
1	28:30	1:10	2	3	2		1	1						
2	29:40	1:40	3	3	1		3	1			1	1		
3	31:20	5:35	7	4	2		3	3			4	2	0/1	0/1
4	36:55	1:26	3	1	1							1		
5	38:21	4:53	7	3	3		3				2	1		
6	43:14	0:55	1	1			2	1			2		0/1	0/1
Task accomplished – Duration: 15 minutes and 39 seconds – 6 episodes														

Table 11. Segmentation table for Session 9

Session 10

No	Time Start	Duration	Give suggestion/ opinion/ orientation		Ask for suggestion/ opinion/ orientation		Agree		Disagree		Shows solidarity		Shows tension/ tension release	
			U1	**U2**	**U1**	**U2**	**U1**	**U2**	**U1**	**U2**	**U1**	**U2**	**U1**	**U2**
1	44:30	1:40	3	2		1	2	2						
2	46:10	2:00	2	2				2				1		
3	48:10	0:59	1	2			1	1			1	1		
4	49:09	2:51	4	3			1	3		1	1	1		
5	52:00	2:00	2	3	1		3	1			2	1		
6	54:00	4:00	6	4	1		2	4				2	0/2	
7	58:00	2:24	2	2	1		1	1		1	1			
8	60:24	3:20	4	3		1	2	2				2	0/1	0/1
Task accomplished – Duration: 19 minutes and 14 seconds – 8 episodes														
1	64:00	1:45	4	2			1	3				1		
2	65:45	1:29	3	2		1	1	1			1	1		
3	67:14	0:50	4	1		1	1	1						
4	68:04	3:56	5	2			2	4	1		1	1		
5	72:00	1:04	3	1			1	2				1	0/1	
Task accomplished – Duration: 9 minutes and 4 seconds – 5 episodes														

Table 12. Segmentation table for Session 10

Appendix 4

Extract from the Java documentation of the DFE Workbench software

Class Workbench

```
java.lang.Object
  |
  +--java.awt.Component
        |
        +--java.awt.Container
              |
              +--java.awt.Window
                    |
                    +--java.awt.Frame
                          |
                          +--javax.swing.JFrame
                                |
                                +--dfe.Workbench
```

All Implemented Interfaces:

javax.accessibility.Accessible, java.awt.event.ActionListener, java.util.EventListener, java.awt.image.ImageObserver, java.awt.MenuContainer, javax.swing.RootPaneContainer, java.io.Serializable, javax.swing.WindowConstants

---

public final class **Workbench**
extends javax.swing.JFrame
implements java.awt.event.ActionListener

Workbench is the main class of the DFE Workbench package. After an user is succesfully logged on to the DFE Workbench, this class allows access to operations such as: Open/Modify/Remove an Assembly in IAS or SAM mode, Generate Reports, Database Interface.

See Also:

Serialized Form

Inner classes inherited from class javax.swing.JFrame
javax.swing.JFrame.AccessibleJFrame

Inner classes inherited from class java.awt.Frame
java.awt.Frame.AccessibleAWTFrame

Inner classes inherited from class java.awt.Window
java.awt.Window.AccessibleAWTWindow

Inner classes inherited from class java.awt.Container
java.awt.Container.AccessibleAWTContainer

Inner classes inherited from class java.awt.Component
java.awt.Component.AccessibleAWTComponent

Field Summary

static boolean	**bProE** Boolean field indicating if the current session was started from Pro Engineer (if DFE Workbench is opened from PRO Engineer NEVER use System.exit(0);).
static java.lang.String	**columnsCOMP** String field indicating the names and order of the columns for the COMP table associated with an assembly (used when a new assembly is created).
static java.lang.String	**columnsIAE** String field indicating the names and order of the columns for the IAE table associated with an assembly (used when a new assembly is created).
static java.lang.String	**columnsSAM1** String field indicating the names and order of the columns for the SAM1 table (the table holding all the joints data of an assembly) associated with an assembly (used when a new assembly is created).
static java.lang.String	**columnsSAM2** String field indicating the names and order of the columns for the SAM2 table (the table holding all the obstructions data of an assembly) associated with an assembly (used when a new assembly is created).
static int	**COMPONENTS** Field number indicating that the components table is being updated.
static java.sql.Connection	**conDfeData** Database connection to the data tablespace.
static java.sql.Connection	**conDfeWork** Database connection to the working tablespace.
static int	**EOL** Field number indicating that the eol table is being updated.
static int	**frameCount** Field number that counts the internal frames opened/created.
static java.lang.String	**home**

