

**Adaptive Learning and Metacognitive Regulation Support for Ill-structured
Problem Solving Processes in Web-Based Learning**

Robert Obwocha Oboko

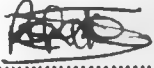
**A thesis submitted in fulfillment of the requirements for the award of the
degree of Doctor of Philosophy in Computer Science**

School of Computing and Informatics, University of Nairobi

26 April 2012

Declaration

This thesis is my original work and has not been presented for a degree in any other university


.....

Date: 26/4/2012
.....

Robert Obwocha Oboko

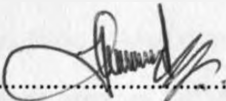
This thesis has been submitted for examination with our approval as supervisors

Signed 
.....

Date: 26th April 2012
.....

Dr. Peter W. Wagacha


School of Computing and Informatics, University of Nairobi

Signed 
.....

Date: 26th April 2012
.....

Dr. Elijah I. Omwenga

School of Computing and Informatics, University of Nairobi

Signed 
.....

Date: 26/4/2012
.....

Prof Arno Libotton,

Faculty of Psychology and Educational Sciences, Free University of Brussels

Abstract

This research was set up to investigate the effect of adaptive user interface and self-regulation learning supports, in a web-based non-collaborative learning system, on the learners' cognition, metacognitive awareness and application of problem solving processes. The research was designed as a mixed study with two quantitative experimental studies and a qualitative cross-case comparative study. Study 1 investigated the effect of adaptive user interface supports on the learner's cognitive performance, measured as test scores, and perceived value of the adaptive features of the learning environment. Study 2 investigated the effect of self-regulation scaffolds on the learner's level of metacognitive awareness and level of application of problem solving processes. The qualitative study further investigated the effects of self-regulation scaffolds by using the think aloud method.

Results from Study 1 indicated no significant difference in mean test scores between learners using and those not using adaptive features. Regarding perceived value of the adaptive features, both groups "starting with" and "starting without" adaptive features perceived the features as useful. The results show that the order of sequencing course content as well as the strategy of presenting adaptive user interface supports has an impact on learner performance and experience. From Study 2, learners who used the scaffolds had a significantly higher metacognitive awareness and application of problem solving processes, compared to the control group. However, whereas the scaffolds were effective for knowledge of regulation of cognition, they had minimal effect on knowledge of cognition. Hence, there is need for a different set of scaffolds for knowledge of cognition. Study 2 also showed that the effect of scaffolds on the learners' cognitive processes is better measured using the self-report method followed by solving of a task.

Keywords: Scaffolding, cognitive load, metacognition, cognition, ill-structured problem solving, adaptive web-based learning, Machine Learning algorithms, instructional design

Acknowledgements

First and foremost, I wish to thank God for His sufficient grace during the journey that the research was. For everything, I give thanks.

I wish to acknowledge the VLIR-IUC-UON programme for funding the research right from its inception. I wish to acknowledge all the staff involved in the programme for giving me logistical support both when in Brussels and in Nairobi. I also give thanks to IDRC for funding the latter stages of my research project.

To the University of Nairobi administration, I appreciate the support I received in the form of resources for my research and leave days granted whenever I had to travel in connection with the research.

I am also deeply indebted to my three promoters: Prof. Arno Libotton of Free University of Brussels, Dr. Elijah I. Omwenga and Dr. Peter W. Wagacha both of University of Nairobi.

Prof. Libotton: I appreciate your support throughout my research project. In 2007, during my study visit to the Free University of Brussels, you gave me the most critical support, through reading my work, listening and giving feedback, and also by being inspirational with your words. Your support continued to the end and I am grateful.

Dr. Omwenga: Thank you for your support through incisive and thought provoking comments as well as giving attention to detail. Your contribution was critical in steering this work forward.

Dr. Wagacha: Thank you for always being available, for listening and prompting me to find out details about my progress and challenges faced. You uplifted me many times with words of encouragement. Your comments on all the chapters were invaluable.

Special thanks to Prof. Timothy Waema for supporting me to get funding from IDRC for my research and for inquiring about my progress regularly.

Special thanks also to Prof. Joseph Kizza for opening my eyes further on the theory of my research.

I wish to acknowledge Prof. Bernard Manderick, for maintaining interest in my progress and even getting time to give feedback on this work.

To the staff of Prof. Libotton's research laboratory at the Free University of Brussels for always being kind, understanding and accommodative, and for offering me support beyond obligation.

I would like to appreciate my colleagues Muthoni Masinde, Christopher Chepken, Evans Miriti and Christine Ronge for all their support at different stages of the research.

I would also like to appreciate the staff of ICT centre, University of Nairobi (Veronica Mutua, Caroline Chebet, Benjamin Kimele, Jerusha Kimani, Jared Onyari, among others), for participating in the initial testing of the course used in the research. Special thanks to Michael Mukiri for giving me technical support during the installation and testing of the system. I am indebted to the Director, ICT Centre, University of Nairobi, Dr. Omwenga, and Peter Mate, Manager of ICT support services, for providing me with access to a computer laboratory when I needed one for one of my cohorts.

To all the students who participated in the research project, from the School of Computing and Informatics (SCI) of University of Nairobi, Inoorero University and Jomo Kenyatta University of Agriculture and Technology (JKUAT), thank you.

To my wife, Lucille, thank you for being understanding and for reminding me constantly of the need to put in extra hours so as to complete the research in good time. Thanks too to my two mothers and children Stacy and Imani for bearing with me, and giving me unconditional love and support.

Table of Contents

Declaration	i
Abstract	ii
Acknowledgements	iii
Table of Contents	v
List of Tables	xii
List of Figures.....	xv
Abbreviations	xvii
Glossary	xviii
CHAPTER 1 : GENERAL INTRODUCTION	1
1.1 Introduction.....	1
1.1.1 Ill-structured Problems.....	1
1.1.2 Need for Scaffolding during Problem Solving.....	2
1.1.3 Need for Scaffolding in Online Learning with Ill-structured Problem Solving.....	4
1.1.4 Adaptive Scaffolds	4
1.1.5 Metacognitive Regulation Scaffolds.....	6
1.1.6 Learning and Information System Theories for Web-based Learning Design.....	6
1.2 Problem Statement	8
1.3 Aim of the Study	8
1.4 Research Objectives	9
1.5 Research Questions	9
1.6 Structure of the Thesis	11
1.7 Language Conventions	12
CHAPTER 2 : LITERATURE REVIEW.....	14

