Department of Computer Science and Information Systems

An Investigation into the Learnability of Object-Oriented CASE Tools for Computing Education

Brenda Scholtz

Supervisor: Prof Janet Wesson

December 2007

Submitted in fulfilment of the requirements for the Degree of Magister Scientiae
in the Faculty of Science
at the Nelson Mandela Metropolitan University

Acknowledgements

I would like to thank Prof. Janet Wesson for her invaluable guidance, supervision and advice over the duration of this dissertation. I would also like to thank all the staff in the Department of Computer Science and Information Systems for allowing me to conduct this research and for providing moral and technical support. Special mention must be made to Dr Lynette Barnard and Dr Charmain Cilliers who also provided me with invaluable assistance.

Last but not least thanks must be made to my darling husband Blayne for all his support and encouragement, without which I would not have been able to complete this project so successfully.

Table of contents **Pg**

Summary

The use of Computer Aided Software Engineering (CASE) tools for teaching object-oriented systems analysis and design (OOSAD) has many potential benefits, but there are also several problems associated with the usage of these tools. A large portion of these problems relate to the usability and learnability of these tools. Learnability is one of the most important attributes of usability and refers to the capability of the system to enable the user to learn its application.

The main research question that this study aims to address is "*How can the learnability of OO CASE tools for computing education in South Africa be evaluated?"*. In order to answer this question several frameworks for evaluating CASE tool usability and learnability were investigated. One of these frameworks, as proposed by Senapathi, was selected as being the most appropriate for evaluating CASE tool learnability for computing education. This framework maintains that the learnability of a CASE tool is dependent on context of use factors such as the tool used, as well as user characteristics such as gender. The primary aim of this research was thus to validate Senapathi's framework for CASE tool learnability in a South African context. A secondary aim of the research was to extend the implementation of the framework in order to enable the comparison of two CASE tools and to support the inclusion of other user characteristics.

An experiment was performed at the Nelson Mandela Metropolitan University (NMMU) in 2006. The participants recruited for this experiment were second year computing students at NMMU. During this experiment, the learnability of two OO CASE tools, namely IBM's Rational Software Modeller and Microsoft's Visio, was evaluated and compared. The quantitative and qualitative results supported Senapathi's results and showed that her framework could be used to evaluate CASE tool learnability and could be adapted to evaluate two CASE tools. The results also showed that the majority of the participants rated the learnability of Microsoft Visio higher for both tasks and that the main reasons participants preferred Visio was due to its simplicity, familiarity and recoverability.

The participants also found the use of CASE tools in computing education to be beneficial and that it helped them to understand the underlying concepts of OO and UML. Several problems with the learnability of both CASE tools were identified, and recommendations were made that could provide valuable information to designers of CASE tools, as well as lead to improved rates of learning of these tools.

Keywords: Learnability, CASE tools, context of use, contextual factors, usability evaluation

List of figures

List of tables

Chapter 1 Introduction

1.1 Background

System developers need to understand objects and object-based tools, since these are regarded as the future of Information Technology (IT) (Brewer and Lorenz 2003). Object modelling is regarded as an essential component of system development and also serves as a form of documentation. In this way the understanding of a system can be passed on from the original developers. This is important since software often lives longer than the original developers (Fenstermacher 2004).

The Unified Modelling Language (UML) has been made an OMG (Object Management Group) standard and is generally accepted as the predominant object modelling language by the industry (Senapathi 2005). UML helps analysts specify, visualise and document models of software systems, including their structure and design, in such a way that all the requirements of the system can be met (Brewer *et al.* 2003). Another reason for creating good-quality UML models (especially in the requirements and analysis phases) is that these can greatly improve the chances of success of the project (Kemerer 1992; Bolloju and Leung 2006).

Higher education institutions (HEIs) need to teach students object-oriented systems analysis and design (OOSAD) (Brewer *et al.* 2003) using UML notation (Senapathi 2005). Students need to have a practical hands-on approach to learning about UML diagrams by actually drawing these diagrams (Burton and Bruhn 2004).

Computer-Aided Software Engineering (CASE) tools are software programs that automate or support the drawing and analysis of system models and provide for the translation of these models into application programs (Whitten 2001). The importance of a practical hands-on approach to learning OOSAD and the use of a CASE tool for the effective teaching of UML are well recognised (Booch, Rambaugh and Jacobson 1999; Douglas and Hardgrave 2000; Burton *et al.* 2004; Tabrizi, Collins, Ozan and Li 2004)

IT educators must continually review their teaching and learning styles and methodologies in order to respond to developments in technology and to meet changing demands from society and industry (Fowler, Allen, Armarego and Mackenzie 2000).

These educators should provide support in teaching tools and techniques which are popular in the workplace and put the students at the forefront of new and leading-edge technologies (Mynatt and Leventhal 1990; Linder, Steele and Dorling 1994; Costain 1998; Senapathi 2005).

The complexity of UML means that it is very difficult for students to learn to develop UML diagrams (Frosch-Wilke 2003). Developing good-quality conceptual models is regarded as a challenge for many analysts, since although UML is widely used, its usability is not highly rated (Bolloju *et al.* 2006). In addition, the problem of choosing the right CASE tool in the OO development process has presented a serious obstacle to OO practitioners (Juric and Kuljis 1999). Most commercial CASE tools have a long learning curve and do not cater well to student requirements (Jankowski 1995). The adoption and usage of CASE tools have been hindered by many problems; a large portion of these problems relate to the usability of the tools (Lending and Chervany 1998; Fowler *et al.* 2000; Post and Kagan 2000).

Usability has been defined by many authors, but the following definitions show a marked similarity:

- *"The measure of the quality of the user experience when interacting with something – whether a web site, a traditional software application, or any other device the user can operate in some way or another" (Nielsen 1993).*
- *"The extent to which a product can be used by specified users to achieve specified goals in a specified context of use with effectiveness, efficiency and satisfaction" (ISO 1997).*

A software system's usability can be determined by the user's perception of the quality of a system. This quality is based on the user's ease of use, ease of learning and relearning, the system's intuitiveness for the user, as well as the user's perceived usefulness of the system (Barnum 2002). Learnability is one of the most important and fundamental attributes of usability *"since most systems need to be easy to learn, and since the first experience most people have with a system is that of learning to use it"* (Nielsen 1993).

1.2 Relevance of research

Teaching OO CASE tools can assist computing students with comprehending and learning OOSAD and UML. However, there are many problems with implementing and using an OO CASE tool for computing education. The majority of problems that students encounter with CASE tool usage relate to the usability of the interface and learning to use the tool.

In order to select a CASE tool that is easy to use and learn, guidance is needed in evaluating the usability and the learnability of these systems. There are many guidelines available in the literature relating to UML and CASE tools, particularly with regard to the usability of CASE tools (Jarzabek and Huang 1998; Booch *et al.* 1999; Johnson and Wilkinson 2003). Insufficient research exists, however, on the evaluation of the usability of CASE tools. Existing studies performed on CASE tool evaluations focus mostly on the utility of the tools (Phillips, Mehandjiska, Griffin, Choi and Page 1998). Those that have been conducted on the usability of CASE tools have been performed in commercial environments and may therefore not be suitable for computing education.

Knowledge of the factors that favourably influence the rate of learning can be obtained by performing CASE tool evaluations. This can lead to improved approaches to teaching CASE tools and, hence, to the uptake of these tools within industry (Fowler *et al.* 2000). Results of studies performed on students us ing a CASE tool showed that usability may be affected by user characteristics like gender, computer experience and learning styles (Senapathi 2005).

Studies undertaken by Senapathi address some of the gaps in CASE tool usability research and have resulted in the design of a framework for evaluating the learnability of CASE tools (Senapathi 2005). This framework was tested by Senapathi at the University of Auckland using undergraduate students to evaluate the learnability of IBM's Rational Rose CASE tool. Senapathi's framework has not been validated and applied to other educational environments and the study was limited to only one CASE tool. Further research is thus required to validate Senapathi's framework in a South African educational context and to expand it to allow for the comparison of the learnability of two different CASE tools.

1.3 Problem Statement

The main problem that this study aims to address is the lack of guidance for evaluating and comparing the learnability of OO CASE tools for use in computing education in South Africa. A possible partial solution to this problem is a framework proposed by Senapathi for evaluating the learnability of CASE tools (Senapathi 2005). The primary goal of this research is therefore to validate and extend Senapathi's framework in order to evaluate and compare two CASE tools in a South African context.

Certain recommendations relating to the framework will be made which could be applied to computing education at HEIs in South Africa. A secondary goal is to determine the relationship between certain factors and the learnability of CASE tools. These results will then be analysed in order to determine the types of problems encountered by the students during the evaluation of the CASE tools, as well as the factors which affect the learnability of these tools. This could provide valuable information to designers of CASE tools, as well as lead to improved rates of learning of these tools. It could also assist lecturers with selecting an OO CASE tool that is easier for the students to learn to use.

1.3.1 Research questions

The primary research question to be answered in this study is therefore as follows, *"How can the learnability of OO CASE tools for computing education in South Africa be evaluated?"*.

Several subsidiary research questions were then identified from the primary research question. These questions are listed in Table 1-1 together with the research methods used to answer the different questions. The chapter in which each research question is addressed is also included in this table.

No	Subsidiary Research Questions	Type of Research	Chapter
1.	What guidelines exist for selecting and using CASE tools?	Literature Review	2
2.	What principles and frameworks exist for evaluating the usability and learnability of CASE tools?	Literature Review	3
3.	How can the learnability of an OO CASE tool be evaluated at NMMU?	Experimental Design	4
4.	What learnability problems occur with OO CASE tools for computing education at NMMU?	Research Results	5
5.	Is there a relationship between factors (such as user characteristics and tool complexity) and CASE tool learnability at NMMU?	Analysis of Results	6
6.	How should the selected framework for the evaluation of OO CASE tool learnability be adapted for computing education at HEIs in South Africa?	Recommendations and Conclusions	7
7.	What are the future research possibilities of this research?	Conclusions	7

Table 1-1 Research questions and methods used to answer these questions

1.3.2 Contribution of this research

The main difference between this study and previous research is that it will focus specifically on frameworks for CASE tool learnability in computing education. It will also include the comparison of the learnability of two CASE tools of different complexity, whereas previous studies have only evaluated the learnability of one CASE tool.

The research will be applied to computing education at an undergraduate level at NMMU, since no documented research of similar studies on frameworks for CASE tool learnability at HEIs in South Africa could be found.

1.3.3 Research scope

The scope of this research will be limited to OO CASE tools which cater primarily for UML diagrams produced during the requirements analysis phase. Only those tools which have the necessary features for drawing the UML diagrams taught by OOSAD modules at NMMU will be considered. The main focus of the study will be on evaluating the learnability of CASE tools used for computing education at an undergraduate level at HEIs in South Africa.

Utility of CASE tools will not form part of this study, except where it is relevant as background or where it affects either usability or learnability.

1.3.4 Research goals

A number of research goals were identified as contributing towards answering the research questions as outlined in Section 1.3.1. These are briefly listed here.

- Identify guidelines for selecting and using OO CASE tools for computing education;
- Investigate principles, standards and frameworks relating to the evaluation of usability and learnability of software systems (especially CASE tools) in computing education;
- Review existing frameworks for evaluating the usability and learnability of CASE tools in order to select a suitable framework;
- Evaluate two selected CASE tools at NMMU using the selected framework; and
- Make recommendations regarding the CASE tool learnability evaluation framework so that it can be applied to computing education at HEIs in South Africa.

1.4 Outline of dissertation

Chapter 1 outlines the context of the research and includes a definition of the problem to be solved as well as the methodology followed. The scope of the research is also defined. The background of the research is discussed in Chapter 2, which focuses on the literature review that was done. An overview is given of UML and CASE tools and the benefits, problems and guidelines associated with their usage in both commercial environments and computing education.

Several usability definitions and classifications are discussed and compared in Chapter 3. This chapter also identifies usability standards as well as attributes which should be used when evaluating the usability and learnability of CASE tools. It also includes a discussion of principles to apply when evaluating the usability and learnability of a system. Principles which apply specifically to CASE tools are then investigated. Several frameworks for evaluating the usability and learnability of CASE tools are examined and their shortcomings identified.

Chapter 4 discusses the research design for the experiment conducted at NMMU to evaluate the selected framework for CASE tool learnability. This includes an explanation of the process of participant and task selection for this experiment. A description of the test tasks and the research instruments are also included.

The research results are discussed in Chapter 5 with the analysis of the results given in Chapter 6. Recommendations regarding existing frameworks for the evaluation of the learnability of CASE tools in computing education are also made in Chapter 6. Chapter 7 concludes the dissertation with a summary of achievements, problems encountered and the identification of several areas for future research.

Chapter 2 CASE tools

2.1 Introduction

This chapter attempts to answer the first research question in Table 1.1, namely *"What guidelines exist for selecting and using CASE tools?"*. It provides an overview of CASE tools and discusses and compares various classifications of CASE tools. It also addresses some of the benefits and problems of using CASE tools both in commercial environments and in computing education. Several guidelines for the use and selection of CASE tools, relating to both the learning environment and the user interaction with the CASE tool, are also provided.

2.2 An overview of CASE tools

One definition of CASE tools was provided in Section 1.1. A CASE tool can also be defined as *'a software product that can assist software engineers by providing automatic support for one or more software lifecycle activities' (ISO 14102).* CASE technology provides software process support by automating some process activities and by providing information about the software system which is being developed (Sommerville 2001). Examples of activities which can be automated using a CASE tool include:

a) The development of graphical system models as part of the requirements specification or the system design;
b) Understanding a design using a data dictionary which holds information about the entities and relations in a design;
c) The automated translation of programs from an older version of a programming language (such as COBOL) to a more recent version;
d) Program debugging through the provision of information about an executing program; and
e) The generation of user interfaces from a graphical interface description which is created interactively by the user.

This study will be limited to CASE tools which automate the activities described in (a) to (c) above, since these are the primary activities taught in OOSAD modules at NMMU.

2.3. CASE classifications

Computer Aided Software Engineering (CASE) is the name given to software that is used to support software process activities such as requirements engineering, design, program development and testing. CASE tools can include design editors, data dictionaries, compilers, debuggers and system-building tools (Jarzabek *et al.* 1998; Sommerville 2001). Some CASE tools address specific phases of the system development lifecycle (SDLC) without supporting others (Mannino 2001). The five phases of the SDLC include systems analysis, conceptual design, physical design, implementation and conversion, and operations and maintenance. An upper-CASE tool provides support for the first few stages in the SDLC, while a lower-CASE tool provides support for the later stages in the life cycle such as physical design and implementation. Integrated CASE tools support both the early and later stages (Lending *et al.* 1998). Recent terminology refers to upper-CASE tools as merely CASE, and lower-CASE tools as Application Development Environment (ADE) or Integrated Development Environment (IDE) tools (Whitten 2001). This research will focus on upper-CASE tools, since computing students at NMMU currently only use CASE tools in the early stages of the SDLC.

There are various ways in which to classify CASE tools. Three main methods of CASE tool classification include the following (Sommerville 2001):

- By perspectives;
- By methodology support; and
- By support offered for the software process.

This section will discuss each of these methods in more detail.

2.3.1 Classification by perspectives

CASE classifications can help us understand the different types of CASE tools and their role in supporting software process activities. There are various ways of classifying CASE tools, each of which gives us a different perspective on these tools. The three primary perspectives of CASE tools are (Sommerville 2001):

- A functional perspective;
- A process perspective; and
- An integration perspective.

2.3.1.1 A functional perspective

A functional perspective is used when CASE tools are classified according to their function. Table 2-1 contains a classification of CASE tools according to function and gives examples of each tool. It does not represent a complete list of CASE tools. Specialist tools, such as tools to support reuse, are not included.

Tool type	**Examples**
Planning tools	PERT tools, estimation tools, spreadsheets
Editing tools	*Text editors, diagram editors, word processors*
Change management tools	*Requirements traceability tools, change control systems*
Configuration management tools	*Version management systems, system building tools*
Method support tools	*Design editors, data dictionaries, code generators*
Prototyping tools	Very high-level languages, user interface generators
Language-processing tools	Compilers, interpreters
Program analysis tools	Cross-reference generators, static analysers, dynamic analysers
Testing tools	Test data generators, file comparators
Debugging tools	Interactive debugging systems
Documentation tools	Page-layout programs, image editors
Re-engineering tools	Cross-reference systems, program re-structuring systems

Table 2-1 Functional classification of CASE tools (Sommerville 2001: 66)

This research will be limited to CASE tools in the four functional areas listed in italics in Table 2-1; namely Editing, Change management, Configuration management and Method support tools, since these are the primary functions used by computing students for OOSAD courses.

2.3.1.2 A process perspective

A process perspective uses the process activities supported by a CASE tool to classify the tool. Figure 2-1 illustrates how CASE tools are classified according to process activities and how these are grouped into four main activities, namely specification, design, implementation and lastly, verification and validation.

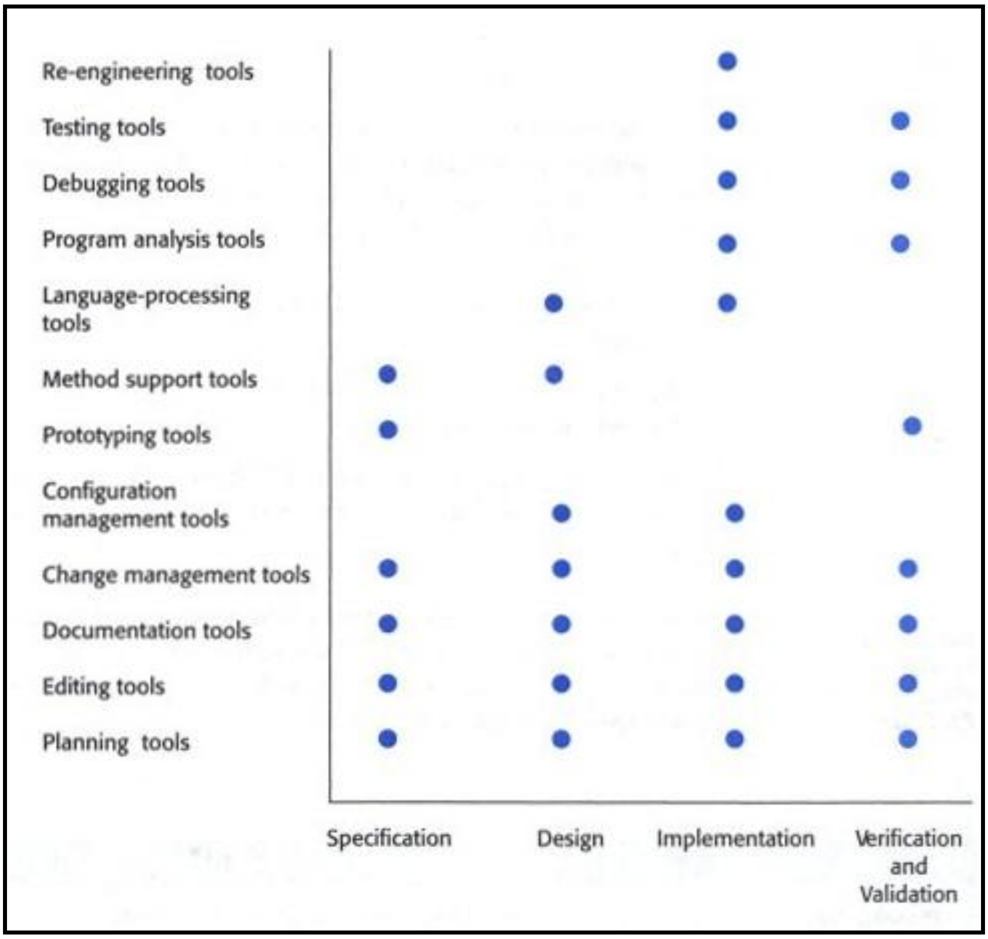

Figure 2-1 Process activity-based classification of CASE tools (Sommerville 2001:67)

This research will focus primarily on the first two of these activities, namely specification and design, since computing students spend most of their time in OOSAD courses on these activities.

2.3.1.3 An integration perspective

An integration perspective classifies CASE tools in terms of how they are organised into integrated units which provide support for one or more process activities.

2.3.2 Classification by methodology support

OO modelling and UML notation are not supported by all CASE tools. Some tools only implement traditional data modelling using entity relationship diagrams (ERDs) and data flow diagrams (DFDs), but there are some tools that implement both. Research has shown that some developers prefer CASE tools which are specifically OO based, rather than traditional CASE tools (Post *et al.* 2000).

An inheritance hierarchy of CASE tools is illustrated in Figure 2-2 (Phillips *et al.* 1998). This is based on the tools support of a methodology or system development process. At the top of the hierarchy are OO CASE tools. These can be divided into methodology-dependent CASE tools, which support only one methodology, and multi-methodology tools, which can support more than one methodology. Multi-methodology CASE tools are either Meta CASE tools, which provide automated or semi-automated support for developing CASE tools, or normal CASE tools, which support more than one methodology. Meta-CASE tools can be classified as either CASE tool generators or modifiable CASE tool environments (Phillips *et al.* 1998). A modifiable CASE tool environment can include generic, parametrizable CASE tools like Rational Rose as well as CASE tool frameworks like Eclipse-EMF/GEF (Taentzer 2007).

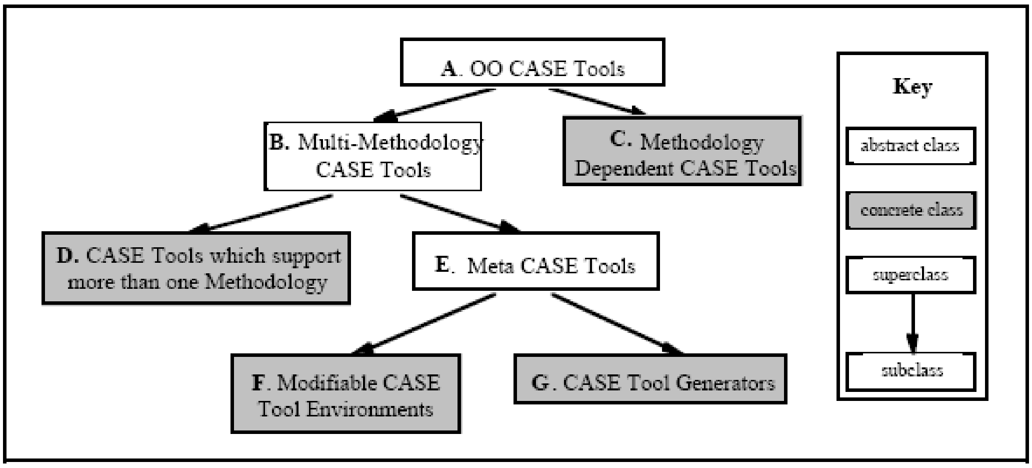

Figure 2-2 A CASE tool inheritance hierarchy (Phillips *et al.* 1998)

2.3.3 Classification by support offered for the software process

CASE systems can also be classified according to the support offered for the software process (Sommerville 2001). Figure 2-3 illustrates this classification and shows some examples of these different classes of CASE support. The three main categories according to this classification are:

- Tools;
- Workbenches; and
- Environments.

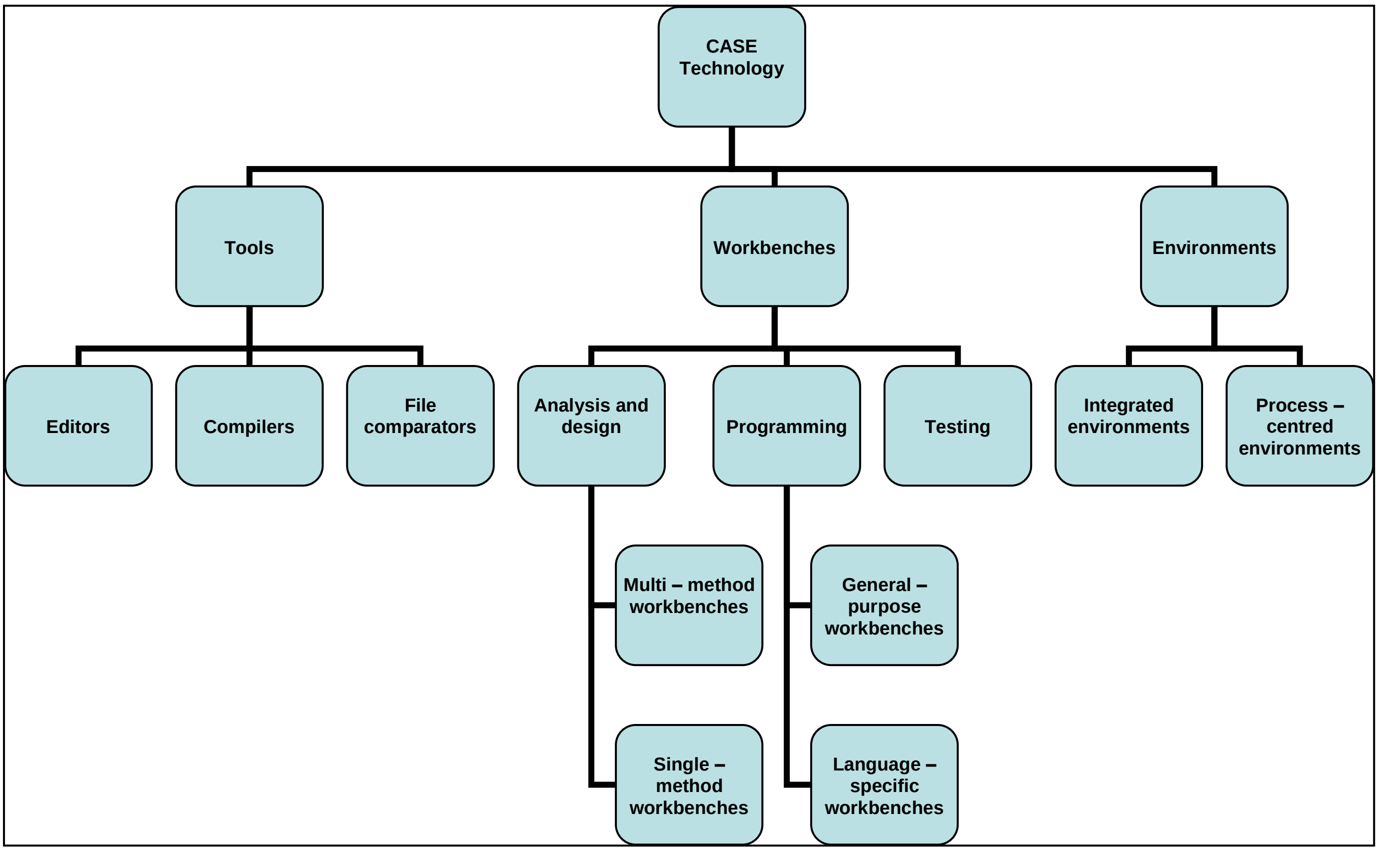

Figure 2-3 Tools, workbenches and environments (Sommerville 2001: 67)

2.3.3.1 Tools

Tools support the individual process tasks such as checking the consistency of a design, compiling a program or comparing test results. The three main types of tools are editors, compilers and file comparators.

2.3.3.2 Workbenches

The three main types of workbenches support tasks related to the different phases in the SDLC. They usually support some method, which includes a process model and a set of rules or guidelines, which apply to the software being developed. They often include a set of tools with some greater or lesser degree of integration.

2.3.3.3 Environments

Environments support all or a major part of the software process and normally include several different workbenches integrated in some way. Environments can be classified as integrated environments or process-centred environments.

Integrated environments provide infrastructure support for data, control and presentation integration. Process-centred environments are more general. They include software process knowledge and a process engine that uses this process model in order to advise engineers on what tools or workbenches to apply and when they should be used.

In practice, the boundaries between these three different categories in the classification of CASE tools are blurred; therefore, it may not always be easy to position a tool using this classification. For example, a CASE tool may be sold as a single system but may embed support for different activities. However, it does provide a useful first step in assisting with understanding the extent of process support which a tool is likely to provide (Sommerville 2001).

The scope of this research will be limited to CASE tools that can be categorised as workbenches, which support the tasks of analysis and design, since this is what is currently required by the OOSAD modules at NMMU.

2.4 Benefits of CASE tools

A significant percentage of the reported failures of developed systems are linked to faulty requirements. It is, therefore, extremely important for analysts and developers to ensure the quality of the conceptual models that they develop in the early phases of system development (Bolloju *et al.* 2006). The quality of these models can be improved by using a CASE tool (Section 1.1). CASE tools can reduce the time and cost of software development and the maintenance of the systems developed (Kemerer 1992; Lending *et al.* 1998). CASE tools can offer automated assistance in the development process that significantly contributes towards improving software productivity (Juric *et al.* 1999).

Using a CASE tool – as opposed to drawing by hand - has the following advantages (Burton *et al.* 2004):

- It improves the understanding of a difficult topic, by representing it visually;
- Diagrams can be more easily drawn, manipulated and modified with a CASE tool;

- A good CASE tool tracks changes to labels, attributes and operations across a set of diagrams;
- Using a CASE tool allows the iterative transition of models from conceptual to logical level to be made easily (because of the above); and
- Electronic diagrams are easy to exchange over a network or the Internet, which facilitates communication.

Some of the reasons for the importance of the teaching of CASE tools to computing students in HEI were highlighted in Section 1.1. Additional benefits for using CASE tools in computing education include:

- The role of a CASE tool as a pedagogical instrument in teaching and learning a system development methodology (Linder *et al.* 1994; Jankowski 1995); and
- The potential of a CASE tool to enhance the student's learning experience (Mynatt *et al.* 1990).

It has also been shown that teaching OOSAD in a computer laboratory where a CASE tool is installed is beneficial, particularly for students whose backgrounds are not academically strong (Burton *et al.* 2004).

2.5 Problems with CASE tool usage

Despite increased popularity of CASE tools, there are some problems with the adoption of these tools. Considerable research has been done relating to these problems. Many CASE tools have not delivered (Lending *et al.* 1998), and have been slower to be adopted by industry than expected (Fowler *et al.* 2000).

Some of the problems with CASE tools include the costs of the software and usability problems (Post *et al.* 2000). The problem of the high cost of these tools can be resolved over time as more tools become available and more developers gain experience with the technologies (Fowler *et al.* 2000; Post *et al.* 2000).

Some studies have shown that CASE tools are complex, difficult to use and time consuming (Kemerer 1992; Lending *et al.* 1998; Fowler *et al.* 2000). CASE tools are often not consistent with the needs of CASE users. Some of the features provided are not considered relevant by users and users also struggle to find the desired diagram shape (Post *et al.* 2000).

Teaching UML with industrial CASE tools (such as Rational Rose or Together CASE Tool) is not always satisfying, since the focus of these tools is on professional development rather than on computing education (Alfert, Pleumann and Schroder 2004). This implies that there is a high risk of the focus being on the tool itself rather than on the language or notation.

2.6 Guidelines for CASE tools

Many recommendations on CASE tools have been made by the research community. The majority of the problems with CASE tool usage relate to usability issues, which is the primary focus of this study. Problem areas were, therefore, grouped into three main categories, and existing guidelines and recommendations were discussed accordingly. These categories are:

- Tool complexity;
- Ease of use problems; and
- Learning environment.

2.6.1 Tool complexity

The main user interface of a CASE tool is usually a work area which looks like a sketch pad. This work area is used by the developer to build graphical structures consisting of a limited number of predefined symbols. These structures are used to model the system under development (Phillips *et al.* 1998). In an OO CASE tool, these structures would comprise various UML diagrams.

The size and complexity of UML is beyond what needs to be taught to most computing students, and it is not necessary to teach all the diagrams or all of the methods (Burton *et al.* 2004; Flint, Gardner and Boughton 2004). The majority of authors agree that the most important UML diagrams for students to learn are the following (Brewer *et al.* 2003; Frosch-Wilke 2003; Burton *et al.* 2004):

- Use case diagram;
- Class diagram;
- Activity diagram; and
- Sequence diagram.

An example of each of these diagrams is shown in Figures 2-4 to Figure 2-7. A use case diagram is a diagram that depicts the interactions between the system and external systems and users (Figure 2-4).

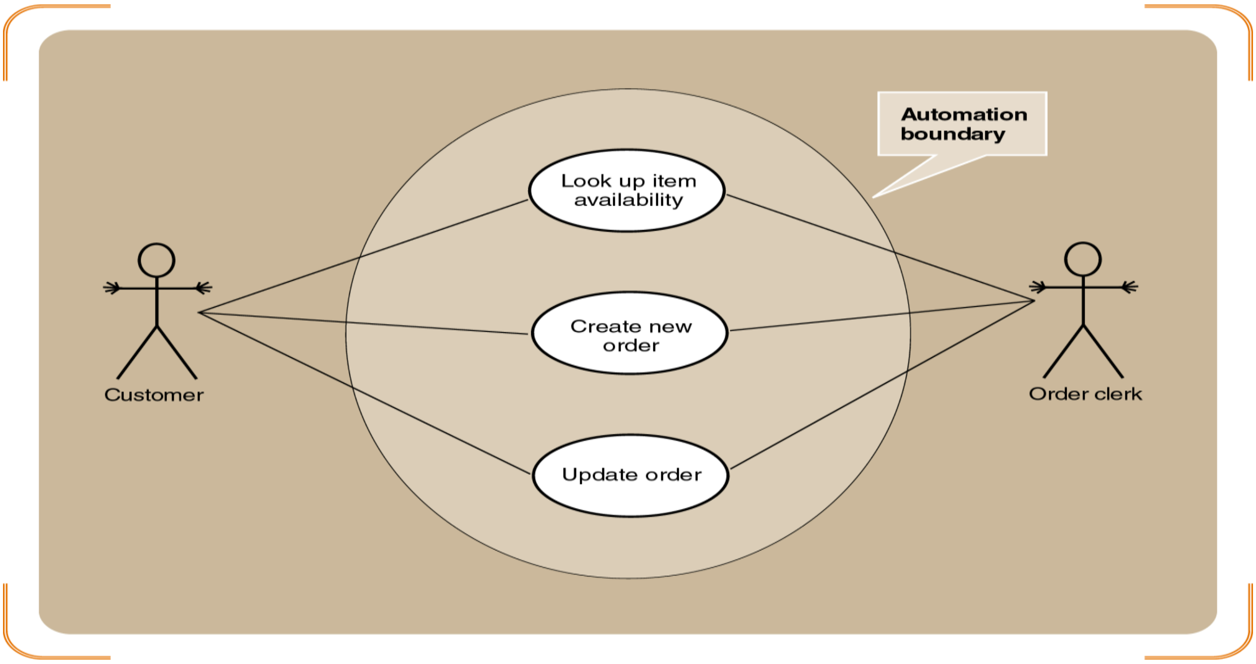

Figure 2-4 Use case diagram example (Satzinger, Jackson and Burd 2005: 216)

A class diagram is a graphical depiction of a system's static object structure, showing object classes that the system is composed of as well as the relationship between those object classes (Figure 2-5).

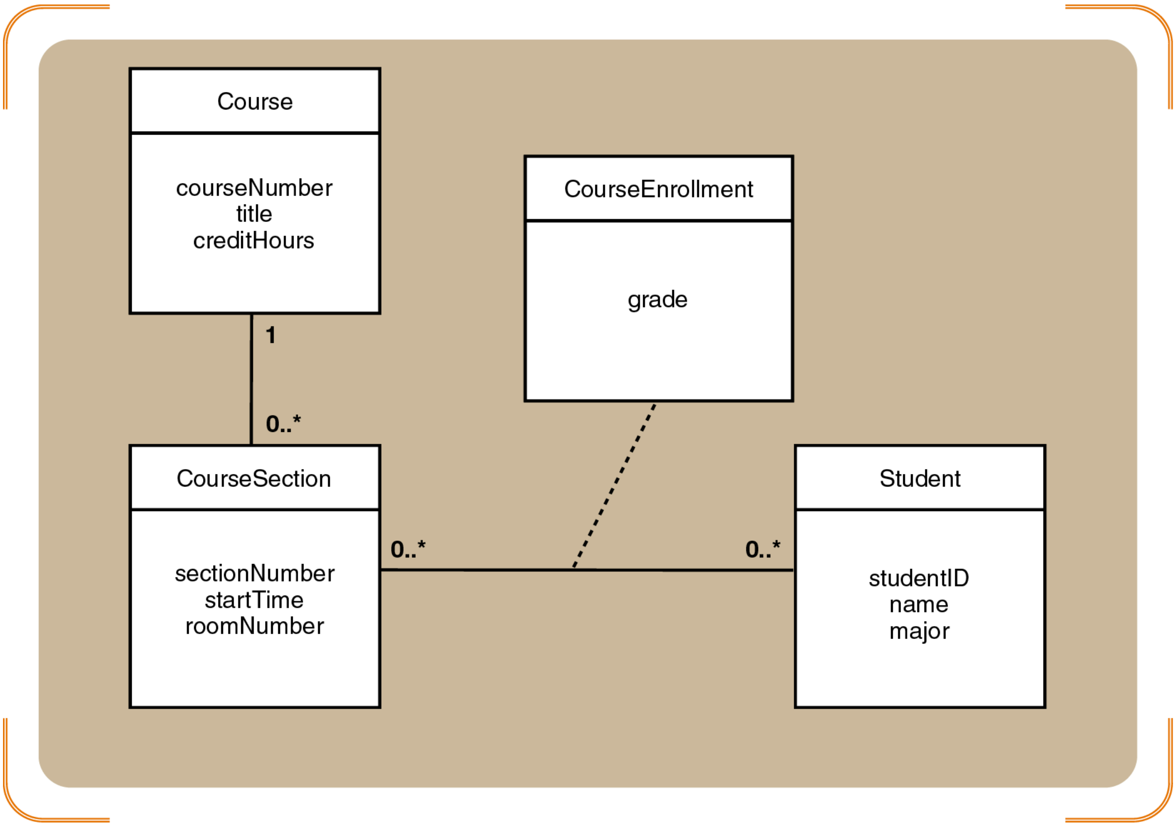

Figure 2-5 Class diagram example (Satzinger *et al.* 2005: 189)

Activity diagrams can be used to graphically depict the flow of a business process, the steps of a use case, or the logic of an object method (Figure 2-6). Another diagram used by some OO methodologies in the logical design phase is the sequence diagram which depicts how objects interact with each other via messages in the execution of a use case or operation (Figure 2-7).

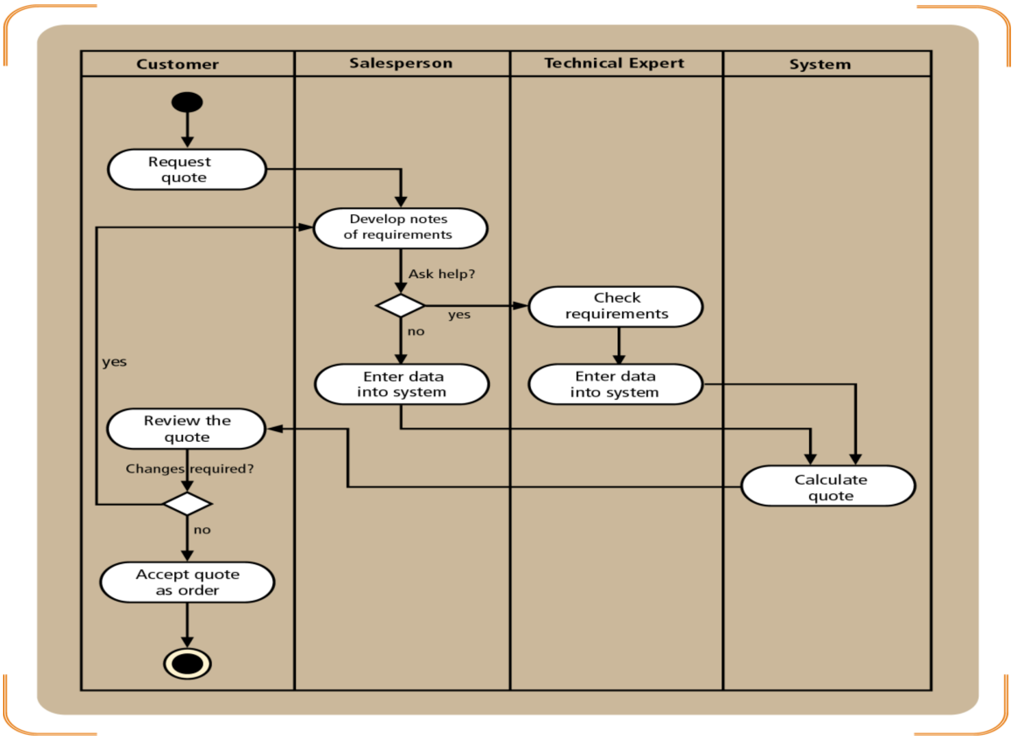

Figure 2-6 Activity diagram example (Satzinger *et al.* 2005: 146)

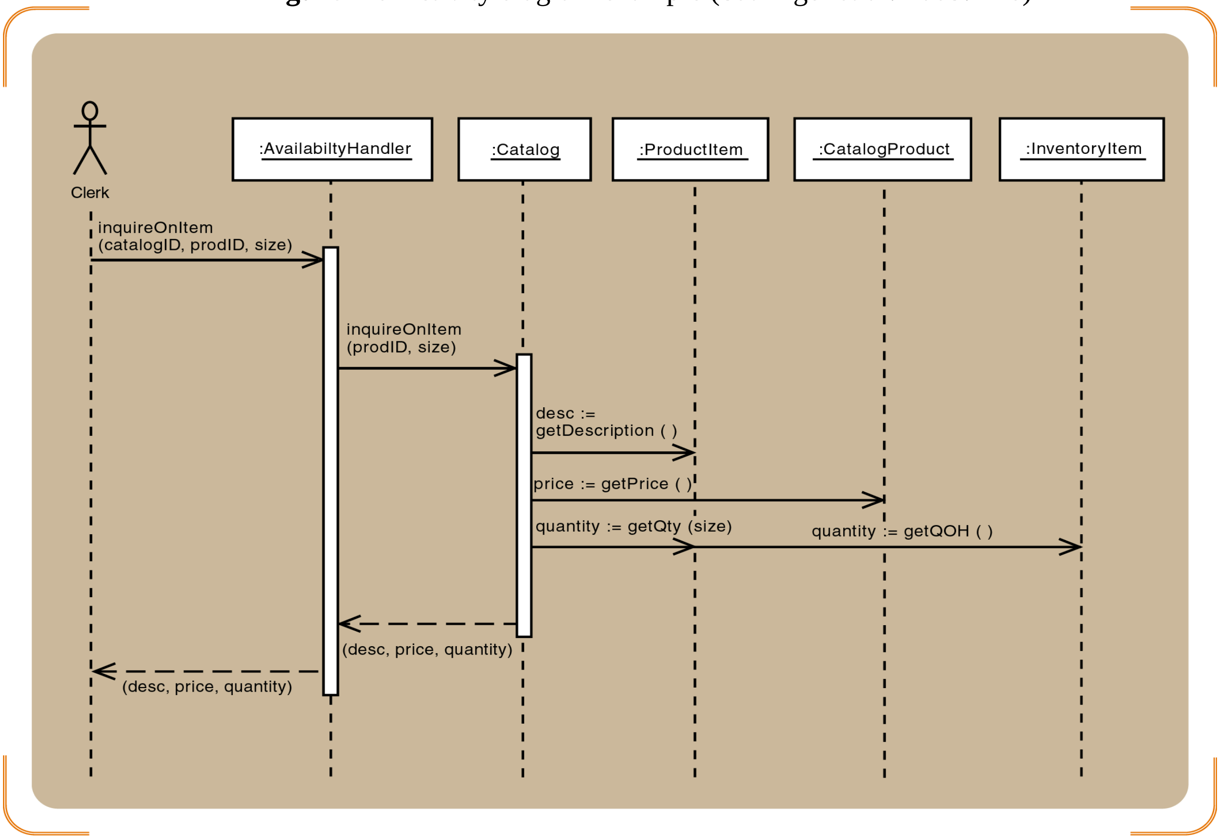

Figure 2-7 Sequence diagram example (Satzinger *et al.* 2005: 318)

Typically only a small portion of a CASE tool's functionality is used by the developers, mainly the diagram-editing and documentation-support features. Less frequently used functionality includes the detection of diagram inconsistencies (Lending *et al.* 1998) and more advanced features like reverse engineering and code generation (Maccari and Riva 2002). One way of improving the relevancy of features is by using CASE tools which are specifically OO based rather than traditional CASE tools (Post *et al.* 2000).

Another way of reducing complexity is to remove some of the unnecessary, sophisticated features found in industrial CASE tools and use a simpler, less cluttered user interface (Maccari *et al.* 2002; Auer, Tschurtschenthaler and Biffl 2003; Alfert *et al.* 2004; Feyt 2004). Interface complexity often results from feature overload; thus, the reduction of features that are not used can help reduce interface complexity (SAP 2007).

Including more advanced features in a CASE tool can, however, improve the quality of the diagrams produced (Brooks and Scott 2001; Van der Straeten 2002; Choi and Kim 2004; Flint *et al.* 2004). One of the disadvantages of adding these advanced features is the resultant increase in complexity of the CASE tool, which may aggravate the problem of ease of use. These advanced features can be classified into the following areas:

- Methodological constraints; and
- Automated processes for developing diagrams.

2.6.1.1 Methodological constraints

Different views exist regarding constraining users from performing UML errors when drawing a diagram in an OO CASE tool. One view is that CASE tools are deficient in providing methodology support (Juric *et al.* 1999) and should include consistency checking of the UML models (Van der Straeten 2002). If these constraints are not implemented then less maintainable, reusable and understandable models may be produced, which are then used merely for system documentation. A CASE tool should thus implement support for a methodology through the automated control of rules and informative feedback on the violation of these rules (Juric *et al.* 1999). Any violation of these rules may result in the CASE tool preventing a user or developer from proceeding in a non-methodological way or may simply issue a warning or error message.

Another view held by some is that tolerating rule violations or inconsistencies in diagrams may be desirable in order to avoid unnecessary restrictions on the development process (Auer *et al.* 2003).

CASE tools which are restricted by methodology rules, may not be attractive enough to the users, and should provide a natural process-oriented development framework rather than a methodology-oriented one. Some authors that hold this view maintain that a CASE tool should support the soft creative, problem-solving aspects of software development, as well as rigorous modelling, but also contain a knowledge base of standard solutions to well-known software development problems (Jarzabek *et al.* 1998).

A possible solution to these opposing views could be an "ideal" methodological constraint environment for CASE tools, which allows the users to create quality designs without alienating them by being too restrictive (Brooks *et al.* 2001). Studies on this "ideal" environment were limited to the area of traditional process-modelling CASE tools, and may, therefore, not be relevant to OO CASE tools.

2.6.1.2 Automated processes for developing diagrams

Students need advice on how to develop UML diagrams (Frosch-Wilke 2003). Developers of CASE tools should incorporate facilities to provide guidance to novice analysts in preventing typical novice errors during the modelling process (Bolloju *et al.* 2006). Some studies were performed on a student diagram-assessment system for UML interaction diagrams (Song 2001) and use-case diagrams (Hoggarth and Lockyer 1998; Rosenberg 1999).