	String field containing the name of the folder in which the DFE Workbench package is saved.
static int	**margin** Field number indicating the margin allow between two internal frames opened in this frame.
static int	**MATERIAL** Field number indicating that the material table is being updated.
static int	**maxNoCompInChart** Field number indicating the maximum number of components allowed in the bar chart of the environmental impact for an assembly.
static int	**maxNoMaterial** Field number indicating the maximum number of materials to be displayed in the Material Variety panel.
static int	**MEOL** Field number indicating that the meol table is being updated.
static int	**MMATCH** Field number indicating that the mmatch table is being updated.
static int	**MPROCESS** Field number indicating that the mprocess table is being updated.
static int	**PROCESS** Field number indicating that the process table is being updated.
static java.sql.PreparedStatement	**psAllAssembliesByName** Prepared Statement for returning all the assemblies created order alphabetically.
static java.lang.String	**sOracleURL** String field containing the connection string to the Oracle database.
static int	**TRANSPORT** Field number indicating that the transport table is being updated.
static int	**USE** Field number indicating that the use table is being updated.

Fields inherited from class javax.swing.JFrame
accessibleContext, EXIT_ON_CLOSE, rootPane, rootPaneCheckingEnabled

Fields inherited from class java.awt.Frame
CROSSHAIR_CURSOR, DEFAULT_CURSOR, E_RESIZE_CURSOR, HAND_CURSOR,

ICONIFIED, MOVE_CURSOR, N_RESIZE_CURSOR, NE_RESIZE_CURSOR, NORMAL, NW_RESIZE_CURSOR, S_RESIZE_CURSOR, SE_RESIZE_CURSOR, SW_RESIZE_CURSOR, TEXT_CURSOR, W_RESIZE_CURSOR, WAIT_CURSOR

Fields inherited from class java.awt.Component

BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT

Fields inherited from interface javax.swing.WindowConstants

DISPOSE_ON_CLOSE, DO_NOTHING_ON_CLOSE, HIDE_ON_CLOSE

Fields inherited from interface java.awt.image.ImageObserver

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary

Workbench()

Constructor for Workbench.

Workbench(java.lang.String user)

Constructor for Workbench called for a user name.

Method Summary

void	**actionPerformed**(java.awt.event.ActionEvent e) Method invoked when an action occurs.
void	**closeAllProjects**() Closes all opened frames (assemblies/subassemblies).
void	**closeDB**() Closes all the connections created to the database.
void	**closeProject**() Closes the current frame (assembly/subassembly).
void	**createNewVersion**(java.lang.String projectName) Creates a new version of the specified project.
static void	**createProjectTables**(java.lang.String name) Creates the 4 necessary tables for a new project (analysis): - IAE - "nameIAE" - SAM1 - "nameSAM1" - SAM2 - "nameSAM2" - COMP - "nameCOMP"
static java.lang.String	**getCOMPTable**(java.lang.String s)

	Returns the name of the COMP table coresponding to a String value.
javax.swing.JDesktopPane	**getDesktopPane()** Returns the desktop pane that holds all the internal frames.
java.util.Vector	**getFramesOpened()** Returns the vector of all the internal frames opened.
static java.lang.String	**getIAETable**(java.lang.String s) Returns the name of the IAE table corresponding to a String value.
java.util.Hashtable	**getProjectMethod**() Returns the hashtable of all the internal frames opened and the mode in which they were opened (IAS or SAM).
java.util.Vector	**getProjectsFromDb**() Returns a vector of all projects created using a prepared statement.
static java.lang.String	**getSAM1Table**(java.lang.String s) Returns the name of the SAM1 table corresponding to a String value.
static java.lang.String	**getSAM2Table**(java.lang.String s) Returns the name of the SAM2 table corresponding to a String value.
static java.lang.String	**getTMPTable**(java.lang.String s) Returns the name of a temporary table corresponding to a String value.
java.lang.String	**getUser**() Returns the name of the user logged in to the DFE Workbench.
static void	**initDB**() Performs all the database initialisation necessary (establishes the connection to both working and data tablespaces).
static boolean	**isStringAllowed**(java.lang.String s) Returns true if the String given is allowed to be used as the name of an assembly/subassembly.
static void	**loadOracleDriver()** Register the JDBC driver.
static void	**main**(java.lang.String[] arg) Main function of DFE Workbench.
void	**newProject**() Creates a new assembly/subassembly (referred to as a project).