2.1 Introduction.....	14
2.2 Problem Solving.....	16
2.2.1 Introduction to Problem Solving	16
2.2.2 Well-structured Problems	18
2.2.3 Ill-structured Problems.....	19
2.2.4 Problem Solving Processes.....	21
2.3 Problem Solving in Relation to Cognition and Metacognition	23
2.3.1 Components of Ill-structured Problem Solving	23
2.3.2 Cognition	23
2.3.3 Metacogniton	25
2.4 Scaffolding.....	29
2.4.1 Introduction.....	29
2.4.2 Fading	30
2.4.3 Scaffolding in Online Learning Environments	30
2.4.4 Scaffolding Cognition.....	31
2.4.5 Scaffolding Metacognition	31
2.5 Cognitive Load Theory.....	32
2.5.1 Introduction.....	32
2.5.2 The Human Cognitive Architecture and Cognitive Load	32
2.5.3 Types of Cognitive Load.....	34
2.5.4 Expertise Reversal Effect and Adaptive Learning.....	37
2.5.5 Influence of Cognitive Load Theory on Instructional Design.....	38
2.6 Tools for Supporting Metacognition During Ill-Structured Problem Solving in Online Learning.....	41
2.6.1 Introduction.....	41

2.6.2 Use of Task Schedule	41
2.6.3 Question Prompts.....	45
2.7 Machine Learning and Adaptive Cognitive Learning Support	48
2.7.1 Introduction.....	48
2.7.2 Suitability of Adaptive User Interfaces for Web-based Learning	49
2.7.3 Techniques for Adaptive Learning Support for Hypermedia.....	49
2.7.4 Machine Learning Techniques.....	51
2.8 Theories of Learning and Instructional Design.....	56
2.8.1 Introduction.....	56
2.8.2 Learning Theories	56
2.8.3 Instructional Design Models.....	57
2.8.4 Instructional Design Theories.....	59
2.9 Summary.....	61
2.9.1 Summary of Relevant Concepts and Their Relationships.....	61
2.9.2 Conceptual Framework	63
CHAPTER 3 : METHODOLOGY – DESIGN OF LEARNING SUPPORT SYSTEM.....	68
3.0 Introduction to Integrated Design Process	68
3.1 Phase 1: Needs Analysis	72
3.1.1 Process 1: Requirements Specification	73
3.1.2 Process 2: Component and Feature Specification.....	85
3.1.3 Process 3: Goal Design.....	89
3.2 Phase 2: Conceptual Design	92
3.2.1 Process 1: Development of Design Scenarios and Use Cases	93
3.2.2 Process 2: Information Design.....	97
3.2.3 Process 3: Navigation Design.....	103

3.2.4 Process 4: Page Design	116
3.2.5 Process 5: Structure Design.....	118
3.2.6 Process 6: Presentation Design	121
3.2.7 Process 7: Data Design	124
3.3 Phase 3: Development.....	124
3.3.1 Process 1: Development of the Course and Related Tools and Items	124
3.3.2 Process 2: Development of the Low-fidelity Prototype	130
3.3.3 Process 3: Design Walkthrough.....	131
3.3.4 Process 4: Towards High-fidelity Prototype	132
3.4 Phase 4: Formative Evaluation	132
3.4.0 Introduction.....	132
3.4.1 Formative Evaluation of the Sub-systems	132
3.4.2 Those who Participated in the Formative Evaluation of the System	133
3.4.3 Process 1: Evaluation of the Instructional System	133
3.4.4 Process 2: Evaluation of the Adaptive Learning Support System.....	141
3.4.5 Process 3: Evaluation of the Problem Solving Support System.....	150
3.5 Summary.....	152
CHAPTER 4 : METHODOLOGY - MACHINE LEARNING AND ADAPTIVE LEARNING SUPPORT DESIGN	155
4.0 Introduction.....	155
4.1 Specification of the Adaptive User Interface Components.....	155
4.1.1 The Learner Model Components.....	155
4.1.2 Data Collection Techniques	156
4.1.3 The Domain Model Components	157
4.1.4 Multiple Representations:.....	157

4.1.5 Inference.....	158
4.1.6 Adaptation of the User Interface	158
4.2 Application of Machine Learning Algorithms	161
4.2.0 Introduction.....	161
4.2.1 Initialization of Learner Model Concept Scores Using HVDM	161
4.2.2 Adaptive Presentation of Additional Learning Materials Using NBC	169
4.3 Summary.....	172
CHAPTER 5 : METHODOLOGY – EXPERIMENTAL DESIGN	175
5.0 Introduction.....	175
5.1 Participants.....	175
5.2 Overall Research Design	178
5.3 Study 1 – Learning OOP Concepts with Adaptive Support.....	180
5.3.0 Introduction to Study 1	180
5.3.1 Design of the Experiment	181
5.3.2 Treatments	182
5.3.3 Measures to Assure Validity of Results	182
5.3.4 Instruments	183
5.4 Study 2 – Problem Solving with Scaffolding for Self-regulation.....	186
5.4.0 Introduction.....	186
5.4.1 Design of the Experiment	187
5.4.2 Treatments	188
5.4.3 Measures to Assure Validity of Results	191
5.4.4 Treatment Materials and Instruments for the Experimental Study.....	192
5.3 The Qualitative Study	198
5.3.0 Introduction.....	198

5.3.1 Participants in the Qualitative Study.....	198
5.3.2 Data Collection Techniques	199
5.3.3 Data Collection	200
5.3.4 Analysis of Qualitative Data.....	201
5.4 Summary.....	203
CHAPTER 6 : RESULTS AND DISCUSSION	206
6.0 Introduction.....	206
6.1 The Experimental Study 1 Results	206
6.1.1 Introduction.....	206
6.1.2 Statistical Data Analysis for Study 1	207
6.1.3 Summary of Findings for Research Questions 1-2	212
6.2 The Experimental Study 2 Results	216
6.3 Results of the Comparative Case Study.....	236
CHAPTER 7 : CONCLUSIONS AND FUTURE WORK.....	253
7.1 Overview of the Findings.....	253
7.2 Achievements	254
7.3 Limitations of the Study	260
7.4 Further Work	262
Bibliography.....	265
Appendix.....	272
Appendix A Summary of comments from one evaluator during the design walkthrough.....	272
Appendix B Goal concepts and how they are related through prerequisite links.....	273
Appendix C Summary of Knowledge modules per concept	274
Appendix D Sample concept with its knowledge modules	275
Appendix E Sample pretest for the concept “Data hiding and Encapsulation”	278

Appendix F	Those who participated in the validation of instruments and evaluation of the system for learning.....	280
Appendix G	Sample Ill-structured programming problem for object oriented programming	281
Appendix H	Class Design Test	282
Appendix I	Scaffold for preparation of task schedule	286
Appendix J	Question Prompts.....	287
Appendix K	Brief Description of the Problem Solving Process	289
Appendix L	Sample Coding Scheme for the Element Declarative Knowledge under Knowledge of Cognition	292
Appendix M	Post Survey (After written exams in study 1).....	293
Appendix N	Metacognitive Awareness Inventory.....	295