The student diagram-assessment system was designed specifically for academic users who require assistance in the underlying methods for developing use cases, as well as the usage of the actual CASE tool (Hoggarth *et al.* 1998). This tool was developed using both Computer-Aided Learning (CAL) and CASE technology and received positive feedback. This research showed that it was possible to generate valid and valuable feedback for the production of systems analysis and design diagrams.

2.6.2 Ease of use problems

CASE tools could be improved by providing a list of diagrams using specified classes and, in this way, assist users who struggle with finding the required objects (Post *et al.* 2000). A CASE tool should provide more guidance for a novice developer than for an expert, but the majority of CASE tools appear almost the same to both novice and expert developers.

Psychological tests reveal that the human ability to cope with complexity is highly restricted (7+-2 principle) (Preece, Rogers and Sharp 2007). Even in the case of an expert user, there is little advantage to exposing many system functions at any one time (Jarzabek *et al.* 1998). If all the items can be scanned and rescanned visually (for example, a list of menu items) then they do not have to be recalled from short-term memory and the 7+-2 principle does not apply (Preece *et al.* 2007). The application of this 7+-2 principle means that user interfaces should promote recognition rather than recall by using menus, icons and consistently placed objects.

System interfaces should also provide users with a variety of ways of encoding digital information (for example files, emails and images) to help them remember where they have stored them, through the use of categories, colour flagging, time stamping, etc. The user should also not be overloaded with complicated procedures for carrying out tasks. It is also important to design interfaces which encourage exploration and constrain and guide users to select appropriate actions when initially learning (Preece *et al.* 2007).

An "active" CASE tool is, therefore, recommended which guides the developer in the development process, determines developers' intentions and customises tool functions to the task at hand. These "active" CASE tools must be based on sound knowledge from different types of developers and their expectations (Jarzabek *et al.* 1998).

2.6.3 Learning environment

Interest in the adoption of CASE tools is escalating, due to the significant role they play in supporting the software development process (Fowler *et al.* 2000). The effort involved in learning to use these tools has made their adoption by industry slower than expected (Section 2.5). The learnability of the CASE tool could be affected by the learning environment (Senapathi 2005). The learning environment is concerned with how and when the CASE tool is taught and includes learning methodologies, teaching methods and resources used.

Several recommendations have been made which relate to the learning environment of CASE tools. These have been grouped into four categories, namely:

- The use of UML within the context of a methodology;
- Integration into real-life projects;
- Provision of appropriate training; and
- Understanding the learning styles of users.

2.6.3.1 The use of UML within the context of a methodology

UML can be used with any software process, however, if UML techniques are put in the context of a specific process, then it can greatly assist the student's understanding of the value of diagrams for certain tasks (Frosch-Wilke 2003). This is important both for the theoretical teaching of UML and CASE tools, as well as the practical application thereof.

Different perspectives exist within the system development process that can be used to draw a UML diagram, namely, the conceptual, specification and implementation perspectives (Cook, 1994). Students need to understand these perspectives in order to be able to know how to use UML diagrams for the particular purpose required (Frosch-Wilke 2003).

Some research has been done regarding the relationship between learning a CASE tool and learning the underlying methodology. Some authors propose that developers should delay adopting a CASE tool until they are comfortable with the methodology (Kemerer 1992).

2.6.3.2 Integration into real-life projects

It is important that students use examples that are part of a practical, real-life project, rather than using isolated examples, in order to fully understand the benefits of using UML in software development and to produce effective diagrams (Frosch-Wilke 2003).

2.6.3.3 Provision of appropriate training

One of the reasons that CASE tools have not been adopted in industry as well as they should have, is that companies do not train the users sufficiently, or that training is not available (Fowler *et al.* 2000). Many tools are initially self taught and are then perceived to be complex, which adds to the difficulty of learning CASE tools (Fowler *et al.* 2000). More extensive educational programs for UML training are also needed, both to increase the number of developers and analysts familiar with UML and to provide ongoing support to help them make better use of its capabilities (Dobing and Parsons 2006).

2.6.3.4 Understanding the learning styles of users

Knowledge of the factors that favourably influence the rate of learning can lead to improved approaches to teaching CASE tools and hence the uptake of these tools within industry (Fowler *et al.* 2000). The Index of Learning Styles is an instrument to assess preferences on four dimensions (Felder and Soloman 2005), namely:

- Active/Reflective;
- Sensing/Intuitive;
- Visual/Verbal; and
- Sequential/Global.

Active learners like to try things out and see how they work and like to work with others. Reflective learners prefer to think about it quietly first. Sensing learners like to learn facts, use well-established methods and are practical and careful. Intuitive learners often prefer discovering possibilities and relationships. Visual learners like diagrams, pictures, graphs and films while verbal learners get more out of words, written and spoken explanations. Everyone learns more when information is presented both visually and verbally. Sequential learners tend to gain understanding in linear steps, with each step following logically from the previous one. Global learners like to jump in, absorb material nearly at random and then get the big picture.

Studies were performed on the learning styles of software engineering students using the CASE tool, Rational Rose. The results showed that usability issues may be affected by user characteristics like gender, computer experience and learning styles (Fowler *et al.* 2000). Their studies produced the following results:

- Fifty-seven percent (57%) of students are active learners, yet teaching is often passive;
- Seventy percent (70%) of students are sensors, yet we often teach intuitively;
- Eighty-three percent (83%) of students are visual, yet a lot of material is presented verbally or in written form; and
- Thirty-nine percent (39%) of students are global learners, yet our teaching is often sequential and does not focus on the big picture.

2.7 Selecting CASE tools for computing education

The previous section discussed the problems associated with using CASE tools, particularly with learning to use these tools. Some guidelines relating to these problems were also discussed. The key guidelines relevant to the usage of CASE tools in computing education were identified and are summarised in Table 2-2. These can assist in the initial, preliminary selection process to draw up a short list of CASE tools prior to performing a usability evaluation, as well as in the implementation process.

Phase	Category		Guidelines for selecting and implementing OO CASE tools in computing education
SELECTION	Tool Complexity	1	Select a CASE tool which is specifically OO based.
		2	The CASE tool should focus on the core UML diagrams – use case diagram, class diagram, activity and sequence diagram.
		3	The CASE tool should have a simple, uncluttered user interface.
		4	The CASE tool should provide basic checking of UML models but not be too restrictive and have too many constraints.
		5	The CASE tool should provide simple assistance for the underlying process and methods.
	Ease of use problems	6	The CASE tool user interface must provide a list of diagrams using specified classes.
		7	The CASE tool user interface should promote recognition rather than recall by using menus, icons and consistently placed objects.
		8	The user interface of the CASE tool should encourage exploration and constrain and guide users to select appropriate actions when initially learning.
IMPLEMENTATION	Learning environment	9	Teach and use UML within the context of a systems development methodology.
		10	Integrate the CASE tool into real-life projects.
		11	Provide appropriate training.
		12	Understand the learning styles of the users.

Table 2-2 Guidelines for selecting and implementing OO CASE tools in computing education

The first two categories in Table 2-2 relate to the selection process and the third category relates to the actual implementation of the tool and the associated learning environment. It must be kept in mind that these guidelines are primarily related to usability and learning issues and not to the utility of the CASE tools.

2.8 Conclusions

CASE tools are software which can provide automatic support for one or more activities of the SDLC (Section 2.2). Upper-CASE tools are CASE tools which support the first few phases of the SDLC. The scope of this research is limited to these upper-CASE tools (Section 2.3). The scope was further limited to CASE tools in four functional areas namely Editing, Change management, Configuration management and Method support tools (Section 2.3.1.1). CASE tools that support the various phases of the SDLC are also known as workbenches. This research will be limited to workbenches supporting the analysis and design tasks occurring in the first three phases in the SDLC (Section 2.3.3.3), as these are the types of CASE tools required for OOSAD modules at NMMU.

There are several benefits to using CASE tools in computing education (Section 2.4). The key benefit is that CASE tools can improve the understanding of a difficult topic, such as UML, by representing it visually. The main problems with CASE tools relate to usability and learnability problems. One of the main reasons for problems with CASE tool usage in computing education is that the tools were designed for use by professional developers in industry and not for students of OOSAD modules. Several guidelines for overcoming these problems were compiled and grouped into three categories, namely tool complexity, ease of use and the learning environment (Section 2.6). Twelve of these guidelines were identified as being relevant for computing education, and should be used to select the CASE tools which are going to be used in a learnability evaluation and to implement them in the learning environment (Section 2.7).

The usability and learnability of CASE tools is investigated further in Chapter 3.

Chapter 3 Usability and learnability of CASE tools

3.1 Introduction

Chapter 2 discussed the importance and benefits of UML and CASE tools for the success of IT projects, as well as for computing education. The types of problems and difficulties encountered with the usage of these tools were also discussed. A large majority of the problems with CASE tools relate to the usability of these tools. Usability should, therefore, be high on the priority list when selecting a CASE tool. This study is however, concerned with the learnability aspect of usability and the primary research question *"How can the learnability of an OO CASE tool for computing education be evaluated?*

In order to answer the primary research question, several subsidiary research questions were compiled. This chapter attempts to address the second subsidiary research question *"What principles and frameworks exist for evaluating the usability and learnability of a CASE tool*?". It does this by firstly discussing the process of evaluating usability and the need for specifying measurable usability requirements. Secondly usability attributes which should be included in the usability requirements are reviewed. Those attributes which contribute to the learnability aspect (the learnability attributes) will be examined. Next, principles which should be used when measuring the usability and learnability of a CASE tool are reviewed. A comparison of methods that can be used for evaluating usability will be provided. Lastly, frameworks for evaluating CASE tool usability and learnability will be analysed.

3.2 Usability requirements

The term usability was introduced and some definitions provided in Section 1.1. Other definitions of usability can be found, for example, *"ensuring that interactive products are easy to learn, effective to use and enjoyable from a user's perspective"*. Usability involves optimising the interactions people have with interactive products to enable them to carry out their activities at work, school and in their everyday life (Preece *et al.* 2007). One of the foremost authors of human-computer interaction (HCI), Jakob Nielsen, considers usability to be part of system acceptability and defines it as how well users can use the functionality or utility of the system (Nielsen 1993).

When incorporating usability in a quality system, measurable usability requirements need to be specified (ISO 2001; Jokela, Iivari, Matero and Karukka 2003). Having measurable usability requirements will reduce the risk of different stakeholders having different views of quality (Kitchenham and Pfleeger 1996).

Another benefit of having measurable usability requirements is that the usability aspect of projects becomes more recognised and goal-driven and leads to greater acceptance of usability work (Jokela *et al.* 2003). These usability requirements should include the purpose of the evaluation, the types of system(s) and the specification of the quality model (ISO 2001). Quality models for software evaluation generally represent the totality of software quality attributes classified in a hierarchical tree structure of characteristics and sub-characteristics. Figure 3-1 illustrates this concept. The highest level of this structure consists of software quality characteristics and the lowest level consists of software quality attributes. An attribute is an entity which can be verified or measured in a software system. Sufficient internal and external attributes should be identified for each sub-characteristic. An external attribute represents the external perspective of software quality when the software is in use. Internal attributes of the software relate to design and code (ISO 2001).

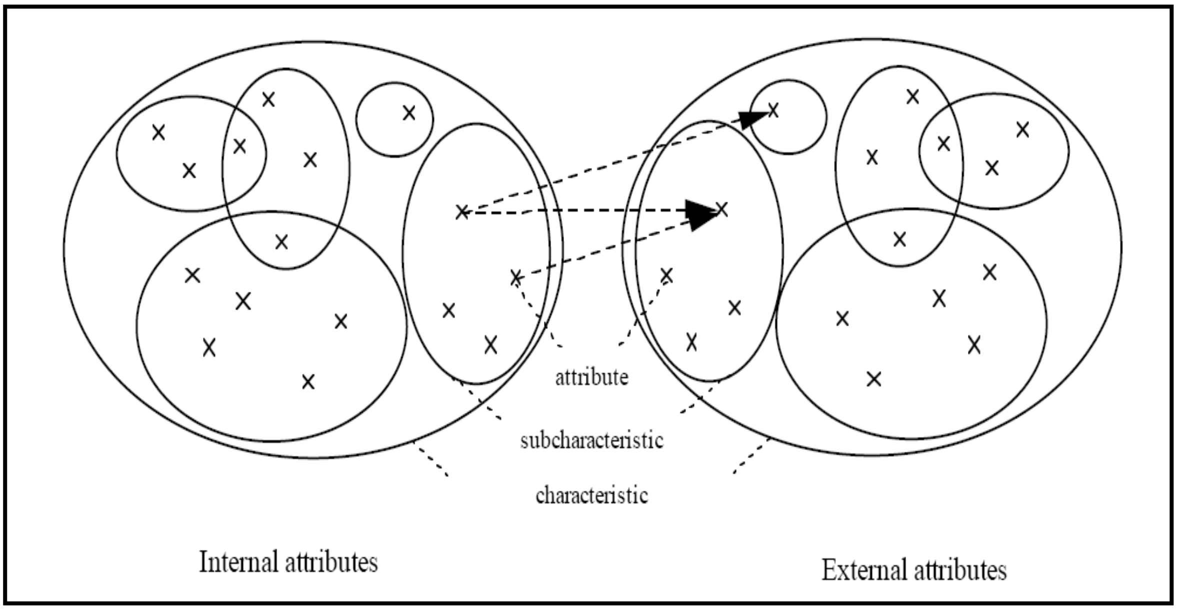

Figure 3-1 Quality characteristics, sub-characteristics and attributes (ISO 2001)

Usability standards and metrics defined by the software engineering community for HCI and usability are developed under the auspices of the International Organisation for Standardisation (ISO) and the International Electrotechnical Commission (IEC).

The *ISO/IEC 9126-1: Software product evaluation (2001)* standard refers to usability as one of six software quality characteristics which can be used for evaluating the quality of software systems (ISO 2001). The six quality characteristics of software are:

- Functionality;
- Reliability;
- Usability;
- Efficiency;
- Maintainability; and
- Portability.

These quality characteristics are, in turn, further divided into sub-characteristics which can consist of attributes which can be measured by internal or external metrics. The latest definition of usability provided is as follows:

***Usability**: the capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions (ISO 2001).*

The phrase *"when used under specified conditions"* is equivalent to *"context of use*" in the earlier standard ISO 9241-11 (ISO 1998). It was added to make it clear that a software system has no intrinsic usability, only a capability to be used in a particular context. Quality in use is the combined effect of the six categories of software quality when the system is in use, both for the end user and the support user (Bevan 2001). The definition of quality in use provided in the ISO 9241-11 standard is as follows:

***Quality in use:** the capability of the software product to enable specified users to achieve specified goals with effectiveness, productivity, safety and satisfaction in specified contexts of use (ISO 2001).*

The ISO/IEC 9126-1 definition of usability is concerned with attributes of the software system that make it understandable, learnable, easy to use and attractive (ISO 2001), and is closer to previous definitions of usability (Nielsen 1993), where usability was identified in terms of the ease of use and learning, and excludes utility. The definition of usability provided by ISO/IEC 9126-1 is slightly different from the ones provided earlier, where usability was considered as having goals relating to effectiveness, efficiency and satisfaction.

The more recent definition provided by ISO no longer considers these as aspects of usability. Efficiency is regarded as an attribute of external and internal quality together with usability, whereas effectiveness and satisfaction are regarded as attributes of quality in use (Bevan 2001).

Two activities relate to the determination of usability requirements: 'Understand and specify the context of use', and 'Specify the user and organisational requirements' (Jokela *et al.* 2003). The context of use includes the characteristics of the intended users, the tasks the users are to perform and the environment in which the users are to use the system. The user and organisational requirements should provide a clear statement of the human-centred design requirements so that the usability evaluation can measure the system against these requirements.

Insufficient research exists, however, on how to determine usability requirements (Jokela *et al.* 2003). The existing standards recommend using usability attributes and then measuring the attributes directly (Kitchenham *et al.* 1996). These attributes must be operationalised and expressed in measurable ways (Jokela, Koivumaa, Pirkola, Salminen and Kantola 2006). It may be necessary to specify criteria both for the minimum acceptable level of usability and for the target level of usability (ISO 1998). User preference questionnaires using Likert scales can provide a subjective metric for the related usability attribute (Jokela *et al.* 2006).

Several different classification schemes for quantifying and assessing usability have been proposed within the HCI and software engineering communities (Gould and Lewis 1985; Nielsen 1993; Nielsen and Levy 1994; Sommerville 2001; Barnum 2002; Dix, Finlay, Abowd and Beale 2004). The most popular schemes are listed below in chronological order.

1. Attributes of usability (Nielsen 1993);
2. Sub-characteristics of usability, which can in turn be comprised of several attributes (ISO 2001) ;
3. Categories of principles of usability (Dix *et al.* 2004);
4. Factors broken down into measurable criteria or sub-factors (Seffah, Donyaee, Kline and Padda 2006); and
5. Usability goals (Preece *et al.* 2007).

The term, "attributes of usability" will be used throughout this study. Table 3-1 lists the recommended usability attributes of each scheme identified.

Scheme	Author	Term	Attributes
1	(Nielsen 1993)	Attributes of usability	• Learnability • Efficiency • Memorability • Few Errors • Satisfaction
2	(ISO 2001)	Sub-characteristics of usability	• Learnability • Understandability • Operability • Attractiveness
3	(Dix *et al.* 2004)	Categories of usability principles	• Learnability • Robustness • Flexibility
4	(Seffah *et al.* 2006)	Factors of usability	• Efficiency • Effectiveness • Satisfaction • Productivity • Learnability • Safety • Trustfulness • Accessibility • Universality • Usefulness
5	(Preece *et al.* 2007)	Usability Goals	• Learnability • Efficiency • Memorability • Safety • Utility

Table 3-1 Attributes of usability

One of the more recent proposals consists of a consolidated model of usability measurement and metrics (Seffah *et al.* 2006). This proposal lists ten factors, each of which corresponds to a specific aspect of usability that is identified in an existing standard or model. These ten factors are decomposed into a total of 26 sub-factors or measurable criteria that can be further decomposed into 127 specific metrics. This consolidated model is called the Quality in Use Integrated Measurement (QUIM) model.

A comprehensive list of 14 usability attributes based on those proposed by the authors in Table 3-1, is provided in Table 3-2 in chronological order. Related terms are shown in the second column.

Attribute of usability	Related term
Learnability	Understandability
Efficiency	
Memorability	
Satisfaction	
Operability	
Attractiveness	
Robustness	Few Errors
Flexibility	
Effectiveness	Productivity
Safety	
Trustfulness	
Accessibility	
Universality	
Usefulness	Utility

Table 3-2 Consolidated list of attributes of usability

Some authors reference the ISO standards (Bevan 2001; Torchiano, Jaccheri, Sørenson and Wang 2002; Senapathi 2005; SAP 2007), whereas others reference well-known usability experts (Phillips *et al.* 1998; Dix *et al.* 2004). It should be acknowledged that the ISO standards were designed for large monolithic organisations (Torchiano *et al.* 2002), and may therefore not be appropriate for educational purposes. The ISO standards do not specify the detailed usability attributes, as these can vary between different organisations and types of systems (ISO 1998).

Once usability requirements have been established and specified, the second step in usability evaluation is to specify the evaluation details. This includes selecting metrics, establishing rating levels for metrics and establishing criteria for assessment. A metric is defined as a quantitative scale and method which can be used for measurement (ISO 1998). Care should be taken when setting criteria that an appropriate weight is given to each measurement item. For example, to set criteria based on errors, it may be necessary to assign weightings to reflect the relative importance of different types of error (ISO 1998). The priority with which each evaluation metric is weighted will depend on the characteristics of the intended user of the system, the tasks, the environment and the specific dialogue technique used.

The definitions and principles provided in the ISO standards can be used to describe explicit measurements for usability (Dix *et al.* 2004). However, there are a number of limitations to these standards. Firstly, they often describe principles and not useful solutions. These standards must be taken into account together within the intended context of use of the system. Secondly, standards on specific details of the user interface can quickly go out of date and should therefore be assessed and updated on a regular basis (Bevan 2001). In addition, evaluators without usability experience have great difficulty applying these standards. Another problem with these standards is that several usability attributes are not included (Torchiano *et al.* 2002).

A growing number of publications in the literature have addressed the problem of how to measure software usability (Sauro and Kindlund 2005; Seffah *et al.* 2006; Jokela *et al.* 2006). It is not clearly indicated how to specify these requirements or how to go about determining the relevant attributes. However, it is better to determine usability requirements somehow, than not to determine them at all (Jokela *et al.* 2003).

This section has highlighted the importance of having measurable usability requirements for a usability evaluation. Some guidance has been provided on identifying the usability attributes necessary for specifying these requirements. The next section will define and discuss the consolidated list of usability attributes identified in Table 3-2.

3.3 Usability attributes

The previous section discussed the importance of specifying usability requirements in order to measure the usability of a system. In order to specify the usability and learnability requirements, the relevant attributes must be identified. This section will review the attributes identified in Table 3-2 as contributing to the usability of a system.

3.3.1 Learnability

The system should be easy to learn so that the user can rapidly start getting some work done with the system (Faulkner 2000; Sommerville 2001). Learnability concerns the features of the interactive system that allow novice users to understand how to use it initially and then how to attain a maximal level of performance. It is also defined as the length of time it takes a new user to become productive with the system (Sommerville 2001), and the capability of the software system to enable the user to learn its application (ISO 2001).

Another definition of learnability refers to it as the ease with which new users can begin effective interaction and achieve maximal performance (Dix *et al.* 2004). When users encounter new systems, learnability is the first attribute they will become aware of since they will be endeavouring to learn how to use the system (Faulkner 2000). How easy it is to learn to use the system will affect their attitudes towards the system. It is necessary for systems to be learned quickly because the cost of training is high and employers would like their employees to be productive with the system as quickly as possible. How easy a system is to learn can be measured in terms of a novice user's experience of learning how to operate it. A system that is easy to learn will enable the user to carry out a large amount of tasks in a short space of time, thus reflecting how quickly the user learns to operate some areas of the system. Some systems can be said to aim for zero-learning time. These systems will be highly usable and transparent and will endeavour to use the user's knowledge of the world in order to make them easy to learn. In other words, they will build on what the user already knows or provide clear cues as to what needs to be done (Faulkner 2000).

The concept of understandability can be related to learnability and is defined as the capability of the software to enable the user to understand whether the software is suitable and how it can be used for particular tasks and conditions of use. This will depend on the documentation and initial impressions given by the software (ISO 2001).

3.3.2 Efficiency

The system should be efficient to use, so that once the user has learned the system, a high level of productivity is possible. Efficiency is also referred to as the resources expended in relation to the accuracy and completeness of goals achieved (ISO 1997).

3.3.3 Memorability

The system should be easy to remember, so that the casual user is able to return to the system after some period of not having used it, without having to learn everything all over again (Nielsen 1993).

3.3.4 Satisfaction (Subjectively Pleasing)

The system should be pleasant to use, so that users are subjectively satisfied when using the system. Satisfaction is also defined as the comfort and acceptability of the working system to its users and other people affected by its use (ISO 1997).

It is the capability of the software system to satisfy users in a specified context of use. It can also be described as the user's response to interaction with the system, and includes attitudes towards use of the system (ISO 2001).

3.3.5 Operability

Operability is the capability of the software system to enable the user to operate and control it (ISO 2001).

3.3.6 Attractiveness

Attractiveness is defined as the capability of the software system to be attractive to its user. This refers to attributes of the software intended to make the user interface more attractive to the user, such as the use of colour and the nature of the graphical design (ISO 2001).

3.3.7 Robustness

Robustness can be defined as how tolerant the system is of user error (Sommerville 2001). It is the level of support provided to the user in determining successful achievement and assessment of goals (Dix *et al.* 2004). This can be related to Nielsen's concept of *Few Errors*, whereby the system should have a low-error rate, so that users make few errors during the use of the system. If the user does make an error, he/she should be able to easily recover from it. Further, catastrophic errors should be prevented (Nielsen 1993).

3.3.8 Flexibility

Flexibility refers to the multiplicity of ways in which the user and system exchange information (Dix *et al.* 2004). Five main aspects affecting flexibility are identified, namely:

- ***Dialogue initiative:*** The extent to which the user is in control of the dialogue and is free from artificial constraints imposed by the system.
- ***Multi-threading:*** A measure of the ability of the system to support more than one task thread at any time.
- ***Customisability***: A measure of the extent to which the system can be adapted, either by the user or by the system.
- ***Task migratability***: The ability to pass control for the execution of a given task so that it becomes either internalised by the user or the system or shared between them.

- ***Substitutivity:*** Allows equivalent values of input and output to be arbitrarily substituted for each other.

3.3.9 Effectiveness

Effectiveness is the capability of the software system to enable users to achieve specified tasks with accuracy and completeness. This is related to the concept of productivity, which is the level of effectiveness achieved in relation to the resources (i.e. time to complete tasks, user efforts, materials or financial cost of usage) consumed by the users and the system. In contrast with efficiency, productivity concerns the amount of useful output that is obtained from user interaction with the software system (Seffah *et al.* 2006).

3.3.10 Safety

Safety is concerned with whether or not a software system limits the risk of harm to people or other resources, such as hardware or stored information (Seffah *et al.* 2006). Safety involves protecting the user from dangerous conditions and undesirable situations (Preece *et al.* 2007).

3.3.11 Trustfulness

Trustfulness refers to the faithfulness a software system offers to its users. This concept is most relevant to e-commerce websites but can be applied to other software products (Seffah *et al.* 2006).

3.3.12 Accessibility

Accessibility is the capability of a software system to be used by persons with some type of disability (Seffah *et al.* 2006).

3.3.13 Universality

This is concerned with whether or not a software system accommodates a diversity of users with different cultural backgrounds (Seffah *et al.* 2006).

3.3.14 Usefulness

Usefulness refers to whether a system enables users to solve real problems in an acceptable way (Seffah *et al.* 2006). The concept of usefulness relates to the term utility used by other authors.

Usefulness implies that a system has practical utility, which in part reflects how closely the system supports the user's own mental model. Utility refers to the extent to which the system provides the right kind of functionality so that users can do what they need or want to do.

3.4 Learnability attributes

This section will discuss the specific measurable attributes of learnability. Some attributes are directly related to the concept of learnability, whereas others may indirectly affect it. The learnability attributes identified are listed in Table 3-3 and originate primarily from the work done by Dix (Dix *et al.* 2004) and Phillips (Phillips *et al.* 1998) .

3.4.1 Direct attributes

Five direct attributes of learnability are identified (Dix *et al.* 2004). These are:

- Familiarity;
- Consistency;
- Predictability;
- Synthesizability; and
- Generalizability.

LEARNABILITY	
DIRECT	Familiarity
	Consistency
	Predictability
	Synthesizability
	Generalizability
INDIRECT	Feedback
	Robustness • Error Prevention • Recoverability • Provision of Help

Table 3-3 Attributes of learnability

3.4.1.1 Familiarity

Familiarity is defined as the extent to which a user's knowledge and experience in other real-world or computer-based domains can be applied when interacting with a new system. New users of a system bring with them a wealth of experience across a wide number of application domains. This experience is obtained both through interactions in the real world and through interaction with other computer systems. For a new user, the familiarity of an interactive system is measured in terms of the correlation between the user's existing knowledge and the knowledge required for effective interaction (Dix *et al.* 2004).

Familiar concepts enable people to get started and become productive quickly (Galitz 2002). When designing interfaces it is, therefore, important to employ familiar concepts and use a language that is familiar to the user. The interface must be kept natural, mimicking the user's behaviour patterns. Real-world metaphors should be used. Familiarity has to do with a user's first impression of the system, and whether or not the user can determine how to initiate any interaction.

There are affordances (or intrinsic properties) of any visual object that suggest to us how they can be manipulated. The appearance of an object stimulates a familiarity with its behaviour. Effective use of the affordances that exist for interface objects can enhance the familiarity of an interactive system (Dix *et al.* 2004). Perceived affordance is the characteristic of a system that gives clues of how it can be used (Norman 1999). The notion of perceived affordance is widely recognised by the user interface design research community, where it is believed that "*perceived affordances can directly affect usability, reduce the need for instruction, user manuals, online help and support, and can also promote familiarity with the interface*" (Seffah and Rilling 2001).

3.4.1.2 Consistency

Consistency can be defined as relating to the likeness in behaviour arising from similar situations or similar task objectives (Dix *et al.* 2004). Consistency must be applied relative to something, for example, consistency in command naming or in command/argument invocation. Familiarity could be considered as consistency with respect to previous real-world experience (Dix *et al.* 2004). Consistent interfaces are easier to learn and use (Preece *et al.* 2007), which in turn assists the users in gaining more confidence in using the system and encourages them to try out exploratory learning strategies (Nielsen 1993).

Consistency is important because it can reduce requirements for human learning by allowing skills learned in one situation to be transferred to another like it (Galitz 2002). While any new system will impose some learning requirements on its users, it should avoid encumbering productive learning with non-productive, unnecessary activity. Consistency also aids learning of the system's conceptual model.

3.4.1.3 Predictability

Predictability can be defined as support for the user to determine the effect of future action based on past interaction history. It is a user-centered concept and is deterministic behaviour from the perspective of the user (Dix *et al.* 2004). This means that in an interactive system, the user's knowledge of the interaction history is sufficient to determine the result of his future interaction with it. It is not enough for the behaviour of the computer system to be determined completely from its state, as the user must be able to take advantage of the determinism. This notion of predictability is concerned with the user's ability to determine the effect of operations on the system (Dix *et al.* 2004).

Predictability reduces mistakes and enables tasks to be completed more quickly (Galitz 2002). The user should be able to anticipate the natural progression of each task. This can be done in the following ways (Galitz 2002):

- By providing distinct and recognisable screen elements; and
- By providing cues to the result of an action to be performed.

Another form of predictability has to do with the user's ability to know which operations can be performed and is termed *operation visibility*. Operation visibility is explained as how the user is shown the availability of operations that can be performed next (Dix *et al.* 2004). The principle of operation visibility also supports the superior ability of humans for *recognition over recall*. Without this ability, the user will have to remember when an operation can be performed and when not (Dix *et al.* 2004). Likewise, the user should understand from the interface if an operation cannot be performed.

3.4.1.4 Synthesizability

Synthesis is the ability of the user to assess the effect of past operations on the current state (Dix *et al.* 2004).

The principle of honesty relates to the ability of the user interface to provide an observable and informative account of a change to some aspect of the internal state caused by an operation. This notification should preferably come immediately, requiring no further interaction by the user, or failing that, it should come eventually, after explicit user directives to make the change observable.

For example, in a graphical user interface (GUI), a Move operation results in an icon representing a file being dragged from its original directory and placed in its destination directory where it remains visible. The user need not expend any more effort to assess the result of the operation, and the GUI is thus immediately honest.

3.4.1.5 Generalizability

Generalizability of an interactive system supports the activity of users extending their knowledge of specific interaction behaviour to situations that are similar and have not been previously encountered. Generalization can be applied to situations in which the user wants to apply knowledge that helps achieve one particular goal to another situation where the goal is in some way similar.

Generalizability can be seen as a form of consistency with respect to experience with the same system or set of applications on the same platform. An example of generalizability would be the attempt to provide Cut/Paste/Copy operations in all applications in the same way (Dix *et al.* 2004).

3.4.2 Indirect attributes

Some attributes of usability may also indirectly affect learnability. These are divided into two aspects, namely:

- Feedback; and
- Robustness.

3.4.2.1 Feedback

Effective feedback refers to providing informative, positive and continuous feedback about what a system is doing and how it is interpreting the user’s input and facilitates the activities of the learners (Nielsen 1993).

The presence of appropriate and relevant feedback, specifically in the context of handling error messages, is considered to have a significant effect on the learnability and understandability of a system (Jankowski 1995; Norman 1999). Providing feedback should also enable the user to be aware of the visibility of system status at all times (Nielsen 1994). The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

3.4.2.2 Robustness

Three attributes contribute to robustness (Phillips *et al.* 1998). These include:

- Error prevention;
- Recoverability; and
- Provision of help.

Error prevention: Error prevention is the degree to which the user interface of a system minimises the possibility for errors to occur. The interface should prevent errors from occurring, and this could contribute to improved learnability (Nielsen 1993; Phillips *et al.* 1998; ISO 2006; SAP 2007; SERC 2007). However, people like to explore and learn by trial and error (Galitz 2002). A system oversensitive to erroneous inputs will discourage users from exploring and trying new things. Learning will be inhibited, and people will be overcautious, working slowly and carefully to avoid mistakes. Productivity will then suffer (Galitz 2002).

Systems should prevent errors from occurring by anticipating where mistakes may occur and designing to prevent them (Galitz 2002). However, automated systems must be careful of not removing the feeling of control from the user, which can lead to increased work stress. In general modes, for example, a changed mouse pointer shape should be avoided since it constrains the actions available to the user at any given time. If modes are necessary then they should be easy to learn and easy to remove (Galitz 2002).

A well designed user interface should prevent a problem from occurring in the first place. This can be achieved by either eliminating error-prone conditions completely or checking for them and presenting users with a confirmation option before they commit to the action (Nielsen 1994).

Recoverability: Recoverability is referred to as how good the system is at recovering from user errors (Sommerville 2001). It is also defined as a measure of the ease with which the user can take corrective action once an error has been recognised (Dix *et al.* 2004).

A fear of making a mistake and not being able to recover from it has always been a primary contributor to fear of dealing with computers (Galitz 2002). Systems should permit people to review, change and undo actions whenever necessary. When errors do occur, the system should present clear instructions on how to correct these.

Error messages which are more specific, positive in tone, easy to understand and constructive can aid recoverability (Shneiderman 1998; Galitz 2002). All errors should be displayed on the screen and identified to the user through a highlighting display technique such as high-intensity or contrasting colour (Galitz 2002). The error messages should describe what error occurred and how it should be corrected. This may lead to lower error rates and increased subjective satisfaction. Using interfaces that permit the easy reversal of actions can assist in the learning process of the user (Shneiderman 1998). Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution (Nielsen 1994).

Provision of help: HCI research highlights the importance of the provision of help facilities in software systems for improving usability and learnability (Nielsen 1993; Shneiderman 1998; Seffah *et al.* 2001; Preece, Rogers and Sharp 2002; SERC 2007). On-line documentation or on-line help should be available (Nielsen 1993).

Three additional attributes are identified as contributing to robustness (Dix *et al.* 2004). These are:

- Observability;
- Responsiveness; and
- Task conformance.

Observability: is defined as a measure of the ease with which the user can evaluate the internal state of the system from its perceivable representation (Dix *et al.* 2004). This is very similar to the concept of feedback (Section 3.4.2.1).

Responsiveness: refers to how the user perceives the rate of communication with the system, and is also referred to as the concept of stability and efficiency (Dix *et al.* 2004).

***Task conformance*:** is defined as the degree to which the system supports the tasks the user wishes to perform in the way in which the user understands them and wants to carry them out (Dix *et al.* 2004). It can therefore be deduced that this concept is related to the utility and usefulness of the system.

This study will not include these three attributes as being relevant to the learnability of CASE tools, since they are similar concepts to other attributes which have already been identified as contributing to learnability.

This section has discussed the attributes to be used for specifying the usability requirements of software. The next two sections will review the specific usability principles which should be used when evaluating CASE tools, in terms of usability and learnability.

3.5 Usability principles for CASE tools

Usability principles should be used as rules of thumb for evaluators in a usability evaluation, particularly if the evaluators are not experts (Usability.gov 2007). The ISO standard *ISO 14102: The evaluation and selection of CASE tools* provides a standard process for the evaluation and selection of a CASE tool in an organisation (ISO 1995). The focus of this study is on the evaluation of the learnability of CASE tools and not utility, so only the relevant sections of the ISO 14102 standard will be reviewed.

The principles and attributes of usability, which should be used when measuring the usability of a CASE tool, are identified in ISO 14102. These are summarised in Table 3-4. One of the items recommended in this standard was not included in this table, namely, *Ease of Installation*. This is because installation is not one of the requirements for computing education since students are not involved with the installation of a CASE tool.

The last column in Table 3-4 indicates the related attribute in the consolidated list of usability attributes identified in Table 3-2 as well as the related learnability attribute in Table 3-3 (where applicable). This link was made in order to ensure that each of the recommended principles of ISO 14102 was covered by the usability attributes identified.

As can be seen from Table 3-4, there were no CASE tool usability principles which did not have a related usability attribute from the list in Table 3-2. This supports the premise that Table 3-2 is a comprehensive list. It is also evident from this table that the majority (six out of nine) of the principles relate to learnability, which provides additional evidence of the importance of learnability to the usability of a CASE tool.

One of the usability principles, namely Ease of learning (Item 8), had no related attribute in the list of learnability attributes in Table 3-3. Ease of learning refers to the novice user's experience on the initial part of the learning curve while trying to learn a new system (Nielsen 1993). It can be deduced that Ease of learning should be added to the list of learnability attributes in Table 3-3.

Item	ISO Principle (ISO 1995)	Usability Attribute
1	***User Friendliness:*** attributes relating to its ability to integrate into the tool user's activities, taking into account the user's level of experience and expertise, and the concepts, information, representations and procedures that are part of the user work domain and culture (professional and individual).	Usefulness
2	***User Guidance:*** attributes relating to the provisions to allow the tool user to know the status of tool operation, to establish the causal relationship between user actions and tool status, and to assess and direct user actions on the tool. Capabilities may be provided in the form of on-line help features, and diagnostic and error messages.	Learnability • Feedback • Provision of help
3	***Homogeneity:*** attributes relating to the consistency of logic within an application or across applications, at the procedural level as well as for the presentation of information.	Learnability • Consistency
4	***Adaptability:*** attributes relating to the ability of its interface to adapt to various task requirements, strategies, habits, and cultural modes. Adaptability has several aspects: the ability to adapt to users with differing levels of experience, the ability of the user to customise input and output methods, and in the number of procedures, options and commands available to a user to achieve a given objective.	Flexibility
5	***Claritv of Control:*** attributes relating to the extent to which the semantics of the dialogue steps used to control the tool, reflect the resulting action, and the predictability of the action.	Learnability • Predictability
6	***Error Handling***: attributes relating to its abilities to help and guide the user in identifying and correcting errors, and to maintain tool integrity (avoiding incorrect data and process changes).	Learnability • Error prevention • Recoverability
7	***Conciseness:*** attributes which decrease the required number of steps to identify and memorise, and which increase the efficiency of the dialogue.	Efficiency
8	***Ease of Learning***: attributes relating to the amount of time and effort required for a user to understand normal CASE tool operations and to become productive.	Learnability • Ease of learning
	The availability and quality of on-line tutorials may be a factor in ease of leaming. These features should be integrated within the presentation and structure of the data on the screen (or reports).	Learnability • Provision of help
9	***Tool Documentation Quality:*** attributes relating to the overall quality of the documentation provided with the tool. If a tool implements a methodology, descriptions of the methodology should accompany the tool.	Learnability • Provision of help

Table 3-4 Principles of usability for CASE tools in computing education

3.6 Learnability principles for CASE tools

The *ISO 9241-110:2006(E) Ergonomics of human-system interaction – Part 110: Dialogue Principles* standard provides design principles formulated in general terms and provides a framework for applying these principles to the analysis, design and evaluation of interactive systems (ISO 2006). Seven principles are identified as being important for the design and evaluation of interactive systems, which serve as a set of general goals for the design and evaluation of dialogues. For each of these principles, a list of illustrative recommendations is provided, but these should be applied within a specific context of use. The main focus of this research is on evaluating the learnability of CASE tools, and not on usability in general (Table 1-1 Question 3). For this reason not all of the principles identified in this standard are listed, but only those which correspond to *Suitability for learning* (ISO 2006).

A dialogue principle is described as suitable for learning when it supports and guides the user in learning to use the system. When evaluating the usability of a system, the focus is on optimising the system for a given context. The way in which each dialogue principle in ISO 9241-10 should be applied to the evaluation of a system will thus depend on its context of use. When applying the dialogue principles, particular consideration should be given to tasks which would typically be performed by the intended users in order for them to meet the goals of their organisation (ISO 1998).

The principles are summarised in Table 3-5. The last column of this table identifies the related attribute of learnability identified in Table 3-3 (Section 3.4). This was done to ensure that the dialogue principles relevant to CASE tools in computing education were covered by a relevant learnability attribute. It is evident that all of the recommended principles have an associated learnability attribute, which further supports the premise that Table 3-3 is a comprehensive list. The three main attributes covered by these principles are Provision of Help, Feedback, Error prevention and Predictability. No exact match was found for principle 7 which relates to operation visibility. It was therefore linked to Predictability (Secion 3.4.1.3) since these two concepts are similar.

Table 3-5 was compiled mainly from the ISO 9241 standard, and does not include principles for the learnability attributes of Consistency, Synthesizability and Generalizability. Consistency was included in Table 3-4 (Item 3). This list of principles should therefore be expanded to include all the learnability attributes.

A new attribute, namely Simplicity was identified, that was not in the original list in Table 3-3. This decision is supported by other research which maintains that the simplicity of a user interface contributes to the learnability of a system (Feyt 2004; SAP 2007). It can be deduced that Simplicity should be added to the list of learnability attributes identified in Table 3-3.

	Principle	**Example**	**Learnability Attribute**
1	Rules and underlying concepts which are useful for learning should be made available to the user.	A CASE tool explains the concepts of a class diagram to the user as part of an initial tutorial.	Provision of Help
2	If infrequent use or user characteristics require relearning of the dialogue, then appropriate support should be provided.	A CASE tool provides a help system that guides the user through the required dialogue steps of creating a new class diagram.	Provision of Help
3	Appropriate support should be provided to assist the user in becoming familiar with the dialogue.	A CASE tool explains the use of individual menu items when the user presses the assigned help key.	Familiarity Provision of Help
4	Feedback or explanations should assist the user in building a conceptual understanding of the interactive system. This should differ between novice and experienced users.	A CASE tool shows all the steps required and the current step reached by the user when drawing a UML diagram.	Feedback
5	The dialogue should provide sufficient feedback about the intermediary and final results of an activity so that the user learns from successfully accomplished activities.	When a user draws a UML diagram, the user receives step-by-step feedback to refine his/her diagram and details about any potential UML violations.	Feedback
6	If appropriate to the tasks and learning goals, the interactive system should allow the user to explore dialogue steps without negative consequences.	• A CASE tool allows a user to evaluate potential variations of a UML diagram to allow the user to foresee negative impacts before changes are applied. • Allow an “Undo” facility.	Error Prevention
7	The system should enable the user to perform the tasks with minimal learning by entering only the minimum amount of information required.	A CASE tool provides all shapes for the necessary UML diagrams and they are clearly visible and available for drag and drop, using the minimum number of steps, using sensible default settings for all options.	Predictability Simplicity

Table 3-5 Dialogue principles: Suitability for learning of CASE tools

This section has reviewed principles which should be applied when evaluating CASE tool learnability. The previous two sections reviewed usability principles specifically for evaluating CASE tools, as well as the attributes to be used for measuring the usability and learnability of software systems. The following section will discuss some existing usability evaluation methods.

3.7 Usability evaluation methods

The previous sections discussed the attributes and principles to be used when evaluating the usability and learnability of a system. This section will discuss some of the methods used for usability evaluation.

There are many methods proposed for evaluating usability, each with its limitations, advantages and disadvantages. Which method(s) should be used depends on a number of different factors including the purpose of the evaluation and the available resources. There may not be one best method or technique for any given situation.

In some cases, combinations of methods may be more beneficial. Every situation must be assessed individually in order to determine which methods to apply. Two types of methods of doing usability evaluations are (Wania, Atwood and McCain 2006):

- Analytical methods; and
- Empirical methods.

Analytical methods are normally conducted by experts or designers to inspect potential design problems. The finding of a usability problem depends on two factors: First, the subject has to experience the problem, and second, the experimenter has to realise that the user experienced the problem (Nielsen and Landauer 1993). Most analytical methods do not involve users, and use structured approaches to conduct evaluations. A popular analytical method is heuristic evaluation (Te'eni, Cary and Zhang 2007), which is mostly suited for use in the early stages of the SDLC (Dix *et al.* 2004), and for mobile and ambient devices (Preece *et al.* 2007). It is very popular as a discount evaluation method that can be used when resources are limited (Nielsen 2007). However, it does not involve real users and thus may not find actual problems related to real users in a real context (Te'eni *et al.* 2007).

Empirical methods are normally conducted by involving users and collecting facts about users interacting with the system. Data can be collected which is either qualitative or quantitative in nature (Te'eni *et al.* 2007). A popular empirical method is usability (or user) testing, which can be done either in a laboratory or in the field, in other words, the place where users actually perform their tasks with the system (Barnum 2002).

More than one of these methods may be used in the same evaluation. Although there are many individual methods for evaluating usability; these are not well integrated into a single, conceptual framework that facilitates their usage (Seffah *et al.* 2001).

Usability testing can be described as the measurement of user performance on typical tasks, while satisfaction can be measured through questionnaires and interviews (Preece *et al.* 2007). Usability testing is also sometimes referred to as user testing and is based on bringing real users in and observing them as they interact with the system in order to perform a given set of tasks (Nielsen *et al.* 1993).

Usability testing has also been described as *"a process that employs participants who are representative of the target population to evaluate the degree to which a product meets specific usability criteria*" (Rubin 1994). It is the process of learning from users about a system's usability by observing them using the system. Usability testing determines whether users can find and use the features in the amount of time and effort they are willing to expend searching for the relevant features (Barnum 2002).

The goal of usability testing is to test whether the system being developed is usable by the intended population to achieve the tasks for which it was designed. Usability testing generally has the following characteristics (Dumas and Redish 1999):

- The primary goal is to improve the usability of a system. For each test, there must be specific goals and concerns that are articulated when planning the test.
- The participants represent real users.
- The participants do real tasks.
- The team observes and records what participants do and say.
- The team analyses the data, diagnoses the problems and recommends changes to fix these problems.

The attributes selected during the requirements process can be used as measures for evaluating usability and fall into two broad categories; namely subjective user-preference measures, assessing how much the users like the system; and objective-performance measures, which usually measure how capable the users are at using the system (Nielsen *et al.* 1994; Rosson and Carroll 2002). Most usability evaluation techniques examine performance or satisfaction with the current version of the system.

Most usability tests also gather users' subjective reactions to a system. Users may be queried in a general fashion (for example, "What did you [dis]like most?") or they may be asked to rate the usability of specific tasks or features. An interesting and challenging aspect of user testing is that subjective reactions do not always correspond to performance data. A feature may improve efficiency but also annoy users, or it may slow users down but make them feel more comfortable (Rosson *et al.* 2002).

The validity of the data gathered from a usability evaluation to predict the usability achieved when a system is actually used will depend upon the extent to which the users, tasks and context of use are representative of the real situation and the nature of the measures chosen (Rosson *et al.* 2002).

Usability testing was considered the most appropriate method for this study, since it involves real users and tasks and is one of the methods most suitable for systems that are in the implementation phase of the SDLC (Te'eni *et al.* 2007). Another reason for using usability testing for this study was that other studies involving CASE tool evaluations used this method (Phillips *et al.* 1998). These studies proposed frameworks for evaluating CASE tool usability (Phillips *et al.* 1998) and for CASE tool learnability (Senapathi 2005) and will be discussed in more detail in the next two sections.