void	**openProjectIAE**() Opens a project (assembly/subassembly) selected from a combo box in IAS.
void	**openProjectIAE**(java.lang.String projectName) Opens the specified project (assembly/subassembly) in IAS.
void	**openProjectSAM**() Opens a project (assembly/subassembly) selected from a combo box in SAM.
void	**openProjectSAM**(java.lang.String projectName) Opens the specified project (assembly/subassembly) in SAM.
void	**removeProject**(int option) Removes the project (assembbly/subassembly) selected from a combo box from the database.
void	**removeProject**(java.lang.String projectName) Removes the specified project without deleting the contained subassemblies.
void	**removeProject**(java.util.Vector v) Removes all projects (assemblies/subassemblies) from the specified vector.
void	**removeProjectWithSubs**(java.lang.String projectName) Removes the specified project, as well as all contained subassemblies.
java.lang.String	**renameProject**(java.lang.String oldName) Renames the specified assembly/subassembly.
void	**saveProjectAs**(java.lang.String projectName, boolean bVersion) Saves the specified project (assembly/subassembly) with another name.
void	**setDesktopPane**(javax.swing.JDesktopPane desktop) Sets the desktop pane that holds all the internal frames.
void	**setSelectedProject**(java.lang.String name) Sets the active assembly/subassembly to the one indicated.
void	**setStatus**(java.lang.String status) Sets the status label displayed in the main window of the DFE Workbench.
void	**setUser**(java.lang.String user) Sets the name of the user logged in to the param value.
void	**showAll**() Displays the List All window.

void	**showPasswordDialog**(int db) Displays the password window required for access to the database interface
void	**showReportConsole**() Displays the Report Console window.
void	**showReportConsole**(java.lang.String projectName) Displays the Report Console window for a specific assembly/subassembly.
void	**showSFastener**() Displays the Fastener Search window.
void	**showSMaterial**() Displays the Material Search window.
void	**showSProcess**() Displays the Process Search window.
void	**showSTransport**() Displays the Transport Search window.

Methods inherited from class javax.swing.JFrame

addImpl, createRootPane, frameInit, getAccessibleContext, getContentPane, getDefaultCloseOperation, getGlassPane, getJMenuBar, getLayeredPane, getRootPane, isRootPaneCheckingEnabled, paramString, processKeyEvent, processWindowEvent, remove, setContentPane, setDefaultCloseOperation, setGlassPane, setJMenuBar, setLayeredPane, setLayout, setRootPane, setRootPaneCheckingEnabled, update

Methods inherited from class java.awt.Frame

addNotify, finalize, getCursorType, getFrames, getIconImage, getMenuBar, getState, getTitle, isResizable, remove, removeNotify, setCursor, setIconImage, setMenuBar, setResizable, setState, setTitle

Methods inherited from class java.awt.Window

addWindowListener, applyResourceBundle, applyResourceBundle, dispose, getFocusOwner, getGraphicsConfiguration, getInputContext, getListeners, getLocale, getOwnedWindows, getOwner, getToolkit, getWarningString, hide, isShowing, pack, postEvent, processEvent, removeWindowListener, setCursor, show, toBack, toFront

Methods inherited from class java.awt.Container

add, add, add, add, add, addContainerListener, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getInsets, getLayout, getMaximumSize, getMinimumSize, getPreferredSize, insets, invalidate, isAncestorOf, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, processContainerEvent, remove, removeAll, removeContainerListener, setFont, validate, validateTree