List of Tables

1.1 Summary of key features of IIM and LTM	33
1.2 Classification of experimental inductive machine learning techniques	52
1.3 Hierarchical structure of the top/middle level of the learner model	80
1.4 Features and components of the instructional system	86
1.5 Features and components of the adaptive user interface system	87
1.6 Features and components of the problem solving support system	88
1.7 Design goals for instructional system	90
1.8 Design goals for the general user interface system	91
1.9 Design goals for the adaptive user interface system	91
1.10 Design goals for the problem solving support system	92
1.11 Main tasks for system users	93
1.12 Identifying roles of key user roles and their sub-roles for the learning system	94
1.13 Examples of user tasks and design scenarios for the different roles in the system ..	95
1.14 Contents for the instructional system	99
1.15 Contents for the adaptive interface system	100
1.16 Contents for the problem solving support system	101
1.17 Contents for the general interface system	102
1.18 Tables for awarding marks per question	129
1.19 Results of analysis of the trial questionnaire on the perceived value of the adaptive UI	142
1.20 Features considered during formative evaluation of the constructivist learning UI system	151
1.21 Showing how different elements of data were collected	157
1.22 Evaluation of the research sample size during the Research	177

Table 5-2 : Overall Research Design	178
Table 5-3 : Experimental Design for study 1 showing groups, treatments and observations	181
Table 5-4 : Experimental Design showing groups, treatments and observations for study 2	188
Table 5-5 : Aspects of product evaluation and their corresponding factors.....	197
Table 6-1 : Means, variances and test for equality of variances for CAT 1	208
Table 6-2 : Means, variances and the student's t-test for adaptive and non-adaptive groups ..	208
Table 6-3 : Means, variances and test for equality of variance for CAT 2.....	209
Table 6-4 : Test two means, standard deviation of adaptive and non-Adaptive groups	209
Table 6-5 : Mann Whitney U test of the difference in perceived value of the system's adaptive features between adaptive and non-adaptive groups.....	210
Table 6-6 : Correlation of perceived value of adaptive features of online system with performance score	212
Table 6-7 : Summary of the results from Study 1.....	213
Table 6-8 : Descriptive statistics and reliability levels for the pre-test and post-test of the 8 elements of metacognitive awareness.....	217
Table 6-9 : Descriptive statistics and reliability levels for the pre-test and post-test of the 4 problem solving processes	218
Table 6-10 : Showing means and Matched-Pairs Wilcoxon Signed-Ranks Test for the elements of Metacognitive Awareness	219
Table 6-11 : Showing means and Matched-Pairs Wilcoxon Signed-Ranks Test for the Problem Solving Processes.....	220
Table 6-12 : Test of gains between pre-test and post-test in terms of level of awareness of the 8 elements of metacognitive knowledge among the three treatment groups.....	222
Table 6-13 : Test of gains between pre-test and post-test in the level of awareness of the problem solving processes among the three treatment groups.....	225
Table 6-14 : Comparison of the effect of the scaffolds on the quality of the system developed using OOP concepts in terms of the three treatment groups.....	227
Table 6-15 : Comparison of product quality scores across the three scaffolding methods using ANOVA.....	228

Table 6-16 : Summary of the Results from Study 2 for Question 3.....	230
Table 6-17 : Summary of results from Study 2 for Question 4.....	232
Table 6-18 : Summary of frequencies for the elements of Knowledge of Cognition across the three treatments as manifested in the think-aloud protocols.....	237
Table 6-19 : Summary of frequencies for elements of Knowledge of Regulation of Cognition as manifested in the think-aloud protocols for the three treatments	240
Table 6-20 : Summary of frequencies of the codes for the problem solving processes as manifested in the think-aloud protocols for the three treatments	242
Table 6-21 : Summary of the code frequencies for the 8 elements of metacognition as found in the think aloud protocols for cases from the three treatment groups.....	245
Table 6-22 : Summary of the code frequencies for the problem solving processes as found in the think aloud protocols from the three treatment groups	249

List of Figures

Figure 2-1: Example checklist	48
Figure 2-2 : Learning Theories and Instructional Design Principles in the Eclectic Design Model	58
Figure 2-3 : Conceptual Framework: Variables for the research and their relationships	64
Figure 3-1 : Integrated Design Process without Adaptive Support and Problem Solving Support features, adapted from Nam & Smith-Jackson (2007).....	69
Figure 3-2 : Extended IDP with Adaptive Support and Problem Solving Support features.....	71
Figure 3-3 : The Phases of the Extended Integrated Design Process and their processes.....	73
Figure 3-4 : Learner model showing the top, goal and concept levels	81
Figure 3-5 : Interaction model for the adaptive user interface system	84
Figure 3-6 : General audience class hierarchy for the learning system	104
Figure 3-7 : Main track structure based on the general audience hierarchy.....	106
Figure 3-8 : The notation used for navigation design	107
Figure 3-9 : Task model for selecting concept to study.....	108
Figure 3-10 : Task model for studying a concept by an Advanced learner	109
Figure 3-11 : Task model for actual problem solving	110
Figure 3-12 : Advanced Learner Audience class track.....	112
Figure 3-13 : Conceptual structural model using structural links following the audience class hierarchy.....	113
Figure 3-14 : Components of a lesson page	118
Figure 3-15 : Hierarchical structure of the system.....	120
Figure 3-16 : Wireframe used for page layout design.....	122
Figure 3-17 : Hierarchical organization of the course into goals, concepts and knowledge modules.....	126
Figure 3-18 : Two questions for the first level of the pre-test for the concept “Introduction to object oriented programming and to key concepts”	128

Figure 3-19 : Presentation plan for pre-test questions	129
Figure 4-1 : Learner model showing domain independent and domain dependent attributes .	156
Figure 4-2 : Link annotation, adaptive presentation and additional material presentation for intermediate knowledge level learner	159
Figure 4-3 : Link annotation, adaptive presentation and additional material presentation for advanced knowledge level learner.....	160
Figure 4-4 : Adaptive link annotation and adaptive text presentation using initialized scores ..	168
Figure 5-1 : Warning message about number of attempts and time per level of the pre-test ..	183
Figure 5-2 : Screen for capturing sub-tasks in the task schedule	189
Figure 5-3 : Features of the Task Schedule Scaffold.....	190
Figure 6-1: Box-plots showing comparison of means and distribution of student scores in CAT 1 for the adaptive and non-adaptive groups.....	207
Figure 6-2 : Box-plots showing comparison of means and distribution of student scores in CAT 2 for the adaptive and non-adaptive treatment conditions	208
Figure 6-3 : Histogram of the perceived value score for the non-adaptive first group	210
Figure 6-4 : Histogram of the average perceived value score for the adaptive first group	211

Abbreviations

CDT	Component Display Theory
CLDM	Constructivist Learning Design Model
ELDM	Eclectic Learning Design Model
ET	Elaboration Theory
IDP	Integrated Design Process
K-NN	k-Nearest Neighbour
LTM	Long Term Memory
MAI	Metacognitive Awareness Inventory
NBC	Naïve Beyes Classifier
OLDM	Objectivist Learning Design Model
OOP	Object Oriented Programming
WBPT	Web-based Pedagogical Tools
WM	Working Memory
ZPD	Zone of Proximal Development

Glossary

Adaptation: This refers to the concept of making adjustments in an educational environment in order to accommodate diversity of learner needs and abilities and to maintain the appropriate context for interaction and achieve personalization