3.8 Phillips et al.'s framework

3.8.1 Overview

Methods for evaluating OO CASE tools can quickly become outdated due to the fact that OO technology has a high rate of change (Phillips *et al.* 1998). It is important, therefore, to use up-to-date, proven methods. Very little research has been done on the evaluation of OO CASE tools, and that which does exist, focuses on utility rather than usability (Phillips *et al.* 1998). A framework was designed by Phillips *et al*. for evaluating the usability of OO CASE tools (Phillips *et al*. 1998). This framework is based on an inheritance hierarchy of OO CASE tool categories discussed in Section 2.3.2 (Figure 2-2); as well as the usability criteria-classification hierarchy illustrated in Figure 3-2. The usability hierarchy was derived from earlier work by Nielsen and Dix (Nielsen 1993; Dix *et al.* 2004). Using this hierarchy to evaluate CASE tool usability can facilitate the easy identification of usability problems. This hierarchy may be pruned or expanded depending on the requirements of the evaluation (Phillips *et al.* 1998).

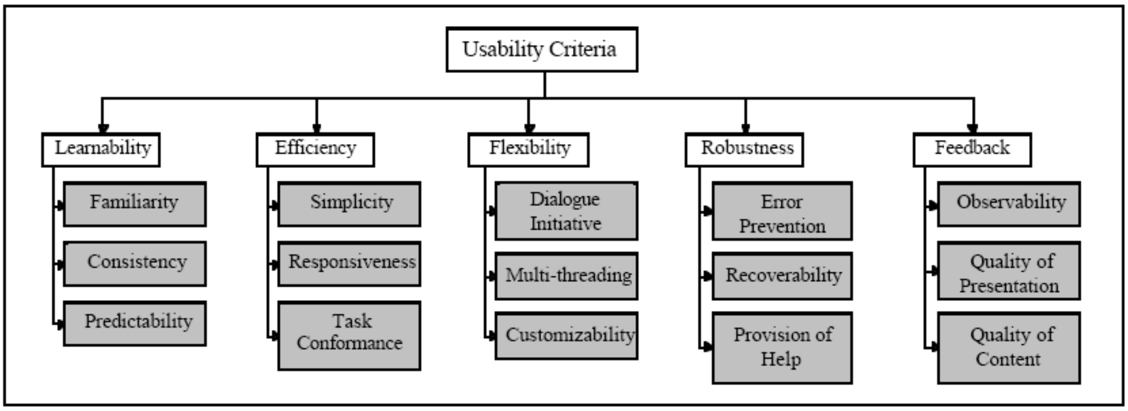

Figure 3-2 Usability criteria classification hierarchy (Phillips *et al.* 1998)

3.8.2 Discussion

The classification of usability criteria presented in Figure 3-2 was compared with those identified by Dix since this was the most popular classification of usability found. This framework uses the term criteria rather than the term attributes, which is used in this study. Four attributes which are identified in the Dix classification are not included in Phillips' usability classification. The first two attributes are learnability attributes, namely *Synthesizability* and *Generalizability* (Section 3.4.1), but no other evidence was found to support the inclusion of these two attributes.

Phillips *et al.'s* framework conforms to the recommendation made by Sommerville that a usability evaluation should be conducted against a usability specification based on usability attributes (Sommerville 2001). Two attributes of Robustness identified by Phillips and supported by other research, are not listed in the Dix classification. These are *Error Prevention* and *Provision of Help* (Section 3.4.2). *Feedback* was identified by Phillips as contributing to usability, and is supported by other authors (Section 3.4.2.1), although this is not included in the Dix classification. However, one of the attributes of feedback (as identified by Phillips), namely *Observability*, is listed in the Dix classification, but under *Robustness.*

Phillips recommends two additional attributes of feedback, which are not included in the Dix classification (Phillips *et al.* 1998), namely:

- *The Quality of presentation* which relates to the extent to which the interface is clear, subjectively pleasing and well structured; and
- The *Quality of content* which is a measure of the quality of the system's responses to the user.

The findings of this section were used to update the list of attributes contributing to learnability (Table 3-3), and an updated list of attributes which can be used for CASE tool learnability evaluation is provided in Table 3-6.

The Feedback attribute identified in Table 3-3 has been sub-divided into three component attributes, namely *Observability*, *Quality of presentation* and *Quality of content*, based on the research performed by Phillips *et al*.

Ease of learning and Simplicity were added as attributes contributing directly to learnability based on the literature survey (Section 3.5) and are thus listed in italics in the table. *Feedback* and *Robustness* have been included under learnability as evidence was found in literature to support their inclusion as indirect attributes of learnability (Section 3.4.2).

LEARNABILITY	
DIRECT	**INDIRECT**
• Familiarity	**Feedback**
• Consistency	• Observability
• Predictability	• Quality of presentation
• *Ease of learning*	• Quality of content
• *Simplicity*	**Robustness**
	• Error prevention
	• Recoverability
	• Provision of help

Table 3-6 New classification of learnability attributes for CASE tools

Phillip's framework evaluates the usability of a CASE tool. It does not focus specifically on the learnability of the tool. The list of learnability attributes proposed in Table 3-6 are very similar to those proposed by Phillips. The main differences are the addition of the *Ease of learning* and *Simplicity* attributes and the linking of *Feedback* and *Robustness* attributes to contribute to Learnability (albeit indirectly) and not only to usability as shown in Phillips' framework.

The next section will discuss a framework designed for evaluating the learnability of a CASE tool in computing education.

3.9 Senapathi's framework

3.9.1 Overview

A framework for evaluating CASE tool learnability was proposed by Senapathi, which was adapted from the ISO9241 usability definition (Senapathi 2005). This framework is depicted in Figure 3-3 and was used to determine the effect of contextual (context of use) factors on CASE tool learnability (Senapathi 2005). These contextual factors will be discussed in more detail in the next section. The framework also provides insight into whether the CASE tool allows learners with different characteristics to successfully learn to use the tool.

Prior to the Phillips and Senapathi studies, methods and frameworks for evaluating CASE tools were mainly suitable for commercial environments and specified an exhaustive set of evaluation criteria (Mosley 1992; ISO 1997).

The research done by Phillips focused on usability in general, whereas Senapathi's framework is concerned only with those usability attributes which affect learnability (Section 3.9.2.2).

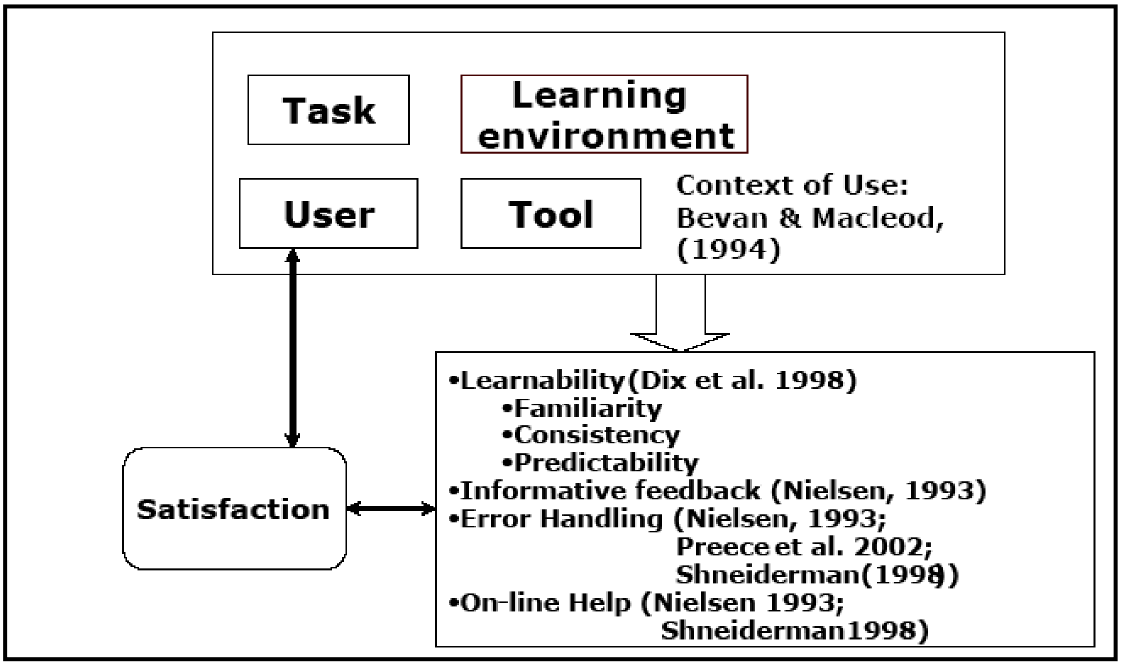

Figure 3-3 A framework for the evaluation of CASE tool learnability (Senapathi 2005)

As can be seen in Figure 3-3, Senapathi's framework consists of three main sections which are indicated by blocks in the diagram.

The first block on the top represents the C*ontext of use* or contextual factors, the block on the bottom right contains the measurable attributes of learnability and the third block on the bottom left represents the users' *Satisfaction* with the CASE tool. These three aspects of the framework will now be discussed in more detail.

3.9.2 Discussion

Each of the three sections of Senapathi's framework will be discussed in more detail in this section.

3.9.2.1 Context of use

Senapathi suggests that a better understanding of the issues that students confront while learning a CASE tool can be gained from investigating the contextual factors that affect CASE tool learnability (Senapathi 2005). These contextual factors include:

- Tasks;
- Learning environment;
- User characteristics; and
- Tool.

Tasks

Task analysis comprises a list of all the tasks that the users want to carry out using the system, the pre-conditions required to achieve these goals, the various steps involved and the interdependencies between these steps, and the various outputs and reports that need to be produced (Nielsen 1993).

In an educational context, all the activities and assessments that require the use of a CASE tool are recognised as tasks and include the following:

- Assessment activities that mandate the use of the tool;
- Goals of the tutorials and exercises used to support the learning of the tool; and
- Dependencies, if any, between the tutorials/formative assessments and the output requirements of the summative assessments that mandate the use of the tool.

Learning Environment

Senapathi's framework recognises the significance of the learning environment in the learnability evaluation of CASE tools in educational environments (Senapathi 2005). The background and context in which the course is delivered should be studied and analysed. This includes the learning methodologies, teaching methods and resources used (Section 2.6.3). An understanding of how and when the CASE tool is taught will have an effect on usability.

User characteristics

CASE tools should be learnable in a short timeframe and support a wide range of different learner characteristics due to the time constraints of students (Senapathi 2005). Senapathi's study explored the effects of five user characteristics on learnability. The five user characteristics investigated were the following (Section 3.9.3):

- Gender;
- General level of computer experience;
- Previous experience with CASE tools;
- Number of hours spent per week with CASE tools; and
- Attitude and motivation.

Tool

The focus of attention in any usability evaluation is on the specific system to be evaluated in a given context. The tool aspect of the framework, therefore, relates to the complexity of the selected CASE tool and how it affects learnability.

The participants in Senapathi's study were undergraduate students at the University of Auckland, New Zealand. The tool selected for the evaluation was Rational Rose as this was the CASE tool used in an undergraduate Systems Analysis and Design course presented at this university (Senapathi 2005).

3.9.2.2 Learnability

A system's learnability can be evaluated by measuring specific measurable attributes in a real-life context (Senapathi 2005). Senapathi proposes seven attributes that should be used to measure learnability (Figure 3-3).

These attributes are:

- Familiarity;
- Consistency;
- Predictability.
- Informative Feedback;
- Error Handling; and
- On-Line Help.

Senapathi's framework addresses all three of the direct attributes affecting learnability, namely Familiarity, Consistency and Predictability (Table 3-6). Informative feedback could be more adequately measured if it was broken down into the proposed attributes of Observability, Quality of presentation and Quality of content (Section 3.8.2). Error prevention is not included in the framework at all, and should therefore be added to it, as it is an important measure of Robustness which is an indirect attribute of learnability (Section 3.8.2). Table 3-6 would thus provide a more comprehensive list of all the direct and indirect attributes of learnability to be measured in CASE tool evaluations, and would provide greater detail and accuracy in identifying learnability problems in CASE tools.

3.9.2.3 Satisfaction

The last aspect of Senapathi's framework addresses the user's satisfaction, and how it is affected by the context of use factors as indicated by the measurable attributes of learnability. Satisfaction was used as a direct measure of evaluating whether or not a CASE tool was learnable in a given context (Senapathi 2005).

3.9.3 Framework implementation

Senapathi used an empirical usability evaluation method where participants of the study assessed the learnability of a CASE tool. The type of usability evaluation method used was usability testing (Section 3.8). The tasks completed by the participants consisted of drawing three types of UML diagrams, namely:

- Use case diagrams;
- Activity diagrams; and
- Class diagrams.

In order to record the five user characteristics, two questionnaires were given to the participants to complete. These were:

- A demographic questionnaire; and
- A post-test attitude and motivation questionnaire.

The demographic questionnaire included questions relating to the first four user characteristics identified (Section 3.9.2.1), namely:

- *Gender:* male or female;
- *General level of computer experience*: low, moderately low, moderately high and high;
- *Previous experience with CASE tools:* none/less than one month, one month or more, but less than six months, six months to one year, and more than one year; and
- *Number of hours spent per week with CASE tool*: none, less than one hour, one to less than four hours, four to less than ten hours and over ten hours.

The last user characteristic was *Attitude and motivation*, which was recorded by means of a post-test questionnaire. This attitude and motivation questionnaire comprised a set of questions for measuring the way students felt about the use of the CASE tool in their course. A copy of these questions is shown in Figure 3-4.

A post-test learnability questionnaire was administered at the end of the term when students had completed all of their course work that required the use of a CASE tool. This questionnaire consisted of a closed-ended section as well as an open-ended section. Students were asked to rate each question on a Likert scale of 1 to 5 where 1 = “Poor” and 5 = “Excellent”, for the attitude and motivation questionnaire as well as the closed-ended section of the learnability questionnaire. The closed-ended section of the learnability questionnaire was divided into six main sections.

LEARNABILITY OF CASE TOOLS						
ATTITUDE AND MOTIVATION QUESTIONNAIRE						
Student Number:						
Student Name:						
Instructions: Mark your selection with an X in the relevant box.						
No	**Question**					
		Poor				**Excellent**
1	The use of a CASE tool for this course was a good idea.	1	2	3	4	5
2	The CASE tool made my work more interesting.	1	2	3	4	5
3	The use of a CASE tool enabled me to complete my tasks more quickly.	1	2	3	4	5
4	The use of CASE tool helped me to understand the underlying concepts better.	1	2	3	4	5
5	Correct understanding and use of the CASE tool helped me to perform better in the UML section of the course and assignments.	1	2	3	4	5

Figure 3-4 Attitude and motivation questionnaire (Senapathi 2005)

A copy of this section of the learnability questionnaire is shown in Figure 3-5. Participants were required to provide their subjective ratings of each of these sections. Five of the six sections related to the measurable attributes of learnability proposed by Senapathi's framework (Section 3.9.2.2 and Figure 3-3). *Section 1: Ease of Learning* in the questionnaire had no related attribute in the framework. *Section 2: Familiarity, Section 3: Consistency* and *Section 4: Predictability* correspond to the respective attributes of learnability as proposed by Senapathi. *Section 5: Informative Feedback* corresponds to the second measurable attribute of learnability in Senapathi's framework, whereas *Section 6: Error Messages* corresponds to the *Error Handling* attribute. The last measurable attribute of learnability in Senapathi's framework, namely, *On-Line Help*, did not have any related questions in the closed-ended part of the questionnaire.

A modified Questionnaire for User Interaction Satisfaction (QUIS) was included at the end of the learnability questionnaire, which was used to measure overall user reactions to the system, using a 5-point Likert scale (Senapathi 2005).

SURVEY QUESTIONS						
Instructions: Rate the following between 1 (Very poor) and 5 (Excellent). Indicate your choice with an X.						
SECTION 1: LEARNABILITY						
1	**Ease of Learning**					
		Poor				**Excellent**
	a. It was easy for me to get started and to learn how to use the tool.	1	2	3	4	5
	b. I was able to use the tool right from the beginning, without having to ask my tutors or my peers for help.	1	2	3	4	5
	c. The system encouraged me to try out new system functions by trial and error.	1	2	3	4	5
	d. It was easy for me to remember commands from one session to another.	1	2	3	4	5
	e. The explanations provided helped me to become more and more skilled at using it.	1	2	3	4	5
2	**Familiarity**					
		Slightly				**Strongly**
	a. Was your prior knowledge of other computer based systems useful in the learning of the CASE tool?	1	2	3	4	5
	b. Was your prior knowledge of other CASE tools useful in the learning of the CASE tool?	1	2	3	4	5
3	**Consistency**					
		Poor				**Excellent**
	a. The tool is consistently designed, thus making it easier for me to do my work.	1	2	3	4	5
	b. I find that the same function keys are used throughout the program for the same functions.	1	2	3	4	5
4	**Predictability**					
		Poor				**Excellent**
	a. The tool behaves similarly and predictably in similar situations.	1	2	3	4	5
	b. When executing functions, I get results that are predictable.	1	2	3	4	5
5	**Informative Feedback**					
		Poor				**Excellent**
	a. Animated cursors keep you informed.	1	2	3	4	5
	b. Performing an operation leads to a predictable result.	1	2	3	4	5
6	**Error Messages**					
		Poor				**Excellent**
	a. If I make a mistake while performing a task, I can easily undo the last operation.	1	2	3	4	5
	b. Error messages clarify the problem.	1	2	3	4	5
	c. I perceive the error messages as helpful.	1	2	3	4	5

Figure 3-5 Post-test learnability questionnaire (Senapathi 2005)

3.9.4 Results

The major findings of Senapathi's study were the differences between groups of students who had different levels of general computer experience and previous experience of CASE tools. The results showed that three of the user characteristics had significant effects on learnability, namely:

- Computer experience;
- Previous experience with CASE tools; and
- Attitude and motivation.

Significant differences were found in these two groups for the categories, Ease of Learning, Consistency and Familiarity. The remaining two characteristics, namely, Gender and Number of hours spent did not show a significant effect. Senapathi's study also revealed that all groups rated the Error handling feature of Rational Rose lower than the other features, regardless of their user characteristics (Senapathi 2005). This supported the qualitative results where complaints about error messages were the most frequently reported complaint. It also agrees with other research done relating to error messages which found that the presence of appropriate and relevant feedback, specifically in the context of handling error messages, is considered to have a significant effect on the learnability of a system (Section 3.4.2.1).

3.9.5 Limitations of the framework

The questionnaires completed by the participants satisfy the requirements for good questionnaire design (Preece *et al.* 2007). Both open and closed-ended questions were used with correct rating scales where 1 was the lowest and 5 was the highest rating. This conforms to recommendations for using scales in questionnaires (Preece *et al.* 2002), as well as using the accepted Likert scale of either 5 or 7 points for assessing satisfaction (Te'eni *et al.* 2007). However, some shortcomings with regard to the post-test learnability questionnaire and the background questionnaire were found.

The background demographic questionnaire included some user characteristics of the participants, which form part of the contextual factors (or context of use) aspect of Senapathi's framework. Questions which required details of the participant's computer experience were therefore included. However, more user characteristics such as frequency of use are recommended (Te'eni *et al.* 2007), specifically for the type of application being evaluated. Similarly, details relating to CASE tool experience should include frequency of use and details regarding the participants' use of CASE tools outside of the class environment (Te'eni *et al.* 2007).

An additional user characteristic, namely home language, should be included in the background questionnaire, particularly in a country like South Africa where the home language of most of the participants is probably not the language of the CASE tool. This may have a significant effect on the learnability of the CASE tools, since users who were raised with different home languages from English or have different cultures may have different preferences (Shneiderman and Plaisant 2005). The academic performance of the students was not taken into account; therefore, the intelligence effect may have influenced the results of the study.

Minor shortcomings in the post-test learnability questionnaire (Figure 3-5) were identified as well as gaps between Senapathi's framework and the questionnaire. Shortcomings in the learnability questionnaire related to the way in which certain questions were asked. The two questions in the Familiarity section require Y*es/No* answers and since a rating of 1 to 5 must be selected, this could be confusing to the respondents and cause them to make an incorrect selection. Mono-operational bias, i.e. measuring constructs using only a single item or question should be avoided and scales or indexes should rather be used (Mouton 2001).

Other problems with the questionnaire related to the unsuitability of a question for the associated attribute which it was supposed to measure. Question 5 (b) *'Performing an operation leads to a predictable result'*, relates to Predictability, since predictability is the ease with which a user can determine the effects of future interactions based on past interaction history (Section 3.4.1.3). It should therefore not be included under the heading Informative Feedback. The last measurable attribute of learnability in the framework, namely, On-line help, does not have any related questions in the closed-ended part of the questionnaire (Section 3.9.3 and Figure 3-5).

Gaps between Senapathi's framework and the research design were found. The questionnaires did not allow for testing the section of the framework relating to tasks and tools (Figure 3-3). The Tasks section of the framework is not addressed since the learnability questionnaire was only completed at the end of the course and not after each task was performed with the CASE tool. The experience of the participants' use of the CASE tool for the different tasks may not be the same. For example, a participant may have had more problems trying to draw certain UML diagrams than other diagrams. The differences between the learnability of the CASE tool depending on the task, or the different UML diagrams, would therefore not have been evaluated.

The Tools section of the framework was not addressed as only one CASE tool was evaluated in Senapathi's study. The learnability questionnaire was not designed for comparing two CASE tools and therefore did not include a section for filling in the name of the CASE tool, nor for questions relating to the participants' tool preference. The questionnaire, therefore, does not allow for the proper validation of the Tools section of Senapathi's framework as proposed in Figure 3-3.

Another discrepancy between the framework and the questionnaire was found in that the Ease of learning attribute was not listed in the framework.

The question content for the Ease of Learning questions (questions 1a to 1e) all related to Ease of Learning in general and not to any other learnability attribute. This implies that Ease of Learning should be an attribute on its own in the framework. This is confirmed by other literature (Nielsen 1993; ISO 1995).

The framework does not allow for learnability principles which are recommended for incorporating usability in a quality system (Section 3.2). This lack of learnability principles poses a risk to the validity of the test results as without these the evaluation results are dependent on the evaluators' skills and experience (Nielsen 1995). It is thus recommended that learnability principles be incorporated in the framework, since this could affect the users' rating of Satisfaction.

3.10 Conclusions

Usability was defined as the capability of the software system to be easy to learn, effective to use and enjoyable from a user's perspective, when used under specified conditions (Section 3.2). Usability is regarded as one of the six software quality characteristics which can be used for evaluating the quality of software (Section 3.2).

Several classifications of usability were compared and contrasted (Table 3-1). Based on this comparison, 14 general attributes for usability (including learnability) were identified which are applicable to all software systems (Table 3-2). Since learnability of CASE tools is the primary focus of this research, several attributes of learnability were identified (Section 3.4). In order to evaluate the learnability of a CASE tool, metrics, rating scales, minimum acceptable levels and target levels should be identified for each attribute (Section 3.2). Evaluators should be given usability principles upon which to evaluate the attributes, especially if they are not expert evaluators. Several principles, which should be applied specifically for evaluating the usability and learnability of CASE tools were identified (Section 3.5).

Usability testing was determined to be the most appropriate method for evaluating the usability of CASE tools (Section 3.7). Two frameworks for evaluating CASE tools were then investigated. The first framework investigated was proposed by Phillips *et al.* and can be used to evaluate CASE tool usability, but it does not focus specifically on learnability (Section 3.8). The second framework as proposed by Senapathi focuses on CASE tool learnability, but has several shortcomings (Section 3.9). The main shortcoming relates to the fact that the framework does not take into account any learnability principles that should be used when evaluating the learnability of CASE tools.

In addition, although the learnability attributes are included in the framework, the evaluation details are not included.

Other shortcomings in Senapathi's framework result from problems in the research design used to validate the framework and the scope of the framework. These problems relate to the design of the post-test learnability questionnaire that does not allow for the recording of satisfaction ratings of all the attributes of learnability. The questionnaire also does not support the comparison of two CASE tools (Section 3.9.5).

In order to implement the framework in a South African context, some modifications to the background questionnaire should be made so that the home language of the participant can be recorded, as this could affect his/her satisfaction rating of the learnability of the CASE tool.

Chapter 4 discusses the design of an experiment to evaluate CASE tool learnability at NMMU, so that Senapathi's framework can be validated and extended for a South African context.

Chapter 4 Research design

4.1 Introduction

The goal of this chapter is to discuss the design of an experiment in order to refine and validate Senapathi's framework in a South African context. The empirical evaluation methods that were used in this research project are discussed. An overview of the research hypotheses and the research instruments used is also discussed. The empirical evaluation section describes the usability testing process and the questionnaires used.

4.2 Research methods

Research hypotheses and research instruments can be used to structure a research project (Dee Medley 2001). Research hypotheses are formulated in order to understand the problem that should be addressed. Once the hypotheses are formulated, they should be tested in order to either accept or reject them. The research method used in this study was an empirical usability evaluation and the specific type of method was usability testing, which is the same method that was used by Senapathi (Section 3.9.3).

4.2.1 Research hypotheses

The research question that needs to be addressed in this section is as follows: *How can the learnability of an OO CASE tool be evaluated at NMMU?* (Table 1-1, Question 3). In order to answer this question, several hypotheses were formulated.

The hypotheses were based on the tasks a user typically performs when drawing a UML diagram and are listed in Table 4-1. H_0 is the null hypothesis. None of the user characteristics used in Senapathi's research was investigated in this study. This study aims to extend Senapathi's research by investigating the Tool aspect of Senapathi's framework (Figure 3-3). In addition, the relationship between two user characteristics, namely home language and frequency of computer use, and CASE tool learnability was investigated. The reason that these characteristics were investigated in this study was because they were identified as relevant to computing education in South Africa (Section 3.9.5), and were not tested by Senapathi.

Since Senapathi's study included the user characteristics of gender, general computer experience, previous experience with CASE tools as well as attitude and motivation, it was deemed unnecessary to repeat the investigation of these characteristics.

The following hypotheses were formulated for examination and tested for significance at the 95% significance level ($\alpha = .05$):

H_0: No relationship exists between the learnability of a CASE tool at NMMU and the context of use.

H_1: A relationship exists between the learnability of a CASE tool at NMMU and the context of use.

Consequently, the null hypothesis (H_0) above is refined to produce the sub-hypotheses listed below in Table 4-1.

$H_{0.1}$		No learnability problems exist at NMMU in either of the selected CASE tools.
	$H_{0.1.1}$	No learnability problems exist at NMMU with Microsoft Visio.
	$H_{0.2.1}$	No learnability problems exist at NMMU with Rational Software Modeller.
$H_{0.2}$		No relationship exists between the learnability of a CASE tool at NMMU and the tool used.
$H_{0.3}$		No relationship exists between the learnability of a CASE tool at NMMU and the user characteristics.
	$H_{0.3.1}$	No relationship exists between the learnability of a CASE tool at NMMU and the user's home language.
	$H_{0.3.2}$	No relationship exists between the learnability of a CASE tool at NMMU and the user's frequency of computer use.

Table 4-1 Research hypotheses

The learning environment section of the framework was excluded from this study since the focus of this research is on the learnability of CASE tools and not pedagogical issues. Thus, two of the four aspects of the context of use section of the framework as proposed by Senapathi (Figure 3-3), were tested using these hypotheses, namely the User and the Tool.

4.2.2 Research instruments

Questionnaires were used as the research instruments for collecting data, as these were the instruments used by Senapathi. Both qualitative and quantitative data was recorded by means of questionnaires. Questionnaires were used to gather background information on the participants as well as user satisfaction of the CASE tools once the evaluations were completed (Section 4.4.4).

One way of understanding the usability problems associated with software is to perform a usability evaluation (Section 3.8). Usability testing was considered as an appropriate method for this research since this was the approach used by Senapathi (Section 3.9.3) and one of the goals of this research was to validate Senapathi's framework (Section 1.7).

4.3 Empirical evaluation

The goal of an empirical evaluation is to discover whether and how a system meets its usability requirements, and to develop suggestions for improving the design (Rosson *et al.* 2002). The empirical evaluation consisted of informal usability testing of two different tasks, using two different CASE tools and three sets of questionnaires. The CASE tool evaluations were scheduled during the students' normal practical sessions in the computer laboratories used for these practical sessions.

An important concern in any test with human participants is that they be treated fairly (Rosson *et al.* 2002). It is necessary to obtain informed consent from each participant. The study's goals and procedures must be summarised for each participant, any questions should be answered, and the participants must be asked to sign a form affirming that they are participating voluntarily and can withdraw at any time without penalty. This procedure was followed at the first week's session and each participant was given an NMMU Ethics Preamble letter and an NMMU Informed Consent form to sign to indicate that his/her consent was voluntary. Copies of these forms can be found in Appendix E and Appendix F respectively.

The following sections discuss the tool and participant selection, the task design as well as the design and use of each of the evaluation instruments.

4.3.1 Tool selection

Hypothesis $H_{0.2}$ in Table 4-1 states that no relationship exists between the CASE tool used and the learnability of those tools. In order to test this hypothesis, the research design included a comparison of two different CASE tools. The two tools selected were of different complexity in order to evaluate this hypothesis. A comparative evaluation of the features of CASE tools was performed at NMMU in 2003 (Scholtz 2003). As a result of this study, Microsoft Visio was selected as the CASE tool to be used by computing students at NMMU as from 2004. Based on these results, Microsoft's Visio was selected as the first tool to be used in this study. IBM's Rational Software Modeller was selected as the more complex tool, as it has more advanced features than Visio (Scholtz 2003). Other studies have shown that Visio is currently the preferred tool of choice for systems modelling (Davies, Green, Rosemann, Indulska and Gallo 2006).

4.3.2 Participant selection

For a usability test, the participants should be representative of the actual users, and for this reason the participants selected were students of the Information Systems II module (WRI201) at NMMU in 2006. All students registered for this module were invited to take part in the study. These students attend practical sessions every alternate week and come to one of two sessions; either on a Monday or a Tuesday. The main aim of these sessions is to demonstrate the concepts of UML and CASE tools taught in the lecture sessions. One of the learning outcomes of this module is for students to be able to draw UML diagrams using a CASE tool. Sixty-two students of the Information Systems II (WRI201) module agreed to take part in the study. The participants were not compensated for their participation, as the tasks formed part of their regular academic activities.

Stratified sampling was used to divide the participants into two groups, namely, the Visio-Rational (VR) group and the Rational-Visio (RV) group. The group names were based on the order in which the tools were evaluated. The VR group evaluated Visio first and Rational second, whereas the RV group evaluated Rational first and Visio second. The students were randomly assigned to each group, while making sure that the proportion of both gender and academic performance per group was representative of the actual student population. Both groups were taught the same material and given the same instructions and information regarding the CASE tools and tasks. All the participants had no previous experience of CASE tools prior to starting the experiment. Table 4-2 shows the demographic profile of the participants according to gender and academic performance.

A graphical representation of the gender profile is depicted in Figure 4-1 and the academic performance profile in Figure 4-2. For the purpose of the research analysis, the 62 students were divided into two different strata, namely Average and Above Average. Average included all students with a mark of less than or equal to 65% for their End User Computing (WRU102) module in 2005, and Above Average included all students with marks greater than 65%. The two groups had approximately the same proportion of male and female students (Figure 4-1). The majority of both groups were male (68% for the VR group and 77% for the RV group).

	Group					
	VR		**RV**		**TOTAL**	
	n	**%**	**n**	**%**	**n**	**%**
Gender						
Male	21	68	24	77	45	73
Female	10	32	7	23	17	27
TOTAL (n=62)	**31**	**100**	**31**	**100**	**62**	**100**
Academic Performance						
Average (<= 65%)	19	61	19	61	38	61
Above average (> 65 %)	12	39	12	39	24	39
TOTAL (n=62)	**31**	**100**	**31**	**100**	**62**	**100**

Table 4-2 Demographic profile of selected participants

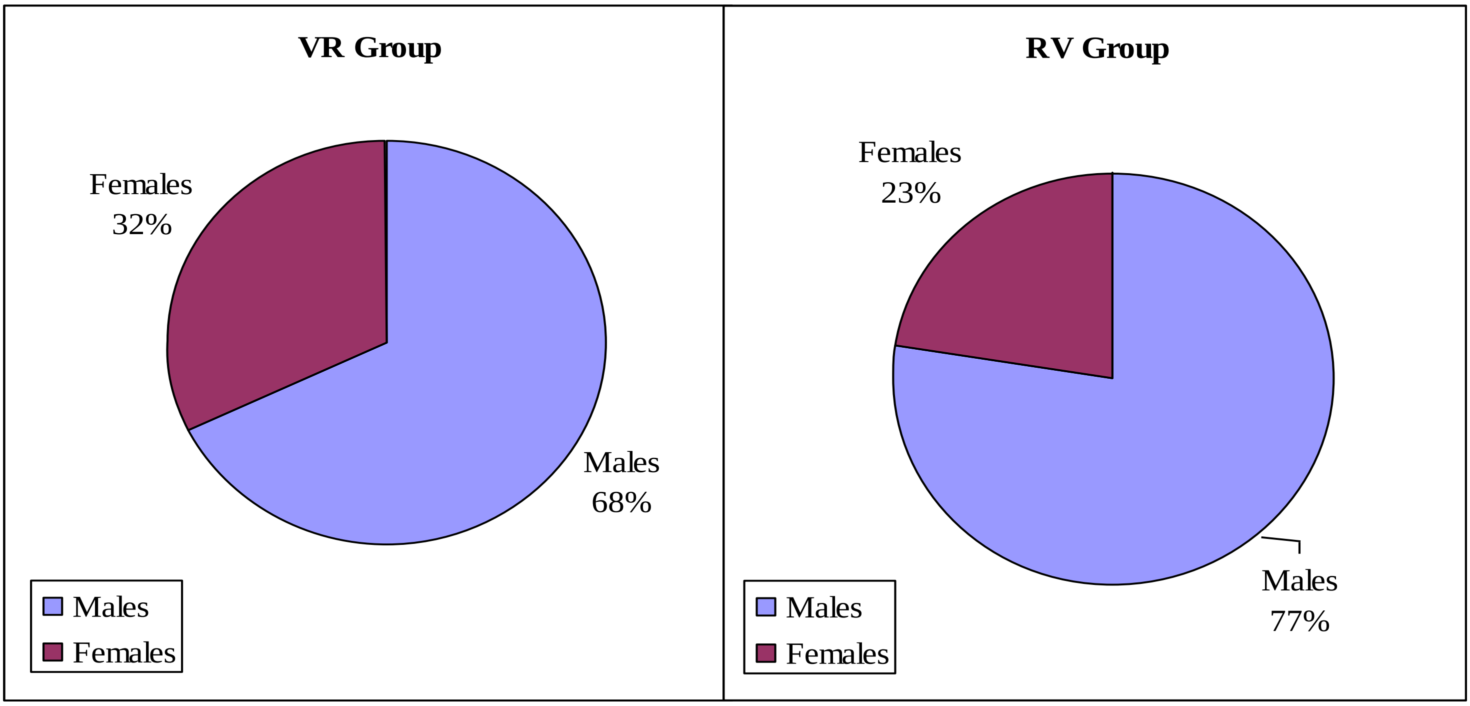

Figure 4-1 Gender profile of selected participants (n = 62)

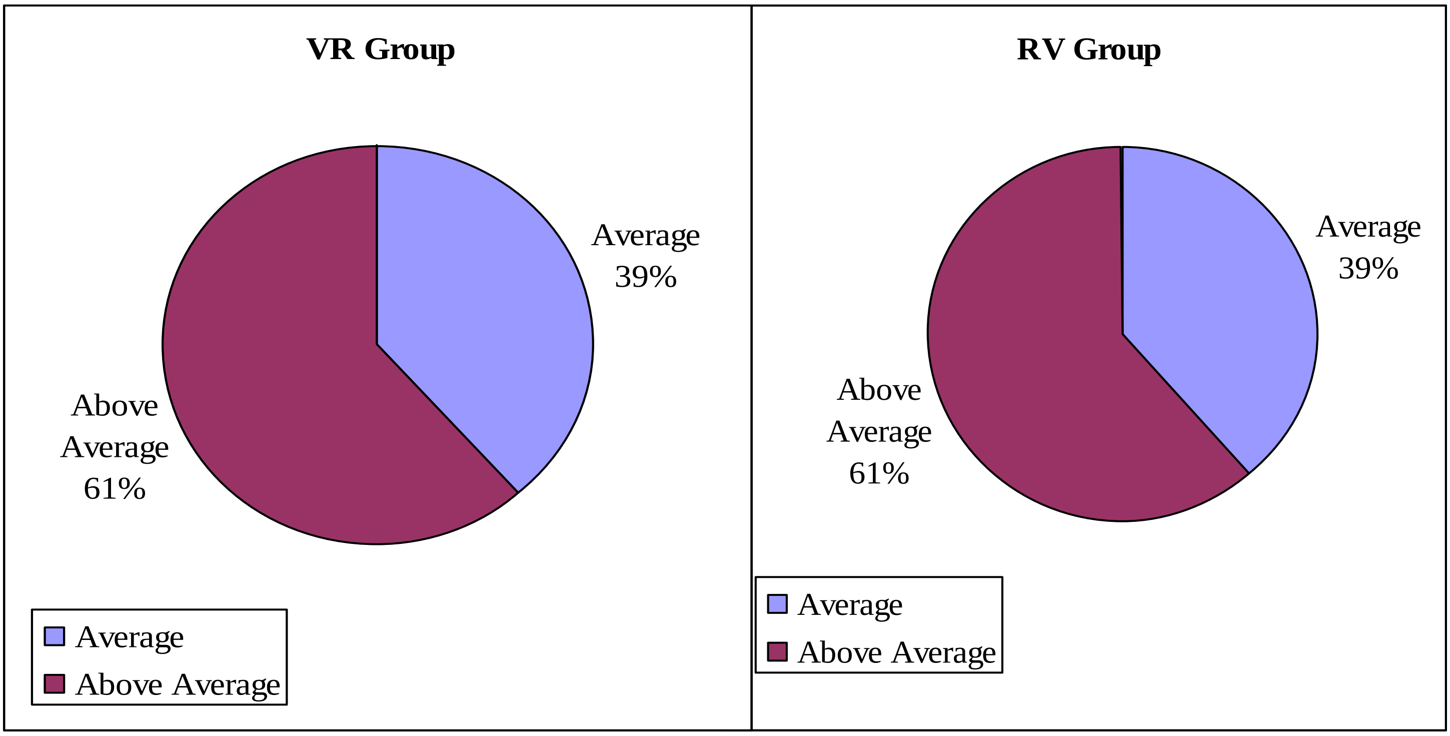

Figure 4-2 Academic performance profile of selected participants (n = 62)

4.3.3 Test tasks

Each participant was required to perform two tasks and therefore two different sessions were held on different days so that only one task had to be performed at each session. This was designed so that the total task time for each task was limited to the recommended one to one and a half hours per task (Barnum 2002). Each task consisted of drawing one UML diagram in each of the two CASE tools. The tasks selected were designed to closely match the typical tasks required in the WRI201 module. The diagrams which the participants were required to draw in each task were taken from an actual case study from the WRI201 module in 2006.

The two tasks selected for the participants to perform were as follows:

- Task A: Draw a UML use case diagram; and
- Task B: Draw a UML class diagram.

These tasks were selected as they are the two key UML diagrams taught to WRI201 students. Task instructions and a copy of the diagram which had to be drawn were given to the participants.

Copies of Task A and Task B, as given to the VR group, are included in Appendix A and Appendix B respectively. The tasks were identical for both groups, except for the order of the CASE tools used. Performance can be affected by the order in which the systems are presented to the subjects; this is called the order effect (Faulkner 2000). The subjects could get better at the task as the task is repeated, and this is known as the practice effect. On the other hand, they may get tired and do worse. This is known as the fatigue effect. For this reason, the order in which the CASE tools were used was varied. The VR group used Visio first and the RV group used Rational first. The experimental design method used was therefore a counterbalanced design and is illustrated in Table 4-3. With counterbalancing, each participant works under both conditions and the possible effects of learning from the first task is neutralised (Preece *et al.* 2007). The same experimental design was used for both tasks.

Each diagram comprised two versions; one with an error and one with the corrected diagram. The reason for this was to enable the participants to experience drawing a UML diagram which had an error, since in order to detect a problem the users need to experience a problem (Section 4.3.2). This would allow for the Error Handling feature of the CASE tools to be evaluated, which was identified by Senapathi as one of the attributes contributing to CASE tool learnability (Section 3.9.2.2).

Before the participants performed the task in the required CASE tool, a Getting Started Guide for the specific CASE tool which they were about to use, was provided. This guide comprised a brief two-page overview of the system and an introduction to its key functions.

Group	**Session Details**	
	First Tool	**Second Tool**
VR	1. Getting Started with Microsoft Visio 2. Draw diagram with Microsoft Visio	1. Getting Started with Rational Software Modeller 2. Draw diagram with Rational Software Modeller
RV	1. Getting Started with Rational Software Modeller 2. Draw diagram with Rational Software Modeller	1. Getting Started with Microsoft Visio 2. Draw diagram with Microsoft Visio

Table 4-3 Experimental design for both tasks

During the usability evaluation, the tasks performed by the participants were monitored by a test monitor and intervention was only allowed if the participant either asked for assistance or could not complete a task without assistance.

4.3.4 Evaluation instruments

Questionnaires were used as evaluation instruments for this research for two reasons. Firstly, questionnaire data is regarded as reliable and valid for assessment of user satisfaction (Mouton 2001). Secondly, this was the method used by Senapathi, and one of the purposes of this study was to verify and validate Senapathi's research (Section 3.9.3).

Subjective data, both qualitative and quantitative, was recorded using questionnaires, since the objective was to measure the satisfaction of the users with the CASE tools. The questionnaires were based on the ones used in Senapathi's research (Section 3.9.3), and included a background demographic questionnaire, as well as two post-test questionnaires. The two post-test questionnaires used were:

- A learnability questionnaire (Appendix D); and
- An attitude and motivation questionnaire (Figure 3-4).

The following sections discuss the different questionnaires, their objectives and design.

4.3.4.1 Background questionnaire

The main purpose of the background questionnaire was to gather biographical data and other user characteristics required for testing the hypotheses. The questionnaire requested the following information:

- Student number and name;
- Degree;
- Gender;
- Number of years using a computer;
- Number of years using a CASE tool (if any);
- Frequency of use of CASE tool;
- Name of CASE tool(s) used;
- Purpose of using CASE tool(s);
- Frequency of computer use;
- Study of computers at school; and
- Home language.

Five additional user characteristics were added to the original questionnaire used by Senapathi, based on the shortcomings identified (Section 3.9.5). These are the last five characteristics in the list. The questions relating to prior experience of CASE tools was required in order to ensure that none of the participants had previously worked on a CASE tool, as this could skew the results of the study. A copy of the modified background questionnaire given to the participants can be found in Appendix C.

4.3.4.2 Post-test learnability questionnaire

The post-test learnability questionnaire was used to determine the satisfaction of the participants with the different CASE tools used, and was based on the one used by Senapathi. Some minor modifications were made based on the gaps identified (Section 3.9.5) and a copy of the revised questionnaire is included in Appendix D. The modifications made to the questionnaire included the following:

- The name of the CASE tool used was recorded (since two CASE tools were evaluated);
- Questions 2(a) and 2(b) were reworded so as to remove confusion and require a rating rather than a *Yes/No* answer;
- Question 5(b) was removed as it was a duplication of Question 4 (b); and
- A new section was added, namely Section 4, where the participant could record their CASE tool preferences.

The modified questions 2(a) and 2(b) are shown in Figure 4-3.

2	**Familiarity**					
		Slightly				**Strongly**
	a. My prior knowledge of other computer based systems was useful in the learning of the CASE tool.	1	2	3	4	5
	b. My prior knowledge of other CASE tools was useful in the learning of the CASE tool.	1	2	3	4	5

Figure 4-3 Modifications to Question 2 of the post-test learnability questionnaire (see Figure 3-5)

The attributes that were taken into account were those identified by Senapathi as contributing to learnability (Section 3.9.2.2). Satisfaction measures were recorded per CASE tool, using a 5-point rating scale, where 1 represented *"Poor"* and 5 represented *"Excellent"*. The learnability questionnaire also included open-ended questions so that qualitative data could be obtained.

Each participant had to complete two learnability questionnaires, one for each CASE tool used. The two questionnaires were identical, except that the one which had to be completed after the second CASE tool had an additional section, namely *Section 4,* where the participant was asked to rate:

- Which CASE tool was easier to use and why; and
- Which CASE tool was easier to learn and why.

Section 1 of the learnability questionnaire comprised questions relating to each of the attributes affecting learnability (Section 3.9.2.2). Section 2 included a section for recording qualitative information with open-ended questions relating to what the participants liked about the CASE tool and what they did not like. Section 3 consisted of six questions and recorded ratings related to overall satisfaction.

4.3.4.3 Attitude and motivation questionnaire

The attitude and motivation questionnaire used was based on the one used by Senapathi (Figure 3-4) and consisted of the same set of five questions. Students were asked to rate the way they felt about the use of the CASE tool in their course by means of this set of questions. A Likert scale of 1 to 5 where 1 = *"Poor"* and 5 = *"Excellent"* was used for these questions.

4.4 Data capturing, collection and editing

A sequential number was recorded on each questionnaire, in order to check for missing questionnaires or duplicate questionnaires. These were then captured into a Microsoft Excel spreadsheet, cross-checked and verified by an independent person and analysed for inconsistencies and errors.

4.5 Data analysis techniques

4.5.1 Quantitative techniques

There are many statistical tests that can be used to assess the significance of the patterns observed in usability testing. Statistics such as totals, means and standard deviations of the responses to the questions in the post-test questionnaires were calculated. STATISTICA V7.0 (StatSoft Inc, 2001), a data analysis software package, was used for exploratory data analysis. In particular, STATISTICA was used to compute the standard descriptive statistics such as means and standard deviation.

The study explored the relationships between two user characteristics (home language and frequency of computer use), two context of use factors (the CASE tool and the task), and the perceived learnability of the CASE tools.

Sequence effects were checked for, in order to detect if there was a difference between using one CASE tool first and the same CASE tool second. Outliers were also identified and removed where necessary. Repeated Analysis of Variance (ANOVA) tests were used to conduct statistical analyses on participants' ratings. These methods were selected as appropriate, since they were used for Senapathi's study upon which this work is based (Section 3.9.3).

4.5.2 Qualitative techniques

Qualitative data was collected from the open-ended questions in the learnability questionnaires and from observation of the participants during the tasks. As the qualitative data was collected, it was coded and structured into categories (Ely, Vinz, Downing and Anzul 1999). This process served to organise the qualitative data into a meaningful context.

The final analysis of qualitative data involved the search for and determination of themes from the identified categories. A theme is defined as a statement of meaning that (Ely, Anzul, Friedman, Garner and MacCormack Steinmetz 1995; Ely *et al.* 1999):

- Runs through all or most of the pertinent data; or
- Carries heavy emotional or factual impact.

Themes can be identified as the explicit or implied attitudes towards an observed behaviour. Thematic analysis was consequently used to present the findings assembled from the various qualitative sections of the questionnaires administered. The themes identified were based on the learnability attributes for CASE tools identified (Table 3-6). The thematic analysis was complemented by frequency counts in order to compare the quantitative results in the identified categories and themes.

4.6 Shortcomings and risks

One of the risks of this study is the sample size, which although it exceeds the recommended size for a usability test (5-8), is still rather small (n=62). Tight experimental controls can, however, warrant valid statistical analysis with samples as small as 15 per group (Applin 2001).