Methods inherited from class java.awt.Component

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addPropertyChangeListener, addPropertyChangeListener, bounds, checkImage, checkImage, coalesceEvents, contains, contains, createImage, createImage, disable, disableEvents, dispatchEvent, enable, enable, enableEvents, enableInputMethods, firePropertyChange, getBackground, getBounds, getBounds, getColorModel, getComponentOrientation, getCursor, getDropTarget, getFont, getFontMetrics, getForeground, getGraphics, getHeight, getInputMethodRequests, getLocation, getLocation, getLocationOnScreen, getName, getParent, getPeer, getSize, getSize, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, imageUpdate, inside, isDisplayable, isDoubleBuffered, isEnabled, isFocusTraversable, isLightweight, isOpaque, isValid, isVisible, keyDown, keyUp, list, list, list, location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll, prepareImage, prepareImage, printAll, processComponentEvent, processFocusEvent, processHierarchyBoundsEvent, processHierarchyEvent, processInputMethodEvent, processMouseEvent, processMouseMotionEvent, removeComponentListener, removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, repaint, requestFocus, reshape, resize, resize, setBackground, setBounds, setBounds, setComponentOrientation, setDropTarget, setEnabled, setForeground, setLocale, setLocation, setLocation, setName, setSize, setSize, setVisible, show, size, toString, transferFocus

Methods inherited from class java.lang.Object

clone, equals, getClass, hashCode, notify, notifyAll, wait, wait, wait

Methods inherited from interface java.awt.MenuContainer

getFont, postEvent

Field Detail

MATERIAL

public static final int MATERIAL

Field number indicating that the material table is being updated.

PROCESS

public static final int PROCESS

Field number indicating that the process table is being updated.

EOL

public static final int EOL

Field number indicating that the eol table is being updated.

USE

public static final int USE

Field number indicating that the use table is being updated.

TRANSPORT

public static final int TRANSPORT

Field number indicating that the transport table is being updated.

MPROCESS

public static final int MPROCESS

Field number indicating that the mprocess table is being updated. Mprocess is the table that holds data indicating the processes associated with each material in the material table.

MEOL

public static final int MEOL

Field number indicating that the meol table is being updated. Meol is the table that holds data indicating the EOLs associated with each EOL in the eol table.

MMATCH

public static final int MMATCH

Field number indicating that the mmatch table is being updated. Mmatch is the table that holds data indicating the materials that have similar properties.

COMPONENTS

public static final int COMPONENTS

Field number indicating that the components table is being updated. Components is the table that holds environmental information on those components for which the material and process are unknown.

maxNoCompInChart

public static final int maxNoCompInChart

Field number indicating the maximum number of components allowed in the bar chart of the environmental impact for an assembly.

margin

public static final int margin

Field number indicating the margin allow between two internal frames opened in this frame.

maxNoMaterial

public static final int maxNoMaterial

Field number indicating the maximum number of materials to be displayed in the Material Variety panel.

columnsIAE

public static final java.lang.String columnsIAE

String field indicating the names and order of the columns for the IAE table associated with an assembly (used when a new assembly is created).

columnsCOMP

public static final java.lang.String columnsCOMP

String field indicating the names and order of the columns for the COMP table associated with an assembly (used when a new assembly is created).

---

columnsSAM1

public static final java.lang.String columnsSAM1

String field indicating the names and order of the columns for the SAM1 table (the table holding all the joints data of an assembly) associated with an assembly (used when a new assembly is created).

---

columnsSAM2

public static final java.lang.String columnsSAM2

String field indicating the names and order of the columns for the SAM2 table (the table holding all the obstructions data of an assembly) associated with an assembly (used when a new assembly is created).

---

bProE

public static boolean bProE

Boolean field indicating if the current session was started from Pro Engineer (if DFE Workbench is opened from PRO Engineer NEVER use System.exit(0);).

---

frameCount

public static int frameCount

Field number that counts the internal frames opened/created.

---

sOracleURL

public static final java.lang.String sOracleURL

String field containing the connection string to the Oracle database.

---

home

public static final java.lang.String home

String field containing the name of the folder in which the DFE Workbench package is saved. This value is required for generating a new report (at the moment replaced with class.getResource()).

---

conDfeWork

public static java.sql.Connection conDfeWork

Database connection to the working tablespace.

---

conDfeData

public static java.sql.Connection conDfeData

Database connection to the data tablespace.

---

psAllAssembliesByName

public static java.sql.PreparedStatement psAllAssembliesByName

Prepared Statement for returning all the assemblies created order alphabetically.

Constructor Detail

Workbench

public Workbench()

Constructor for Workbench. The frame is initialised. The frames opened vector and hashtable are initialised. The Oracle connection is established.