Adaptive Navigation Support: The goal of the adaptive navigation support technology is to support the learner in hyperspace orientation and navigation by changing the appearance of visible links. Adaptive navigation support makes it easier for the student to select the next link to follow by adaptively sorting, annotating, or partly hiding some of the links of the current page

Adaptive Presentation : The goal of the adaptive presentation technology is to adapt the content of a hypermedia page to the user's goals, knowledge and other information stored in the user model. In a system with adaptive presentation, the pages are not static, but adaptively generated or assembled from pieces for each user. For example, with several adaptive presentation techniques, expert users receive more detailed and deep information, while novices receive more additional explanation

Adaptive user interface: This is software that improves its ability to interact with a user by constructing a user model based on experience when interacting with the user

Cognition: Refers to domain-specific knowledge and strategies for information and problem manipulation

Cognitive Load: This is the demand placed on the learner's memory, in particular, working memory, as a result of processing novel information coming in through sensory memory and which the learner tries to make sense of during learning

Component Display Theory: This is a model for designing domain content for each concept. Each learning objective is related to the required content and to the desired performance outcome (simple to complex)

Constructivism: A philosophy of learning founded on the premise that, by reflecting on our experiences, we construct our own understanding of the world we live in.

Domain Model: This is the model of the course showing organization of the domain knowledge components

Elaboration Theory: A model for sequencing and organizing instruction. It is aimed at helping select and sequence content in a way that will help optimize the attainment of learning goals. It operates at the levels of learning goals and concepts

e-Learning: This is an umbrella term referring to learning that is facilitated and supported through information and communications technology (ICT). It is a broad set of applications and processes on a digital medium, which include online learning, web-based learning, computer-based learning, virtual classrooms, and digital tools and content. Delivery takes place through the Internet, intranets, audio- and videotape, satellite broadcast, interactive TV, and CD-ROM.

Ill-structured Problems : These are problems which are complex i.e. not clearly and fully defined, with information for solving them not adequately provided in the problem statement. The goals of solving the problems are not well defined, there are multiple ways of solving them, and no optimal solution among the many that are possible; there is a possibility of no solution at all

Knowledge of Cognition: One of the categories of metacognitive knowledge that refers to the activities that involve conscious reflection on one's cognitive abilities and activities

Learner Model: A learner model consists mainly of information about the individual knowledge (what knowledge the learner has, what knowledge he does not have and what knowledge he has wrongly acquired), preferences and interests, which determine the learner's behaviour. It reflects what the system believes about these aspects of a learner.

Learner Modeling: The process that involves the acquisition and representation of learner characteristics such as the knowledge of the learner about a given domain of knowledge, the learner's beliefs, learning style, and demographic information

Learning: Is the relatively permanent change in an individual's behavior or behavior potential (or capability) as a result of experience or practice

Learning Theories/Epistemology: These theories deal with how what exists can be known. They influence the formulation of specific teaching/learning strategies

Link Annotation: Aims at the visual augmentation of links in accordance to their importance for the user. Annotations can be realized in textual form but also visually, e.g. by using different icons, colours, or font sizes

Link Hiding: The purpose of navigation support by hiding is to restrict the navigation space for the user by hiding, removing, or disabling links to irrelevant pages. A page can be considered irrelevant for several reasons: for example, if it is not related to the user's current learning goal or presents materials which the user is not yet prepared to understand. Hiding protects users from the complexity of the whole hyperspace and reduces their cognitive overload

Machine Learning: This is an area of Artificial Intelligence which is concerned with the development of techniques (algorithms) which can be used to enable a system "learn" from what it "experiences", represented in the form of some data sets; the system uses this learned knowledge to perform better in solving similar problems

Metacognition: Is awareness of one's thinking and learning processes, enabling the student to direct one's learning in productive ways

Object Oriented Programming: This is a programming paradigm that provides mechanisms that support the object-oriented style of programming well. Objects that communicate with each other are used to create a model of a real world

Objectivism: Is the belief that there exists reliable and stable knowledge about the world and concepts within it; that this knowledge and skills can be concretely specified and structured; means can be determined to instill such knowledge and skills into learners effectively and efficiently; that is when learning is seen to have occurred

Problem Solving: Is the process of searching solution paths through a problem space. What is of interest is the learning as a result of this search process

Question Prompts: Refers to a set of static or adaptive questions or hints, which are designed to puzzle students at different stages of learning in order for them to reflect on the status of their learning and the activities they are engaging in, in order to achieve learning

Reflection: With reflection, the students look back over their past efforts to complete a task and analyze their own performance

Scaffold : Is a temporary support to facilitate learner comprehension and reflection on complex tasks

Scaffolding: Refers to support for learners while they participate in activities that are beyond their current capabilities

Self-regulation: Also known as regulation of cognition, self-regulation refers to the activities regarding self-regulatory mechanisms applied in an ongoing basis while attempting to learn or solve problems

Task Schedule: Is a personal task management scaffold for student learning and it contributes to the achievement of self-organized learning when using problem solving approaches such as project management

Web-based Learning: Refers to a paradigm of learning that uses the World Wide Web (WWW) as its' platform and medium. The web is a very suitable tool for distance learning (both adaptive and non adaptive) because of ease of access, wide visibility and hypermedia structure

Well-structured Problems: These are problems whose initial and goal states, together with the set of local operators for transition from initial to goal states, are clearly defined beforehand

The greatest obstacle to discovery is not ignorance; it is the illusion of knowledge.

—Daniel J. Boorstin

Chapter 1 : General Introduction

1.1 Introduction

Over the past two decades, a lot of research and development has been done focusing on experiential and student-centred learning, based on solving of complex and authentic problems (Brickell & Herrington, 2006; Jonassen, 2000, 2006). This is in recognition of the fact that if students solve authentic, real-life problems during learning, then the transfer of knowledge to their places of work after qualifying increases. This happens because the cognitive challenges faced in academic and real-life situations become similar (Savery & Duffy, 1995).

1.1.1 Ill-structured Problems

Ill-structured problems have characteristics which make them different from well-structured problems. The characteristics make these problems challenging to handle due to the cognitive pressure they cause the learner while solving them.

Characteristics of complex ill-structured problems

Complex, ill-structured, real-world problems are usually not fully defined and do not provide adequate information to guide problem solving. They also have multiple optimal solution paths. They are the type of problems regularly solved by people in their everyday and professional lives (Jonassen, 2000). In learning with such problems, the student plays a key role in problem definition, information seeking, and retrieving relevant prior knowledge.

Complex, real-world problems create opportunities for learners to acquire different types of knowledge and skills which are useful in real life.

- Students are able to understand facts and concepts in the context of a conceptual framework of a given subject area. The conceptual framework enables the learners to organize knowledge during learning in such a way that

retrieval from memory, application and transfer to solve similar and dissimilar problems in other contexts can occur (Bransford, Brown, & Cocking, 2000).