It was, therefore, important to ensure that extraneous variables, such as previous training and experience of the participants, were controlled and kept constant, since these could affect the dependent variables. For this reason, a process of participant allocation that initially maximised and equalised the population sizes of the strata from which the samples were drawn for each group, was followed.

One of the shortcomings of this study was that, due to time constraints, the tasks had to be limited to the drawing of two UML diagrams. In addition the tasks were performed by computing students in the computer laboratories at NMMU and, therefore the results may only be applicable to users in a similar educational context.

Another risk of this study is the limited expertise of the participants selected for the usability evaluation. None of the participants were HCI experts and may not have a full understanding of the questions posed in the post-test learnability questionnaire, particularly as Senapathi's framework did not include any learnability requirements. This could possibly skew the results (Section 3.9.5).

4.7 Conclusions

The fourth research question for this study related to identifying what learnability problems occur with OO CASE tools used for computing education at NMMU (Section 4.2.1). In order to answer this question several research hypotheses were identified relating to the relationship between CASE tool learnability and various independent variables (Table 4-1). In order to test these hypotheses an experiment was performed. The experiment included a usability evaluation of two CASE tools at NMMU.

Usability testing was considered the most appropriate method for this research since this was the approach used by Senapathi (Section 3.9.3) and one of the goals of this research was to validate Senapathi's framework (Section 1.7). The independent variables selected were home language, frequency of computer use, the CASE tool used and the task since these were not tested by Senapathi (Section 4.2.1).

Sixty-two WRI201 students agreed to participate in the study and stratified sampling was used to divide these students into two equivalent groups (Section 4.3.2). In order to test the hypothesis regarding the relationship between the CASE tool used and learnability, two CASE tools were selected (Section 4.3.1).

The CASE tools selected for the study were Microsoft Visio and Rational Rose (Section 4.3.1). The tasks the participants were required to perform involved the drawing of two UML diagrams, which are key tasks in the WRI201 module (Section 4.3.3). For Task A, the participants were required to draw a UML use case diagram, and for Task B, a UML class diagram.

The participants were required to complete several questionnaires, which were used as evaluation instruments for this research (Section 4.3.4). Prior to commencing either task, a background questionnaire was completed by each participant. After each task, the participants completed a post-test learnability questionnaire. On completion of both tasks, an attitude and motivation questionnaire was completed (Section 4.3.4.3).

The data analysis techniques used to analyse the quantitative data included elementary descriptive statistics, such as means and standard deviations (Section 4.5). The qualitative data collected from responses to the open-ended questions were coded and structured into categories. Themes were then identified from these categories and matched with one of the relevant usability and learnability attributes identified (Table 3-6). Frequency counts for each theme were calculated in order to determine the most common themes. The research results obtained from the empirical evaluation are discussed in Chapter 5.

Chapter 5 Research results

5.1 Introduction

Chapter 4 outlined the research design. The research method consisted of an empirical usability evaluation and resulted in qualitative and quantitative data. The primary objective of this chapter is to report on the process of participant and task selection as well as the results obtained during the empirical evaluation.

5.2 Dependent and independent variables

The dependent variable for this study was the users' satisfaction rating of the attributes comprising learnability proposed in Senapathi's framework (Section 3.9.2.2), namely:

- Learnability – Ease of learning, Familiarity, Consistency, Predictability;
- Informative feedback;
- Error handling; and
- On-line help.

Four independent variables were identified as being relevant for the current investigation (Section 4.2.1), and were included in the background questionnaire. These variables were selected in order to test the hypotheses listed in Table 4-1 and were the CASE tool (H_1), the task (H_3), home language ($H_{2.1}$), and frequency of computer use ($H_{2.2}$).

5.3 Selection of participants

Of the 62 students who volunteered for the study, only 46 participants and 92 post-test questionnaires could be included in the data analysis. The data for 16 participants had to be removed from the data analysis, due to incomplete questionnaires or incomplete tasks. A breakdown of the number of participants whose data was removed and the reasons for removal are shown in Table 5-1.

Reason	**n**
Participants who used only one of the CASE tools for a task, or completed only one task (This includes participants who left out a large percentage of the questions)	12
Participants who did not complete the WRI201 module	4
Total participants removed	**16**

Table-5-1 Number of participants removed from data analysis

A breakdown of the demographic profile of the 46 participants who participated in both tasks (and were therefore included in the data analysis), is shown in Table 5-2.

	Group					
	VR		RV		TOTAL	
	n	%	n	%	n	%
Gender						
Male	16	67	16	73	**32**	**70**
Female	8	33	6	27	**14**	**30**
TOTAL (n= 46)	**24**	**100**	**22**	**100**	**46**	**100**
Academic Performance						
Average (<= 65%)	13	54	14	64	**27**	**59**
Above average (> 65%)	11	46	8	36	**19**	**41**
TOTAL (n=46)	**24**	**100**	**22**	**100**	**46**	**100**

Table 5-2 Demographic profile of actual participants

The gender profile of these participants is shown graphically in Figure 5-1, and the profile of their academic performance is illustrated in Figure 5-2. The additional participant information gathered from the background questionnaires (Section 4.4.4.1) is given in Table 5-3.

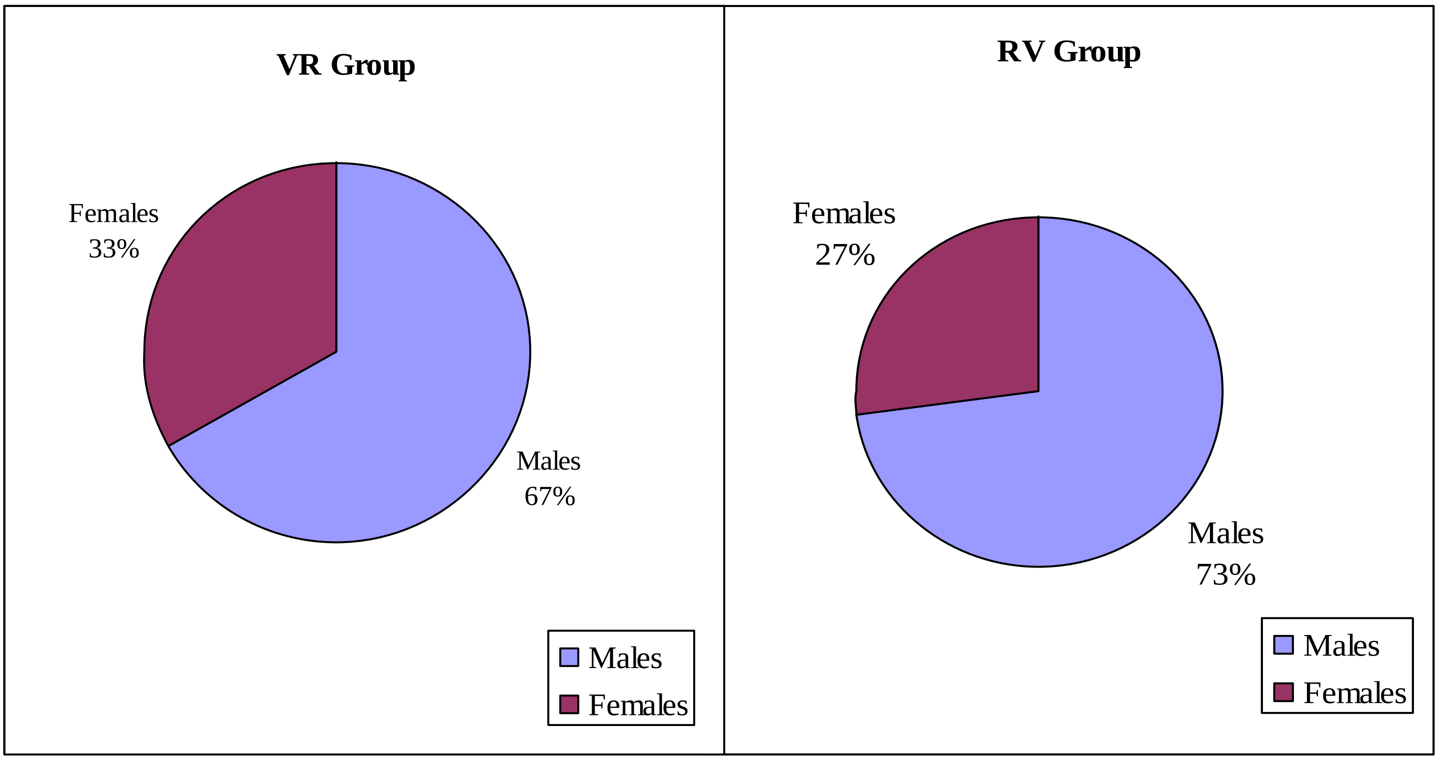

Figure 5-1 Gender profile of actual participants (n = 46)

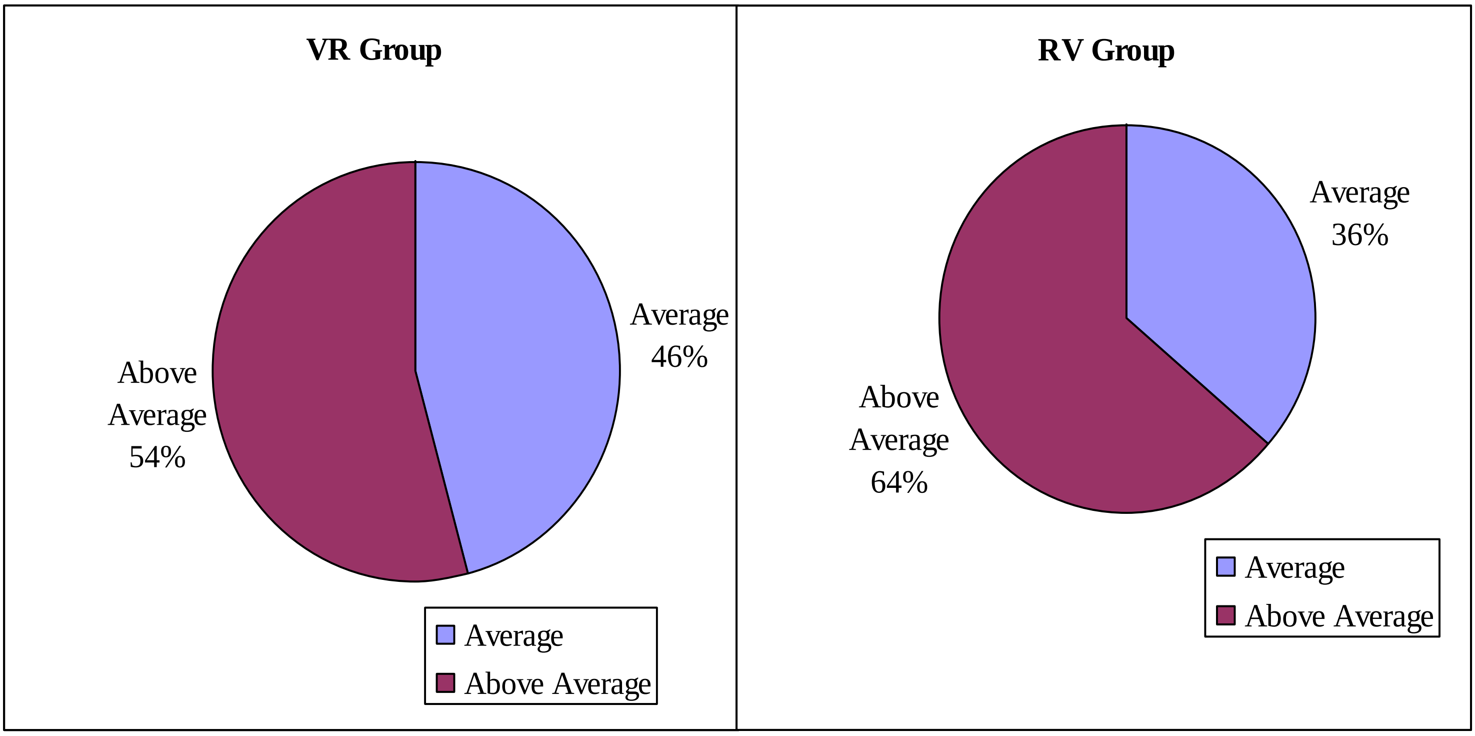

Figure 5-2 Academic performance profile of actual participants (n = 46)

Additional Demographic Information		n	%
Home Language	English	13	28
	Afrikaans	12	26
	Other African Languages	21	46
	TOTAL	**46**	**100**
General Computer Experience	Less than 3 Years	6	13
	3 – 5 Years	6	13
	6 – 10 Years	23	50
	More than 10 Years	10	22
	Missing	1	2
	TOTAL	**46**	**100**
Frequency of Computer Use	Very Low (< 5 Hours per week)	3	7
	Low (5 – 10 Hours per week)	18	39
	High (11 – 20 Hrs per week)	12	26
	Very High (> 20 Hours per week)	12	26
	Missing	1	2
	TOTAL	**46**	**100**

Table 5-3 Additional background information of actual participants

A graphical representation of three user characteristics, namely home language, general computer experience and frequency of computer use, is illustrated in Figures 5-3, 5-4 and 5-5 respectively. None of the participants had any prior experience with CASE tools which meant that no one had to be eliminated from the study for this reason (Section 4.3.4.1).

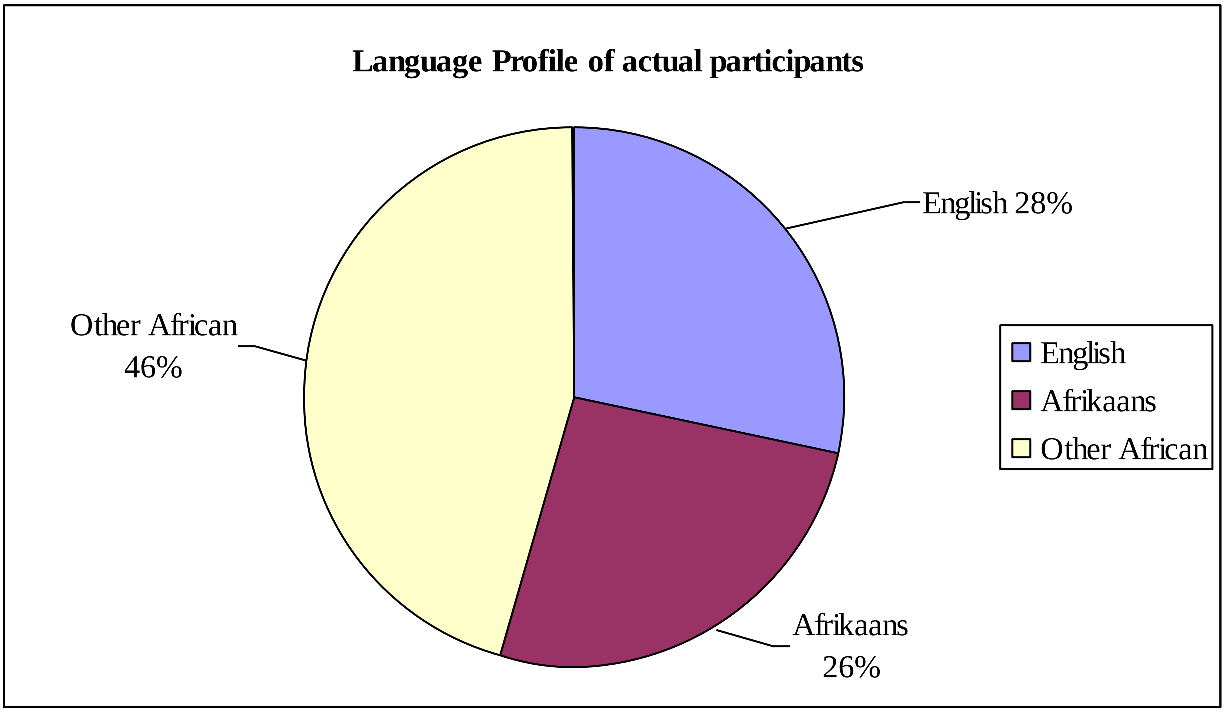

Figure 5-3 Language profile of actual participants (n = 46)

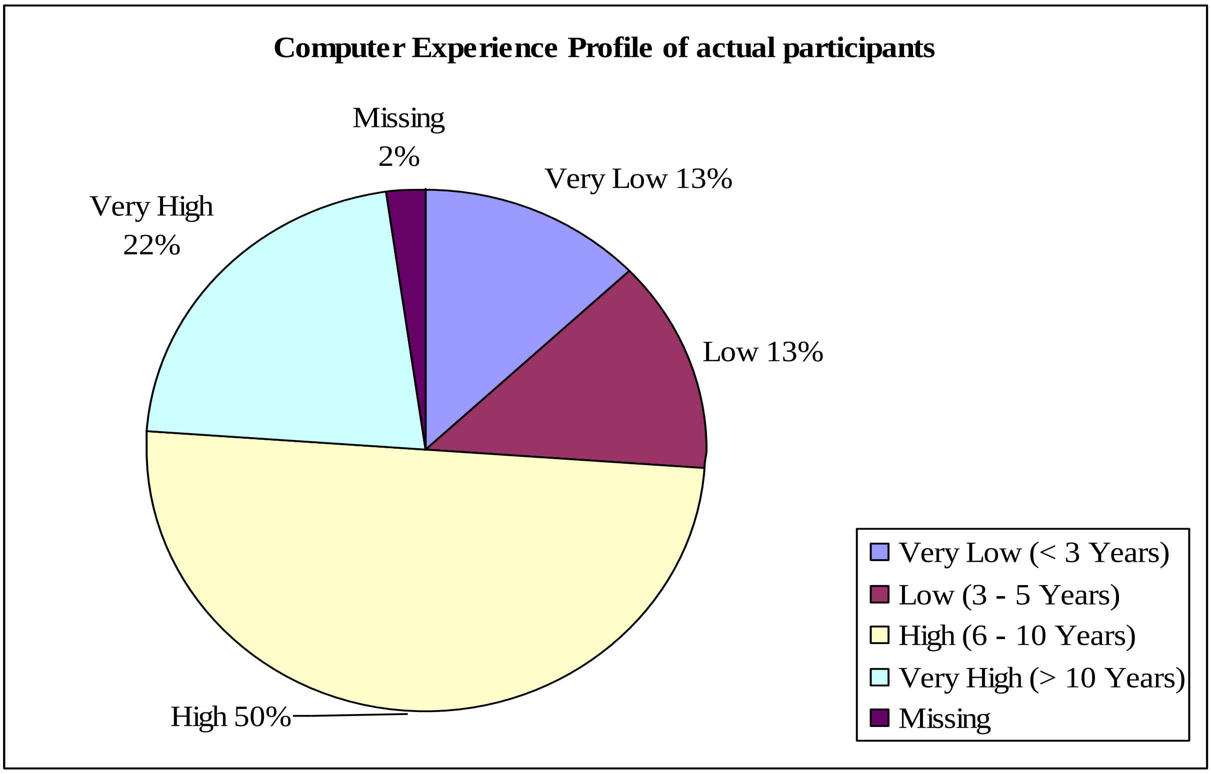

Figure 5-4 General computer experience profile of actual participants (n = 46)

It can be seen from Table 5-3 that only 28% of the participants indicated that English was their home language. This provides further justification for the need to test for a relationship between home language and CASE tool learnability.

The majority of the participants (72%) indicated that they had a High or Very High level of computer experience (Figure 5-4). This provides further justification for the decision not to investigate the relationship between computer experience and CASE tool learnability.

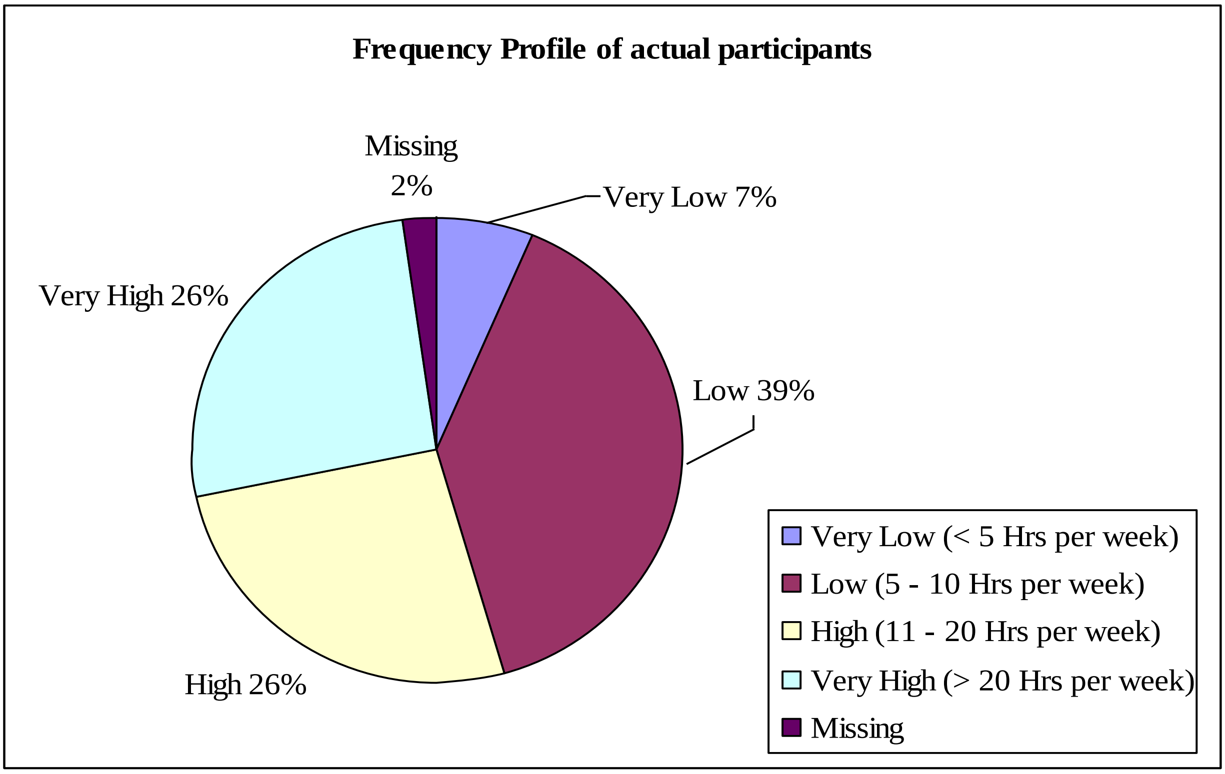

Figure 5-5 Frequency profile of actual participants (n = 46)

Only half (52%) of the participants indicated that they had a High or Very High frequency of computer use (Figure 5-5). This implies that there was a reasonably even split of participants using a computer for more than 10 hours per week, and those using a computer for less than or equal to 10 hours per week. This provides further justification for the decision to investigate the relationship between the frequency of computer use and CASE tool learnability.

5.4 Quantitative results

In order to eliminate bias due to demographic influence, the participants of each group (VR and RV), were analysed as a single group in this study.

The reliability and internal consistency of the data obtained from the quantitative responses to the two post-test questionnaires were measured using Chronbach's alpha (Nunnelly 1978). A copy of the Chronbach's alpha coefficients for the mean difference and mean standard deviation of each of these metrics for both questionnaires is shown in Table 5-4.

The values varied from 0.64 to 0.86 for Task A and from 0.68 to 0.87 for Task B, while the most frequent value was 0.86. Informative Feedback was also part of the learnability questionnaire but could not be calculated since it comprised only one question. The attitude and motivation questionnaire was completed after both tasks and is, therefore, not task specific. It can be seen that the inner consistency of the data was moderately valid, since the alpha coefficients fall within the acceptable range (> 0.6).

Cronbach's ∞		
	Task A	**Task B**
LEARNABILITY QUESTIONNAIRE		
SECTION 1 – LEARNABILITY		
Ease of Learning	0.86	0.86
Familiarity	0.76	0.87
Consistency	0.64	0.81
Predictability	0.86	0.83
Error Handling	0.73	0.68
SECTION 3 – OVERALL REACTIONS		
Overall Reactions	0.84	0.82
ATTITUDE AND MOTIVATION QUESTIONNAIRE		
Attitude and Motivation	0.90	

Table 5-4 Results of Chronbach's Alpha test

The next section presents the comparative quantitative analysis of satisfaction ratings recorded for both tasks in the study.

5.4.1 Learnability results

Section 1 of the post-test learnability questionnaire consisted of 15 questions. These questions were grouped into six of the learnability attributes as proposed by Senapathi (Section 3.9.2.2). These were measured with Likert rating scales of 1 to 5. A comparison of the sample sizes, means and standard deviations of each CASE tool for each question in Section 1 is included in Appendix G.

The mean ratings for all questions for each CASE tool for each task is listed in Table 5-5. This table also shows the results of the tests of statistical and practical significance.

For the test of practical significance, Cohen's d statistic was used and for statistical significance, the t-test was used and a p-value of < 0.05 was regarded as significant. The results clearly show that Visio scored higher than Rational for learnability for both tasks, although the difference in mean ratings between the two CASE tools dropped slightly from 0.64 to 0.59. The mean rating for both CASE tools increased slightly from Task A to Task B. This is to be expected since for Task B the participants had already had prior experience with the two CASE tools.

Learnability							
Task	**CASE Tool**	**n**	**Mean**	**SD**	**t-test**	**p-value**	**d**
A	Visio	42	3.63	0.43			
	Rational	42	2.99	0.63			
	Difference		0.64	0.72	5.76	.000	0.89 large
B	Visio	44	3.75	0.43			
	Rational	44	3.16	0.79			
	Difference		0.59	0.88	4.45	.000	0.67 moderate

Table 5-5 Learnability results

The mean ratings for each learnability attribute were calculated per tool and compared for each task. These ratings are included in the tables in Appendix H and shown in graphical format in Figures 5-6, 5-7 and 5-8. All learnability results in this section were checked for both statistical and practical significance.

5.4.1.1 Task A – Use case diagram

For every learnability attribute in the questionnaire, Visio had a higher mean rating than Rational for Task A, except for Familiarity. In Task A, the difference in mean ratings for the two CASE tools for all attributes was of practical and statistical significance, except for Familiarity. For Familiarity, both CASE tools had the same mean rating of 3.3. This was also the highest rated attribute for Rational. The second highest rated attribute for Rational was Error Handling with a mean rating of 3.03. Visio's highest rated attribute was Predictability (with a mean rating of 3.89) and the second highest rating was for Ease of Learning (with a mean rating of 3.82).

The lowest attribute for Rational was Informative Feedback (with a mean rating of 2.74) and the second lowest was Ease of Learning (with a mean rating of 2.77). Visio's lowest rating was for Familiarity (with a mean rating of 3.3) and its second lowest rating was Informative Feedback (with a mean rating of 3.52).

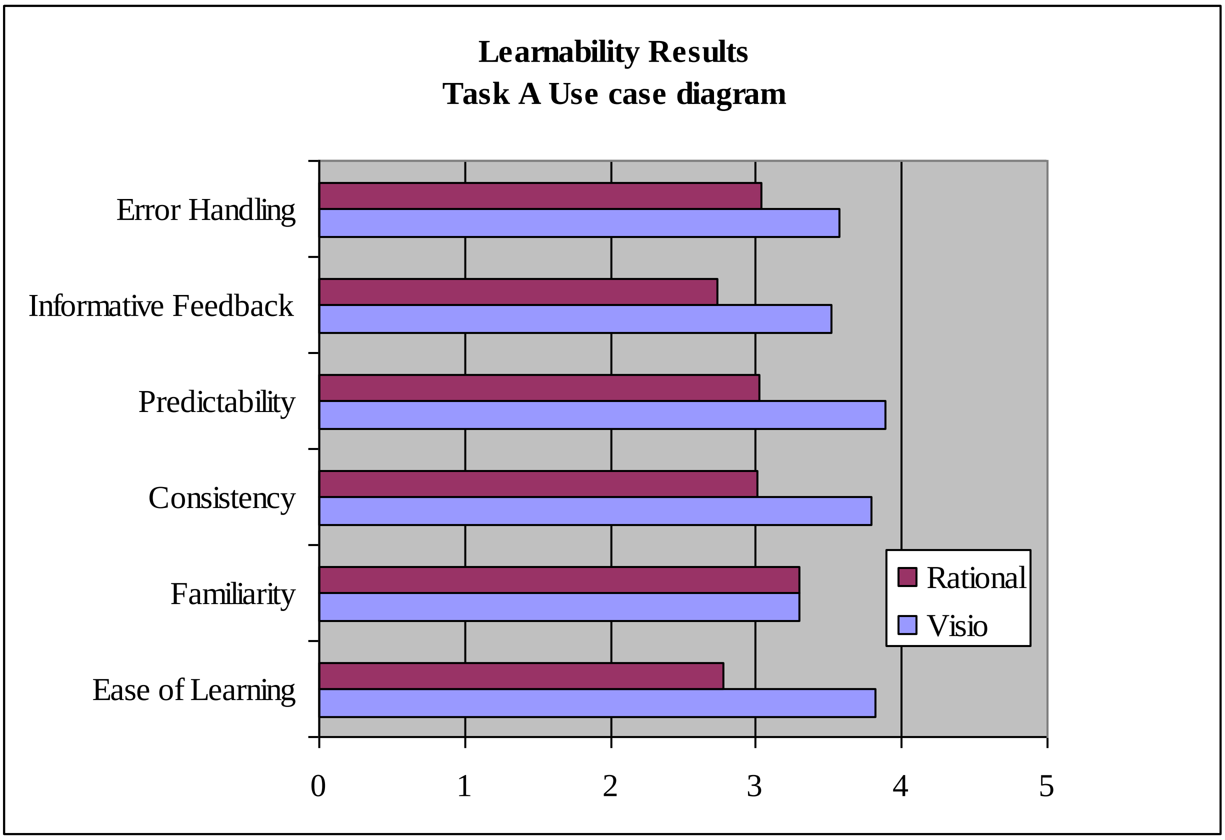

Figure 5-6 Learnability mean ratings for Task A (n = 46)

.

From these results it is evident that for the majority of learnability attributes for Task A, there was a difference between the learnability of the two CASE tools and the participants clearly preferred the learnability of Visio to Rational. It can also be seen that in Task A there was a difference between the preferred attributes of learnability for the two tools, as the highest and lowest rated attributes for the two tools were not the same.

5.4.1.2 Task B – Class diagram

For every attribute of learnability in the questionnaire Visio had a higher mean rating than Rational for Task B. The highest rated attribute for Rational was Error Handling (with a mean rating of 3.27) and the second highest was Familiarity (with a mean rating of 3.22). The highest rated attribute for Visio was Consistency (with a mean rating of 3.95) and Predictability was second highest (with a mean rating of 3.93).

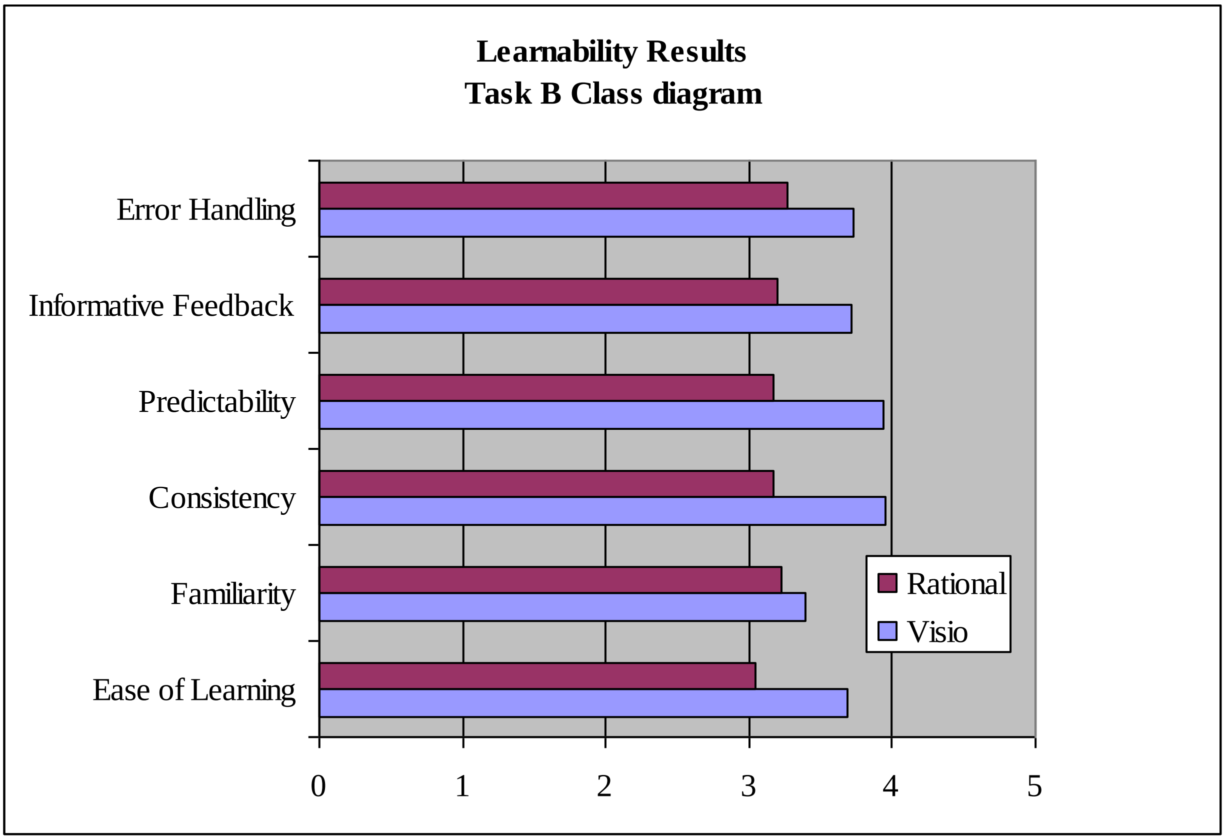

Figure 5-7 Learnability mean ratings for Task B (n = 46)

The lowest rated attribute for Rational was Ease of Learning (with a mean rating of 3.04), and the second lowest was Consistency and Predictability (with a mean rating of 3.17 respectively). Familiarity was the lowest rated attribute for Visio (with a mean rating of 3.39) and the second lowest was Ease of Learning (with a mean rating of 3.69).

For Task B, the differences in mean ratings between the two CASE tools, were of practical and statistical significance, except for Familiarity and Error Handling. It is evident from these results that similar to Task A, Visio was clearly the preferred CASE tool for the majority of learnability attributes. The highest and lowest rated attributes were not the same for the two tools.

5.4.1.3 Both tasks

A graphical representation of the mean ratings per learnability attribute for both tasks is shown in Figure 5-8.

It is evident from these results that Visio scored higher than Rational for every attribute of learnability in the questionnaire. The highest rated attribute for Rational was Familiarity (with a mean rating of 3.27). This was unexpected as the participants had not worked with Rational or any other IBM product before. However they had worked with Microsoft products before so it was expected that Visio would have a high Familiarity rating. This is a concern and leads one to question whether or not the participants truly understood what Familiarity meant or if the underlying questions were incorrectly phrased. Providing a list of clear learnability requirements to the participants could have assisted in reducing any misunderstanding (Section 3.9.5).

The second highest rated attribute for Rational was Error Handling (with a mean rating of 3.14). The highest rated attribute for Visio was Predictability (with a mean rating of 3.91) and the second highest was Consistency (with a mean rating of 3.87).

The lowest rated attribute for Rational was Ease of Learning (with a mean rating of 2.9) and the second lowest was Informative Feedback (with a mean rating of 2.97). Familiarity was the lowest rated sub-section for Visio (with a mean rating of 3.34) and the second lowest was Informative Feedback (with a mean rating of 3.62).

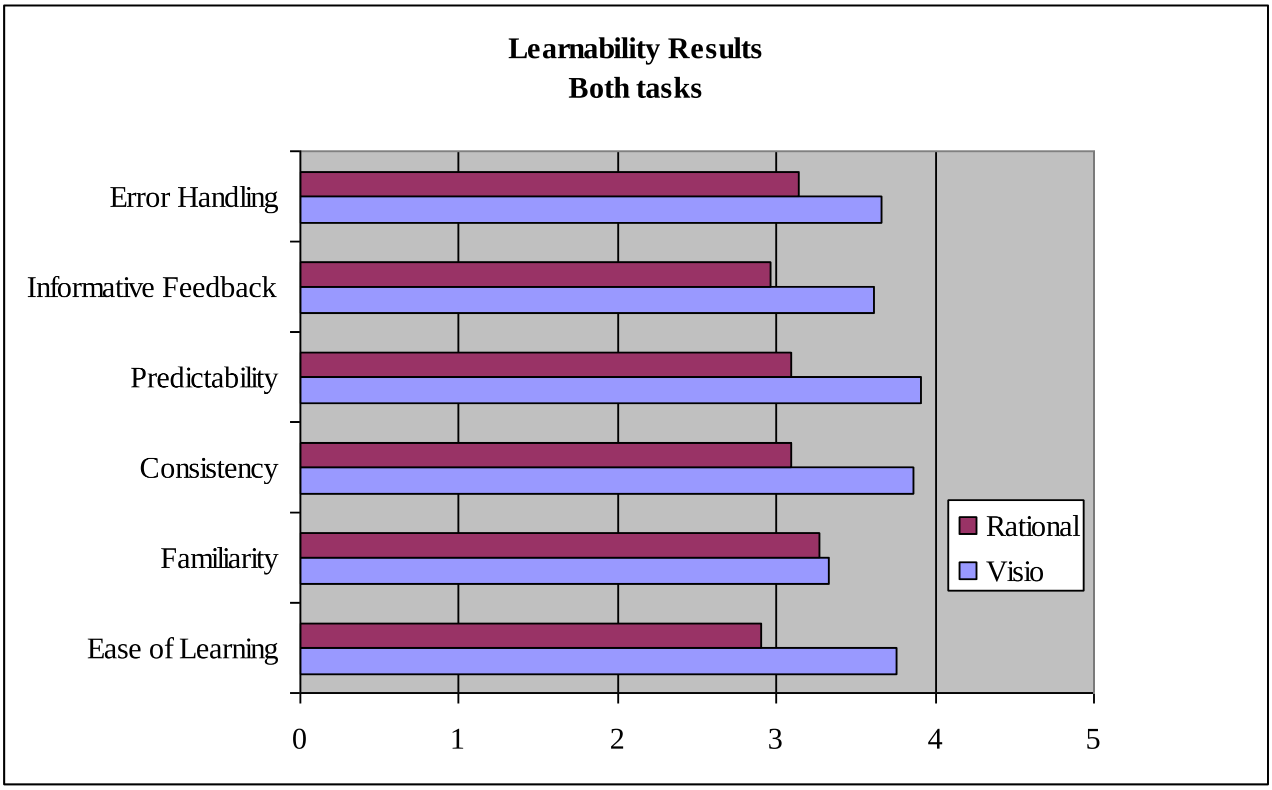

Figure 5-8 Learnability mean ratings for both tasks (n = 46)

When looking at both tasks, the difference in mean ratings between the two CASE tools for all attributes in the learnability questionnaire were of practical and statistical significance, except for Familiarity. Predictability and Consistency were the top two rated attributes for Visio, whereas Error Handling and Familiarity were the top two attributes for Rational.

From the initial descriptive statistics, we can conclude that for both tasks there was a difference in mean ratings between the CASE tool used and that there is a relationship between the CASE tools used and the learnability of the CASE tool. This seems to be true irrespective of the UML task performed. In addition, since all learnability attributes had a mean rating of 2.5 or higher, we can conclude that the learnability of both CASE tools was rated positively.

5.4.2 Overall learnability

The mean rating for all 15 questions for Section 1 of the learnability questionnaire represents the overall learnability of the CASE tools. All the questions in Section 1 had five possible scores (1 to 5), which were divided into three categories for analysis purposes, namely Low, Average and High. Low includes mean ratings between 1 and 2.6 inclusive; Average includes mean ratings greater than 2.6 and less than or equal to 3.4; and High includes mean ratings greater than 3.4. The frequency counts of all scores falling into each of these three categories are included in Appendix I. A graphical representation of these figures is provided for Task A in Figure 5-9 and for Task B in Figure 5-10.

For Task A, 71% of the participants rated the learnability of Visio as High, whereas only 26% of participants rated the learnability of Rational as High. For Task B, the percentage of participants who rated Visio as High increased to 80% (an increase of 9%), while the number of participants who rated Rational as High increased to 36% (an increase of 10%). The number of scores in the Low category for Rational decreased from 26% in Task A to 23% in Task B (a decrease of 3%). The ratings in the Low category for Visio dropped from 2% to 0% (a decrease of 2%). There was a general trend of an improvement in learnability ratings for both tools from Task A to Task B. This is to be expected due to the learning effect.

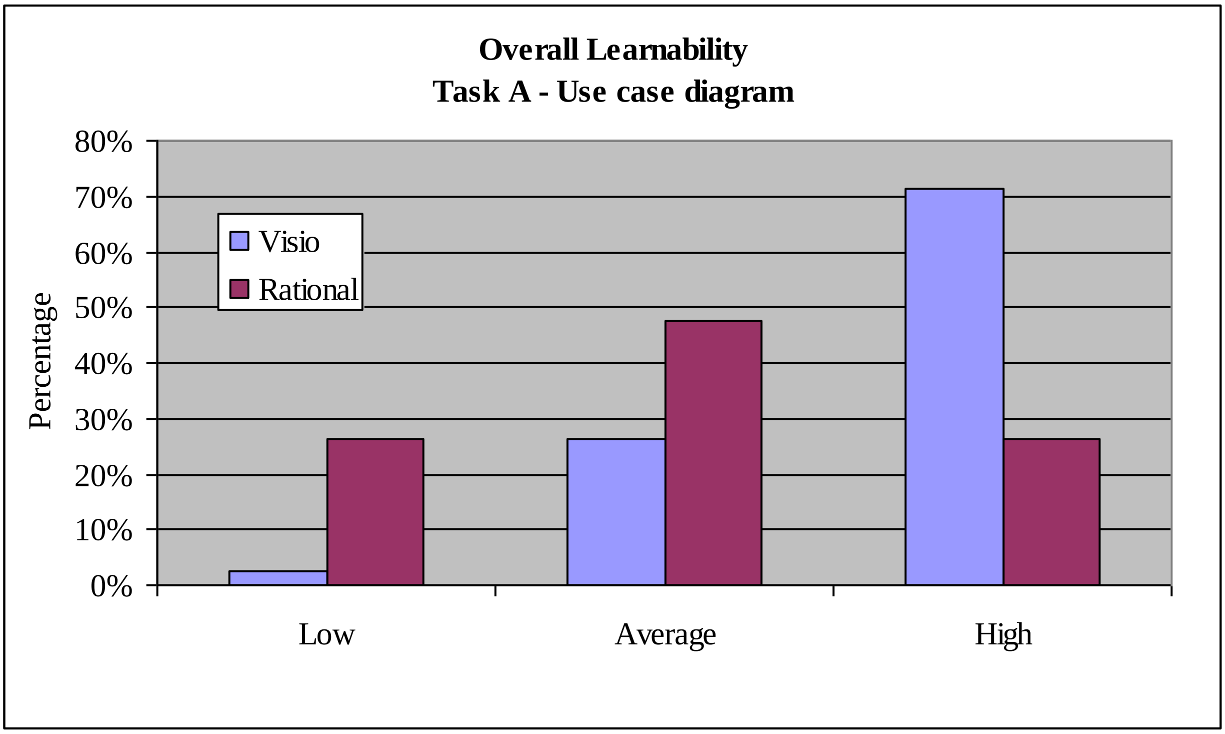

Figure 5-9 Learnability ratings for Task A (n = 46)

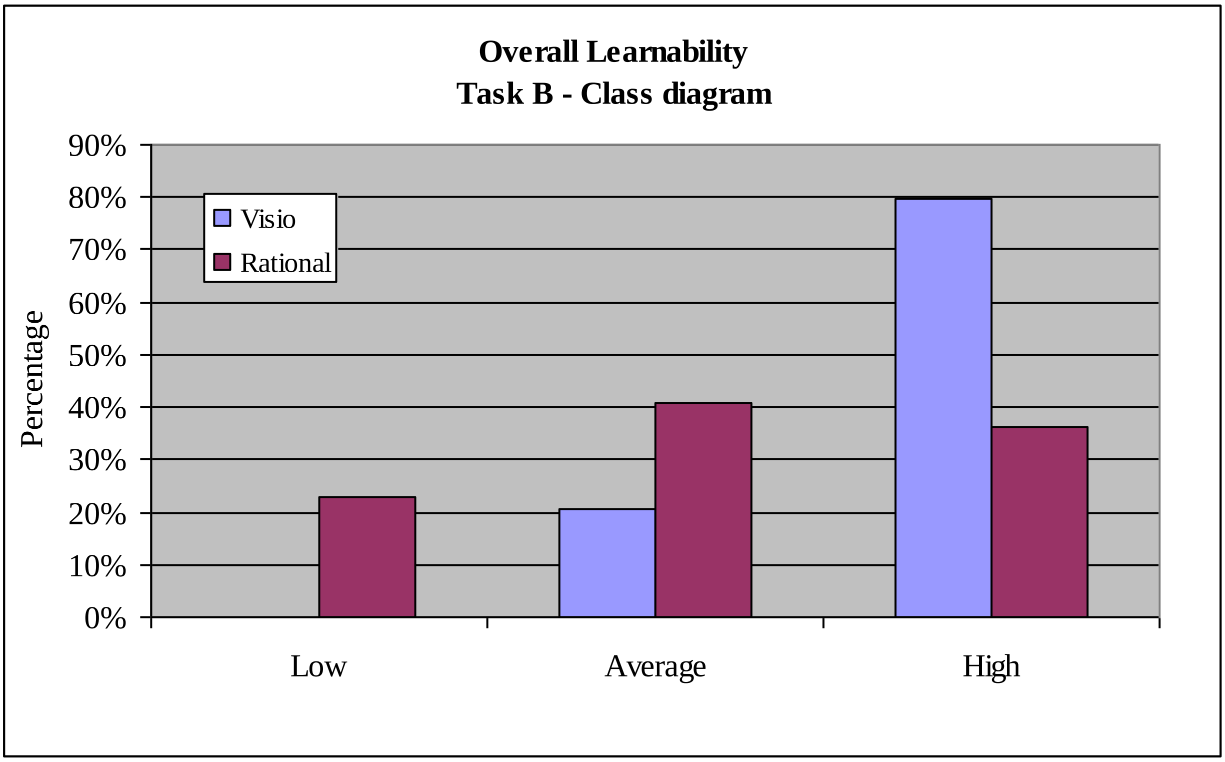

Figure 5-10 Learnability ratings for Task B (n = 46)

We can conclude that for overall learnability, the majority of participants preferred Visio and that there is a difference between the two tools in terms of user satisfaction.

5.4.3 Overall reactions

Section 3 of the learnability questionnaire (Question 8) recorded the participants' overall reactions and consisted of six questions (Section 4.3.4.2). The sample size, overall mean and standard deviation for all six questions for each CASE tool and task is summarised in Table 5-6. The results of the tests for practical and statistical significance are also shown in this table. Data from two participants who did not complete Section 3 for both tools had to be removed from this data analysis, so that the sample sizes were the same. This meant that for Task A the sample size dropped to 44, whereas all 46 participants completed this section for Task B. It can be seen from Table 5-6 that Visio scored higher than Rational for Overall Reactions for both tasks, and that the differences between the means and standard deviations for the two CASE tools were both practically and statistically significant.