Workbench

public Workbench(java.lang.String user)

Constructor for Workbench called for a user name.

Parameters:

user - A String value representing the name of the user logged in.

Method Detail

getUser

public java.lang.String getUser()

Returns the name of the user logged in to the DFE Workbench.

Returns:

A String value representing the name of the user logged in.

---

setUser

public void setUser(java.lang.String user)

Sets the name of the user logged in to the param value.

Parameters:

user - A String value representing the name of the user logged in.

---

getFramesOpened

public java.util.Vector getFramesOpened()

Returns the vector of all the internal frames opened.

Returns:

A vector.

---

getProjectMethod

public java.util.Hashtable getProjectMethod()

Returns the hashtable of all the internal frames opened and the mode in which they were opened (IAS or SAM).

Returns: A hashtable

---

getDesktopPane

public javax.swing.JDesktopPane getDesktopPane()

Returns the desktop pane that holds all the internal frames.

Returns:

A desktop pane.

---

setDesktopPane

public void setDesktopPane(javax.swing.JDesktopPane desktop)

Sets the desktop pane that holds all the internal frames.

Parameters:

desktop - A new desktop pane value.

---

loadOracleDriver

public static void loadOracleDriver()

Register the JDBC driver. This needs to be done only once in a Java application.

Throws:

java.sql.SQLException - if a database related exception has occurred.

---

initDB

public static void initDB()

Performs all the database initialisation necessary (establishes the connection to both working and data tablespaces).

---

closeDB

public void closeDB()

Closes all the connections created to the database.

---

setStatus

public void setStatus(java.lang.String status)

Sets the status label displayed in the main window of the DFE Workbench.

Parameters:

status - A String value representing the status value (the name of the assembly/subassembly activated and the mode in which it was opened)

---

actionPerformed

public void actionPerformed(java.awt.event.ActionEvent e)

Method invoked when an action occurs. Implements the actionPerformed method of the ActionListener interface.

Specified by:

actionPerformed in interface java.awt.event.ActionListener

---

showReportConsole

public void showReportConsole()

Displays the Report Console window.

---

showSMaterial

public void showSMaterial()

Displays the Material Search window.

---

showSProcess

public void showSProcess()

Displays the Process Search window.

---

showSTransport

public void showSTransport()

Displays the Transport Search window.

---

showSFastener

public void showSFastener()

Displays the Fastener Search window.

---

showReportConsole

public void showReportConsole(java.lang.String projectName)

Displays the Report Console window for a specific assembly/subassembly.

Parameters:

projectName - A String representing the name of the assembly/subassembly for which reports are to be generated.

---

showAll

public void showAll()

Displays the List All window. Access to all assemblies/subassemblies created and different actions is facilitated.

---

showPasswordDialog

public void showPasswordDialog(int db)

Displays the password window required for access to the database interface

Parameters:

db - An integer value indicating the table for which direct access was required (material, process, transport, eol, etc.).

---

newProject

public void newProject()

Creates a new assembly/subassembly (referred to as a project). A project can contain a number of components (Subassembly). CHANGE - 02.10.2001 - a project contains now components but also other subassemblies. It can be viewed as an assembly, but it can also be part of another assembly and then it is a subassembly. When a new project is created the class PROJECT is called with the name of the new project - all the necessary tables will be created there.

---

getProjectsFromDb

public java.util.Vector getProjectsFromDb()

Returns a vector of all projects created using a prepared statement.

Returns:

A vector.

---

setSelectedProject

public void setSelectedProject(java.lang.String name)

Sets the active assembly/subassembly to the one indicated.

Parameters:

name - A String representing the name of the assembly/subassembly to be activated.

---

openProjectIAE

public void openProjectIAE()

Opens a project (assembly/subassembly) selected from a combo box in IAS.

---

openProjectIAE

public void openProjectIAE(java.lang.String projectName)

Opens the specified project (assembly/subassembly) in IAS.

Parameters:

projectName - A String representing the name of the assembly/subassembly to be opened in IAS mode.

---

openProjectSAM

public void openProjectSAM()

Opens a project (assembly/subassembly) selected from a combo box in SAM.

---

openProjectSAM

public void openProjectSAM(java.lang.String projectName)

Opens the specified project (assembly/subassembly) in SAM.

Parameters:

projectName - A String representing the name of the assembly/subassembly to be opened in SAM mode.