- Students are also able to acquire and apply vital skills such as generation of hypotheses, formulation of strategies for inquiry, problem formulation, evaluation of information, critical thinking and appropriate decision making (Schmidt, 1989; Sternberg, 2009).
- The students also acquire the ability to control and regulate their problem solving and learning (Valcke, 2002; Van Marrienoer & Sweller, 2005).

Provision of complex real-world problems does not guarantee success

The provision of complex real-world problems for the learners to solve in order to learn does not always succeed. The learners can acquire knowledge in a given domain but are not able to apply it to solve problems in other situations, unless they are prompted or provided with hints (Brickell & Herrington, 2006). This is due to lack of or not being aware of their metacognitive skills or even not knowing when to apply these skills during problem solving. Metacognition is the type of knowledge that enables students to reflect upon, understand, control and regulate their learning and problem solving.

Students suffer cognitive load if they have not mastered the domain knowledge and skills which they are to apply when solving an ill-structured problem. The same also applies if they have inadequate metacognitive skills or indeed if they cannot apply these skills. Usually, an instructor compensates for some of these weaknesses by intervening at critical points, giving hints, prompts or guidelines to enable the student to progress during problem solving.

1.1.2 Need for Scaffolding during Problem Solving

Scaffolds are needed to reduce the cognitive load associated with acquisition of the domain knowledge to be applied during problem solving. They are also needed to reduce the cognitive load experienced during problem solving.

Cognitive load

Cognitive load is the demand placed on the learner's memory, in particular, working memory (WM), as a result of processing huge amounts of novel information or information with a lot of interaction among its elements or information that is presented poorly such as by separating related information elements in time or space (Van Marrienoer & Sweller, 2005). WM is the part of memory that holds the information or knowledge elements currently being processed by the learner. With high cognitive load levels, the student gets frustrated and disoriented, and cannot succeed in ill-structured problem solving.

The instructor usually filters some of this information and provides it in bits, using a suitable sequence or presentation format, to ensure successful acquisition of knowledge. Therefore, in the absence of an instructor, there is need to ensure that necessary knowledge representations are formed in memory before the student commences problem solving, or else give support, such as hints or prompts, to steer the learner's problem solving forward (Brickell & Herrington, 2006).

Scaffolding

The hints or prompts are scaffolding messages to the learner during ill-structured problem solving. Scaffolds are temporary supports to enable the learner to successfully participate in activities that are beyond his current capabilities.

Cognitive and Metacognitive Scaffolds

There are scaffolds for cognition used during the learning of domain knowledge and for metacognition used during problem solving (Azevedo & Hadwin, 2005).

Cognitive scaffolds activate the student's domain knowledge in memory that is relevant to the new information being learned, retrieve the knowledge and enhance construction of new schemata in memory to represent new knowledge acquired as a result of integrating the new information with knowledge already in memory.

Metacognitive or self-regulation scaffolds improve the student's level of metacognition by making his thinking processes explicit, and prompting and guiding him to apply the components of metacognitive knowledge during problem solving. These include planning, processing information available for problem solving, monitoring progress, revising strategies to improve performance, and evaluating one's performance (Ge, 2001; McLoughlin & Hollingworth, 2001). Externalization of problem solving steps also provides opportunities to investigate one's thinking, reflect on progress, and also enables the learner to do self-regulation and take control over one's problem solving process (Brickell & Herrington, 2006; Valcke, 2002).

1.1.3 Need for Scaffolding in Online Learning with Ill-structured Problem Solving

The advantages of learning by solving complex, real world problems are also being tapped in online learning environments (Azevedo & Cromley, 2004; Brickell & Herrington, 2006; McLoughlin & Hollingworth, 2001). However, the challenges are that learners are mostly on their own, especially in non-collaborative learning and therefore, there is no one to guide or control the student's learning or problem solving. The students also face the challenge of having a lot of information from which to select relevant information. At the same time, the student has to map out the path to follow in learning the domain's concepts under these circumstances (Brusilovsky, 2001). The student also suffers because of too many fragments of information and links on a single page, especially if navigation, presentation or content design is poor.

Without relying on scaffolds in online learning, the students may end up with poor self-regulation skills as well as poor understanding of the domain concepts (Azevedo & Cromley, 2004). This would result in sub-optimal problem solving performance.

1.1.4 Adaptive Scaffolds

Adaptive scaffolds are necessary because learners are unique individuals with unique needs under different learning contexts.

Uniqueness of Individual Learners

In online learning environments, consideration should be given to the uniqueness of students in terms of their characteristics and learning conditions such as geographic location or network bandwidth. Moreover, some of the students' attributes are dynamic, varying with increased learning or interaction with the learning system. For example, learning styles and level of domain knowledge (Oboko, Wagacha, Omwenga, Libotton, & Odotte, 2009b; Papanikolaou, Grigoriadou, Kornilakis, & Magoulas, 2003; Virvou & Tsiriga, 2004; Wolf, 2004).

Therefore, the cognitive and metacognitive scaffolds employed have to vary according to the characteristics of learners in order to be beneficial to the learner's cognitive learning and problem solving. Otherwise, the students are given inappropriate support, making them suffer undesirable cognitive load in trying to reconcile unsuitable prompts, hints or guidelines with their knowledge level, learning style, preferences or orientation during learning. This degrades the learner's performance. Hence, supports need to be made adaptive in order to respond to the learner's current status and associated needs (Kalyuga, 2009; Van Marriënboer & Sweller, 2005).

Adaptive Scaffolds for Cognitive Learning

Adaptive learning techniques can be used to support acquisition of domain knowledge. They change the system's interface to suit the learner's characteristics. The techniques work by reducing the number of information elements to be processed at one time to a manageable level according to the learner's expertise, provide favourable presentation of the learning material and give hints to make it easier for the learner to identify an appropriate navigation path to follow through the web-based course content. The techniques can reduce the cognitive load suffered if used appropriately. Examples of these techniques are adaptive navigation support, adaptive presentation support and adaptive curriculum sequencing (Brusilovsky, 1999).

For instance, with adaptive navigation support through link hiding, some learning material links can appear or disappear automatically to reduce the student's information space to relevant information elements only. This can be based on how the student had previously used the materials in the course (Oboko, Wagacha, Omwenga, & Libotton, 2009c).

1.1.5 Metacognitive Regulation Scaffolds

The scaffolds make the students more aware of the elements of their metacognitive knowledge and problem solving processes. They also provide prompts or hints to stir the learners into applying the knowledge or problem solving processes, such as requiring the student to plan, monitor or evaluate their problem solving. The scaffolds also provide the learners with opportunities to record their plans for and progress in problem solving, a process called externalization (Valcke, 2002). The externalized plans or solution steps become the object of self-monitoring, self-regulation, self-reflection, self-explanation and justification, leading to reduction in cognitive load, better knowledge construction and progress in problem solving.