Overall Reactions							
Task	**CASE Tool**	**n**	**Mean**	**SD**	**t-test**	**p-value**	**d**
A	Visio	44	3.76	0.53			
	Rational	44	3.18	0.80			
	Difference		0.58	1.05	3.66	.001	0.55 moderate
B	Visio	46	3.67	0.60			
	Rational	46	3.19	0.72			
	Difference		0.48	1.09	2.99	.005	0.44 small

Table 5-6 Overall reactions results

The sample size, mean and standard deviation for each question comprising Overall Reactions is included in Appendix J. A graphical representation of the ratings per question for Section 3 Overall Learnability (Question 8) for Task A is provided in Figure 5-11 and in Figure 5-12 for Task B.

On further investigation into each question comprising Overall Reactions, it can be seen that for Task A, Visio scored better than Rational for all questions except question 3 (Appendix J). For question 3 the participants were required to rate the tool from 1=dull to 5=stimulating. Here the difference was very slight, with Rational scoring 3.44 and Visio 3.40. The same trend occurred for Task B, where Visio was the preferred tool for all questions except question 3 (Appendix J).

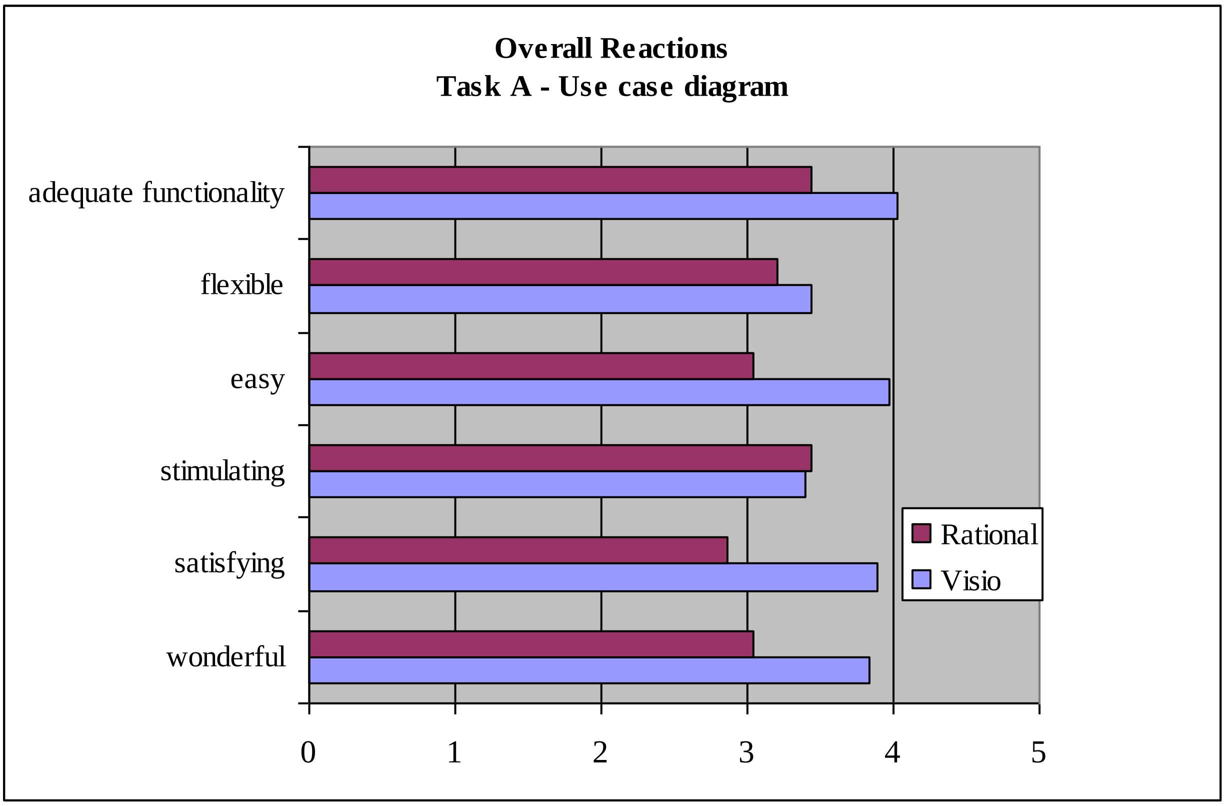

Figure 5-11 Overall reactions ratings for Task A (n = 44)

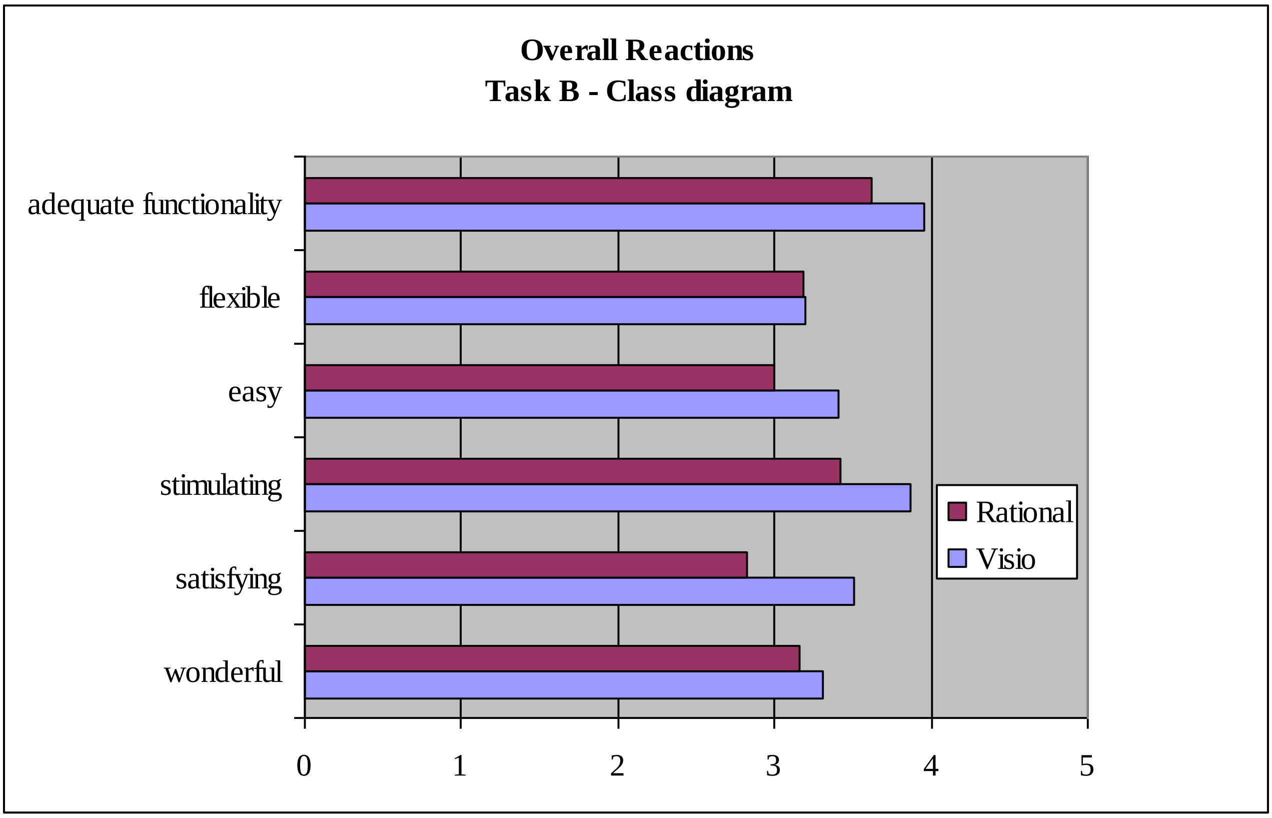

Figure 5-12 Overall reactions ratings for Task B (n = 46)

We can conclude that for Overall Reactions, participants preferred Visio and that there is a difference between the two tools in terms of user satisfaction.

The difference in mean ratings between the two tools dropped from Task A to Task B, which could imply that as the participants got used to the tools, the participants felt less of a difference between using the two tools.

5.4.4 Ease of Use and Learning

Section 4 of the post-test learnability questionnaire comprised both closed-ended questions as well as open-ended questions. This section discusses the closed-ended responses and the next section will discuss the open-ended responses of Section 4 since these were qualitative in nature. For the closed-ended questions the participants had to select their preferred CASE tool for both Ease of Use and Ease of Learning. The Ease of Use question asked the participant *"Which CASE tool did you prefer using?",* and the Ease of Learning question asked the participant *"Which CASE tool was easier for you to learn to use?"*

A summary of the frequency counts of CASE tool preferences is provided in Appendix K for both tasks, and tests for statistical and practical significance were performed on all data in this section. To check for practical significance, a Chi-square goodness-of-fit test and a Cramer's V test was used, which is based on frequencies. The results of these tests showed that for both tasks and tools, the results were of statistical and practical significance (Appendix K).

A graphical representation of the CASE tool preferences for Ease of Use is shown in Figure 5-13 for Task A and Figure 5-14 for Task B. For Task A, five participants did not select a tool preference, and their data could therefore not be included in the analysis. This meant that the sample size for Task A was only 41, whereas for Task B the sample size was 46 as all participants selected either Rational or Visio as their CASE tool preference for Ease of Use.

The number of participants who selected Rational as the preferred tool for Ease of Use increased from 22% to 30% from Task A to Task B. The percentage of participants who selected Visio as the preferred tool for Ease of Use dropped from 78% to 70% from Task A to Task B.

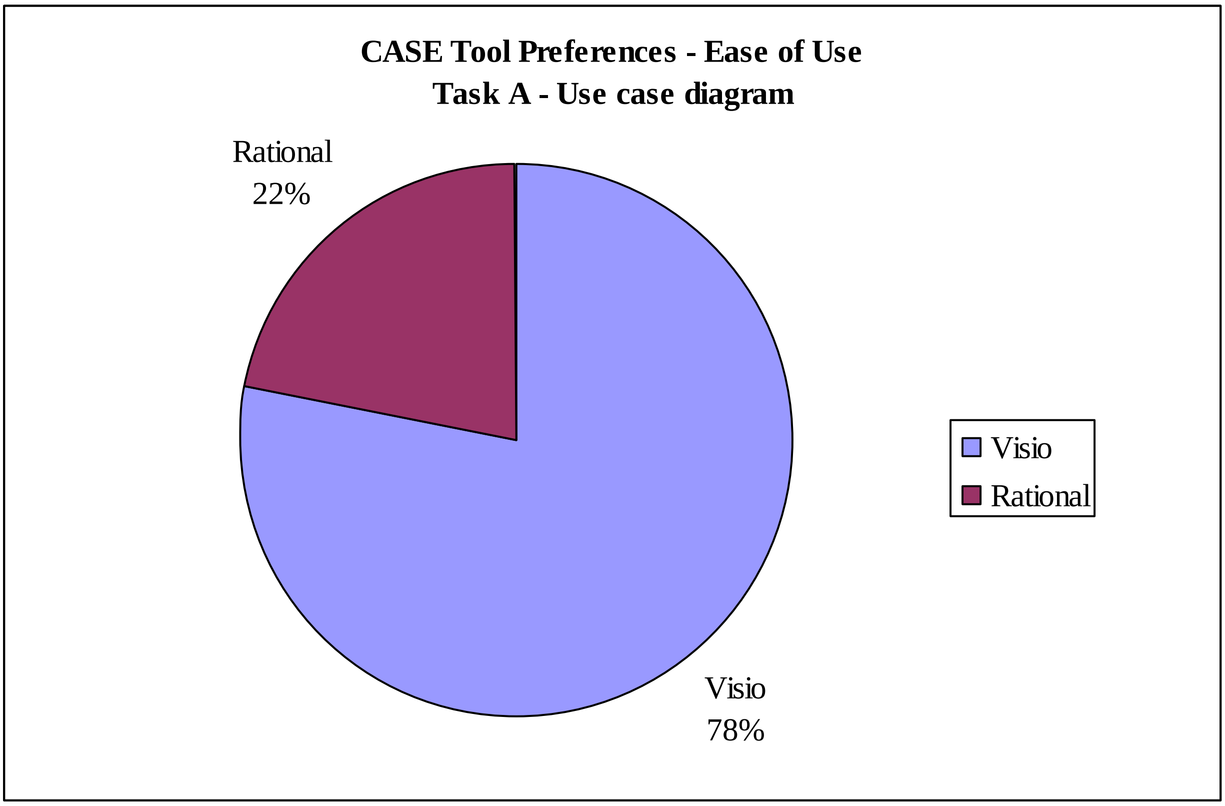

Figure 5-13 CASE tool preferences – Ease of Use for Task A (n = 41)

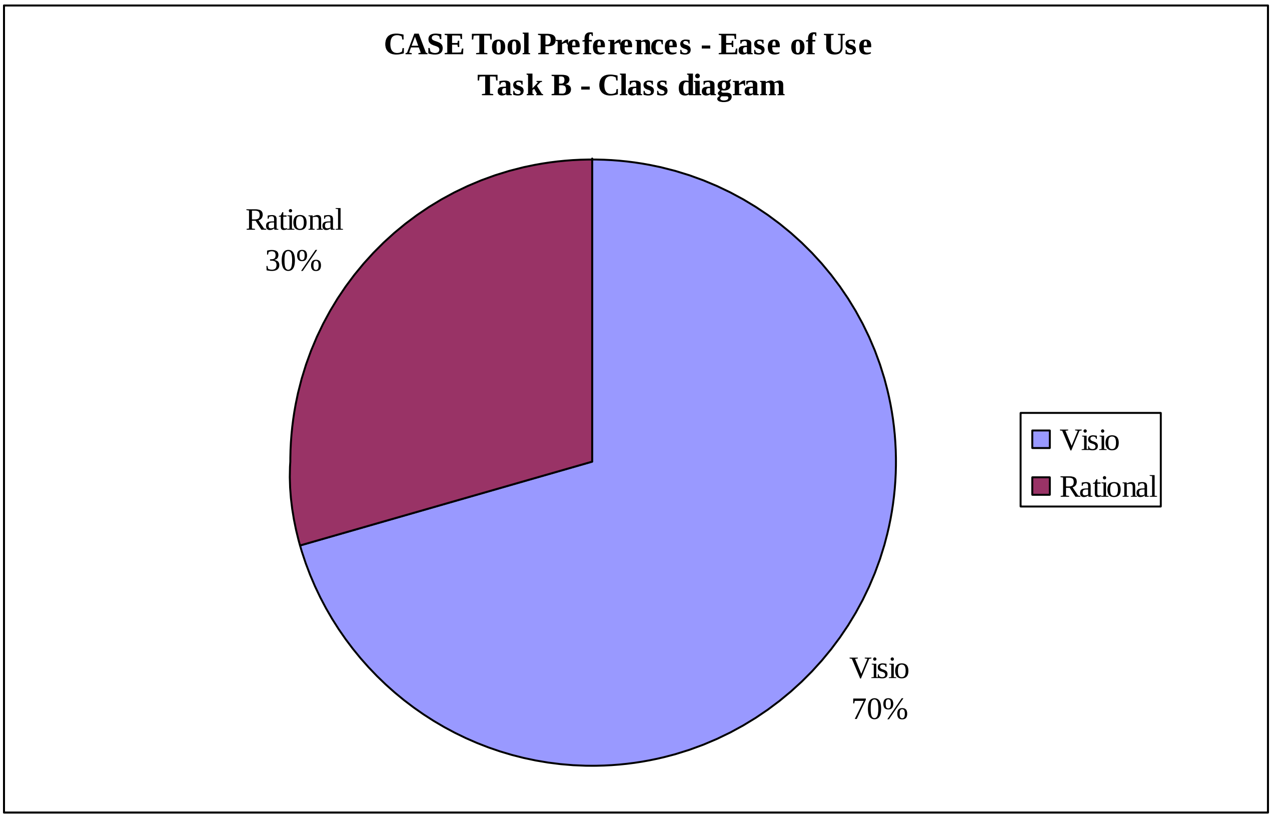

Figure 5-14 CASE Tool preferences – Ease of Use for Task B (n = 46)

Figure 5-15 and Figure 5-16 provide a graphical representation of the CASE tool preferences for Ease of Learning, for Task A and B respectively.

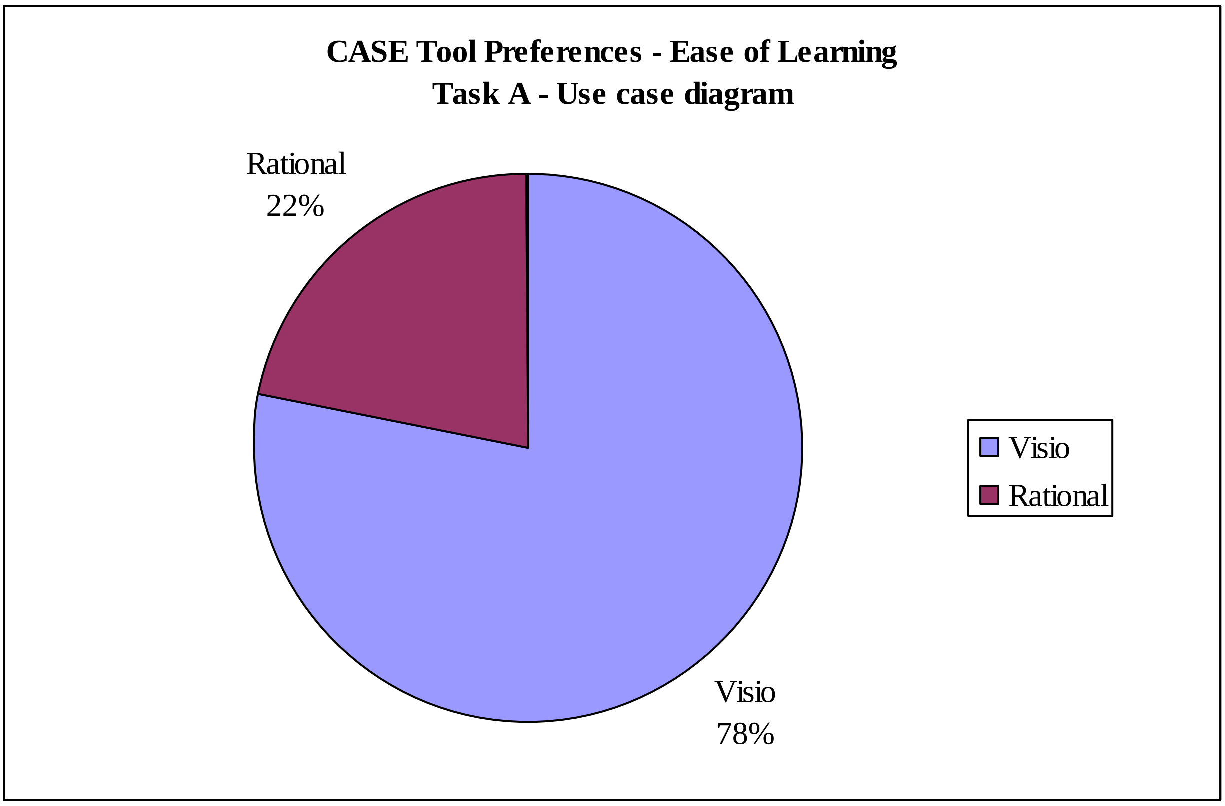

Figure 5-15 CASE tool preferences – Ease of Learning for Task A (n = 41)

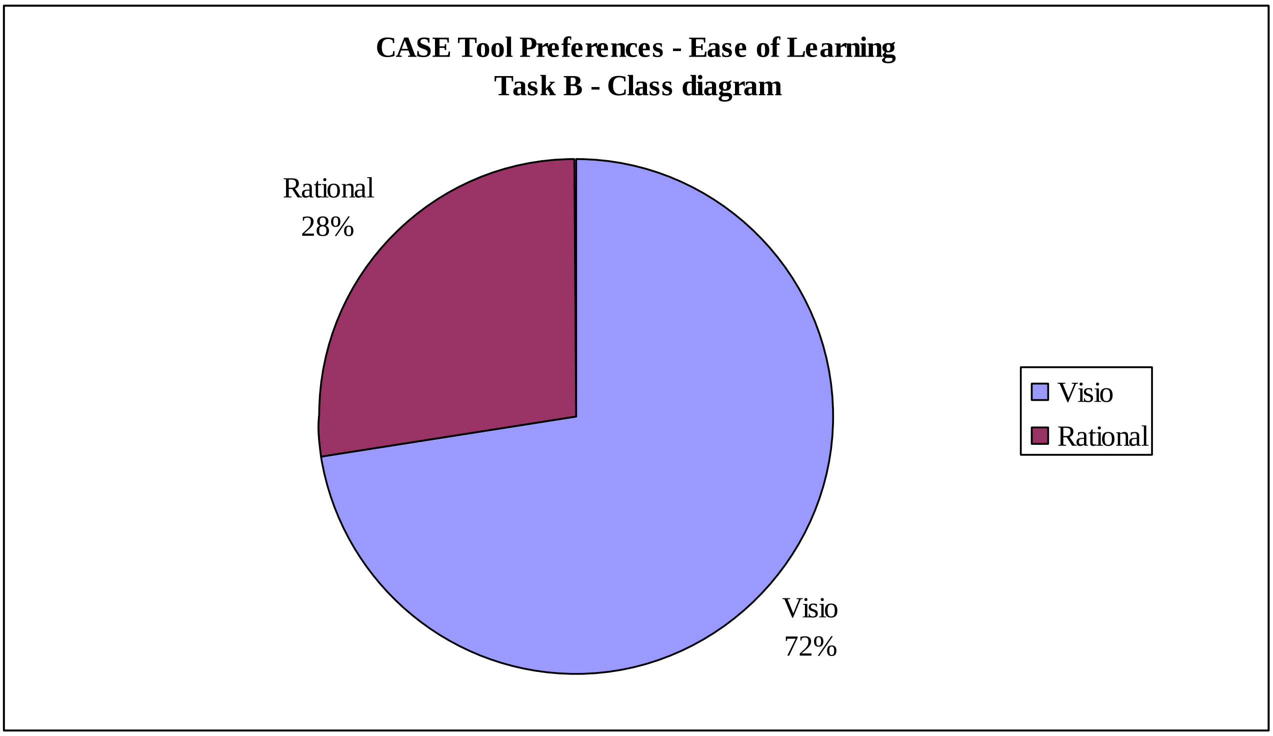

Figure 5-16 CASE tool preferences – Ease of Learning for Task B (n = 40)

The number of no responses for Task A was five and six for Task B. This meant that there was a sample size of 41 and 40 respectively. The number of participants who preferred Rational with respect to Ease of Learning increased from nine (22%) to eleven (28%), from Task A to Task B.

For both Ease of Use and Ease of Learning, Visio was clearly the preferred CASE tool. However, whilst the percentage of participants who preferred Visio dropped from Task A to Task B, those who preferred Rational increased. This could be due to the fact that as the participants learned to use Rational (and got used to it), it became easier to use than Visio. This was confirmed by some of the comments of the participants, one of whom said that Rational was "*Difficult in the beginning but easier as you work with it*" (Section 5.5.2.1).

The results for Task A for Ease of Use and Ease of Learning were exactly the same. For both of these attributes the majority of participants (78%) preferred Visio. For Task B Visio's majority was greater for Ease of Learning (78%) than for Ease of Use (72%). It can therefore be deduced that the Ease of Learning aspect of Visio was preferred to its Ease of Use. This is supported by the results obtained from the learnability ratings when looking at both tasks, where Visio's Ease of Learning had a mean rating of 3.75, whereas Rational's Ease of Learning had a mean rating of 2.9. (Section 5.4.1.3).

5.4.5 Attitude and motivation

Students were asked to describe the way they felt about the use of CASE tools in their module by means of a set of five questions aimed at measuring their attitude and motivation (Section 4.3.4.3), using a 5-point Likert scale. A table of the descriptive statistics comprising mean and standard deviation for each question is provided in Table 5-7.

From these results it can be seen that the highest mean for this questionnaire was for question 1, "*The use of a CASE tool for this course was a good idea",* with a mean of 4.10. From this we can deduce that the participants agreed that a CASE tool is beneficial for OOSAD courses, which agrees with similar studies which have highlighted the pedagogical benefits of using a CASE tool for computing education (Section 2.4). This also supports the qualitative results obtained showing that the learnability of both CASE tools for all attributes were positively rated (Section 5.4.1.3).

Question No	Question	n	Mean	SD
1	The use of a CASE tool for this course was a good idea.	41	4.10	0.80
2	The CASE tool made my work more interesting	41	3.71	0.81
3	The use of a CASE tool enabled me to complete my tasks more quickly.	41	3.68	0.85
4	The use of a CASE tool helped me to understand the underlying concepts better.	41	3.73	0.78
5	Correct understanding and use of the CASE tool helped me to perform better in the UML section of the course and assignments.	41	3.80	0.78
	Overall Attitude and Motivation	**41**	**3.80**	**0.68**

Table 5-7 Attitude and motivation results

5.5 Qualitative results

The next step was the inspection of participants' answers to the open-ended questions in Section 2 and Section 4. Section 2 had to be completed for each CASE tool for both tasks and required the participants to list the positive and negative features of the CASE tools used during the task.

Section 5.5.1 will discuss the positive features of the CASE tools and Section 5.5.2 will discuss the negative features. Section 5.5.3 will discuss the qualitative responses obtained from Section 4 of the questionnaire, in which the participants had to give reasons for their CASE tool preferences.

The qualitative open-ended responses for the questions in Sections 2 and 4 were divided into 17 themes (Section 4.5.2). The first 11 of these themes were based on the learnability attributes for CASE tools as proposed in Table 3-6, and the last six were additional themes which emerged or which were too general to be allocated to a theme. Blank or meaningless responses were not included. The results are discussed in the following sections.

5.5.1 Positive results

The positive features of the CASE tools were extracted from responses to the question "*Briefly describe what you like about the CASE tool?"* (Q7.1). A summary of the frequency counts of these responses per theme identified is provided in Appendix L.

5.5.1.1 Task A – Use case diagram

The key themes relating to Task A are listed in Table 5-8, together with some examples of actual responses. The top themes identified for Visio for Task A were Observability (16%), Familiarity (14%), Simplicity (12%) and Recoverability (11%).

The three key themes for Rational for Task A were Quality of Presentation (36%), Observability (15%) and Error prevention (10%).

One of the key positive sub-themes which emerged was the ease of use of the connectors in both Rational and Visio. These connectors are used to draw associations between objects in the UML diagram. Some of the participants preferred Rational's connectors. One participant said that "*Connecting use cases, extends, includes is easier with Rational"*; and another that *"The association tools are a lot easier to use than Visio"*. Other participants liked Visio's connectors and said that Visio "*Makes associations quite easy to use";* while another participant commented that with Visio it was *"Easy to connect use cases with different shapes."*

Visio			
Theme	**n**	**%**	**Examples of actual responses**
Observability	9	16	*Model explorer provides you with detail to what is really happening with actors and relationships.*
Familiarity	8	14	*It works and is easy to use due to the similarity of the tools of Microsoft.*
Simplicity	7	12	*Simple and easy to use.*
Recoverability	6	11	*Simple to create a use case diagram. Errors are easy to see because they appear in red.*
Rational			
Theme	**n**	**%**	**Examples of actual responses**
Quality of Presentation	14	36	*Easy to work with. Makes work look professional and tools are easily learnt.*
Observability	6	15	*The model explorer window shows you all the relationships that are available in the diagram or project.*
Error Prevention	4	10	*I like the fact that the user is restricted to only draw "includes" and "extends" when appropriate.*
			It won't allow you to make mistakes.

Table 5-8 Key themes for positive responses for Task A (Question 7.1)

Another key positive sub-theme which emerged for both tools was Error Handling (Recoverability), which is an attribute of Robustness. This supports the quantitative results for Rational for Task A where Error Handling was the second highest rated learnability attribute (Section 5.4.1.1). Visio's error handling feature however, scored only fourth in the quantitative results for Task A (Section 5.4.1.1). It is interesting to note the different reactions from participants to these two approaches to error handling. Figure 5-17 illustrates how Visio highlights in red any part of the diagram which is in violation of UML rules. In the sample UML use case diagram drawn in Figure 5-17, an attempt is made by the user to erroneously connect an <<extends>> association line to another association line, which violates UML rules. The related error message is displayed in the error window below the drawing pane.

As can be seen from the comments in Table 5-8, some of the participants liked Visio's handling of errors whilst others did not (Section 5.5.2).

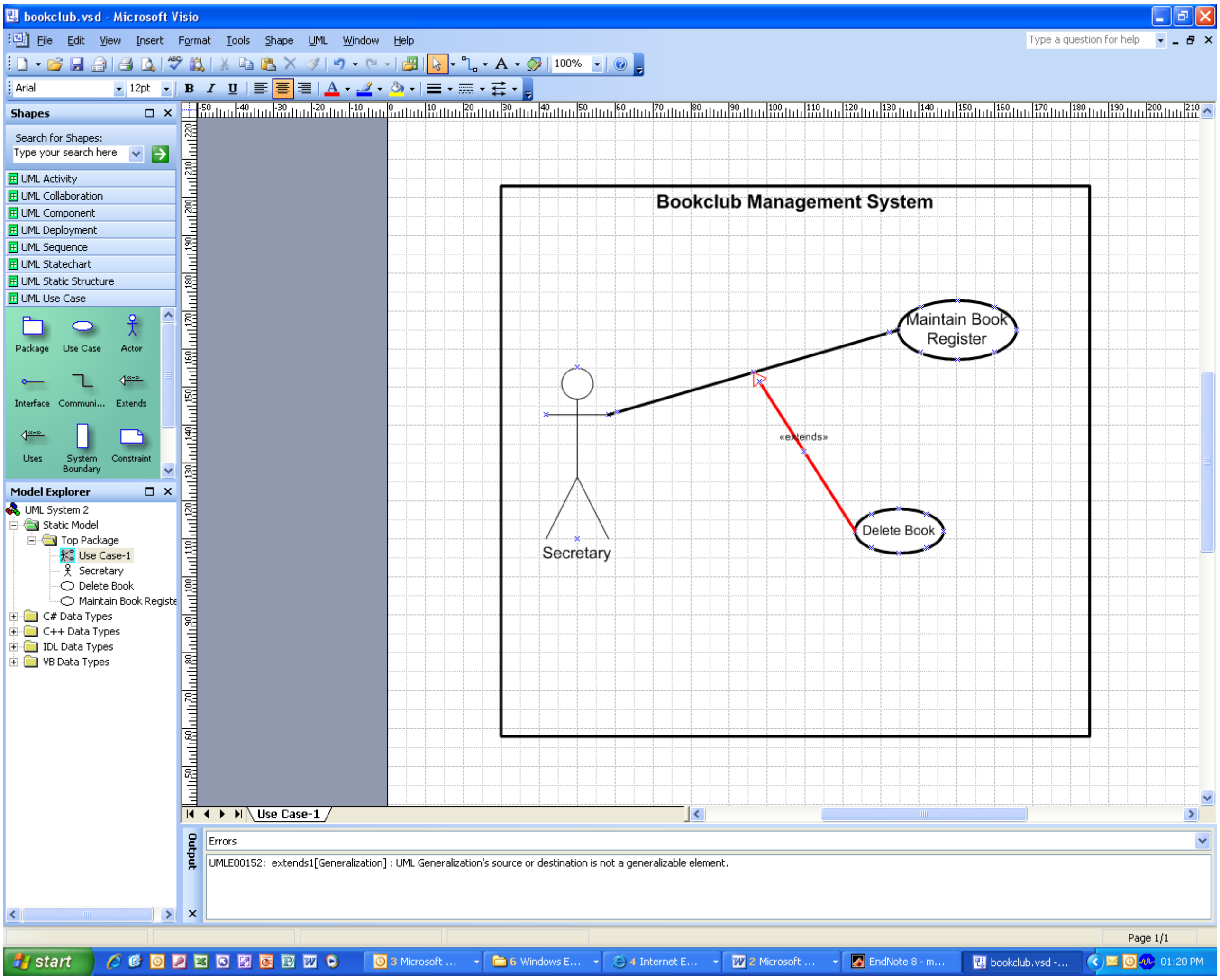

Figure 5-17 Visio Drawing pane showing error handling

The results obtained relating to the high incidence of the Robustness theme (Error prevention and Recoverability) therefore support its inclusion in the list of learnability attributes (Table 3-6). The results also showed that the use of the CASE tools assisted with the understanding of UML, which supports the findings of the literature review (Section 2.4), as well as the attitude and motivation results (Section 5.4.5). One of the participants commented that "*Using the CASE tool helps understand use cases, actors and communication lines*".

The top themes for each tool were totally different. Familiarity was one of the top themes for Visio as would be expected, due to the fact that WRI201 students have worked on a number of different Microsoft products and the user interface for Visio is very similar and therefore familiar.

5.5.1.2 Task B – Class diagram

Examples of actual comments relating to the key themes for Task B are listed in Table 5-9. The key themes for Visio were Recoverability (28%), Observability (19%), and Familiarity (6%). The three key themes for Rational for Task B were Quality of Presentation (35%) and Recoverability (18%).

Visio			
Theme	**n**	**%**	**Examples of actual responses**
Recoverability	15	28	*The case tool is user friendly. It is a very interesting case tool because errors can easily be identified.*
			Errors are pointed out by the shapes turning red.
Observability	10	19	*Easy to get the tools on the working space because you just drag them and they are easy to use.*
			The drag and drop feature of the Case tool. I don't have to draw the boxes and lines, I just use the template. Great.
Familiarity	3	6	*It's easy to learn to use and the fact that the interface is similar to the more popular software like Word and Excel.*
Rational			
Theme	**n**	**%**	**Examples of actual responses**
Quality of presentation	12	35	*Visually pleasing and easy to use. Well laid out and user friendly.*
			It looks very nice. It is easy to use. The functions and buttons are more visible.
Recoverability	6	18	*It tells the user when they make a mistake and explains clearly what the problem is.*
			Good error messages.
			Once you get used to how it works, it is a nice tool to work with. Especially the fact that it gives you an error message when something is wrong. It was fun playing with it.

Table 5-9 Key themes for positive responses for Task B (Question 7.1)

A number of participants commented on the fact that they found the interface of Visio familiar with other Windows products, and this made it easier to use. One of the sub-themes which emerged for both tools was the adding of attributes. Several participants mentioned that they liked the way in which Visio allowed the user to add attributes to classes. This was also true for Rational, however, as a number of participants commented on Rational's ease of use with regard to the adding of attributes.

5.5.1.3 Both tasks

Recoverability, Familiarity and Observability were the top positive themes for Visio for both tasks. Observability had the highest frequency count in Task A for Visio, but only the second highest in Task B, whilst Recoverability moved from fourth highest count to first. This is probably because as participants got to know and understand how the error handling features of Visio worked, so they rated it better.

Recoverability was one of the key positive themes for both CASE tools and tasks, and all three of the additional attributes of Recoverability (Error Prevention, Recoverability and Provision of Help) were listed in the participant's responses. These results support the inclusion of these attributes in Table 3-6.

Quality of Presentation was a top theme in both tasks for Rational. However, the quantitative results show that Familiarity was the highest rated attribute for Rational. This is a concern as it could imply that the participants were not clear as to what is meant by Familiarity, or it could mean that the question was incorrectly phrased. The expertise of the participants could also have affected their ratings (Section 4.6).

Error handling was the second highest scoring attribute for Rational when looking at both tasks in the quantitative results (Section 5.4.1.3). This agrees with the qualitative results which showed Error prevention and Recoverability as one of the key positive themes for Rational. There is no related attribute for the Quality of Presentation theme in Senapathi's framework, but the fact that this was one of the key positive themes supports its inclusion in the proposed list of learnability attributes in the Feedback category (Table 3-6).

5.5.2 Negative results

The responses to the question "*Briefly discuss what you don't like about the CASE tool and any problems that you have encountered with using it"* (Q7.2) were allocated to themes and sub-themes. A summary of the frequency counts of these responses per theme is included in Appendix M.

5.5.2.1 Task A – Use case diagram

The key themes for negative responses for both CASE tools for Task A are listed in Table 5-10, together with some examples of actual responses. The key negative themes for Visio for Task A were Observability (30%), Recoverability (27%) and Simplicity (10%). The key negative themes for Rational for Task A were Simplicity (24%), Recoverability (21%) and Quality of Presentation (18%).

One of the main problems mentioned in Task A for Visio was Recoverability. Other problems included difficulties with labelling the associations or connectors.

Visio			
Theme	**n**	**%**	**Examples of actual responses**
Observability	9	30	*Initial use hard, finding correct structures hard.*
Recoverability	8	27	*The annoying thing is trying to connect the communication lines and trying to change their colour from red to black.*
			The red lines indicate errors but these weren't obvious what they meant.
Simplicity	3	10	*Too many functions and commands*
Rational			
Theme	**n**	**%**	**Examples of actual responses**
Simplicity	9	24	*Too many functions and commands.*
			Too many menus and buttons.
Recoverability	8	21	*When you make a mistake the program does not allow you to perform the action at all. The program is confusing to use. The connector tools are difficult to use in comparison with Visio. It is difficult to undo mistakes made with connector tools.*
Quality of presentation	7	18	*User interface not attractive at first. Too many windows.*

Table 5-10 Key themes for the negative responses for Task A (Question 7.2)

When adding a new association between an actor and a use case, Visio puts default end names on the association, which have to be manually removed for each association and is very time consuming. Other problems experienced with Rational, besides the error handling and complexity of the tool, were the difficulty in finding the right tools, and the fact that you cannot drag objects onto the drawing window.

5.5.2.2 Task B – Class diagram

The key negative themes which emerged for Task B for both CASE tools are listed in Table 5-11, together with some examples of actual responses. The key negative themes for Visio for Task B were Observability (36%), Recoverability (25%), and Provision of Help (18%). The key negative themes for Rational for Task B were Observability (37%), Recoverability (20%), and Simplicity (14%).

Visio			
Theme	**n**	**%**	**Examples of actual responses**
Observability	10	36	*Drawing the association lines for the class diagram was not easy.*
Recoverability	7	25	*The only problem comes with connecting the cases, whereby you get a red line until you've connected properly.*
Provision of help	5	18	*The help feature wasn't helpful. The menu wasn't clear as to where to find the feature I wanted.*
Rational			
Theme	**n**	**%**	**Examples of actual responses**
Observability	13	37	*Associations are extremely difficult to work with. The program is difficult to work with in terms of finding the tools to use.*
			Using associations was confusing and difficult to master.
			When adding attributes to a class it is very frustrating having to right-click "Insert new property" and then having to click on the name to change it.'
			Tedious adding attributes to classes. Takes long getting started.
Recoverability	7	20	*When you are making a mistake it doesn't show you an error message.*
			Errors are well stated but don't continually remind the user or highlight the error.
			Sometimes did not give the error message and did not respond.
Simplicity	5	14	*Too many functions and commands*

Table 5-11 Key themes for the negative responses for Task B (Question 7.2)

Several key themes identified from the negative responses were the same as the key themes identified from the positive responses. For Visio these themes were Recoverability and Observability. For Rational this was Quality of Presentation. This means that sometimes the best rated attributes were often the worst as well. This was definitely true for Recoverability and Error Prevention, where several participants liked the error handling features of Visio (Section 5.5.1.1), but others did not. A similar trend emerged with Rational, since although some participants liked the fact that Rational did not allow an erroneous association to be made, others found it confusing as no message is displayed.

Visio could improve its error handling by providing error messages which are expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution (Section 3.4.2.2). The error message is also displayed at the bottom of the screen where the users do not always see it, and should be more appropriately placed on the screen, possibly by using a confirmation message (Nielsen 1994).

While Visio highlights incorrect connections in red, Rational does not allow the user to make erroneous connections. As with Visio, some of the participants like the way in which Rational did not allow the participant to make an erroneous error, others found it frustrating (Section 5.5.2).

One way in which Rational could improve its error prevention technique, would be by checking for errors and presenting the users with a confirmation option or a warning before they commit to the action (Section 3.4.2.2). This would be preferable to preventing the users from making the error completely without providing any feedback or error message as is done currently.

Since Rational was the lesser preferred CASE tool and the most frequently identified negative theme for Rational was Recoverability, it can be deduced that Rational's approach to error handling is not the desired approach. Designers of CASE tools for use in computing education should take note of this in terms of how to design a CASE tool. This supports the recommendation that Rational show an error message rather then preventing the erroneous connection completely without providing any feedback.

Further research is required, together with a larger sample size, to determine whether or not the recommended modifications are preferred. In addition the relationship between the Robustness attributes and the context of use factors could be tested. This could assist designers and researchers in understanding which participants preferred which error prevention method and why.

The top two negative qualitative themes were similar for both CASE tools in Task B. This was not true for the positive results where the key themes were different for the two tools. All the learnability attributes identified in Table 3-6 had related comments from the participants (both positive and negative) and their inclusion in a learnability evaluation is therefore supported.

5.5.3 CASE tool preferences

Section 4 of the questionnaire related to the participants' preferences of CASE tool with regard to Ease of Use and Ease of Learning. Only the questionnaire for the second CASE tool had a Section 4, as this could section could only be completed after both CASE tools had been used.

5.5.3.1 Ease of Use

The first question in Section 4 related to Ease of Use and asked the participants, *"Which CASE tool did you prefer using?" (Question 9.1).* The second question (Question 9.2), asked the participants to discuss the reasons for their answer in Question 9.1. The key themes identified for participants' preferring one CASE tool over another for Ease of Use for Task A are listed in Table 5-12 and for Task B in Table 5-13. Some examples of actual reasons which were given in their responses are also included.

Ease of Use	
Theme	**Examples of Reasons for Preferring Visio**
Simplicity	*Simpler and easier. Shows error messages. Rational does not.*
	Simpler interface which makes working with it easy. Immediately indicates an error.
Familiarity	*It is familiar territory because of Microsoft. Logical and understandable.*
	Common interface with Microsoft products. Integration with other products.
Recoverability	*Communication tools are easier to use. Mistakes are indicated so they can be corrected Rational does not indicate error but does not let you do the action.*
	Easier to get started and understand and shows you where your errors are.
Theme	**Examples of Reasons for Preferring Rational**
Quality of presentation	*Looks better than Visio and tools are easier to use especially the drawing of lines and naming of elements is better because you can do it all in one form. I like the fact that it looks and feels like Eclipse Java Editor.*
	Presentable, attractive and looks more professional. Easy to interpret. Easy to construct.
Recoverability	*Does not allow you to perform an error in diagram.*
	Interface might not be friendlier but it is easier to use and I enjoyed the fact that it allowed one to easily correct errors.

Table 5-12 Reasons for CASE tool preferences for Ease of Use for Task A

Ease of Use	
Theme	**Examples of Reasons for Preferring Visio**
Familiarity	*It is much easier to learn how to use. The Microsoft interface is familiar. The structure of the program is simple to understand. Rational Software Modeller is not set out well and it is difficult to navigate through the program.*
Simplicity	*Simpler and easier. Shows error messages. Rational does not.*
Recoverability	*Easier to learn how to use. Communication tools are easier to use. Mistakes are indicated so they can be corrected. Rational does not indicate error but does not let you do the action.*
Theme	**Examples of Reasons for Preferring Rational**
Quality of Presentation	*Easier to use, well laid out and less complex. It looks better than Visio and is visually appealing.*
	The whole interface is much better than the other one. Tools are easy to use. Everything is right here in front of you. Error messages are more helpful than the other one.
Recoverability	*Interface might not be friendlier but it is easier to use and I enjoyed the fact that it allowed one to easily correct errors.*
	Does not allow you to perform an error in diagram.

Table 5-13 Reasons for CASE tool preferences for Ease of Use for Task B

The key themes identified from the reasons stated for selecting Visio over Rational for Ease of Use were similar for Task A and Task B. These themes were Simplicity, Familiarity and Recoverability. The key themes which emerged from the reasons why participants selected Rational over Visio were Quality of Presentation and Recoverability for both tasks. The reasons participants preferred Visio over Rational for Ease of Use differed from the reasons Rational was preferred to Visio. The only theme that was the same for both was Recoverability.

5.5.3.2 Ease of Learning

The second qualitative question in Section 4, was Question 9.3 which related to Ease of Learning. Some examples of the reasons given for CASE tool preferences for Ease of Learning for Task A are listed in Table 5-14, and in Table 5-15 for Task B.

Ease of Learning	
Theme	**Reasons for Preferring Visio**
Familiarity	*More familiar. More useful and less time consuming.*
	Easier to learn and use since it has same interface as Word and Excel.
Simplicity	*Straightforward. Easy to create new drawing. Options all presented in one area - system boundary, Rational uses drop menu for "uses" and "extends"*
	Steps are simpler to follow when using Microsoft Visio.
Theme	**Reasons for Preferring Rational**
Provision of Help	*Better access to tutorials which makes learning the tool easier.*
	Tutorials are nice and easy guide to learning the software.

Table 5-14 Reasons for CASE tool preferences for Ease of Learning for Task A

Ease of Learning	
Theme	**Examples of Reasons for Preferring Visio**
Familiarity	*Fewer actions. Familiarity of Microsoft.*
Simplicity	*There aren't any unnecessary windows open when you start using it, whereas Rational has quite a few windows open which you don't know what to do with. The workspace in Visio is much better to manipulate.*
	Not as complicated as Rational.
	Everything you need is in one place. Rational cannot see where you must add attributes or draw an association.
Theme	**Examples of Reasons for Preferring Rational**
Provision of Help	*The tutorials and the fact that you cannot make a mistake makes this tool easier to use and learn.*
Recoverability	*Good error messages.*

Table 5-15 Reasons for CASE tool preferences for Ease of Learning for Task B

The key themes which emerged from the reasons given why participants preferred Visio to Rational for Ease of Learning were Familiarity and Simplicity for both Task A and Task B. The primary theme for participants selecting Rational as the preferred CASE tool for Ease of Learning for Task A was Provision of Help, due to its on-line help facility and tutorials. These are different from the key themes identified for Ease of Use, but because of the low frequency count of participants preferring Rational, no conclusions can be drawn.

The main themes that emerged from the reasons given for preferring Rational to Visio for Ease of Learning for Task B differed slightly from those for Task A and were Provision of Help and Recoverability. These results contradict with the quantitative results which showed Familiarity as the highest rated attribute for Rational.

Since the quantitative results clearly show that Visio was the preferred CASE tool (Section 5.4.4), and the key qualitative themes for preferring Visio for both Ease of Use and Ease of Learning were Familiarity and Simplicity, it can be deduced that the key desired attributes for CASE tool design should be Simplicity and Familiarity. This supports the inclusion of these attributes in the proposed list of learnability attributes (Table 3-6).

5.6 Conclusions

Forty six participants participated in the study and 92 post-test questionnaires were included in the data analysis. Of these participants only half (52%) indicated that they had more than 10 hours per week computer use, whereas only 28% indicated that English was their home language. This provided further justification for testing for a relationship between home language and frequency of computer use, and CASE tool learnability (Section 5.3).