---

saveProjectAs

```
public void saveProjectAs(java.lang.String projectName,
                          boolean bVersion)
```

Saves the specified project (assembly/subassembly) with another name.

Parameters:

projectName - The name of the assembly/subassembly to be saved as.

bVersion - If this param is set to true a new version of the given assembly/subassembly will be created.

---

renameProject

public java.lang.String renameProject(java.lang.String oldName)

Renames the specified assembly/subassembly.

Parameters:

oldName - The name of the assembly/subassembly to be renamed.

---

createNewVersion

public void createNewVersion(java.lang.String projectName)

Creates a new version of the specified project. The current version number is the value of VERSION field in ASSEMBLIES table. The new version of the project will be named projectName + "-" + (version+1). The version of the new version will be 0.

Parameters:

projectName - The name of the assembly/subassembly for which a new version is created.

---

closeProject

public void closeProject()

Closes the current frame (assembly/subassembly).

---

closeAllProjects

public void closeAllProjects()

Closes all opened frames (assemblies/subassemblies).

---

removeProject

public void removeProject(int option)

Removes the project (assembly/subassembly) selected from a combo box from the database.

Parameters:

option - If option is 0 (zero) the assembly and its components but without the contained subassemblies is removed. If option is 1 (one) the assembly and its entire content (components and subassemblies) is removed.

---

removeProjectWithSubs

public void removeProjectWithSubs(java.lang.String projectName)

Removes the specified project, as well as all contained subassemblies.

Parameters:

projectName - The name of the assembly/subassembly to be removed.

---

removeProject

public void removeProject(java.lang.String projectName)

Removes the specified project without deleting the contained subassemblies.

Parameters:

projectName - The name of the assembly/subassembly to be removed.

---

removeProject

public void removeProject(java.util.Vector v)

Removes all projects (assemblies/subassemblies) from the specified vector.

Parameters:

v - The vector containing the names of the assemblies/subassemblies to be removed.

---

isStringAllowed

public static boolean isStringAllowed(java.lang.String s)

Returns true if the String given is allowed to be used as the name of an assembly/subassembly. A String value is considered to be not allowed if it contains one of the following characters: \/:*?"<>|-, where - is not allowed because it's used in versioning.

Parameters:

s - A String value.

Returns:

true if the String is allowed and false otherwise.

---

createProjectTables

public static void createProjectTables(java.lang.String name)

Creates the 4 necesary tables for a new project (analysis): - IAE - "nameIAE" - SAM1 - "nameSAM1" - SAM2 - "nameSAM2" - COMP - "nameCOMP"

Parameters:

name - A String representing the name of a new assembly/subassembly to be created.

---

getIAETable

public static java.lang.String getIAETable(java.lang.String s)

Returns the name of the IAE table coresponding to a String value.

Parameters:

s - A String representing the name of a new assembly/subassembly.

Returns:

A String representing the name of the IAE table corresponding to the given assembly/subassembly.

---

getCOMPTable

public static java.lang.String getCOMPTable(java.lang.String s)

Returns the name of the COMP table coresponding to a String value.

Parameters:

s - A String representing the name of a new assembly/subassembly.

Returns:

A String representing the name of the COMP table corresponding to the given assembly/subassembly.

---

getSAM1Table

public static java.lang.String getSAM1Table(java.lang.String s)

Returns the name of the SAM1 table corresponding to a String value.

Parameters:

s - A String representing the name of a new assembly/subassembly.

Returns:

A String representing the name of the SAM1 table corresponding to the given assembly/subassembly.

---

getSAM2Table

public static java.lang.String getSAM2Table(java.lang.String s)

Returns the name of the SAM2 table corresponding to a String value.

Parameters:

s - A String representing the name of a new assembly/subassembly.

Returns:

A String representing the name of the SAM2 table corresponding to the given assembly/subassembly.

---

getTMPTable

public static java.lang.String getTMPTable(java.lang.String s)

Returns the name of a temporary table corresponding to a String value. This table is used for reportiong and is removed once it's not needed.

Parameters:

s - A String representing the name of an assembly/subassembly.

Returns:

A String representing the name of the TMP table corresponding to the given assembly/subassembly.

---

main

public static void main(java.lang.String[] arg)

The main function of the DFE Workbench. The DFE Workbench window is called.