For instance, a learner can be forced to plan for problem solving by being required to identify sub-tasks of the problem and the knowledge, skills and other resources required to solve them, by using tools such as question prompts, task schedule, calendars, to-do-lists, post-its (Brickell & Herrington, 2006). He can be forced to monitor progress in problem solving by being required to keep a diary or journal of progress in the sub-tasks or by being shown a map or graph of his progress in the sub-tasks (Chris Quintana, Zhang, & Krajcik, 2005). He can be forced to do evaluation by being reminded of the goals set out at the beginning of problem solving.

1.1.6 Learning and Information System Theories for Web-based Learning Design

Web-based learning system developers do not adequately consider learning, instructional design and information system theories in order to determine suitable system development methodologies. Therefore, the selection of system goals,

which can be included to assure effective web-based learning systems include cognitive load theory, connectivism, social constructivism, social network theory and process virtualization theory.

Methodologies for designing and developing web-based learning systems which integrate both learning theories and information system development theories are more complete. They result in web-based learning systems which not only deliver content but also deliver learning. The systems enable learning in a way that is suitable to individual learner characteristics in order to maximize learning or problem solving as well as the usability and user experience for the learners (Nielsen, 2001).

1.2 Problem Statement

Scaffolding across the continuum of knowledge acquisition, beginning with supporting acquisition of domain knowledge at beginner level, to solving complex, authentic problems at expert level, all in one online learning system, in the context of management of associated cognitive load, has not been studied adequately in terms of adaptive, personalized supports for learners.

As a result, students tend to use inappropriate supports. This makes them to experience expertise reversal effect, which increases cognitive load as they learn domain knowledge or solve problems, instead of reducing it.

Without appropriate support, the students learning online on their own through solving complex problems feel frustrated, lose motivation and feel like failures. As a result, they not only have poor self-regulation skills but also a poor conceptual understanding of the subject, leading to poor problem solving performance.

1.3 Aim of the Study

The aim of this study is to explore ways of providing learning support to reduce cognitive load effects in an online learning environment with complex problem solving. The study is to contribute to the practice of design of web-based e-learning systems in

components and features to include in the system is not well guided by instructional design and principles of appropriate user interface design which include provision of affordances to encourage self-regulation in problem solving environments and adapting the interface to suit individual learner characteristics. Systems developed this way also fail to consider learning and problem solving tasks to be performed by the learners, the diverse and changing circumstances under which learning is to take place, the course content as well as individual learner characteristics which are diverse and keep changing (Hadjerrouit, S., 2007; Katia & Granger, 2000; Nam & Smith-Jackson, 2007; De Troyer, 2001).

Examples of web-based learning system features informed by learning theories are assessment integrated into the learning system, meaningful corrective and cognitive feedback, simple to complex sequencing of content and use of multiple content presentation formats such as text, multimedia, diagrams and pictures. Others are affordances on the system's interface such as features facilitating planning, monitoring and externalization of progress to enable learners achieve beyond current ability and nearer their potential. Examples of relevant learning theories are behavioral, cognitive and constructivist learning theories. Others are cognitive load theory, connectivist theory, elaboration theory, component display theory and theory of metacognition.

It is also important to involve prospective users of the systems throughout the design and development stages to ensure that requirements capture is as complete as possible. Such systems should also be developed through evolutionary and participatory development approaches, which allow for continuous formative evaluation, with the feedback being factored in to improve systems before it is too late. Design of such systems should also consider the fact that learning is increasingly becoming virtual and social at both individual and community level. The focus of developers of such systems has to go beyond technical considerations, content delivery and the assumption of universal and static learner characteristics. Examples of the information system theories

which can be included to assure effective web-based learning systems include cognitive load theory, connectivism, social constructivism, social network theory and process virtualization theory.

Methodologies for designing and developing web-based learning systems which integrate both learning theories and information system development theories are more complete. They result in web-based learning systems which not only deliver content but also deliver learning. The systems enable learning in a way that is suitable to individual learner characteristics in order to maximize learning or problem solving as well as the usability and user experience for the learners (Nielsen, 2001).

1.2 Problem Statement

Scaffolding across the continuum of knowledge acquisition, beginning with supporting acquisition of domain knowledge at beginner level, to solving complex, authentic problems at expert level, all in one online learning system, in the context of management of associated cognitive load, has not been studied adequately in terms of adaptive, personalized supports for learners.

As a result, students tend to use inappropriate supports. This makes them to experience expertise reversal effect, which increases cognitive load as they learn domain knowledge or solve problems, instead of reducing it.

Without appropriate support, the students learning online on their own through solving complex problems feel frustrated, lose motivation and feel like failures. As a result, they not only have poor self-regulation skills but also a poor conceptual understanding of the subject, leading to poor problem solving performance.

1.3 Aim of the Study

The aim of this study is to explore ways of providing learning support to reduce cognitive load effects in an online learning environment with complex problem solving. The study is to contribute to the practice of design of web-based e-learning systems in

the form of a design and development methodology for incorporating various ways of providing learning support. Learning is considered to take place beginning from the novice level, involving acquisition of domain knowledge, until the expert level, involving solving complex problems. The study is expected to evaluate a web-based e-learning system developed using this methodology and incorporating various learning supports. In particular, the study is to explore the effect of adaptive learning support technologies on the acquisition of domain knowledge and the learners' perception of the value of web-based learning environments incorporating these technologies. The study is also to investigate the effect of learner-centric scaffolding techniques on the level of metacognitive awareness, the level of application of ill-structured problem solving processes and the quality of the product of problem solving.

1.4 Research Objectives

The objectives for this research are:

- 1) To develop a web-based learning system that reduces learner cognitive load by providing support through adaptive user interfaces for cognitive learning and metacognitive control scaffolds for ill-structured problem solving processes
- 2) To establish whether use of adaptive presentation support and adaptive navigation support features of a learning system improves learning scores and perceived value of the system's adaptive features for the learners using the system
- 3) To establish the effect of scaffolding features on the level of metacognitive awareness of the learners
- 4) To establish the effect of scaffolding features on the level of application of problem solving processes and the quality of the product of problem solving

1.5 Research Questions

The study investigates the effects of cognitive and metacognitive scaffolds on the student's acquisition of domain knowledge, perceived value of the system's features,

level of metacognitive awareness, level of application of problem solving processes and the quality of the product resulting from the solving of ill-structured problems.

The research focused on the following questions:

- 1) Which group of online learners (those with adaptive support or those without adaptive support) achieves higher test scores when learning object oriented programming concepts?
- 2) Does the utilization of adaptive learning support lead to differences in the perceived value of the online learning environment?
- 3) Does the use of metacognitive scaffolds (question prompts or task schedule) have an effect on the students' level of metacognitive awareness while solving the ill-structured task?
 - a) Does the use of question prompts influence students' level of metacognitive awareness while solving the ill-structured task?
 - b) Does the use of a task schedule influence students' level of metacognitive awareness while solving the ill-structured task?
- 4) Does the use of metacognitive scaffolds (question prompts or task schedule) have an effect on student's level of application of problem solving processes when solving an ill-structured task?
 - a) Does the use of question prompts have an effect on students' solving of the ill-structured task in terms of level of application of problem representation, solution process, solution justification and monitoring and evaluation of solutions?
 - b) Does the use of a task schedule have an effect on students' solving of the ill-structured task in terms of level of application of problem representation, solution process, solution justification and monitoring and evaluation of

solutions?