Initial statistics on the responses from the first section of the learnability questionnaire revealed that Visio scored higher than Rational for both tasks (Section 5.4.1). The difference in mean ratings dropped slightly from Task A to Task B. For all attributes of learnability (except Familiarity), Visio was the preferred CASE tool. The highest rated attribute for Rational for both tasks was Familiarity and for Visio it was Predictability.

Seventy one percent (71%) of the participants rated the learnability of Visio for Task A as High, whereas only 26% of the participants rated Rational as High (Section 5.4.2). For Task B, the percentage of participants who rated the learnability of Visio as High, increased by 9%, whereas for Rational it increased by 10%. Visio was the preferred tool for both tasks for Overall Reactions, except for how stimulating the tool was (Section 5.4.3).

Visio was also the preferred CASE tool for both tasks for Ease of Use and Ease of Learning (Section 5.4.4).

There was however, a slight increase in the number of participants who preferred Rational for Ease of Use and Ease of Learning, from Task A to Task B, whilst those preferring Visio dropped.

The participants were also asked to record their attitude towards the use of a CASE tool in the course. The results showed a favourable response towards using a CASE tool for teaching OOSAD and that using a CASE tool helped to understand the underlying UML concepts (Section 5.4.5). The qualitative results showed that for both tasks Familiarity and Recoverability were two key positive themes for Visio; whereas for Rational the key positive theme was Quality of Presentation (Section 5.5.1).

The qualitative data also showed that the participants encountered several learnability problems with both CASE tools (Section 5.5.2), specifically with regard to the error handling features of these tools. Other problems encountered with both tools related to the use of connectors, the finding of tools and shapes and the adding of attributes (Section 5.5.2).

The two most common positive themes listed for preferring Visio for both tasks for Ease of Use and Ease of Learning were the same, namely, Familiarity and Simplicity (Section 5.5.3). Although the majority of the participants selected Visio as the preferred tool, several participants said they preferred Rational's error handling feature. The main theme emerging from the reasons why participants preferred Rational for Ease of Use was Quality of Presentation and for Ease of Learning was Provision of Help. Since the quantitative results clearly show that Visio was the preferred CASE tool (Section 5.4.4), it can be deduced that the key desired attributes for CASE tool design should be Simplicity and Familiarity.

The initial descriptive statistics given in this chapter clearly show that Visio was the preferred CASE tool for both tasks. The next chapter will discuss a more detailed analysis of the relevant quantitative data in order to determine if a relationship exists between several context of use factors and CASE tool learnability. The context of use factors selected for further analysis were home language, frequency of computer use and the tool used. This further analysis will enable the testing of the hypotheses listed in Section 4.2.1.

Chapter 6 Analysis of results

6.1 Introduction

The objective of this chapter is to discuss the further analysis of the quantitative research results documented in the previous chapter. This chapter will also discuss the relationship between selected context of use factors and the satisfaction ratings of learnability given by the participants. This analysis will then be used to answer the hypotheses stated in Chapter 4 (Table 4-1).

6.2 Relationship between context of use factors and learnability

A repeated measures ANOVA test was conducted on the relevant factors for each participant and the overall mean ratings for the different learnability attributes. Statistica V7.0 software was used to perform these tests. The study explored the relationship between three context of use factors and learnability. The first two factors were from the User aspect of the framework, namely home language and frequency of computer use. The last factor was the CASE tool used, which represents the Tool aspect of Senapathi's framework. A repeated measures ANOVA test was performed on each of these independent variables and the mean ratings for all attributes in the learnability section of the questionnaire. The table for the detailed ANOVA results for both tasks is included in Appendix N.

The results for each of these factors are discussed in the following three sections.

6.2.1 Home Language

In order to answer the question, *"Is there a relationship between the student's home language and CASE tool learnability"*, a repeated measures ANOVA test was conducted using home language as the independent variable for each of the learnability attributes.

The participants' home language recorded on the background questionnaires had four choices, namely "Afrikaans", "English", "Xhosa" and "Other African". Due to the small sample size, the Xhosa and Other African language groups were combined into a single group, labelled "African" for purposes of the ANOVA tests. This meant that for the repeated ANOVA tests, three language groups were used.

Table 6-1 lists only the significant results from the repeated measures ANOVA tests for the relationship between home language and each of the various learnability attributes. Significant results were found for the Familiarity and Predictability attributes for Task A. For Task B, home language was not significant for any of the learnability attributes. This could be due to the fact that by the second task (Task B), the participants had gained some experience of CASE tools; thus the difference in home language was neutralised. No further tests for Task B were thus required. The two attributes that were significant for Task A will be discussed in more detail in the next two sections.

Home language						
Task A						
		Sum of Squares	**df**	**Mean Square**	**F**	**p**
Familiarity	**Language**	6.30	2	3.15	3.35	.046
Predictability	**R1*Language**	2.93	2	1.47	3.27	.049

Table 6-1 Significant results for home language

6.2.1.1 Familiarity

According to the ANOVA results, home language was significant for the Familiarity (p = .046) attribute for Task A, which means that certain language groups had a significantly different mean rating from others for this attribute. The mean rating for Familiarity for each language group is shown in Table 6-2.

Language Group	**n**	**Mean**	**SD**
English	13	3.423	0.813
Afrikaans	12	3.479	0.361
African	20	3.063	1.022

Table 6-2 Familiarity ratings for home language groups for Task A

When more than two groups are being analysed and there is a significant difference between these groups, it is necessary to carry out more specific two-group comparisons in order to determine where the major treatment effect is occurring. These two-group comparisons are commonly referred to as individual comparisons (follow up tests, or post-hoc tests). Since there were three home language groups, post-hoc Scheffe tests were performed on home language and Familiarity. The results of the post-hoc Scheffe test performed for Familiarity are shown in Appendix O.

None of the differences between the home language groups were found to be statistically significant based on the Scheffe tests performed. This contradiction could be due to the relatively small sample size used in this study.

6.2.1.2 Predictability

Home language was found to be significant for the repeated measure (R1*Language) of the Predictability attribute in Task A, which means that there was interaction between at least two of the language groups for this attribute. In other words, the change in mean rating for the Predictability attribute for one language group from the first measure (Tool 1) to the second measure (Tool 2) differed significantly from another language group. There are more than two groups for home language; therefore a post-hoc Scheffe test was performed for Predictability to see which pair of means was significantly different. The detailed results of the Scheffe test are given in Appendix O. The results of this post-hoc test confirmed the ANOVA test results showing that there was a significant interaction between two of the home language groups. This interaction occurred between the Afrikaans and the African language groups (p = .030) and is illustrated in Figure 6-1.

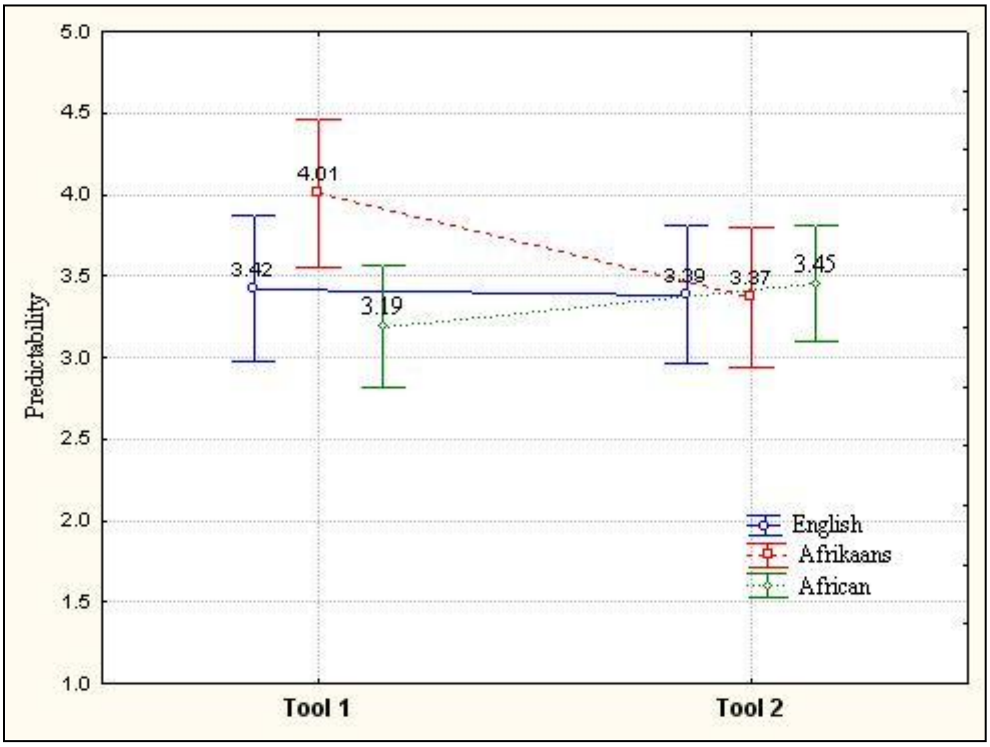

Figure 6-1 Relationship between home language and Predictability for Task A (n=46)

The mean rating for the Afrikaans group decreased from 4.01 for the first measure (Tool 1) to 3.37 for the second measure (Tool 2), but the mean rating for the African language group increased from 3.19 for Tool 1 to 3.45 for Tool 2. The English language group was the only group whose mean rating remained fairly consistent between the two tools. It can be deduced that this is because the language of the user interface of both CASE tools was in English and this was the home language of the participants in this group. The user interface was therefore found to be more consistently predictable for this language group.

6.2.2 Frequency of computer use

The participants' general frequency of computer use was measured by the question, *"What is the frequency of your computer use?"* and had four choices, namely Very Low (< 5 hrs per week), Low (5 – 10 hrs per week), High (11 – 20 hrs per week) and Very High (> 20 hrs per week). Due to the relatively small sample size for frequency of computer use, the Very Low and Low groups were merged into a Low group and the High and Very High groups remained the same. Table 6-3 lists the groupings used for the ANOVA tests for Frequency of computer use.

	Frequency of computer use	
Group No	**Description**	**ANOVA Group**
1	Very Low (< 5 Hrs per wk)	**Low**
	Low (5-10 Hrs per Wk)	
2	High (11-20 Hrs per wk)	**High**
3	Very High (> 20 Hrs per wk)	**Very High**

Table 6-3 Groupings used for frequency of computer use

The ANOVA table for the results for frequency of computer use is shown in Table 6-4. For both tasks, frequency of computer use was significant for the repeated measure (R1*Frequency) of the Error handling attribute of learnability. This means that there was interaction in the mean ratings for Error handling between at least two of the frequency of computer use groups.

Frequency of computer use						
Task A						
		Sum of Squares	**df**	**Mean Square**	**F**	**p**
Error handling	R1*Frequency	6.56	2	3.28	4.70	.015
Task B						
Error handling	R1*Frequency	8.06	2	4.03	4.00	.027

Table 6-4 ANOVA Results for frequency of computer use

These results are illustrated in the graph in Figure 6-2 for Task A and in Figure 6-3 for Task B. Since there were more than two frequency of computer use groups, post-hoc Scheffe tests were run in order to determine which of the groups were significant. The results of these post-hoc tests are included in Appendix O.

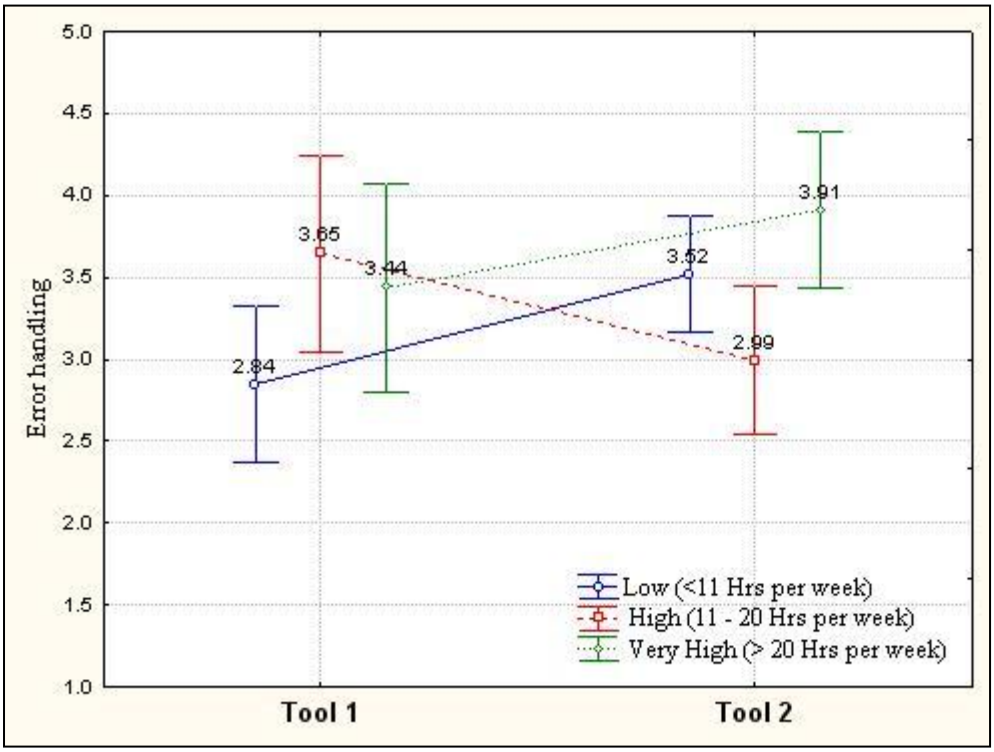

Figure 6-2 Relationship between frequency of computer use and Error handling for Task A (n = 46)

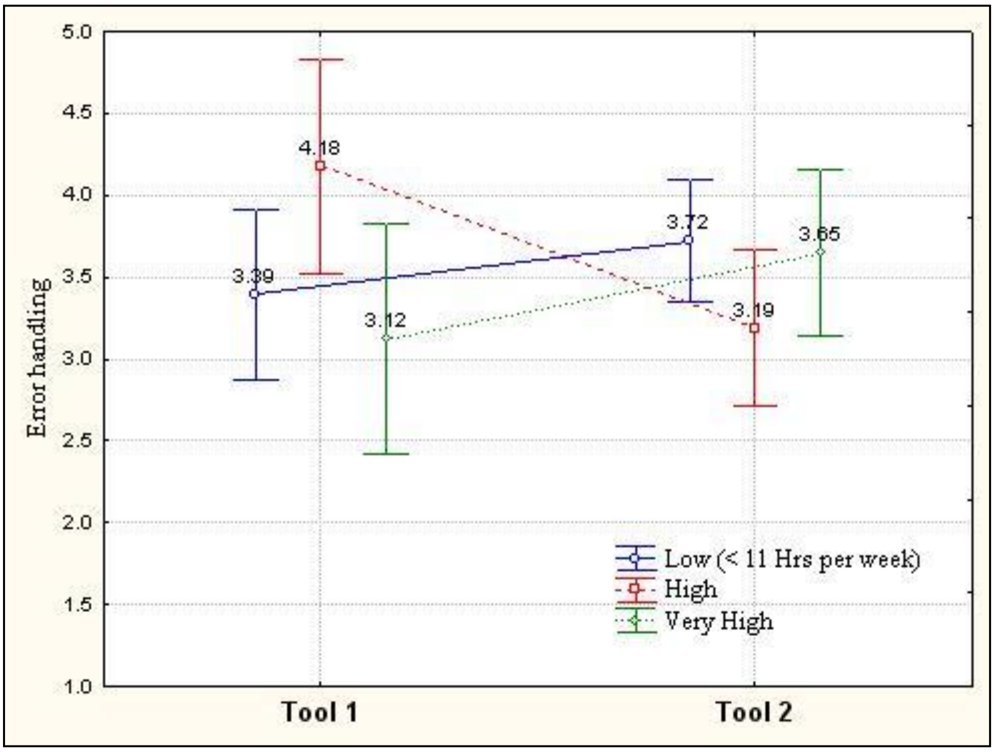

Figure 6-3 Relationship between frequency of computer use and Error handling for Task B (n = 46)

These post-hoc tests did not reveal any significant differences between the different groups. This contradiction could also be caused by the relatively small sample size. The interactions shown in Figure 6-2 and Figure 6-3 are therefore not significant.

For Task A, there was an interaction between the Low and the High frequency groups. The mean rating for Error Handling for the Low group increased from the first tool to the second tool, but the mean rating for the High group decreased. The pattern was the same for Task B. This could imply that the less frequent computer users (Low frequency group) had fewer problems with Tool 2 and thus their satisfaction ratings increased after using the first tool, since their experience of CASE tools had increased. The High frequency group's mean rating dropped as they had slightly more computer experience and frequency of computer use prior to using the CASE tools than the Low group. The fact that the Very High frequency group's mean ratings remained more or less consistent between the two tools could be due to their higher level of computer experience and greater frequency of computer use prior to using the CASE tools.

Error handling was the only attribute which had a significant interaction for frequency of computer use. This means that for all other learnability attributes the mean rating remained fairly consistent from Tool 1 to Tool 2 for the three frequency groups.

6.2.3 Tool

The Tool part of the framework was measured by the two CASE tools used, namely Visio and Rational. Table 6-5 lists the significant results for the repeated measures ANOVA tests performed where tool was the independent variable.

For Task A, the CASE tool used was significant for Familiarity. This is shown graphically in Figure 6-4, where it can be seen that there was no interaction between the two groups, but the difference between the mean ratings for the two groups was significant ($p = .005$). The results of the ANOVA tests showed that Familiarity was not significant for the repeated measure (R1*Familiarity) for either task. This implies that the participants found Tool 1 and Tool 2 to be relatively similar in terms of Familiarity. This agrees with the findings of the initial descriptive statistics where the mean ratings for Familiarity for both CASE tools were very similar (Section 5.4.1).

CASE tool used						
Task A						
		Sum of Squares	**df**	**Mean Square**	**F**	**p**
Ease of Learning	R1*Tool	22.44	1	22.44	41.86	.000
Familiarity	Tool	8.56	1	8.56	9.10	.005
Consistency	R1*Tool	13.69	1	13.69	37.06	.000
Predictability	R1*Tool	13.49	1	13.49	30.05	.000
Informative Feedback	R1*Tool	13.01	1	13.01	12.06	.001
Error Handling	R1*Tool	6.41	1	6.41	9.19	.004
Overall reactions	R1*Tool	7.81	1	7.81	16.47	.000
Task B						
Ease of Learning	R1*Tool	10.62	1	10.62	16.68	.000
Consistency	R1*Tool	14.80	1	14.80	24.91	.000
Predictability	R1*Tool	13.10	1	13.10	28.31	.000
Informative Feedback	R1*Tool	3.62	1	3.62	5.50	.024
Error Handling	R1*Tool	6.88	1	6.88	6.75	.013
Overall reactions	R1*Tool	6.45	1	6.45	10.45	.003

Table 6-5 Significant ANOVA results for CASE tool used

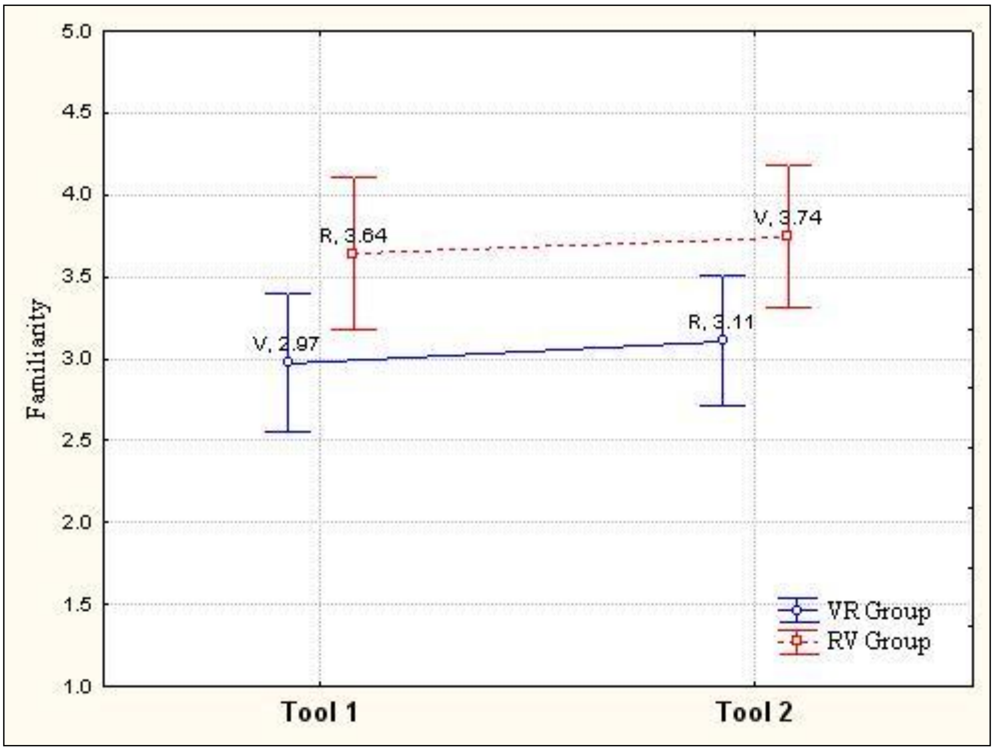

Figure 6-4 Relationship between CASE tool and Familiarity for Task A (n = 46)

The repeated measure for the tool used (R1*Tool) was significant for the following attributes for both Task A and Task B:

- Ease of Learning;
- Consistency;
- Predictability;
- Informative Feedback;
- Error handling; and
- Overall Reactions.

This means that for both tasks, for all of the learnability attributes (except Familiarity), there was an interaction in the mean ratings of both CASE tools from performing the task in the first tool and then in the second tool.

The behaviour for all the significant attributes listed was the same. This behaviour showed that the mean ratings for the two CASE tools, Rational and Visio, were different irrespective of the order in which the tool was used. For each of these attributes showing an interaction, the VR group rated their Tool 1 (Visio) higher than their Tool 2 (Rational). The RV group rated their Tool 1 (Rational) lower and their Tool 2 (Visio) higher. This means that both groups rated Visio higher than Rational.

Since the behaviour patterns were similar for all the learnability attributes, only the graphs for Ease of Learning are shown for Task A in Figure 6-5 and for Task B in Figure 6-6. The graphs for the remaining significant interactions are included in Appendix P. For Task A, the most significant interactions were for Ease of Learning ($p = .000$, $F = 41.80$), Consistency ($p = .000$ and $F = 37.06$) and Predictability ($p = .000$, $F = 30.05$). For Task B, the attributes with the most significant interaction were Predictability ($p = .000$, $F = 28.31$), Consistency ($p = .000$ and $F = 24.91$) and Ease of Learning ($p = .000$ and $F = 16.68$).

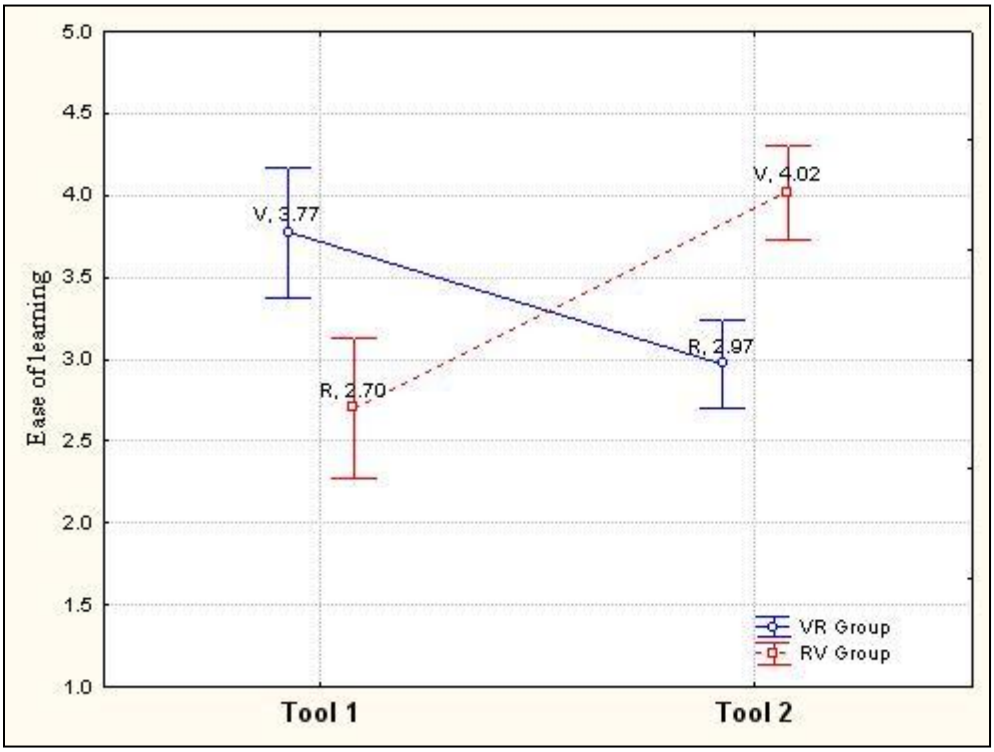

Figure 6-5 Relationship between CASE tool and Ease of learning for Task A (n = 46)

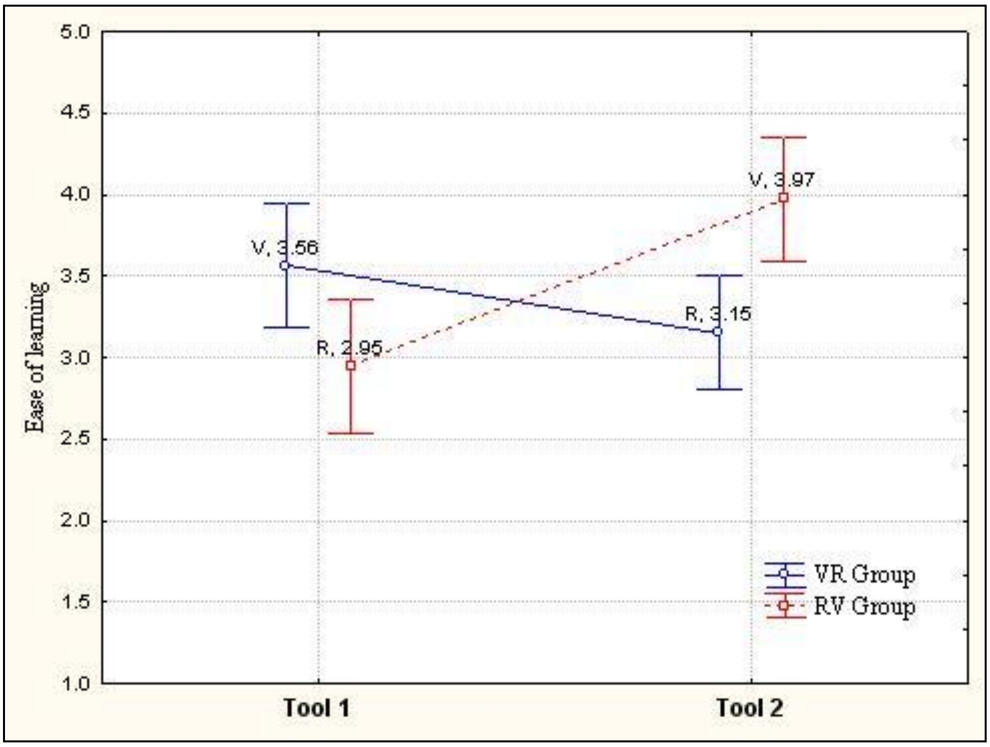

Figure 6-6 Relationship between CASE tool and Ease of learning for Task B (n = 46)

It can be deduced therefore that for every attribute (except Familiarity) for both tasks, Visio was the preferred CASE tool, irrespective of the order in which the CASE tools were used.

This supports the results obtained from the initial descriptive statistics which showed that Visio was rated significantly higher than Rational for all the learnability attributes, except Familiarity (Section 5.4.1).

Since there were only two CASE tools used, post-hoc tests were not required. The results therefore clearly show that there is a relationship between the CASE tool used and learnability.

6.3 Hypotheses Testing

This section discusses the different hypotheses (Table 4-1) as well as the motivation why each hypothesis should be accepted or rejected.

6.3.1 $H_{0.1}$: No learnability problems exist at NMMU in either of the selected CASE tools.

As can be seen in Section 5.5.2, several learnability problems were encountered by participants in both CASE tools (Microsoft Visio and Rational Software Modeller), therefore $H_{0.1}$ can be rejected.

6.3.2 $H_{0.2}$: No relationship exists between the learnability of a CASE tool at NMMU and the type of tool used.

As can be seen from the initial results reported in Sections 5.4.1 to 5.4.4, the learnability of the two CASE tools was not rated equally. This was confirmed by the more detailed repeated measures ANOVA tests showing that the CASE tool used has an effect on learnability (Section 6.2.3). One can thus conclude that $H_{0.2}$ should be rejected.

6.3.3 $H_{0.3}$: No relationship exists between the learnability of a CASE tool at NMMU and the user characteristics.

As shown in Section 6.3.1, there was a relationship between home language and frequency of computer use, on the learnability of the two CASE tools, for some of the learnability attributes. This means that both hypotheses $H_{0.3.1}$ and $H_{0.3.2}$ should be rejected. Thus $H_{0.3}$ can be rejected.

6.3.4 Summary of hypotheses

Table 6-6 summarises the research hypotheses and includes an indication whether the hypotheses can be rejected or not.

Number		**Hypotheses**	**Result**	**Research Question**
$H_{0.1}$		No learnability problems exist at NMMU in either of the two CASE tools.	Rejected	4
	$H_{0.1.1}$	No learnability problems exist at NMMU in Microsoft Visio.	Rejected	4
	$H_{0.1.2}$	No learnability problems exist at NMMU in Rational Software Modeller.	Rejected	4
$H_{0.2}$		No relationship exists between the learnability of a CASE tool at NMMU and the type of tool used.	Rejected	5
$H_{0.3}$		No relationship exists between the learnability of a CASE tool at NMMU and the user characteristics.	Rejected	5
	$H_{0.3.1}$	No relationship exists between the learnability of a CASE tool at NMMU and the user's home language.	Rejected	5
	$H_{0.3.2}$	No relationship exists between the learnability of a CASE tool at NMMU and the user's frequency of computer use.	Rejected	5

Table 6-6 Summary of hypotheses and results

6.4 Conclusions

The results of the repeated measures ANOVA tests revealed that a relationship exists between two of the user characteristics and learnability, namely home language and frequency of use. Significant results were found for home language for the Familiarity and Predictability attributes for Task A (Section 6.3.1). Further post-hoc Sheffe tests revealed that home language was not significant for Familiarity. There was an interaction between the mean ratings for Predictability for two of the language groups (the African and Afrikaans language groups), from Tool 1 to Tool 2 in Task A. The mean ratings for Predictability for the English group remained fairly constant from one tool to the other, which could be due to the fact that the language used by the user interface of both CASE tools is English.

Since the statistics showed that there was a relationship between the home language of participants and some of the learnability attributes, the hypothesis $H_{0.3.1}$ *"No relationship exists between the learnability of a CASE tool at NMMU and the user's home language"* was rejected.

Initial results from the repeated ANOVA tests revealed that for both tasks, frequency of computer use was significant for only one attribute, namely Error Handling (Section 6.3.2). The interaction occurred between the Low and the High frequency group, for both Task A and Task B. The hypothesis $H_{0.3.2}$ *"No relationship exists between the learnability of a CASE tool at NMMU and the user's frequency of computer use"* was therefore rejected. A larger sample size is required in order to obtain more detailed results regarding which groups were significant. $H_{0.3}$ was thus rejected as both $H_{0.3.1}$ and $H_{0.3.2}$ were rejected.

The repeated measures ANOVA tests performed on the CASE tools confirmed the results obtained in Chapter 5 which showed that there is a definite relationship between the CASE tool used and learnability (Section 6.3.3). The hypothesis $H_{0.2}$ "No relationship exists between the learnability of a CASE tool at NMMU and the type of tool used" was thus rejected.

The hypothesis $H_{0.1}$ which states that *"No learnability problems exist at NMMU in either of the two CASE tools"* was rejected (Section 6.3), since several learnability problems were encountered by all participants with both Microsoft Visio and Rational Software Modeller.

The hypothesis H_0 which states that *"No relationship exists between the learnability of a CASE tool at NMMU and the context of use"* was rejected, since all the sub-hypotheses were rejected.

The final chapter revisits the research objectives and summarises the objectives achieved. Problems encountered during the research are discussed. This chapter concludes with future research possibilities and some concluding remarks.

Chapter 7 Recommendations and Conclusions

7.1 Introduction

The aim of this dissertation was to investigate and validate a framework for evaluating the learnability of CASE tools in computing education in South Africa. A secondary aim was to extend this framework to enable the comparison of two CASE tools in a South African context. The objective of this chapter is to briefly revisit what was done in this research project and to make conclusions and recommendations based on the knowledge and insights gained.

7.2 Research objectives revisited

Five main research goals were identified in Section 1.3.4. The first research goal was to identify guidelines for selecting and using OO CASE tools in computing education. The second research goal was to conduct a literature study of standards, principles and frameworks relating to the evaluation of the usability and learnability of CASE tools (Chapter 2). These were studied in order to derive a comprehensive set of CASE tool learnability attributes (Chapter 3). These attributes were required in order to measure the learnability of a CASE tool (Chapter 5), using the usability evaluation methods discussed in Chapter 3. The relationship between the context of use factors and the learnability of a CASE tool was determined by means of an experiment conducted at NMMU (Chapter 6). This experiment consisted of a usability evaluation of two CASE tools. Lastly recommendations to the framework for CASE tool learnability evaluation were proposed so that it can be applied in a South African context (Chapter 7).

Several research questions were posed in Section 1.3.1 in order to address the relevant research goals. In Chapter 2 the background to CASE tools was discussed. This included a definition of a CASE tool as software which can provide automatic support for one or more activities of the SDLC (Section 2.2). Several benefits and problems with CASE tool usage were identified (Sections 2.4 and 2.5). The key benefit of CASE tool usage was considered to be the role of these tools in teaching OOSAD. The main problems with CASE tool usage were identified to be usability and learnability problems. Several guidelines for overcoming these problems were compiled and grouped into three categories, namely tool complexity, ease of use and the learning environment (Section 2.6).

A number of definitions and classifications of usability were discussed and the importance of having measurable usability requirements was highlighted (Section 3.2). Principles which should be applied for CASE tool usability evaluations were reviewed, as well as dialogue principles to be used for evaluating the suitability for learning of CASE tools (Section 3.6). Several methods of usability evaluation were discussed. Usability testing was determined to be the most appropriate method for evaluating the usability of CASE tools (Section 3.7). Two frameworks for evaluating CASE tools were investigated and their shortcomings identified (Section 3.8 and 3.9). The first framework selected was Phillips' framework for evaluating CASE tool usability, as it is relevant to CASE tools in a computing education environment. The second framework was selected because it was designed specifically for evaluating CASE tool learnability.

In order to validate Senapathi's framework in a South African context, an experiment was conducted at NMMU. The participants selected were computing students enrolled for the WRI201 module at NMMU (Section 4.3.2). The CASE tools selected for the usability evaluation were Microsoft Visio and Rational Software Modeller (Section 4.3.1). The tasks the participants were required to perform involved the drawing of two UML diagrams, which are key tasks in OOSAD (Section 4.3.3). The primary evaluation instruments used in this study were questionnaires that recorded both quantitative and qualitative data (Section 4.3.4).

Initial statistics on the responses revealed that for all attributes of learnability (except for Familiarity), Visio scored higher than Rational for both tasks (Section 5.4.1). Visio was by far the preferred tool for both tasks for Overall reactions, except for how stimulating the tool was (Section 5.4.3). Visio was also the preferred CASE tool for both tasks for Ease of Use and Ease of Learning (Section 5.4.4).

Further analysis of the data revealed that there was a relationship between the three context of use factors examined, namely home language, frequency of computer use and the CASE tool, on the participants' ratings of learnability (Section 6.2).

The contributions made by this research, both theoretical and practical, are discussed in Section 7.3 Problems encountered and recommendations are identified in Sections 7.4 and 7.5 respectively.

As can be seen from the discussion in this section, all the objectives of this study were met.

7.3 Research contributions

The contributions of this research can be broken down into theoretical and practical contributions, each of which is discussed in the following sections.

7.3.1 Theoretical contributions

This dissertation has four main theoretical contributions that were produced after an extensive literature study. These are:

- A set of 12 guidelines that should be used when selecting and implementing a CASE tool for computing education (Section 2.6);
- A comprehensive list of 11 attributes that can be used for evaluating the learnability of CASE tools (Table 3-6);
- The analysis and validation of Senapathi's framework for evaluating the learnability of CASE tools (Section 3.9); and
- A set of 15 principles to be applied when evaluating the learnability of OO CASE tools for computing education (Table 7-2).

The analysis and validation of Senapathi's framework included a discussion of the shortcomings of the framework and some recommendations. The first main shortcoming identified in Senapathi's framework was the omission of several learnability attributes. Senapathi's framework lists three main attributes of learnability (Familiarity, Consistency and Predictability), as well as three supporting attributes (Informative Feedback, Error handling and On-Line help) (Figure 3-3). The results of this research support the inclusion of a more comprehensive list of 11 learnability attributes in the framework (Section 5.5.1). These attributes are listed in Table 7-1 together with a comparison to Senapathi's attributes. The five attributes that were added to Senapathi's list are printed in blue in the table.

Ease of learning and Simplicity were added as learnability attributes in the framework as a result of the literature review (Section 3.9.3). The research results also confirmed that these attributes contribute to learnability. The literature review showed that the learnability attributes Observability, Quality of presentation and Quality of content all contribute to Feedback (Section 3.4.2).

For this reason the Feedback attribute in Senapathi's framework was further broken down into these three attributes. Error prevention was also identified as contributing to learnability and was added to the list of learnability attributes (Section 3.4.2.2).

LEARNABILITY	
Attribute	**Senapathi's attribute**
Familiarity	Familiarity
Consistency	Consistency
Predictability	Predictability
Ease of learning	NEW
Simplicity	NEW
Observability	
Quality of presentation	Feedback
Quality of content	
Error prevention	NEW
Recoverability	Error handling
Provision of help	On-line Help

Table 7-1 Comparison of learnability attributes

Other shortcomings in Senapathi's framework related to the questionnaire used for evaluating the learnability of the CASE tool. One of the attributes in Senapathi's framework, On-line Help, had no related questions in the questionnaire and could therefore not be correctly measured. Other questions were worded ambiguously or the question content was unrelated to the learnability attribute in that section of the questionnaire. In addition the research design did not allow for the testing of the Tool part of the framework, as only one CASE tool was evaluated.

A modified framework for evaluating CASE tool learnability is proposed in Figure 7-1. The modifications include the amendment of the learnability attributes identified, as well as the addition of the *Learnability Requirements* block and the *Learnability Principles* block. All modifications are shown in the figure in blue and in italics.

The *Learnability Requirements* block includes the learnability attributes, but should also include evaluation measures such as a metric, target level and minimum acceptable level for each attribute (Section 3.2). These were omitted from Senapathi's framework; thus they have been added to the updated framework.

The *Learnability Principles* block highlights the importance of providing these principles to evaluators prior to the usability evaluation. Principles for each learnability attribute should be identified as these could assist in providing the evaluators with a better understanding of the specific requirements for CASE tools and thereby result in more accurate evaluation results (Section 3.5). Table 7-2 lists the learnability principles which should be used to evaluate OO CASE tools in computing education.

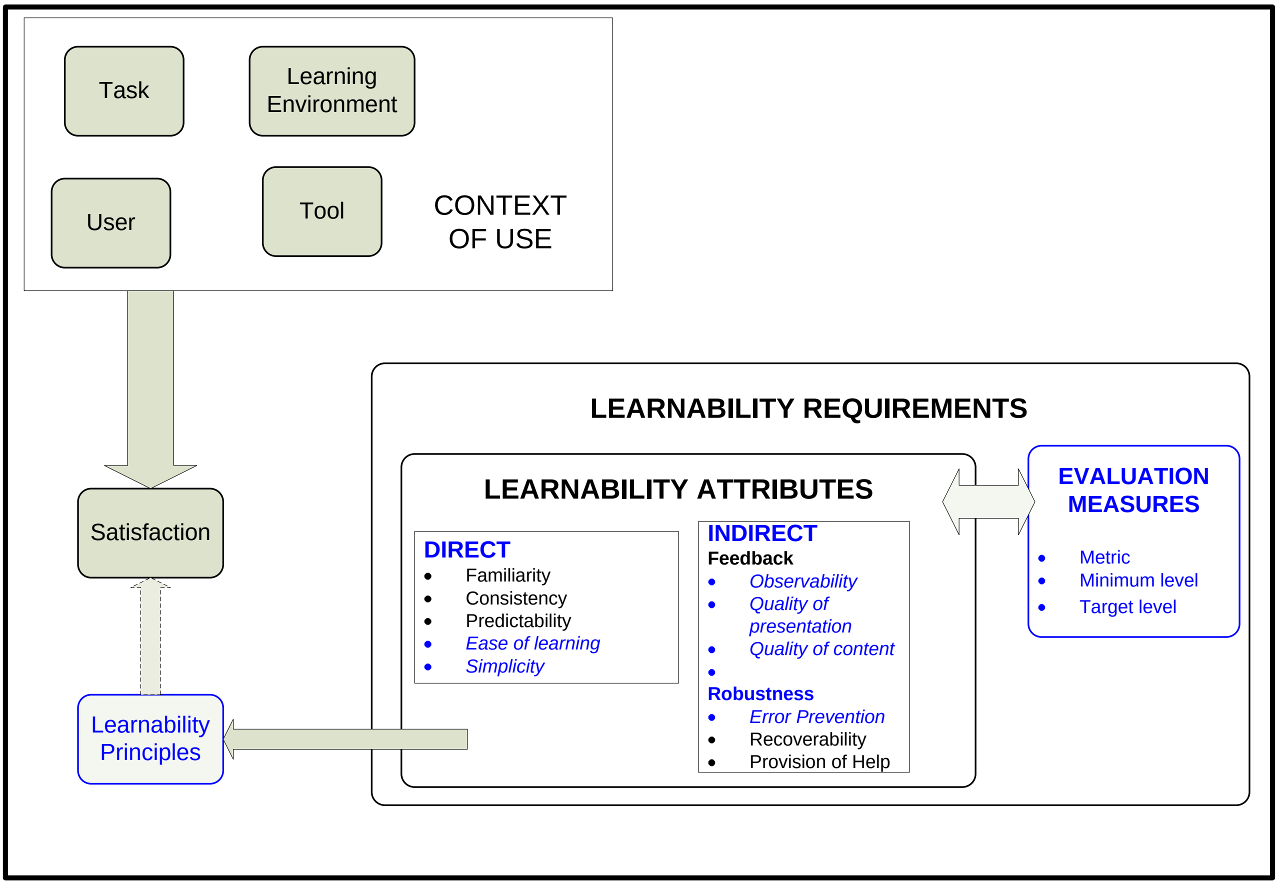

Figure 7-1 Modified framework for evaluating CASE tool learnability

The dotted arrow drawn between the Learnability Principles block and the Satisfaction block indicates that providing evaluators with principles of learnability could affect the rating of satisfaction and provide more accurate results which are consistent with the learnability requirements specified. The arrow is dotted so as to indicate that its relationship is not as strong as the relationships indicated by the other arrows.

Learnability Attribute	**No**	**Principle**
Familiarity	1	Appropriate support should be provided to assist the user in becoming familiar with the user interface of the CASE tool.
Consistency	2	The logic within the CASE tool, at the procedural level as well as for the presentation of information, should be consistent. The CASE tool should use controls and function keys consistently throughout the application.
Predictability	3	The semantics of the dialogue steps used to control the CASE tool should reflect the resulting action, and the predictability of the action.
Ease of learning	4	The amount of time and effort required for a user to understand normal CASE tool operations and to become productive should be acceptable. A computing education student should be able to become productive after a one-hour introductory workshop.
Simplicity	5	The system should enable the user to perform the tasks with minimal learning by entering only the minimum amount of information required and using the minimum number of steps.
Observability	6	The user should know the status of the CASE tool at all times and should easily be able to evaluate the internal state of the system from its external representation
Quality of content	7	Sufficient feedback should be provided about the intermediary and final results of an activity so that the user learns from successfully accomplished activities. This should differ between novice and experienced users. When a user draws a UML diagram, the user should receive step-by-step feedback to refine his/her diagram and details about any potential UML violations.
Quality of presentation	8	The CASE tool user interface must be clear, subjectively pleasing and well structured.
Error prevention	9	The CASE tool should maintain tool integrity and should not allow the user to violate UML rules.
	10	The CASE tool should allow the user to explore dialogue steps without negative consequences. A CASE tool should allow a user to evaluate potential variations of a UML diagram to enable the user to see negative impacts before changes are applied.
	11	The CASE tool should provide an “Undo” facility.
Recoverability	12	The CASE tool should help and guide the user in identifying and correcting errors. A CASE tool should provide error messages for UML violations. These should be easy to understand and displayed as a confirmation message to ensure that the user has seen the message.
Provision of help	13	A tutorial on UML concepts and on how to draw the main UML diagrams should be available.
	14	A tutorial on how to use the CASE tool should be available.
	15	A CASE tool should explain the use of individual functions when the user presses the help key. It should provide a help system that guides the user through the required steps of creating a new UML diagram.

Table 7-2 Principles for evaluating CASE tool learnability in computing education

A revised post-test learnability questionnaire was compiled from the principles in Table 7-2. This could be used as an instrument to measure the learnability attributes of OO CASE tools in computing education environments. A copy of this revised post-test learnability questionnaire is included in Appendix Q. The modifications made to this questionnaire included the addition of several questions for each additional learnability attribute proposed in the updated framework. Each section maps directly to one of the learnability attributes in the updated framework. These questions were derived from the literature review as well as from the list of principles to be used for evaluating CASE tool learnability (Table 7-2).

7.3.2 Practical contributions

In order to validate Senapathi's framework, a usability evaluation of two CASE tools was performed at NMMU. The results of these usability tests clearly validate Senapathi's work and confirmed her hypothesis that there is a relationship between certain context of use factors and the learnability of a CASE tool. The three context of use factors tested in this study, namely home language, frequency of computer use and the CASE tool used, were not tested by Senapathi. Further extensions to Senapathi's study made by this research, allowed the comparison of two CASE tools of different complexity. Results obtained from the testing of these extensions to the framework support the inclusion of the Tool aspect of the framework, since the results showed a relationship between the tool used and the learnability ratings.

The qualitative research results, specifically in the area of error handling, are a significant practical contribution of this study. These results showed that participants had problems with the way in which both CASE tools handled UML errors, even though the approaches were quite different. Rational used error prevention and did not allow the participants to perform any erroneous UML connections. This frustrated the participants as they did not know why they were not allowed to perform the action. Rational's design could be improved by displaying an error message relating to the erroneous connection rather than preventing it altogether (Section 5.5.1.1).

Visio's problems with error handling related to the fact that although the erroneous UML connection was highlighted in red, the participants were not aware of the reason for this as the error message did not use language which was easy to understand and was also hidden at the bottom of the screen. One way of overcoming this problem would be to use a confirmation message (Section 5.5.1.1).

The differences in users' reactions to the two approaches to error handling, as well as any proposed improvements in design, will require further research before any conclusions can be made. More detailed studies will have to be undertaken in order to determine how to resolve this problem and the related issues.