- c) Does the use of metacognitive scaffolds (question prompts or task schedule) have an effect on the quality of the product of ill-structured problem solving?

1.6 Structure of the Thesis

The thesis is organized as follows:

Chapter 1 provides a brief introduction to the study: a brief introduction to the background of the study, the research problem, the research objectives, research questions and presentation of a high level organization of the thesis.

Chapter 2 presents a review of the previous work which informed the research: problem solving and in particular ill-structured problem solving, cognition, metacognition, scaffolding of self-regulation and adaptive scaffolding for acquisition of domain knowledge, cognitive load theory, scaffolding in web-based learning, the influence of theories of learning on instructional design and the conceptual framework for the research.

Chapter 3 presents the first part of research methodology for the study. It provides justification for and amendments to the Integrated Design Process (IDP), a methodology for the development of web-based learning systems. The chapter also illustrates how IDP was used to guide the development of the learning environment. It traces system development through the four phases of the methodology: needs analysis, conceptual design, development and formative evaluation.

Chapter 4 presents the second part of research methodology for the study. It describes how Machine Learning algorithms were applied for making inferences in the application of adaptive learning support technologies to make the user interface of the learning environment adaptive. It presents an introduction to adaptive user interfaces, a brief introduction to components of adaptive user interfaces and a description of how various

Machine Learning algorithms were used to make inferences in order to guide interface adaptation.

Chapter 5 presents the third part of research methodology for the study. It describes how experiments for the study were designed. It describes the two studies in the research, and for each study, it presents the context, research questions answered, research design, treatments and research instruments used.

Chapter 6 presents results and findings from the two studies in the research and their discussion.

Chapter 7 presents the conclusions, contributions and limitations of the research, as well as suggestions for further work.

1.7 Language Conventions

Depending on the context, the following sets of words have been used interchangeably in this thesis:

- Scaffolds and supports
- Learning system, learning environment and system
- Learner, student, problem solver and system user
- Web-based system and online system
- Metacognitive regulation, metacognitive control and self-regulation
- Cognition, domain knowledge and subject matter
- Online learning and web-based learning
- The terms “he”, “his”, “him” and “himself” have been used in reference to both genders.

The symbols used in the description of the Machine Learning algorithms, such as Σ , ∞ , \prod and $P(x_i/v_j)$ should be associated with their classical Mathematical meanings.

A single question can be more influential than a thousand statements.

- Bo Bennett

Chapter 2 : Literature Review

2.1 Introduction

There is need for universities and other institutions of higher learning to produce graduates capable of solving real-world problems; graduates who can anticipate and find realistic solutions to authentic problems (McLoughlin & Hollingworth, 2001). In order for this to be possible, the pedagogy applied in training these students ought to be refocused away from mere delivery of content to the students, or training the students on how to solve well-structured 'textbook' problems for which the requisite heuristics and strategies are either provided or easily obtainable. The focus ought to be on solving authentic, ill-structured problems (Jonassen, 1997) since the acquired skills are easily transferred to solving real-world problems (Savery & Duffy, 1995). Solving ill-structured, real-world problems must become the basis for improving learning.

Further, effective pedagogy needs to include the provision of support for learning problem solving processes, especially the related higher order skills. These skills are required for learning by solving ill-structured problems and are the foundations for effective higher order thinking. They include organizing, evaluation, metacognition (Azevedo & Hadwin, 2005), decision making and research (Oliver & Herrington, 2001).

The need for support during the solving of ill-structured problems is even more serious in distributed learning environments, where there is minimal support for the students due to the physical absence of the instructor and the more able peers of the students. All this is besides the complexity involved in solving ill-structured problems. The complexity is due to the number of information elements and domain specific strategies, and interactions among them, which the students have to contend with, as it causes increase in intrinsic cognitive load (Van Marrienoer & Sweller, 2005).

Students, especially those with low self-regulatory and metacognitive skills face additional difficulties when learning online compared to face to face learning (Dabbagh

& Kitsantas, 2004). They have to plan their problem solving, execute these plans, monitor their progress and determine by themselves when the problem has been adequately solved. Some of these skills, such as metacognition, which are critical for the students' successful learning in such environments, develop late in people and are mostly poorly developed even among adults (Schraw, 1998). Metacognition is important for problem solving because it enables students to be in control of their learning by being able to regulate themselves through planning, information management, making amendments to their approach to ensure continued progress in terms of time spent and progress towards task completion, monitoring and evaluation (Dabbagh & Kitsantas, 2004; Oliver & Herrington, 2001; Schraw & Dennison, 1994).

Learning support in online distributed systems can be achieved by providing adaptive scaffolds which are personalized to the individual learner. However, the scaffolds which have been used have not been adaptive most of the time (Azevedo & Hadwin, 2005). However, the supports which have been adaptive have depended a lot on an external agent, such as an instructor or the student to activate them, calibrate them or even initiate fading (gradual withdrawal of support) when the students are able to manage their learning (Azevedo, Cromley, Winters, Moos, & Green, 2005; Quintana et al., 2005).

Interaction should be between the student (human) and the system (computer) and the interface between these two should recognize the individual student interacting with it and adjust itself accordingly. Hence, the interface adapts to the student by providing support suitable for that particular student, which also spares the student some cognitive load effects such as expertise reversal and redundancy (Kalyuga, 2009; Van Marrienoer & Sweller, 2005). The interface between the student and the system is planned to be interactive, dynamic and flexible, continuously adjusting itself to the individual student as interaction goes on. This is especially critical for distributed learning environments, such as web-based learning systems, where there is no physical presence of an instructor or more able peer or technical person to provide learning support to the student (Azevedo & Hadwin, 2005; Dabbagh & Kitsantas, 2005).

The rest of the chapter is organized as follows:

Section 2.2 introduces problem solving and describes well-structured and ill-structured problems, together with the problem solving processes. Section 2.3 introduces cognition and metacognition and describes the relationship among cognition, metacognition and ill-structured problem solving. Section 2.4 introduces scaffolding and section 2.5 describes the cognitive load theory framework, types of cognitive load and the relationship between scaffolding and cognitive load theory. Section 2.6 introduces approaches for scaffolding metacognition in online learning, and discusses question prompts and task schedule in detail. Section 2.7 introduces adaptive scaffolds for cognitive learning. Section 2.8 presents theories of learning and related instructional design models, as well as two instructional design theories for designing learning objects for an adaptive learning system. Section 2.9 provides a summary of the chapter and presents the conceptual framework for the research.

2.2 Problem Solving

2.2.1 Introduction to Problem Solving

Problem solving is a part of thinking. It is the most complex among intellectual functions and is one of the higher-order cognitive processes. It requires the transition and control of more regular or elementary skills to enable the person solving the problem to know how to proceed from a given state of the problem to a preferred goal state.