Other contributions made by this study include improved approaches to teaching OO CASE tools which could result from following the recommended guidelines relating to the learning environment of CASE tools (Section 2.6). These include using UML within the context of a methodology, integrating the use of the CASE tool into real-life projects, providing appropriate training and understanding the learning styles of the students.

In addition, knowledge of the factors affecting CASE tool learnability (such as home language) could assist with the understanding of these problems and the selection of appropriate teaching methods. It can also assist with the selection of a more suitable CASE tool for South African students. Selecting a CASE tool that is easier to learn will improve the rate of learning for students and improve their understanding of UML. The results from this study have also shown that Simplicity and Familiarity should be key issues when selecting a CASE tool.

7.4 Problems encountered and limitations identified

Several problems were encountered during the course of this study. The first of the problems related to the actual sample size used in the final experiment, which was not as large as one would of hoped and was too small to measure all the possible relationships for some of the repeated measures ANOVA tests. Using students as participants exposed the study to the risk of the normal behaviour of students of missing lectures and practicals; this resulted in the reduction of the initial sample size from 62 to 46.

All the participants were students from the same module; thus all of them had already passed the first year of computing education, and any previous differences in experience and frequency of computer use may have been reduced. This could account for why these user characteristics had a significant result for only one learnability attribute, namely Error Handling (Section 6.2.2).

This study could be repeated in a commercial environment where the analysts and developers come from different HEIs and will have different experience and frequency of computer use profiles.

7.5 Recommendations

7.5.1 Recommendations for theory

The literature study identified that measurable usability requirements should be specified in order to evaluate the usability and the learnability of CASE tools (Section 3.2). These are not taken into account in Senapathi's framework. It is therefore recommended that the extended framework (Figure 7-1) be used to evaluate the learnability of OO CASE tools in computing education. This framework includes the comprehensive list of learnability attributes identified in Table 7-1. This list of learnability attributes should thus also be taken into account when designing an OO CASE tool, as well as when evaluating it. In particular, the quantitative results of this study showed that the attributes of learnability that were rated highest by the students for the preferred CASE tool were Predictability, Consistency and Ease of learning. These are thus desirable attributes in a CASE tool and should be included in the design of such a tool.

7.5.2 Recommendations for practice

The results of this study can be used by developers of CASE tools to incorporate facilities to provide guidance and feedback to users which will assist them in preventing typical novice UML errors during the modeling process. It is important to design a CASE tool which prevents the user from making errors, but which does not constrain them. One way of doing this would be to prevent an erroneous UML connection by displaying a confirmation message prior to preventing the error so that the user is provided with feedback as to why the action is restricted. The error messages must be expressed using simple language that is not technical and can be clearly understood.

Lecturers involved with teaching OOSAD in computing education can use the guidelines provided for selecting OO CASE tools. They can also use the proposed framework (Figure 7-1) when evaluating CASE tool learnability. This framework includes specifying CASE tool learnability requirements prior to performing a learnability evaluation. These requirements should include a list of learnability attributes, as well as the evaluation measures. The evaluation measures include metrics, minimum acceptable levels and target levels for each attribute. The participants should also be provided with criteria or principles for evaluating these attributes. Using this framework should improve the accuracy of the evaluation results. The results of the evaluation can then be used to select the CASE tool with the highest rated satisfaction ratings for learnability.

This could result in reduced learning rates for students and reduce the risks of problems that students face with learning to use a CASE tool.

7.5.3 Recommendations for future research

Several possibilities and opportunities exist for future research, some of which are discussed in this section.

7.5.3.1 Measuring effectiveness and efficiency of learnability

Insufficient empirical research has been done on the topic of measurable usability requirements (Jokela *et al.* 2006). This study has focused on measuring satisfaction. Other indicators of learnability besides satisfaction, such as effectiveness and efficiency, can be measured. These indicators could include the following metrics (ISO 1998, 2001):

- For effectiveness:
  - The number of functions learned; and
  - The percentage of users who managed to learn the function after reading the product description.
- For efficiency:
  - The time taken to learn how to use particular functions; and
  - The effort required to learn one operation – the ratio of time required to learn one operation for a specific task and operation time.

7.5.3.2 Research on error handling

Although this study revealed some interesting qualitative data on the importance of error handling and error messages in CASE tool learnability, it was not conclusive enough to make final recommendations in these areas. Further research on error prevention, recoverability and the provision of help in CASE tools could thus be done which would provide a valuable contribution to this field. A larger sample size could allow for testing if a relationship exists between various context of use factors and the participants' rating of the Recoverability aspect of learnability.

Modifications to Rational could be made with respect to error prevention of UML errors so that it displays an error message explaining why the action is prevented.

A usability evaluation could then be performed which compares the original method of error prevention with the new approach in order to determine which method the users prefer.

Future research could also be undertaken with regard to the proposed change to Visio's error handling approach. Visio could be changed so that it uses a confirmation message when an erroneous UML connection is made. A usability evaluation could then be performed which compares the original method of displaying the error message at the bottom of the screen, to the new approach in order to determine which method the users prefer.

7.5.3.3 Research on the learning environment

Further research could be undertaken regarding the relationship between the learning environment of computing students and CASE tool learnability. This study identified several factors in the learning environment which could affect CASE tool usage, namely the inclusion of the CASE tools in real-life projects, the training provided and the learning styles of the users (Section 2.6.3). The research could also include an investigation into other possible factors in the learning environment that could affect the learnability of CASE tools.

7.5.3.4 Other Research opportunities

Since the sample size of this study was relatively small (n = 46) and was limited to NMMU, this research could be repeated on a larger scale at other HEIs in South Africa. The CASE tools market in South Africa is growing and such a study should be repeated in order to obtain more results. In this way, it might become possible to make predictions regarding the learning behaviour of computing education students using OO CASE tools.

Other research opportunities exist to identify the target and minimum acceptable levels for each learnability attribute for CASE tools in computing education.

o--------------- End -------------------o

REFERENCES

ALFERT, K., PLEUMANN, J. and SCHRODER, J. (2004): Software Engineering Education needs adequate modelling tools. In *Proceedings 17th Conference on Software Engineering Education and Training (CSEET'04)*. 72-77, IEEE CNF.1-3 March, 2004.

APPLIN, A.G. (2001): Second Language Acquisition and CS1: Is * ==**? *ACM SIGCSE Bulletin* 33(1):174-178.February 2001.

AUER, M., TSCHURTSCHENTHALER, T. and BIFFL, S. (2003): A Flyweight UML modelling tool for software development in Heterogeneous environments. In *Proceedings 29th EUROMICRO Conference "New waves in system architecture" (EUROMICRO '03)*. 267-272, IEEE CNF.1-6 September, 2003.

BARNUM, C. (2002): *Usability Testing and Research*. Pearson.

BEVAN, N. (2001): International Standards for HCI and Usability. *International Journal for Human-Computer Studies* 55:533-552.April 2001. http://www.idealibrary.com

BOLLOJU, N. and LEUNG, F. (2006): Assisting novice analysts in developing quality conceptual models with UML. *Communications of the ACM* 49(7):2772-2782.July 2006.

BOOCH, G., RAMBAUGH, J. and JACOBSON, I. (1999): *The Unified Modelling Language User Guide*. Addison-Wesley.

BREWER, J. and LORENZ, L. (2003): Using UML and Agile Development Methodologies to Teach Object-Oriented Analysis & Design Tools and Techniques. In *Proceedings CITC4'03*, Lafayette, Indiana, USA. 54-57, ACM.October 16-18, 2003.

BROOKS, A. and SCOTT, L. (2001): Constraints in CASE tools: results from curiosity driven research. In *Proceedings 13th Australian Software Engineering Conference (ASWEC'01)*. 285-293, IEEE CNF.27-28 August, 2001.

BURTON, P. and BRUHN, R. (2004): Using UML to facilitate the teaching of object-oriented systems analysis and design. *Journal of Computing Sciences in Colleges* **19**(3):1937-4771.January 2004.

CHOI, G. and KIM, T. (2004): UML CASE Tool Supporting OLE Compound Document. In *Proceedings 11th Asia-Pacific Software Engineering Conference (APSEC'04)*. 250-256, IEEE CNF.30 November - 3 December 2004.

COSTAIN, G. (1998): A framework for post-implementation review of use of CASE methodology. Masters Thesis thesis. MSIS, University of Auckland. Auckland.

DAVIES, I., GREEN, P., ROSEMANN, M., INDULSKA, M. and GALLO, S. (2006): How do practitioners use conceptual modeling in practice? *Data & Knowledge Engineering* 58(3):358-380.September 2006.

DEE MEDLEY, M. (2001): Using qualitative research software for CS education research. *ACM SIGCSE Bulletin* 33(3):141-144.

DIX, A., FINLAY, J., ABOWD, G.D. and BEALE, R. (2004): *Human-Computer Interaction*. Third Edn, Prentice Hall.

DOBING, B. and PARSONS, J. (2006): How UML Is Used. *Communications of the ACM* 49(No 5).May 2006.

DOUGLAS, E. and HARDGRAVE, B. (2000): Object-Oriented Curricula in Academic Programs. *Communications of the ACM*.November 2000.

DUMAS, J.S. and REDISH, J. (1999): *A practical guide to usability testing*. England, Exeter, Revised Ed, Intellect.

ELY, M., ANZUL, M., FRIEDMAN, T., GARNER, D. and MACCORMACK STEINMETZ, A. (1995): *Doing Qualitative Research: Circles Within Circles*. Falmer Press.

ELY, M., VINZ, R., DOWNING, M. and ANZUL, M. (1999): *On Writing Qualitative Research: Living by Words*. Falmer Press.

FAULKNER (2000): *Usability Engineering*. London, MacMillan Press Ltd.

FELDER, R. and SOLOMAN, B. (2005): Learning Styles and Strategies. http://ww2.ncsu.edu/unity/lockers/users/f/felder/public/ILSdir/styles.htm (Last accessed: September, 2006).

FENSTERMACHER, K. (2004): If I had a Model, I'd Model in the Mornin'. In *Proceedings OOPSLA'04*, Vancouver, British Columbia, Canada. ACM Press.24 - 28 October 2004.

FEYT, L. (2004): A simplified CASE tool for IS students. Honours Treatise thesis. Department of CS & IS, Nelson Mandela Metropolitan University. Port Elizabeth.

FLINT, S., GARDNER, H. and BOUGHTON, C. (2004): Executable/Translatable UML in Computing Education. In *Proceedings Sixth Australasian Computing Education Conference (ACE2004)*, Dunedin, New Zealand. 30, Raymond Lister and Alison Young, Ed.

FOWLER, L., ALLEN, M., ARMAREGO, J. and MACKENZIE, J. (2000): Learning Styles and CASE tools in Software Engineering. In *Proceedings 9th Annual Teaching Learning Forum*, Perth:Curtin University of Technology.2 - 4 February 2000. http://lsn.curtin.edu.au/tlf/tlf2000/folwer.html

FROSCH-WILKE (2003): Using UML in Software Requirements Analysis - Experiences from Practical Student Project Work. In *Proceedings InSITE - Where Parallels Intersect*. Informing Science.June.

GALITZ (2002): *The Essential Guide to User Interface Design*. John Wiley & Sons, Inc.

GOULD, J.D. and LEWIS, C. (1985): Designing for usability: key principles and what designers think. *Communications of the ACM* 28(3):300-311

HOGGARTH, G. and LOCKYER, M. (1998): An Automated Student Diagram Assessment System. In *Proceedings ITiCSE*, Dublin, Ireland. ACM Press

ISO (1995): ISO/IEC 14102: Information Technology: Guideline for the evaluation and selection of CASE tools. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=23593 (Last accessed: December 2006).

ISO (1997): ISO 9241-1: Ergonomic requirements for office work with visual display terminals. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=21922 (Last accessed: October 2006).

ISO (1998): ISO/IEC 9241-11: Guidance on Usability. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=16883 (Last accessed: April 2007).

ISO (2001): ISO/IEC 9126-1: Software Product Evaluation. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=22749 (Last accessed: July 2006).

ISO (2006): ISO9241-110: Ergonomics of human-system interaction - Part 110: Dialogue Principles. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38009 (Last accessed: March 2007).

JANKOWSKI, D. (1995): Case feedback in support of learning a systems development methodology. *Journal of Information Systems Education* 7(3):88-90

JARZABEK, S. and HUANG, R. (1998): The Case for User-Centered CASE tools. *Communications of the ACM* 41(8):93-99.August 1998.

JOHNSON, H.A. and WILKINSON, L. (2003): CASE tools in object-oriented analysis and design. In *Proceedings Eastern Conference: Consortium for Computing Services in Colleges*.

JOKELA, T., IIVARI, N., MATERO, J. and KARUKKA, M. (2003): The Standard of User-Centered design and the Standard Definition of Usability:Analyzing ISO 13407 against ISO 9241-11. In *Proceedings Latin American Conference on HCI CLIHC'03*. ACM Press.August 2003.

JOKELA, T., KOIVUMAA, J., PIRKOLA, J., SALMINEN, P. and KANTOLA, N. (2006): Methods for quantitative usability requirements: a case study on the development of the user interface of a mobile phone. *Personal and Ubiquitous computing* 10(6):345-355.

JURIC, R. and KULJIS, J. (1999): Building an Evaluation Instrument for OO CASE tool Assessment for Unified Modelling Language Support. In *Proceedings 32nd Hawaii International Conference on System Sciences*, Hawaii. 10, IEEE.May 1999.

KEMERER, C. (1992): How the Learning Curve Affects CASE tool adoption. *IEEE Software* 0740-0759/92/0500/0023.May 1992.

KITCHENHAM, B. and PFLEEGER, S.L. (1996): Software quality: the elusive target [special issues section]. *Software, IEEE* 13(1):12 - 21.January 1996.

LENDING, D. and CHERVANY, N. (1998): CASE tools: Understanding the Reasons for Non-Use. *ACM SIGCPR conference on computer personnel research SIGCPR'98* 19(2).April 1998.

LINDER, N., STEELE, P. and DORLING, R. (1994): Using a commercial integrated CASE tool in an undergraduate computing degree. In *Proceedings Software Education Conference*. 212-218.22-25 November, 1994.

MACCARI, A. and RIVA, C. (2002): On CASE tool usage at Nokia. In *Proceedings 17th IEEE International Conference on Automated Software Engineering (ASE '02)*.

MANNINO, M. (2001): *Database Application Development and Design*. McGraw-Hill.

MOSLEY, V. (1992): How to Assess Tools Efficiently and Quantitatively. *IEEE Software* 9(3):29-32.May 1992.

MOUTON, J. (2001): *How to succeed in your Master's & Doctoral studies*. Pretoria, Van Schaik Publishers.

MYNATT, B.T. and LEVENTHAL, L.M. (1990): An evaluation of a CASE-based approach to teaching undergraduate software engineering. In *Proceedings 21st SIFCSE technical symposium on Computer Science Education*, Washington DC, United States. 48-52.1990.

NIELSEN, J. (1993): *Usability engineering*. Academic Press.

NIELSEN, J. (1994): Ten Usability Heuristics. www.useit.com (Last accessed: February 2007).

NIELSEN, J. (1995): Getting Usability Used. In *Proceedings Human Computer Interaction, Interact '95*. 3 - 12.

NIELSEN, J. (2007): How to conduct a heuristic evaluation. http://www.useit.com/papers/heuristic (Last accessed: April 2007).

NIELSEN, J. and LANDAUER, T.K. (1993): A Mathematical Model of the Finding of Usability Problems. In *Proceedings INTERCHI'93*, Amsterdam, The Netherlands. 206 - 213, ACM Press.1993.

NIELSEN, J. and LEVY, J. (1994): Measuring Usability: Preference vs. Performance. *Communications of the ACM*, ACM. (April 1994, Vol 37 No 4).

NORMAN, D.A. (1999): Affordance, conventions and design. In *Proceedings SIGCHI Interactions*. 6(3):38-42.May 1999.

NUNNELLY, J. (1978): *Psychometric Theory*. New York, Second Edn, McGraw-Hill.

PHILLIPS, C., MEHANDJISKA, D., GRIFFIN, D., CHOI, M.D. and PAGE, D. (1998): The usability component of a framework for the evaluation of OO CASE tools. In *Proceedings Software Engineering, Education and Practice*, Dunedin. 134-141, IEEE.26 - 29 January 1998.

POST, G. and KAGAN, A. (2000): OO-CASE tools: an evaluation of Rose. *Information and Software Technology* 42:383-388.15 April 2000.

PREECE, J., ROGERS, Y. and SHARP, H. (2002): *Interaction Design beyond human-computer interaction*. John Wiley & Sons, Inc.

PREECE, J., ROGERS, Y. and SHARP, H. (2007): *Interaction Design: beyond human-computer interaction*. 2nd Edn, John Wiley & Sons Ltd.

ROSENBERG, D. (1999): *Use Case Driven Object Modelling with UML: A Practical Approach*. Addison-Wesley.

ROSSON, M. and CARROLL, J. (2002): *Usability Engineering*. Morgan Kaufmann Publishers.

RUBIN, J. (1994): *Handbook of usability testing*. New York, John Wiley & Sons.

SAP (2007): Simplifying for Usability. http://www.sapdesignguild.org/resources/simplification/index.htm (Last accessed: February 2007).

SATZINGER, J., JACKSON, R. and BURD, S. (2005): *Object-Oriented Analysis and Design with the Unified Process*. Thomson Course Technology.

SAURO, J. and KINDLUND, E. (2005): A Method to Standardize Usability Metrics Into a Single Score. In *Proceedings CHI 2005*, Portland, Oregon, USA. 407-409, ACM.2 - 7 April 2005.

SCHOLTZ, B. (2003): A Comparative Analysis of CASE Tools. Honours Treatise thesis. Department of CS & IS, University of Port Elizabeth. Port Elizabeth.

SEFFAH, A., DONYAEE, M., KLINE, R. and PADDA, H.K. (2006): Usability measurement and metrics:A consolidated model. *Software Quality Journal* 14:159-178.

SEFFAH, A. and RILLING, J. (2001): Investigating the relationship between usability and conceptual gaps of human-centric CASE tools. In *Proceedings IEEE Symposia on Human-centric computing languages and environments (HCC'01)*. 226-231, SOFTWARE, I. (ed).

SENAPATHI, M. (2005): A Framework for the Evaluation of CASE tool learnability in Educational Environments. *Journal of Information Technology Education* 4.

SERC, S.E.R.C.-. (2007): QUINT2 The Extended ISO Model of Software Quality. www.serc.nl/quint-book/ (Last accessed: February 2007).

SERVICES, U.S.D.O.H.H. (2007): Heuristic Evaluation. http:/www.usability.gov/methods/heuristicseval.html (Last accessed: July 2007).

SHNEIDERMAN, B. (1998): *Designing the user interface: Strategies for effective human computer interaction.* Addison-Wesley.

SHNEIDERMAN, B. and PLAISANT, C. (2005): *Designing the User Interface*. Fourth Edn.

SOMMERVILLE, I. (2001): *Software Engineering*. Sixth Edn, Addison Wesley.

SONG, I. (2001): A Heuristic for Developing Object Interaction Diagrams. In *Proceedings 2001 Informing Science Conference*, Krakow (Poland). 487-492.2001.

TABRIZI, M., COLLINS, C., OZAN, E. and LI, K. (2004): Implementation of Object-Orientation Using UML in Entry Level Software Development Courses. In *Proceedings SIGITE'04*, Salt Lake City, Utah, USA. 128-131.28 - 30 October 2004.

TAENTZER, G. (2007): Overview On Tools. http://wwwcs.upb.de/cs/ag-engels/ag_engl/Segravis/school/overview.pdf (Last accessed: April 2007).

TE'ENI, D., CARY, J. and ZHANG, P. (2007): *Human Computer Interaction:Developing Effective Organisational Information Systems*. John Wiley & Sons, Inc.

TORCHIANO, M., JACCHERI, L., SøRENSON, C. and WANG, A. (2002): COTS Products Characterization. In *Proceedings SEKE '02*, Ischia, Italy. 1-58113-556-4:335-338, ACM

USABILITY.GOV (2007): Heuristic Evaluation. http:/www.usability.gov/methods/heuristicseval.html (Last accessed: July 2007).

VAN DER STRAETEN, R. (2002): Semantic Links and Co-Evolution in Object-Oriented Software Development. In *Proceedings 17th IEEE Conference on Automated Software Engineering (ASE '02)*. 317.

WANIA, C.E., ATWOOD, M.E. and MCCAIN, K.W. (2006): How do Design and Evaluation Interrelate in HCI Research. In *Proceedings DIS 2006*, University Park, Pennsylvania, USA. ACM Press.2006.

WHITTEN, B. (2001): *Systems Analysis and Design Methods*. McGraw-Hill.

Appendices

Appendix A – Task A Instructions

Group: VR **Task Number: A** **Task Name:** Use Case diagram

Required:

1. You need to draw a Use Case diagram for the Book Club system, using two CASE tools in order to evaluate the learnability of the two tools. After each tool has been used, the relevant questionnaires must be completed. The tools must be used in the following order:
   - First tool to be used: Microsoft Visio
   - Second tool to be used: Rational Software Modeller

2. The use case diagram which must be drawn can be seen in Figure 1 below. You may notice that there are errors in the diagram. Try to draw the diagram as shown and take note of how each respective CASE tool handles these errors.

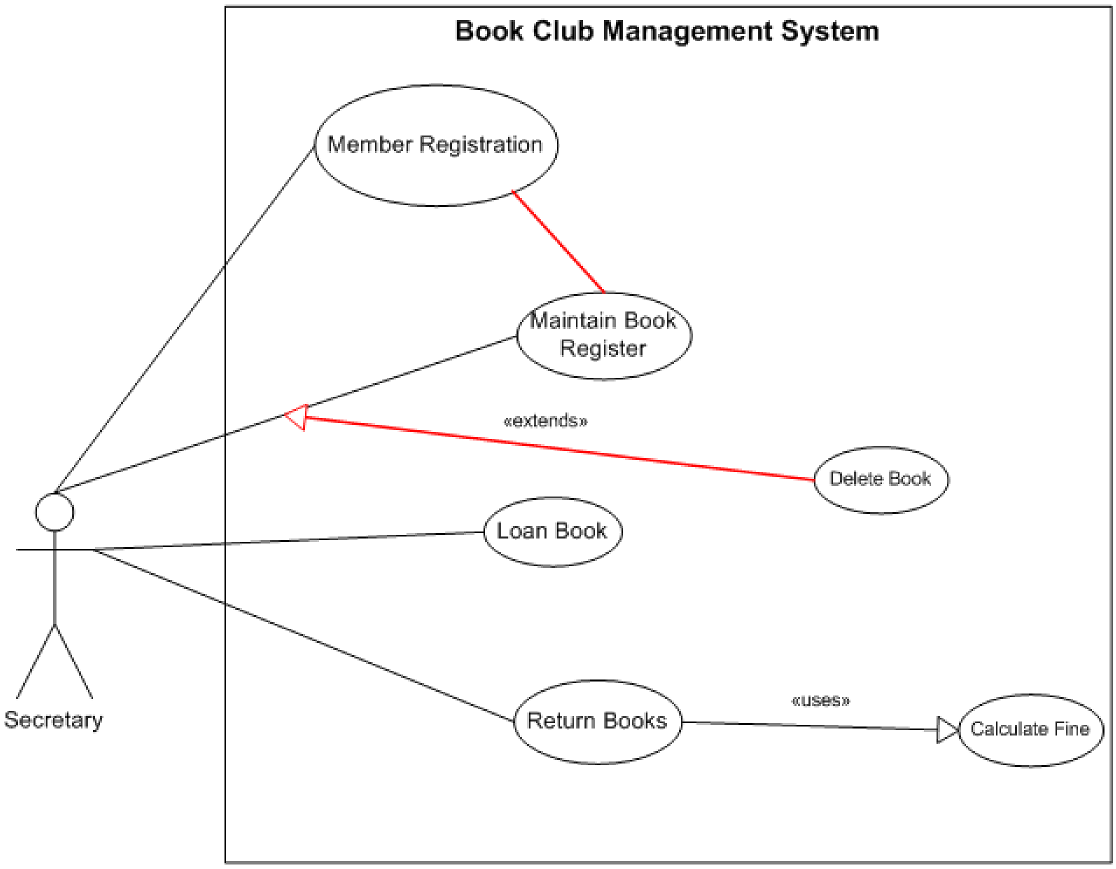

Figure 1 Partial Use case diagram of a Book Club system (with errors)

3. Once you have done this, correct your errors and complete the correct diagram as shown in Figure 2.

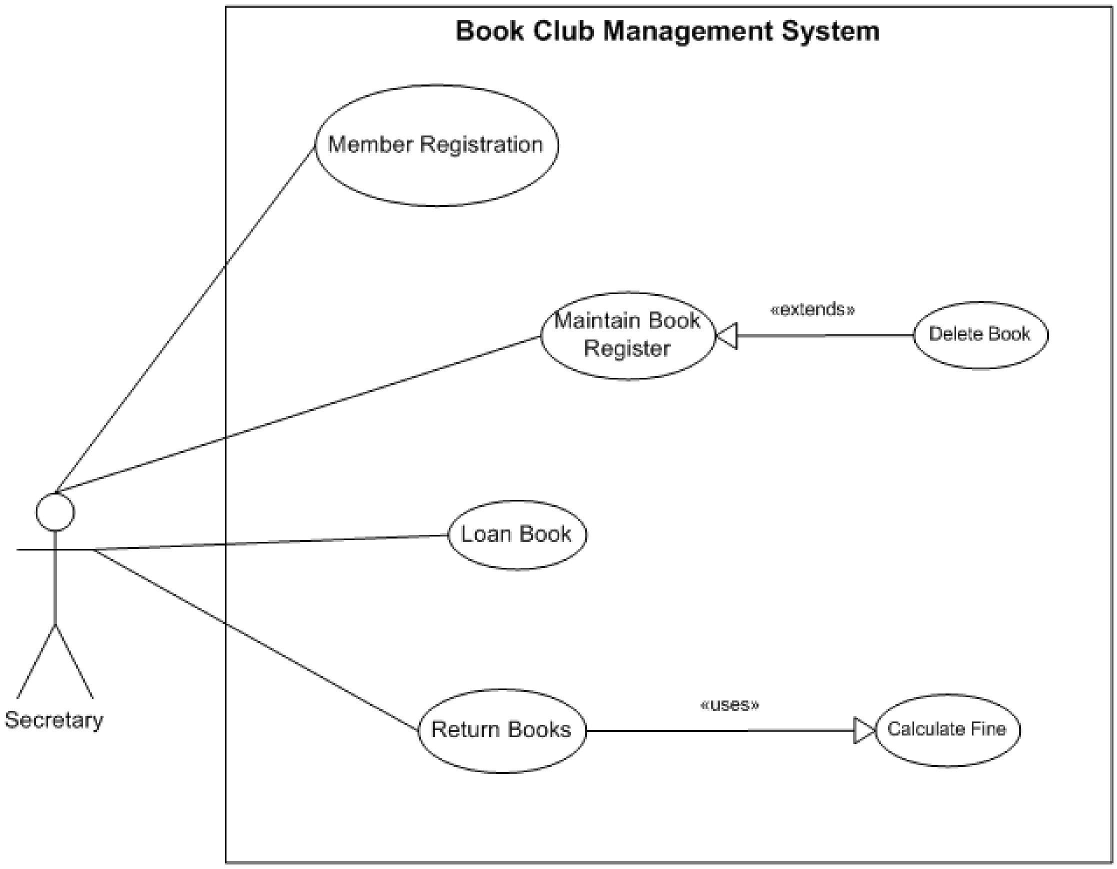

Figure 2 Partial Use Case diagram for the Book Club system (without errors)

Appendix B – Task B Instructions

Group: VR **Task Number:** B **Task Name:** Class diagram

Required:

1. You need to draw a partial class diagram for the Book Club system, using two CASE tools in order to evaluate the learnability of the two tools. After each tool has been used, the relevant questionnaires must be completed. The tools must be used in the following order:
   - First tool to be used: Microsoft Visio
   - Second tool to be used: Rational Software Modeller

2. The class diagram which must be drawn can be seen in Figure 1 below. You may notice that there are errors in the diagram. Try to draw the diagram as shown and take note of how each respective CASE tool handles these errors.

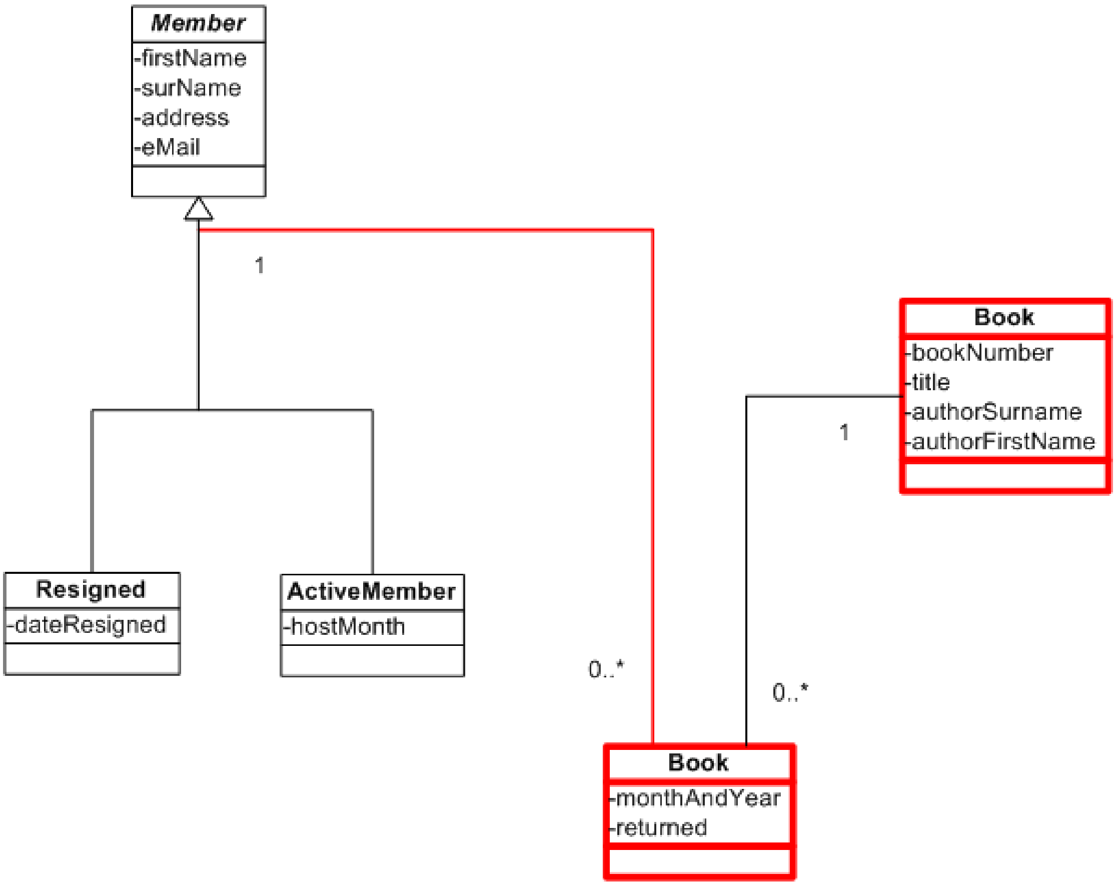

Figure 1 Partial class diagram of a Book Club system (with errors)

3. Once you have done this, correct your errors and complete the correct diagram as shown in Figure 2.

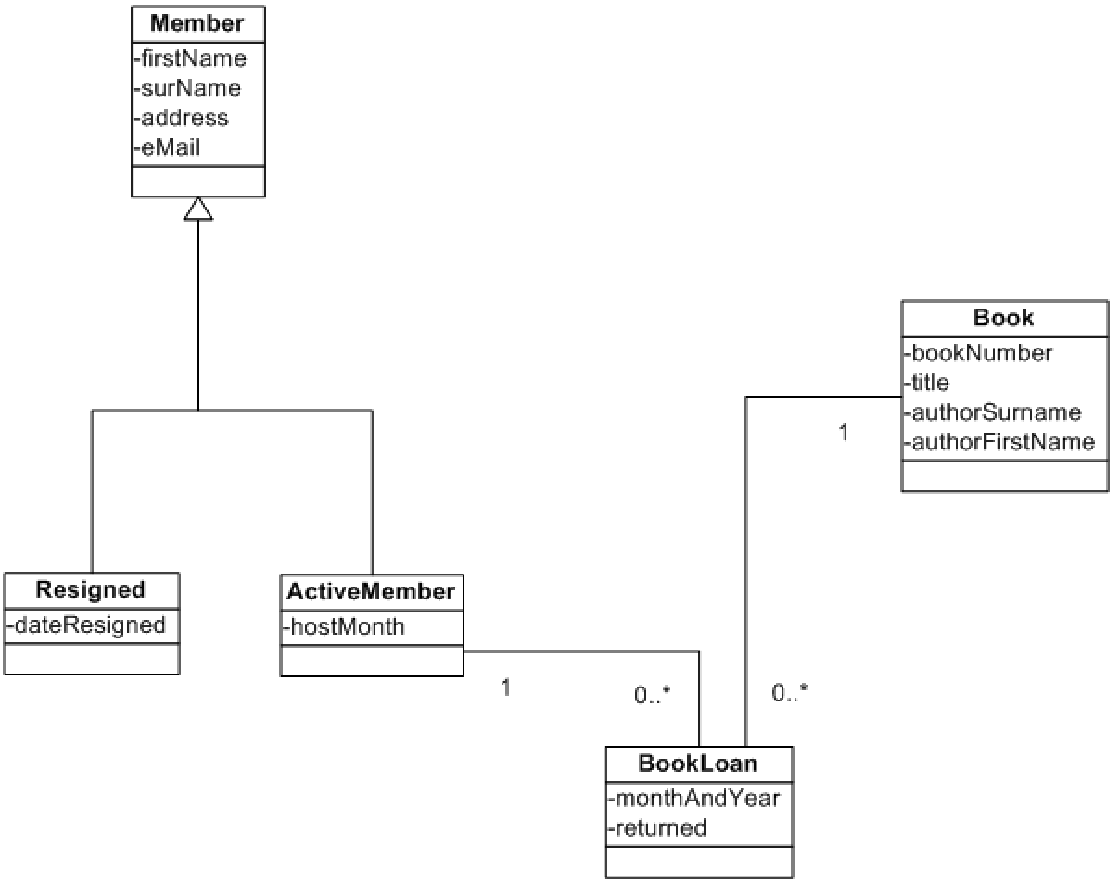

Figure 2 Partial class diagram for the Book Club system (without errors)

Appendix C – Background questionnaire

BIOGRAPHICAL INFORMATION								
Student Number:								
Name:								
Degree:								
Instructions: Mark your selection with an X in the relevant box.								
1. Gender:		Male		Female				
2. Home Language:		Afrikaans		English		Xhosa		Other
COMPUTER EXPERIENCE (PRIOR TO FEBRUARY 2006)								
3. How many years have you been using a computer for?								
	< 3 years		3- 5 years		6 – 10 years		More than 10 years	
4. What is the frequency of your computer use?								
	< 5 hrs per week		5 – 10 hrs per week		11 – 20 hrs per week		> 20 hrs per week	
5. Do you have a computer at home?		Yes		No				
6. Did you study computers at school?		Yes		No				
6.1. If yes to Q6, then please specify:		HG		SG				
7. Have you used a CASE tool before?		Yes		No				
8. If yes to Q7, then specify :								
8.1 The name/s of the CASE tool/s :								
8.2 What the CASE tool was used for		Course Work		Outside work		Both		
8.3 How many times have you used this CASE tool/s?								
		1 to 10 times		between 10 and 20 times		more than 20 times		

Appendix D – Post-test learnability questionnaire

CASE TOOL LEARNABILITY QUESTIONNAIRE B (SECOND CASE TOOL)						
CASE TOOL PRODUCT:						
BIOGRAPHICAL INFORMATION						
Group:	1					
Student Number:		Name:				
SURVEY QUESTIONS						
Instructions: Rate the following between 1 (Very poor) and 5 (Excellent). Indicate your choice with an X.						
SECTION 1: LEARNABILITY						
1	**Ease of Learning**					
		Poor				**Excellent**
	a. It was easy for me to get started and to learn how to use the tool.	1	2	3	4	5
	b. I was able to use the tool right from the beginning, without having to ask my tutors or my peers for help.	1	2	3	4	5
	c. The system encouraged me to try out new system functions by trial and error.	1	2	3	4	5
	d. It was easy for me to remember commands from one session to another.	1	2	3	4	5
	e. The explanations provided helped me to become more and more skilled at using it.	1	2	3	4	5
2	**Familiarity**					
		Slightly				**Strongly**
	a. My prior knowledge of other computer based systems was useful in the learning of the CASE tool.	1	2	3	4	5
	b. My prior knowledge of other CASE tools was useful in the learning of the CASE tool.	1	2	3	4	5
3	**Consistency**					
		Poor				**Excellent**
	a. The tool is consistently designed, thus making it easier for me to do my work.	1	2	3	4	5
	b. I find that the same function keys are used throughout the program for the same functions.	1	2	3	4	5
4	**Predictability**					
		Poor				**Excellent**
	a. The tool behaves similarly and predictably in similar situations.	1	2	3	4	5
	b. When executing functions, I get results that are predictable.	1	2	3	4	5
5	**Informative Feedback**					
		Poor				**Excellent**
	For every user action, the CASE tool responds appropriately in some way eg in web design and DHTML a button makes a clicking sound or changes colour when clicked to show the user something has happened.	1	2	3	4	5
6	**Error Messages**					
		Poor				**Excellent**
	a. If I make a mistake while performing a task, I can easily undo the last operation.	1	2	3	4	5
	b. Error messages clarify the problem.	1	2	3	4	5
	c. I perceive the error messages as helpful.	1	2	3	4	5

Please turn over …

SECTION 2: GENERAL	
7	**General**
7.1	Briefly describe what you like about the CASE tool.
7.2	Briefly discuss what you don't like about the CASE tool and any problems that you have encountered with using it.
SECTION 3: OVERALL REACTION TO THE SYSTEM	
Instruction: For each of the following, select which of the ratings 1 to 5 best represent your overall reactions to the CASE tool. Indicate your choices with an X.	
8	**Overall Reaction to the system**
8.1	terrible wonderful 1 2 3 4 5 N/A
8.2	frustrating satisfying 1 2 3 4 5 N/A
8.3	dull stimulating 1 2 3 4 5 N/A
8.4	difficult easy 1 2 3 4 5 N/A
8.5	rigid flexible 1 2 3 4 5 N/A
8.6	Inadequate functionality adequate functionality 1 2 3 4 5 N/A **Please turn over ……….**

SECTION 4 (Only complete this section, if it is the second CASE tool which you are evaluating)	
9.	**Preferences**
9.1	Which CASE tool did you prefer using?
9.2	Discuss your reasons for your answer in 9.1.
9.3	Which CASE tool was easier for you to learn to use?
9.4	Discuss your reasons for your answer in 9.3.

Appendix E – NMMU Ethics Preamble Letter

Faculty of Science
NMMU
Tel: +27 (0)41 504-2079 Fax: +27 (0)41-504-2081
-mail Faculty Chairperson: brenda.scholtz@nmmu.ac.za

Date : 25 April 2006

Ref: An Investigation into CASE Tool Learnability

Contact person: Brenda Scholtz

Dear Student

You are being asked to participate in a research study. We will provide you with the necessary information to assist you to understand the study and explain what would be expected of you (participant). These guidelines would include the risks, benefits, and your rights as a study subject. Please feel free to ask the researcher to clarify anything that is not clear to you.

To participate, it will be required of you to provide a written consent that will include your signature, date and initials to verify that you understand and agree to the conditions.

You have the right to query concerns regarding the study at any time. Immediately report any new problems during the study, to the researcher. Telephone numbers of the researcher are provided. Please feel free to call these numbers.

Furthermore, it is important that you are aware of the fact that the study has to be approved by the Research Ethics Committee (Services) of the university. The RECH consist of a group of independent experts that has the responsibility to ensure that the rights and welfare of participants, in research are protected and that studies are conducted in an ethical manner. Studies cannot be conducted without RECH's approval. Queries with regard to your rights as a research subject can be directed to the Research Ethics Committee (Services) you can call the Director: Research Management at (041) 504-4536.

If no one could assist you, you may write to: The Chairperson of the Research, Technology and Innovation Committee, PO Box 77000, Nelson Mandela Metropolitan University, Port Elizabeth, 6031.

Participation in research is completely voluntary. You are not obliged to take part in any research. If you choose not to participate in medically related research, your present and/or future medical care will not be affected in any way and you will incur no penalty and/or loss of benefits to which you may otherwise be entitled.

If you do partake, you have the right to withdraw at any given time, during the study without penalty or loss of benefits. However, if you do withdraw from the study, you should return for a final discussion or examination in order to terminate the research in an orderly manner.

If you fail to follow instructions, or if your medical condition changes in such a way that the researcher believes that it is not in your best interest to continue in this study, or for administrative reasons, your participation maybe discontinued. The study may be terminated at any time by the researcher, the sponsor or the Research Ethics Committee (Services) that initially approved the study.

Although your identity will, at all times remain confidential the results of the research study may be presented at scientific conferences or in specialist publications.

This informed consent statement has been prepared in compliance with current statutory guidelines.

Yours sincerely

Brenda Scholtz
RESEARCHER

cc: Prof Janet Wesson

Appendix F – NMMU Informed Consent Form

NELSON MANDELA METROPOLITAN UNIVERSITY

INFORMATION AND INFORMED CONSENT FORM

(Please delete any information not applicable to your project and complete/expand as deemed appropriate)

Title of the research project	An Investigation into the Usability of UML CASE tools in Computing Education
Reference number	
Principal investigator	Brenda Scholtz
Address **Postal Code**	Office 010101c Embizweni Building NMMU
Contact telephone number (private numbers not advisable)	041-5042079

A. DECLARATION BY OR ON BEHALF OF PARTICIPANT (Person legally competent to give consent on behalf of the participant)		Initial
I, the participant and the undersigned I.D. number ***OR*** I, in my capacity as of the participant I.D. number Address (of participant)	(full names)	
A.1 I HEREBY CONFIRM AS FOLLOWS:		
1. I, the participant, was invited to participate in the above-mentioned research project that is being undertaken by of the Department of in the Faculty of of the Nelson Mandela Metropolitan University.		
2. **The following aspects have been explained to me, the participant:** 2.1 **Aim:** The investigators are studying: The information will be used to/for:		
2.2 **Procedures: I understand that**		

2.3 **Risks:**	
2.4 **Possible benefits:** As a result of my participation in this study	
2.5 **Confidentiality:** My identity will not be revealed in any discussion, description or scientific publications by the investigators.	
2.6 **Access to findings:** Any new information/or benefit that develops during the course of the study will be shared as follows:	
2.7 **Voluntary participation/refusal/discontinuation:** My participation is voluntary [ ] YES [ ] NO My decision whether or not to participate will in no way affect my present or future care/employment/lifestyle [ ] TRUE [ ] FALSE	
3. The information above was explained to me/the participant by (name of relevant person) in / and I am [**Afrikaans**] [**English**] [**Xhosa**] [**Other**] in command of this language/it was satisfactorily translated to me by (name of translator) I was given the opportunity to ask questions and all these questions were answered satisfactorily.	
4. No pressure was exerted on me to consent to participation and I understand that I may withdraw at any stage without penalisation.	
5. Participation in this study will not result in any additional cost to myself.	

A.2I HEREBY VOLUNTARILY CONSENT TO PARTICIPATE IN THE ABOVE-MENTIONED PROJECT

Signed/confirmed at ______ on ______ 20

	Signature of witness
Signature or right thumb print of participant	Full name of witness

B.	STATEMENT BY OR ON BEHALF OF INVESTIGATOR(S)

I,……….…………declare that

- **I have explained the information given in this document to**

(name of patient/participant)

and/or his/her representative

(name of representative)

- **he/she was encouraged and given ample time to ask me any questions;**

- **this conversation was conducted in**

Afrikaans		**English**		**Xhosa**		**Other**	

and no translator was used / this conversation was translated into

(language)

- **I have detached Section D and handed it to the participant**

	YES		NO

Signed/confirmed at

Port Elizabeth	on	25th April	2006

Signature of interviewer	Signature of witness
	Full name of witness

C.	DECLARATION BY TRANSLATOR

I,

I.D. number

Qualifications and/or

Current employment

confirm that I

-translated the contents of this document from English into ____

(indicate the relevant language) to the participant/the participant's representative;

- also translated the questions posed by **(name)** ____ as well as the answers given by the investigator/representative; and
- conveyed a factually correct version of what was related to me.

Signed/confirmed at		on		20

I hereby declare that all information acquired by me for the purposes of this study will be kept confidential

Signature or right thumb print of translator	Signature of witness
	Full name of witness

D. IMPORTANT MESSAGE TO PATIENT/REPRESENTATIVE OF PARTICIPANT

Dear participant/representative of the participant

Thank you for your/the participant's participation in this study. Should, at any time during the study:

- **an emergency arise as a result of the research, or**
- **you require any further information with regard to the study, or**
- **the following occur**

(indicate any circumstances which should be reported to the investigator)

Kindly contact

at telephone number

(it must be a number where help will be available on a 24 hour basis, if the research project warrants it)

Appendix G – Learnability results per section

Task A – Use case diagram						
	Rational			Visio		
Question	n	Mean	SD	n	Mean	SD
1. Ease of Learning						
a. It was easy for me to get started and to learn how to use the tool.	45	2.56	1.29	46	4.02	0.88
b. I was able to use the tool right from the beginning, without having to ask my tutors or my peers for help.	45	2.33	1.38	43	3.88	1.05
c. The system encouraged me to try out new system functions by trial and error.	45	2.87	1.06	46	3.48	1.03
d. It was easy for me to remember commands from one session to another.	44	2.98	0.98	45	4.00	0.80
e. The explanations provided helped me to become more and more skilled at using it.	45	3.11	1.03	46	3.63	0.83
2. Familiarity						
a. My prior knowledge of other computer based systems was useful in the learning of the CASE tool.	45	3.18	1.03	46	3.48	1.13
b. My prior knowledge of other CASE tools was useful in the learning of the CASE tool.	46	3.41	1.02	46	3.13	1.17
3. Consistency						
a The tool is consistently designed, thus making it easier for me to do my work.	46	2.91	0.72	46	3.91	0.66
b. I find that the same function keys are used throughout the program for the same functions.	45	3.11	0.86	46	3.67	0.82
4. Predictability						
a. The tool behaves similarly and predictably in similar situations.	46	2.98	0.93	46	3.93	0.74
b. When executing functions, I get results that are predictable.	46	3.07	0.98	46	3.85	0.76
5. Informative Feedback						
For every user action, the CASE tool responds appropriately in some way.	46	2.74	1.06	46	3.52	0.84
6. Error Handling						
a. If I make a mistake while performing a task, I can easily undo the last operation.	45	3.60	1.30	46	4.30	0.79
b. Error messages clarify the problem.	45	2.73	1.30	45	3.13	1.18
c. I perceive the error messages as helpful.	45	2.78	1.17	45	3.29	1.27

Task B – Class diagram						
	Rational			**Visio**		
Question	**n**	**Mean**	**SD**	**n**	**Mean**	**SD**
1. Ease of Learning						
a. It was easy for me to get started and to learn how to use the tool.	46	2.96	1.17	46	3.87	0.83
b. I was able to use the tool right from the beginning, without having to ask my tutors or my peers for help.	46	2.67	1.14	46	3.50	1.21
c. The system encouraged me to try out new system functions by trial and error.	46	3.11	1.14	45	3.51	1.01
d. It was easy for me to remember commands from one session to another.	45	3.27	1.01	46	4.00	0.84
e. The explanations provided helped me to become more and more skilled at using it.	46	3.24	1.23	46	3.52	0.86
2. Familiarity						
a. My prior knowledge of other computer based systems was useful in the learning of the CASE tool.	45	3.22	1.13	46	3.35	0.92
b. My prior knowledge of other CASE tools was useful in the learning of the CASE tool.	45	3.22	1.04	46	3.50	0.89
3. Consistency						
a. The tool is consistently designed, thus making it easier for me to do my work.	46	3.09	0.98	46	4.00	0.73
b. I find that the same function keys are used throughout the program for the same functions.	46	3.26	0.93	46	3.89	0.85
4. Predictability						
a. The tool behaves similarly and predictably in similar situations.	46	3.26	0.95	46	4.00	0.63
b. When executing functions, I get results that are predictable.	46	3.09	1.07	46	3.87	0.81
5. Informative Feedback						
For every user action, the CASE tool responds appropriately in some way.	46	3.20	0.93	46	3.72	0.83
6. Error Handling						
a. If I make a mistake while performing a task, I can easily undo the last operation.	46	3.67	1.28	46	4.39	0.74
b. Error messages clarify the problem.	45	3.00	1.33	45	3.31	1.16
c. I perceive the error messages as helpful.	45	3.13	1.31	45	3.51	1.16

Appendix H –Learnability results per attribute

Task A – Use case diagram									
		Rank*							
		Visio	**Rational**	**n**	**Mean**	**SD**	**t-test**	**p-value**	**d**
1. Ease of Learning	Visio	2		45	3.82	0.69			
	Rational		5	45	2.77	0.90			
	Difference				1.05	1.09	6.46	.001	0.96 large
2. Familiarity	Visio	6		45	3.30	1.04			
	Rational		**1**	**45**	**3.30**	**0.94**			
	Difference				0	1.29	0.00	1.000	
3. Consistency	Visio	3		45	3.79	0.58			
	Rational		4	45	3.01	0.69			
	Difference				0.78	0.90	5.81	.001	0.87 large
4. Predictability	***Visio***	***1***		***46***	***3.89***	***0.70***			
	Rational		3	46	3.02	0.87			
	Difference				0.87	1.04	5.67	.001	0.84 large
5. Informative Feedback	Visio	5		46	3.52	0.84			
	Rational		6	46	2.74	1.06			
	Difference				0.78	1.40	3.78	.001	0.56 moderate
6. Error Handling	Visio	4		45	3.58	0.87			
	Rational		2	45	3.03	1.01			
	Difference				0.54	1.27	2.85	.007	0.43 small

Task B - Class diagram									
		Rank*							
Section		**Visio**	**Rational**	**n**	**Mean**	**SD**	**t-test**	**p-value**	**d**
1. Ease of Learning	Visio	5		46	3.69	0.70			
	Rational		5	46	3.04	0.95			
	Difference				0.65	1.12	3.94	.001	0.58 moderate
2. Familiarity	Visio	6		45	3.39	0.84			
	Rational		2	45	3.22	1.03			
	Difference				0.17	1.19	0.96	.3	
3. Consistency	***Visio***	***1***		***46***	***3.95***	***0.70***			
	Rational		4	46	3.17	0.88			
	Difference				0.78	1.06	4.99	.001	0.74 moderate
4. Predictability	Visio	2		46	3.93	0.63			
	Rational		4	46	3.17	0.94			
	Difference				0.76	0.94	5.48	.001	0.81 large
5. Informative Feedback	Visio	4		46	3.72	0.83			
	Rational		3	46	3.20	0.93			
	Difference				0.52	1.17	3.01	.004	0.44 small
6. Error Handling	Visio	3		45	3.73	0.84			
	Rational		**1**	**45**	**3.27**	**1.18**			
	Difference				0.46	1.55	1 .97	.055	

Both Tasks									
		Rank *							
		Visio	**Rational**	**n**	**Mean**	**SD**	**t-test**	**p-value**	**d**
1. Ease of Learning	Visio	3		45	3.75	0.58			
	Rational		6	45	2.90	0.76			
	Difference				0.85	0.89	6.41	.001	0.96 large
2. Familiarity	Visio	6		44	3.34	0.75			
	Rational		**1**	**44**	**3.27**	**0.84**			
	Difference				0.07		0.46	.645	
3. Consistency	Visio	2		45	3.87	0.52			
	Rational		4	45	3.09	0.62			
	Difference				0.78	0.80	6.54	.001	0.98 large
4. Predictability	***Visio***	***1***		***46***	***3.91***	***0.55***			
	Rational		3	46	3.10	0.75			
	Difference				0.81	0.76	7.23	.001	1.07 large
5. Informative Feedback	Visio	5		46	3.62	0.68			
	Rational		5	46	2.97	0.80			
	Difference				0.65	1.02	4.32	.001	0.64 moderate
6. Error Handling	Visio	4		44	3.66	0.64			
	Rational		2	44	3.14	0.96			
	Difference				0.52	1.22	2.83	.007	0.43 small

NOTE:

* In these tables, the column with the heading “Rank” indicates the rank or position of that attribute according to its mean rating. The highest attribute for Rational is highlighted in bold and in italics for Visio and will thus have a value of one in the rank column. Similarly, the attribute with the lowest mean rating will have a rank of six as there are six learnability attributes.

Appendix I – Overall learnability by rating category

Overall Learnability								
		Low		**Average**		**High**		**TOTAL (n)**
		n	**%**	**n**	**%**	**n**	**%**	
A	Rational	11	26	20	48	11	26	42
	Visio	1	2	11	26	30	71	42
B	Rational	10	23	18	41	16	36	44
	Visio	0	0	9	20	35	80	44

Appendix J – Overall reactions results

Overall Reactions Results				
Task A				
		n	**Mean**	**SD**
Wonderful	Visio	44	3.84	0.68
	Rational	44	3.05	0.99
	Difference		0.79	
Satisfying	Visio	44	3.89	0.81
	Rational	44	2.86	1.19
	Difference		1.03	
Stimulating	Visio	43	3.40	0.88
	Rational	43	3.44	1.14
	Difference		-0.04	
Easy	Visio	45	3.98	0.81
	Rational	45	3.04	0.98
	Difference		0.94	
Flexible	Visio	44	3.43	0.85
	Rational	44	3.20	0.93
	Difference		0.23	
Adequate functionality	Visio	44	4.02	0.85
	Rational	44	3.43	0.90
	Difference		0.59	
Task B				
Wonderful	Visio	45	3.87	0.76
	Rational	45	3.16	0.95
	Difference		0.71	
Satisfying	Visio	46	3.41	1.00
	Rational	46	2.83	1.00
	Difference		0.58	
Stimulating	Visio	45	3.20	1.01
	Rational	45	3.42	0.94
	Difference		-0.22	
Easy	Visio	46	3.96	0.84
	Rational	46	3.00	1.03
	Difference		0.96	
Flexible	Visio	44	3.70	0.82
	Rational	44	3.18	0.87
	Difference		0.52	
Adequate functionality	Visio	44	3.93	0.82
	Rational	44	3.61	0.87
	Difference		0.32	

Appendix K – CASE tool preferences

CASE Tool Preferences			
Ease of Use			
Task	**CASE Tool**	**n**	**%**
A	Visio	32	78
	Rational	9	22
	TOTAL	**41**	**100**
	Significance	(Chi²(1)=12.90, p<.0005; V=0.56 Large)	
B	Visio	31	70
	Rational	13	30
	TOTAL	**46**	**100**
	Significance	(Chi²(1)=7.36, p<.007; V=0.41 Medium)	
Ease of Learning			
Task	**CASE Tool**	**n**	**%**
A	Visio	32	78
	Rational	9	22
	TOTAL	**41**	**100**
	Significance	(Chi²(1)=12.90, p<.0005; V=0.56 Large)	
B	Visio	29	72
	Rational	11	28
	TOTAL	**40**	**100**
	Significance	(Chi²(1)=8.10, p<.004; V=0.45 Medium)	

Appendix L –Positive qualitative results

Positive Qualitative Results by Theme				
Task A - Use case diagram				
	Visio		**Rational**	
	n	**%**	**n**	**%**
LEARNABILITY				
Familiarity	8	14		
Consistency			1	3
Predictability				
Ease of Learning	1	2	2	
Simplicity	7	12	3	8
Feedback				
Observability	9	16	6	**15**
Quality of presentation	5	9	14	**36**
Quality of content				
Robustness				
Error Prevention			4	10
Recoverability	6	11		
Provision of help			1	3
USABILITY				
Flexibility	1	2	1	3
Efficiency				
GENERAL				
Usability - unspecific	12	21	4	10
Learnability - unspecific	7	12	1	3
Understanding of UML	1	2		
Not specified			2	5
TOTALS	**57**	**100**	**39**	**100**

Positive Qualitative Results by Theme				
Task B - Class diagram				
	Visio		**Rational**	
	n	**%**	**n**	**%**
LEARNABILITY				
Familiarity	3	**6**		
Consistency	2	4	1	3
Predictability	1	2		
Ease of Learning	2		2	6
Simplicity	2	4	1	3
Feedback				
Observability	10	**19**	2	6
Quality of presentation	1	2	12	**35**
Quality of content				
Robustness				
Error Prevention			2	6
Recoverability	15	**28**	6	18
Provision of help				
USABILITY				
Flexibility				
Efficiency	1	2		
GENERAL				
Usability - unspecific	8	15	4	12
Learnability - unspecific	5	9	4	12
Understanding of UML				
Not specified	3	6		
TOTALS	**53**	**100**	**34**	**100**

Appendix M –Negative qualitative results

Negative qualitative results by theme				
Task A - Use case diagram				
	Visio		**Rational**	
	n	**%**	**n**	**%**
LEARNABILITY				
Familiarity	1	3		
Consistency				
Predictability	2	7	1	3
Ease of Learning	2	7	2	
Simplicity	3	10	9	**24**
Feedback				
Observability	9	**30**	4	11
Quality of presentation	3	10	7	18
Quality of content				
Robustness				
Error Prevention				
Recoverability	8	**27**	8	**21**
Provision of help	1	3	1	3
USABILITY				
Flexibility				
Efficiency	1	3	1	3
GENERAL				
Usability - unspecific			2	5
Learnability - unspecific			2	5
Understanding of UML				
Not specified			1	3
TOTALS	**30**	**100**	**38**	**100**

Negative qualitative results by theme				
Task B - Class diagram				
	Visio		**Rational**	
	n	**%**	**n**	**%**
LEARNABILITY				
Familiarity	1	4		
Consistency			1	3
Predictability	1	4		
Ease of Learning	2	7	2	6
Simplicity			5	14
Feedback				
Observability	10	36	13	37
Quality of presentation	2	7	4	11
Quality of content				
Robustness				
Error Prevention				
Recoverability	7	25	7	20
Provision of help	5	18	1	3
USABILITY				
Flexibility				
Efficiency				
GENERAL				
Usability - unspecific			1	3
Learnability - unspecific				
Understanding of UML				
Not specified			1	3
TOTALS	**28**	**100**	**35**	**100**

Appendix N –ANOVA results

ANOVA results					
Task A – Use case diagram					
	Sum of Squares	**df**	**Mean Square**	**F**	**p**
1. Ease of Learning					
Intercept	*898.68*	*1*	*898.68*	*1336.13*	*.000*
Tool	0.00	1	0.00	0.00	.956
Lang	0.07	2	0.03	0.05	.952
Freq	1.71	2	0.86	1.27	.292
Error	25.56	38	0.67		
R1	1.33	1	1.33	2.49	.123
R1*Tool	*22.44*	*1*	*22.44*	*41.86*	*.000*
R1*Lang	0.36	2	0.18	0.34	.717
R1*Freq	3.13	2	1.56	2.92	.066
Error	20.38	38	0.54		
2. Familiarity					
Intercept	*910.11*	*1*	*910.113*	*967.87*	*.000*
Tool	*8.56*	*1*	*8.560*	*9.10*	*.005*
Lang	*6.30*	*2*	*3.150*	*3.35*	*.046*
Freq	0.63	2	0.315	0.33	.717
Error	35.73	38	0.940		
R1	0.29	1	0.294	0.33	.568
R1*Tool	0.01	1	0.005	0.01	.939
R1*Lang	0.30	2	0.151	0.17	.844
R1*Freq	1.30	2	0.652	0.74	.486
Error	33.65	38	0.886		
3. Consistency					
Intercept	*925.76*	*1*	*925.76*	*2164.78*	*.000*
Tool	0.05	1	0.05	0.11	.739
Lang	0.60	2	0.30	0.70	.505
Freq	0.17	2	0.09	0.20	.818
Error	16.25	38	0.43		
R1	*1.61*	*1*	*1.61*	*4.35*	*.044*
R1*Tool	*13.69*	*1*	*13.70*	*37.06*	*.000*
R1*Lang	1.10	2	0.55	1.49	.238
R1*Freq	1.00	2	0.50	1.37	.269
Error	14.04	38	0.37		
4. Predictability					
Intercept	*977.00*	*1*	*977.00*	*1429.61*	*.000*
Tool	1.81	1	1.81	2.65	.112
Lang	2.04	2	1.02	1.50	.237
Freq	0.01	2	0.00	0.00	.996
Error	26.66	39	0.68		
R1	0.38	1	0.38	0.85	.364
R1*Tool	*13.49*	*1*	*13.49*	*30.05*	*.000*
R1*Lang	*2.93*	*2*	*1.47*	*3.27*	*.049*
R1*Freq	2.14	2	1.07	2.39	.105
Error	17.50	39	0.45		

5. Informative Feedback					
Intercept	*799.87*	*1*	*799.87*	*954.24*	*.000*
Tool	3.12	1	3.12	3.72	.061
Lang	0.67	2	0.34	0.40	.672
Freq	0.64	2	0.32	0.38	.687
Error	32.69	39	0.84		
R1	0.05	1	0.05	0.05	.834
R1*Tool	*13.01*	*1*	*13.01*	*12.06*	*.001*
R1*Lang	0.40	2	0.20	0.19	.832
R1*Freq	1.30	2	0.65	0.60	.552
Error	42.09	39	1.08		
6. Error Handling					
Intercept	*897.33*	*1*	*897.33*	*960.19*	*.000*
Tool	0.51	1	0.51	0.54	.465
Lang	0.12	2	0.06	0.06	.938
Freq	3.38	2	1.70	1.81	.178
Error	35.51	38	0.94		
R1	0.52	1	0.52	0.75	.394
R1*Tool	*6.41*	*1*	*6.41*	*9.19*	*.004*
R1*Lang	0.49	2	0.25	0.35	.705
R1*Freq	*6.56*	*2*	*3.28*	*4.70*	*.015*
Error	26.53	38	0.70		
8. Overall reactions					
Intercept	*948.61*	*1*	*948.61*	*2320.76*	*.000*
Tool	0.13	1	0.13	0.32	.573
Lang	0.54	2	0.27	0.65	.526
Freq	0.20	2	0.10	0.24	.787
Error	15.12	37	0.41		
R1	1.89	1	1.89	3.99	.053
R1*Tool	*7.81*	*1*	*7.81*	*16.47*	*.000*
R1*Lang	1.05	2	0.53	1.11	.341
R1*Freq	2.45	2	1.23	2.58	.089
Error	17.55	37	0.47		
Task B – Class diagram					
	Sum of Squares	**Df**	**Mean Square**	**F**	**p**
1. Ease of Learning					
Intercept	*941.63*	*1*	*941.63*	*1152.80*	*.000*
Tool	0.22	1	0.22	0.27	.604
Lang	1.08	2	0.54	0.66	.522
Freq	0.04	2	0.02	0.03	.975
Error	31.86	39	0.82		
R1	1.91	1	1.91	3.00	.091
R1*Tool	*10.62*	*1*	*10.62*	*16.68*	*.000*
R1*Lang	1.08	2	0.54	0.85	.435
R1*Freq	0.41	2	0.20	0.32	.728
Error	24.84	39	0.64		

2. Familiarity					
Intercept	*893.63*	*1*	*893.63*	*850.75*	*.000*
Tool	0.01	1	0.01	0.01	.930
Lang	0.26	2	0.13	0.12	.885
Freq	1.57	2	0.77	0.75	.480
Error	39.92	38	1.05		
R1	0.30	1	0.30	0.40	.529
R1*Tool	0.30	1	0.30	0.40	.533
R1*Lang	0.04	2	0.02	0.03	.971
R1*Freq	1.92	2	0.96	1.29	.288
Error	28.30	38	0.75		
3. Consistency					
Intercept	*1043.00*	*1*	*1043.00*	*1343.71*	*.000*
Tool	0.01	1	0.01	0.02	.894
Lang	0.15	2	0.08	0.10	.908
Freq	0.51	2	0.26	0.33	.722
Error	30.27	39	0.78		
R1	1.03	1	1.03	1.74	.195
R1*Tool	*14.78*	*1*	*14.80*	*24.91*	*.000*
R1*Lang	1.25	2	0.63	1.05	.358
R1*Freq	0.21	2	0.11	0.18	.836
Error	23.17	39	0.60		
4. Predictability					
Intercept	*1035.87*	*1*	*1035.86*	*1097.89*	*.000*
Tool	0.06	1	0.06	0.07	.797
Lang	0.42	2	0.21	0.22	.802
Freq	0.48	2	0.24	0.25	.778
Error	36.80	39	0.95		
R1	0.03	1	0.03	0.07	.789
R1*Tool	*13.10*	*1*	*13.10*	*28.31*	*.000*
R1*Lang	1.23	2	0.61	1.33	.277
R1*Freq	0.75	2	0.37	0.81	.454
Error	18.05	39	0.46		
5. Informative Feedback					
Intercept	*986.71*	*1*	*986.71*	*1105.38*	*.000*
Tool	0.52	1	0.52	0.59	.448
Lang	1.73	2	0.87	0.97	.388
Freq	0.89	2	0.44	0.50	.612
Error	34.81	39	0.89		
R1	0.00	1	0.00	0.00	.953
R1*Tool	*3.62*	*1*	*3.62*	*5.50*	*.024*
R1*Lang	0.16	2	0.08	0.12	.887
R1*Freq	1.45	2	0.72	1.10	.343
Error	25.67	39	0.66		
6. Error Handling					
Intercept	*985.32*	*1*	*985.32*	*1116.70*	*.000*

Tool	0.80	1	0.80	0.90	.348
Lang	0.12	2	0.06	0.07	.933
Freq	0.99	2	0.49	0.56	.576
Error	33.53	38	0.88		
R1	0.04	1	0.04	0.04	.842
*R1*Tool*	*6.88*	*1*	*6.88*	*6.75*	*.013*
R1*Lang	4.50	2	2.25	2.21	.124
R1*Freq	*8.06*	*2*	*4.03*	*3.96*	*.027*
Error	38.70	38	1.02		
8. Overall reactions					
Intercept	*955.90*	*1*	*955.90*	*3091.80*	*.000*
Tool	0.35	1	0.35	1.14	.292
Lang	0.10	2	0.05	0.16	.854
Freq	0.17	2	0.09	0.28	.758
Error	12.06	39	0.31		
R1	1.51	1	1.51	2.45	.125
R1*Tool	*6.45*	*1*	*6.45*	*10.45*	*.002*
R1*Lang	0.85	2	0.43	0.69	.507
R1*Freq	0.14	2	0.07	0.11	.895
Error	24.06	39	0.62		

Appendix O – Post-hoc Scheffe test results

Scheffe results for Home language and Familiarity			
Task A – Use case diagram			
Familiarity			
Scheffe test; variable DV_1 (CASE Data1c 070604.sta) Probabilities for Post Hoc Tests Error: Between MS = .94032, df = 38.000			
Language	**1**	**2**	**3**
1		0.997	0.267
2	0.997		0.321
3	0.267	0.321	

Scheffe results for Home language and Predictability						
Task A – Use case diagram						
Predictability						
Language	**1**	**1**	**2**	**2**	**3**	**3**
1		0.83	0.93	0.97	0.32	1.00
1	0.83		0.33	1.00	0.96	0.93
2	0.93	0.33		0.41	***0.03***	0.79
2	0.97	1.00	0.41		0.87	0.99
3	0.32	0.96	***0.03***	0.87		0.22
3	1.00	0.93	0.79	0.99	0.22	

Scheffe results for Frequency of computer use and Error handling						
Task A – Use case diagram						
Scheffe test; variable DV_1 (CASE Data1c 070604.sta) Probabilities for Post Hoc Tests Error: Between; Within; Pooled MS = .95033, df = 75.613						
Frequency	**1**	**1**	**2**	**2**	**3**	**4**
1		0.163	0.276	0.997	0.548	0.134
1	0.163		1.000	0.717	1.000	0.990
2	0.276	1.000		0.624	0.999	0.999
2	0.997	0.717	0.624		0.893	0.481
3	0.548	1.000	0.999	0.893		0.975
3	0.134	0.990	0.999	0.481	0.975	
Task B – Class diagram						
Scheffe test; variable DV_1 (CASE Data1c 070604.sta) Probabilities for Post Hoc Tests Error: Between; Within; Pooled MS = .95033, df = 75.613						
Frequency	**1**	**1**	**2**	**2**	**3**	**4**
1		0.96	0.47	1.00	1.00	1.00
1	0.96		0.90	0.87	0.93	1.00
2	0.47	0.90		0.44	0.48	0.86
2	1.00	0.87	0.44		1.00	0.98
3	1.00	0.93	0.48	1.00		0.99
3	1.00	1.00	0.86	0.98	0.99	

Appendix P –ANOVA graphs

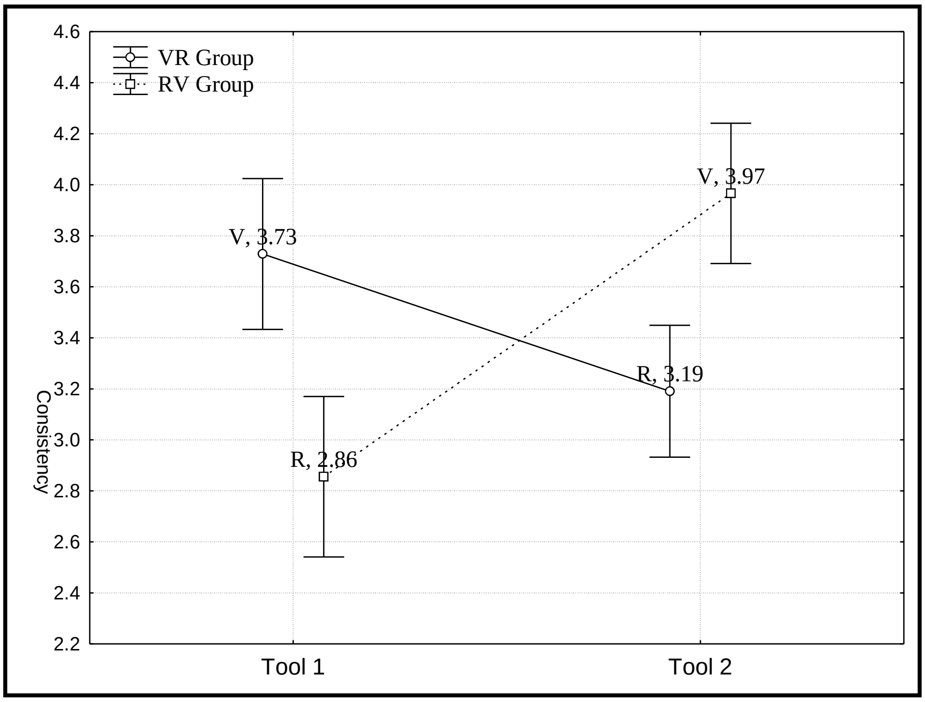

Figure P-1 Relationship between CASE tool and Consistency for Task A

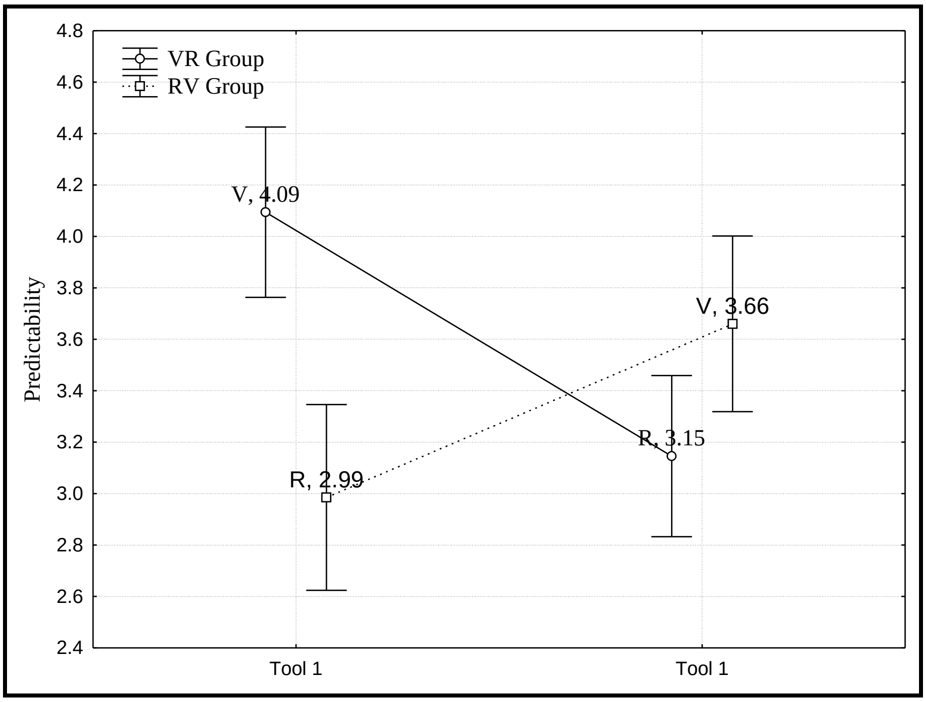

Figure P-2 Relationship between CASE tool and Predictability for Task A

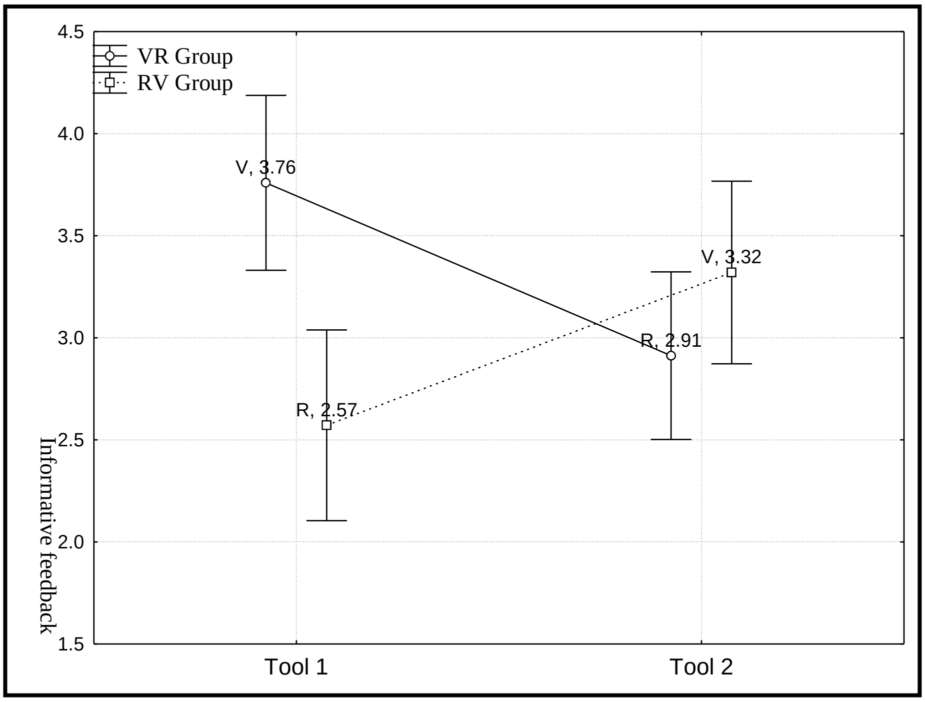

Figure P-3 Relationship between CASE tool and Informative Feedback for Task A

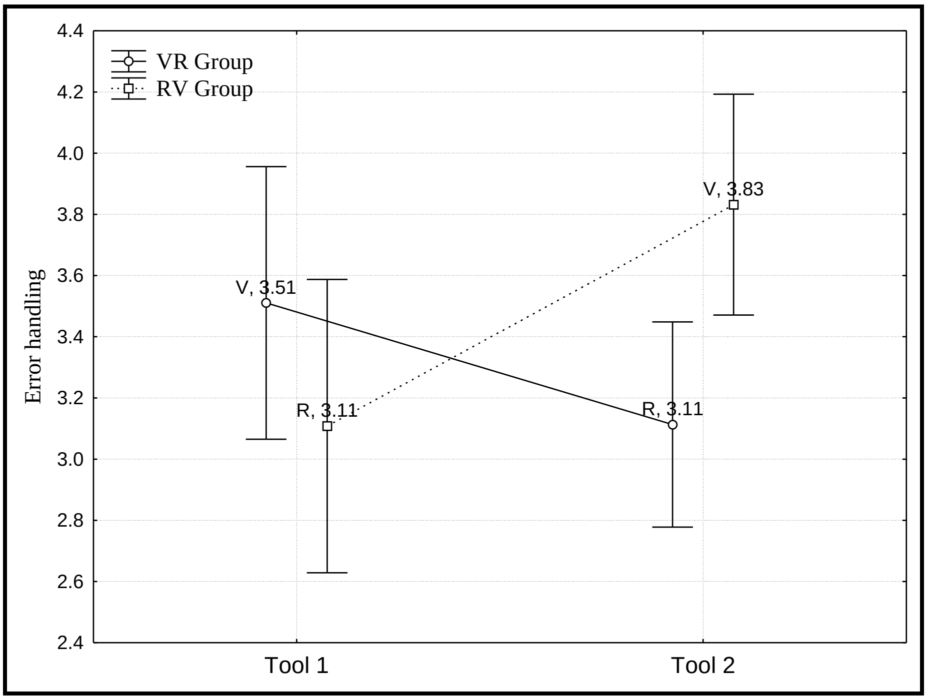

Figure P-4 Relationship between CASE tool and Error handling for Task A

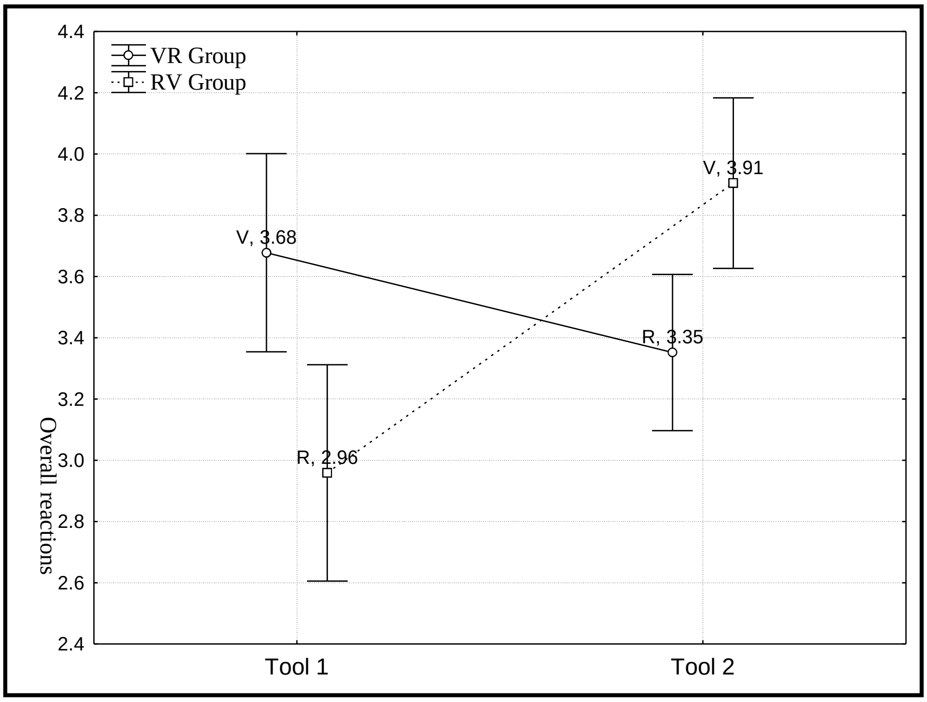

Figure P-5 Relationship between CASE tool and Overall reactions for Task A

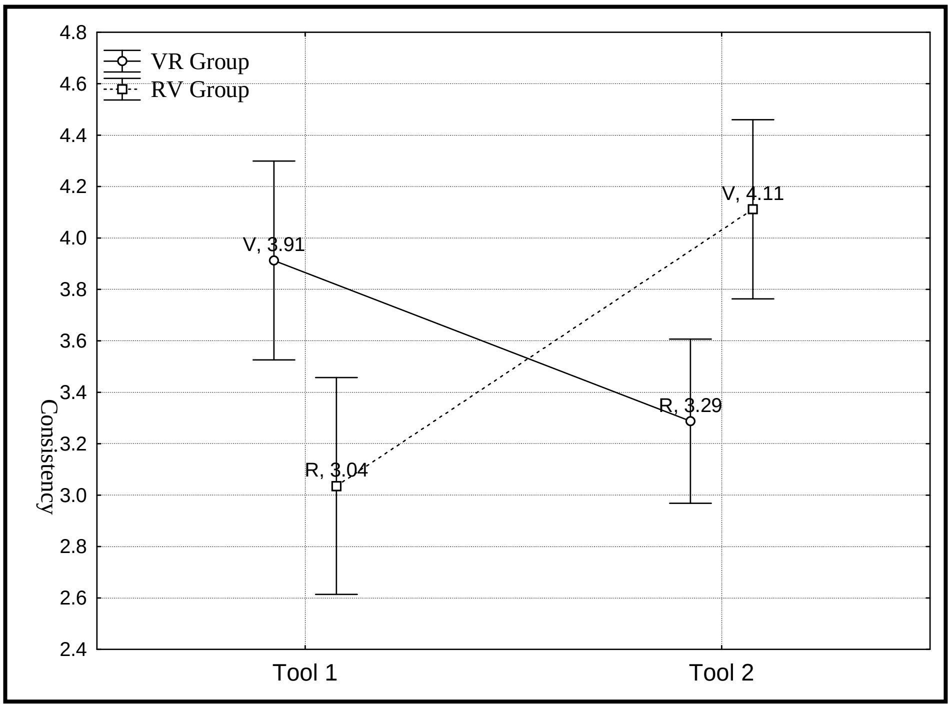

Figure P-6 Relationship between CASE tool and Consistency for Task B

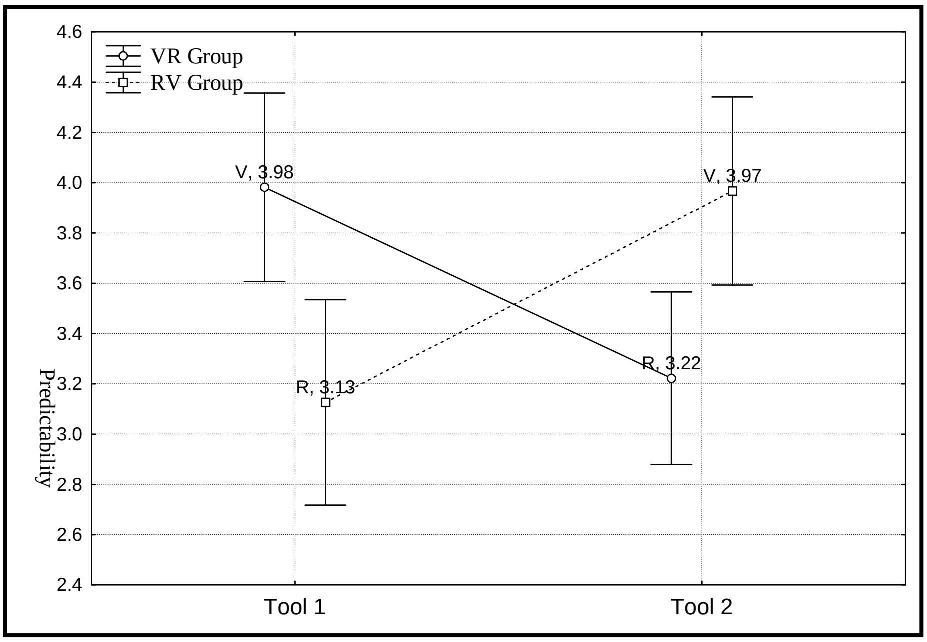

Figure P-7 Relationship between CASE tool and Predictability for Task B

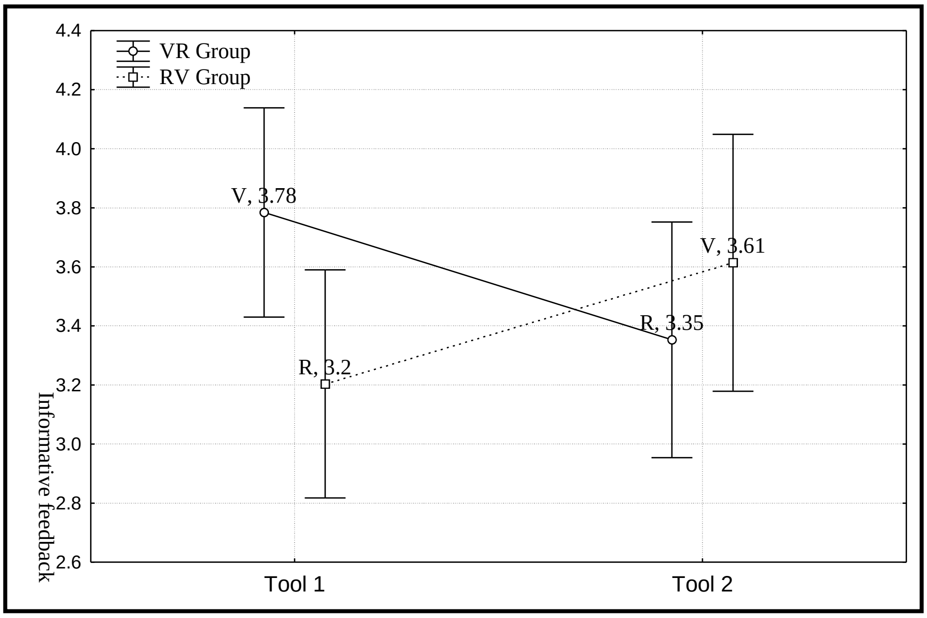

Figure P-8 Relationship between CASE tool and Informative feedback for Task B

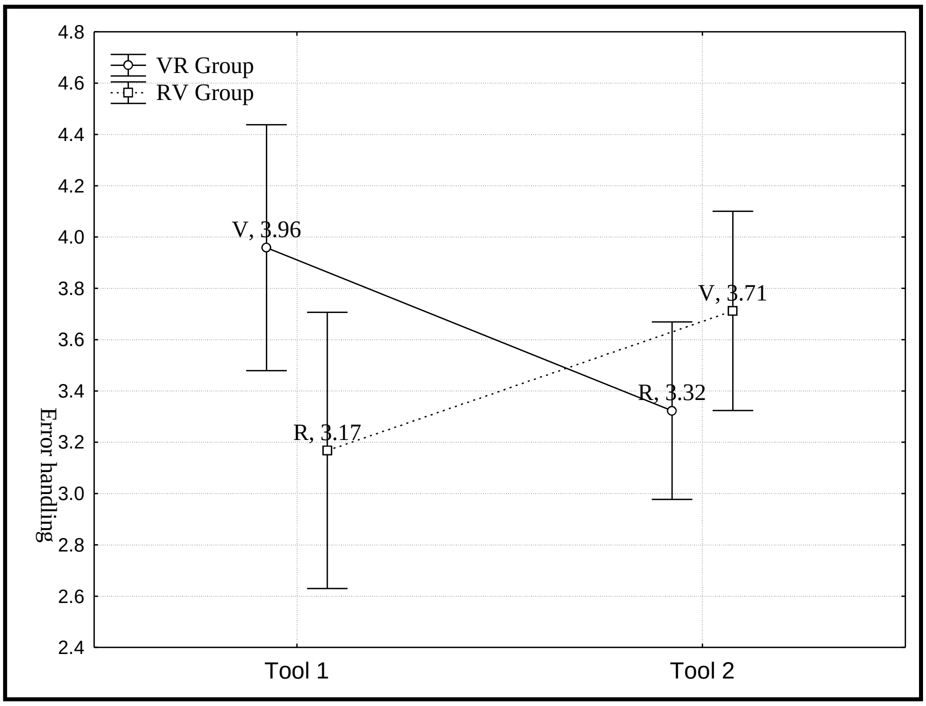

Figure P-9 Relationship between CASE tool and Error handling for Task B

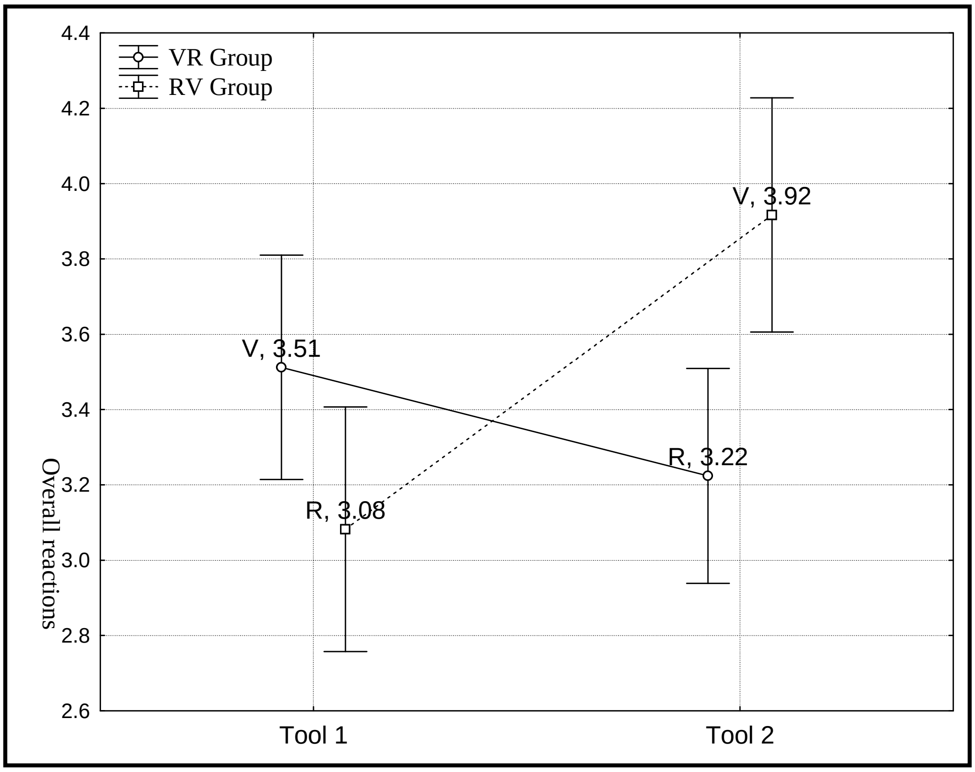

Figure P-10 Relationship between CASE tool and Overall reactions for Task B

Appendix Q –Revised post-test learnability questionnaire

CASE TOOL LEARNABILITY QUESTIONNAIRE A (FIRST CASE TOOL)						
CASE TOOL PRODUCT: MICROSOFT VISIO ENTERPRISE						
BIOGRAPHICAL INFORMATION						
Group:	1					
Student Number:		Name:				
SURVEY QUESTIONS						
Instructions: Rate the following between 1 (Very poor) and 5 (Excellent). Indicate your choice with an X.						
SECTION 1: LEARNABILITY						
1	**Familiarity**					
		Slightly				**Strongly**
	a. This system felt familiar due to my prior knowledge of other computer based systems.	1	2	3	4	5
	b. This system felt familiar due to my prior knowledge of other similar tools.	1	2	3	4	5
	c. This system felt familiar due to my prior knowledge of other CASE tools. (Omit if you have not worked on other CASE tools)	1	2	3	4	5
2	**Consistency**					
		Very Poor				**Excellent**
	a. The tool is consistently designed, thus making it easier for me to do my work.	1	2	3	4	5
	b. I find that the same function keys are used throughout the program for the same functions.	1	2	3	4	5
3	**Predictability**					
		Very Poor				**Excellent**
	a. The tool behaves similarly and predictably in similar situations.	1	2	3	4	5
	b. When executing functions, I get results that are predictable.	1	2	3	4	5
4.	**Ease of Learning**					
		Very Poor				**Excellent**
	a. It was easy for me to get started and to learn how to use the tool.	1	2	3	4	5
	b. I was able to use the tool right from the beginning, without having to ask my tutors or my peers for help.	1	2	3	4	5
	c. The system encouraged me to try out new system functions by trial and error.	1	2	3	4	5
	d. It was easy for me to remember commands from one session to another.	1	2	3	4	5
	e. The explanations provided helped me to become more and more skilled at using it.	1	2	3	4	5
5.	**Simplicity**					
		Very Poor				**Excellent**
	a. The CASE tool has an uncluttered user interface which is simple to use.					
	b. The CASE tools lets me draw a UML diagram in as few steps as possible.					
	Please turn over …					

6. Informative Feedback						
		Poor				**Excellent**
6.1	**Observability**					
	a. For every user action, the CASE tool responds appropriately in some way.	1	2	3	4	5
	b. The CASE tool provides sufficient feedback about intermediate and final results while drawing a diagram.	1	2	3	4	5
	c. The feedback provided is clear and easy to understand.	1	2	3	4	5
		Poor				**Excellent**
6.2	**Quality of Presentation**					
	a. The quality of presentation is aesthetically pleasing.	1	2	3	4	5
	b. The information is presented in easily accessible places.	1	2	3	4	5
6.3	**Quality of Content**					
	a. The content presented on the screen by the CASE tool is relevant.	1	2	3	4	5
	b. The content presented on the screen by the CASE tool is sufficient.	1	2	3	4	5
7. Robustness						
		Poor				**Excellent**
7.1	**Error Prevention**					
	a. The CASE tool prevented me from making errors when appropriate.	1	2	3	4	5
	b. The CASE tool gave clear and understandable warnings of potential errors.	1	2	3	4	5
7.2	**Recoverability (Error Handling)**					
	a. If I make a mistake while performing a task, I can easily undo the last operation.	1	2	3	4	5
	b. Error messages are expressed in an easy to understand language.	1	2	3	4	5
	c. I perceive the error messages as helpful.	1	2	3	4	5
	Provision of Help					
7.3	a. The tutorials provided helped me to understand the CASE tool better.	1	2	3	4	5
	b. The help provided assisted me with understanding the CASE tool better.	1	2	3	4	5

SECTION 2: GENERAL	
8	**General**
8.1	Briefly describe what you like about the CASE tool.
8.2	Briefly discuss what you don't like about the CASE tool and any problems that you have encountered with using it.