Even with increase in the attention being paid to the development of learning environments which support active, student-centered learning, not much attention has been given to the need to support the development of problem solving skills in technology enhanced learning. This is in spite of the possibility of problem solving skills providing a firm foundation for effective higher order thinking (McLoughlin & Hollingworth, 2001). Bloom's taxonomy of learning objectives has six cognitive learning objectives i.e. knowledge, comprehension, application, analysis, synthesis and evaluation (Bloom & Krathwohl, 1956). Of these, analysis, synthesis and evaluation have

elements which are required for solving problem. Nonetheless, this cognitive load motivates the construction of the necessary schemata which are used to guide problem solving (Van Marrienoer & Sweller, 2005).

2.2.2 Well-structured Problems

These are the kind of problems mostly found in school subject tests and textbook exercises, with clear, though not necessarily easy paths to a solution. They are also called move problems since they require a series of moves from an initial goal state to reach a final goal state (Stoyanov, 2001; Woods, 1994). Well-structured problems require a student to apply a restricted number of concepts, rules, and principles which have been learned, to a restricted problem situation (Jonassen, 1997), also called restricted encoding (Van Marrienoer, Clark, & de Crook, 2002).

Such problems have:

- a well defined initial state of the problem
- a well defined end state of the problem, and
- one solution that is considered the best for the problem

Information needed for solving the problem is well defined or sufficiently provided in the statement of the problem (Jonassen, 1997).

The shortcoming of such problems when used for learning in school contexts is that they have little relevance to the students' real life experiences and therefore, the knowledge learned has reduced chances of retention in LTM (Kalyuga, 2009; Van Marrienoer et al., 2002). Moreover, the knowledge acquired is not easily transferrable to real-life situations because the problems themselves are not authentic (Savery & Duffy, 1995; Van Marrienoer & Sweller, 2005).

2.2.3 Ill-structured Problems

Ill-structured problems are typically complex, ill-defined and not well demarcated (Jonassen, 1997). The problems also have several possible solutions (Van Marrienoer & Sweller, 2005). They have the following features:

- Generally, such problems have one or several aspects of the situation not specified.
- Generally, the goals are vaguely defined or unclear
- The descriptions of the problems are, in general, unclear
- The information needed to solve the problems is not entirely contained in the problem statements. Hence, it is not obvious what actions to take in order to solve them.
- The path(s) to the final solution(s) is not clear and/or may not be exclusive.
- The solution to the problems could involve more than one or no solutions at all (Sternberg, 2009)

In solving such problems, the problem solver is initially unsure of how to proceed and also about the connection between the given data/information and the solution or goal-state (Jonassen, 1997; Woods, 1994). Typically, the problem solver works backwards from the goal and searches for connections between the goal and the given information, an approach called means-ends analysis (Stoyanov, 2001). This approach generates cognitive load (Van Marrienoer & Sweller, 2005). If the problem solver is not provided with knowledge or information that is required, then he has to look for them before carrying on with problem solving. This process increases cognitive load because of the many elements of information to be accessed and processed before settling on the appropriate information for solving the given problem.

During the solving of ill-structured problems, the problem solver needs to process a lot of interacting information elements together. These elements include domain knowledge and cognitive strategies, as well as problem solving processes and elements of metacognitive knowledge which apply across domains such as ability to engage in

monitoring and self-regulation (Valcke, 2002). These interacting information elements need to be managed well by the student in order to succeed in problem solving.

The ill-structured problem creates a context for learning. The context provides the student who is solving the problem with sufficient experience to develop skills, knowledge and expertise of domain content and cognitive strategies. This is through participation in exploration, inquiry and application of different types of knowledge and skills in the development of the solution. The context must also include elements to facilitate the application of problem solving processes as well as strategies for investigation, planning, designing, retrieval and application of knowledge, reflection and evaluation in solving the problem and the development of a product.

The challenge with using ill-structured problems for learning is that they are more open-ended and difficult to solve when compared to well-structured problems. The transfer of acquired knowledge and skills across contexts also rarely takes place successfully, especially for domains which are highly complex and ill-structured (Ge, 2001). The students find themselves faced with too much novel information with high levels of interactivity, for which they have no related prior knowledge to help in organizing it. The students also have no knowledge of the problem solving processes to apply, leading to the application of poor problem solving methods such as means-ends analysis, which leads to high cognitive load.

Mostly, the problem solvers also have poor self-regulation skills (Oliver & Herrington, 2001). They should be provided with hints to trigger self-regulation or opportunities for externalization of their problem solving processes and intermediate solution steps to reduce cognitive load. This reduces the amount of information to be processed presently in WM and encourages further reflection and taking of corrective action to ensure progress in problem solving. Otherwise, learning by solving ill-structured problems becomes frustrating.

2.2.4 Problem Solving Processes

The steps of the problem solving cycle include problem identification, problem definition/representation, strategy formulation, organization of information, allocation of resources, monitoring and evaluation (Sternberg, 2009).

Problem identification involves recognizing what the goal of problem solving is, e.g. if one wants to write a term paper, what question will it address. Problem definition and representation involve expressing the problem clearly in measurable terms, including constructing a mental model of the problem space with the sub-problems which can be tackled, defining boundaries for the problem and also defining who owns the program. It may also include providing justification for the decisions. Strategy formulation involves generating the set of steps needed to solve the problem and combining these steps into a workable strategy for solution. Justification for the selected solution is also provided. Organizing information involves decisions on how to represent information about the problem and it should be in such a way that the problem solver will find it easy to implement the formulated strategy for solving that problem.

Resource allocation involves allocating mental and physical resources, such as time, money, equipment and space. For example, allocating more mental resources to planning can save one time and energy and avoid frustration later on. Monitoring involves checking on one's resource usage e.g. time, as well as progress towards the goal in the course of problem solving, instead of waiting until the end. Remedial action can be taken if necessary. Evaluation involves making judgment on the solution after it is finished, and may lead to new problems being recognized, the problem being redefined or new strategies coming to light (Sternberg, 2009).

There are other ways in which problem solving processes have been organized. Voss and Post (1988) identified the stages of problem solving as problem representation, the solution process, monitoring and evaluation, with justification being provided for all the decisions/choices made.

- **Problem representation** involves constructing a problem space, including defining sub-problems, searching and selecting information for solving the problem. It also involves developing justification for the selected definition and representation of the problem.
- The **solution process** involves both **solution generation** and **solution selection**. Solution generation involves generating alternative strategies for solving the problem, including the steps to be followed, the knowledge and skills to be applied and the resources to be expended for each alternative solution. Solution selection involves identifying the best solution from among the alternatives and providing the reasons for it.
- **Justification** is very important for ill-structured problem solving because the problems are complex in nature (Jonassen, 1997; Zhang, 2004) and therefore there is a multiplicity of possible solutions. The problem solver has to provide convincing and well-argued explanations to support the selections as well as provide evidence in the form of facts to back these explanations.
- **Monitoring** involves checking how resources are being utilized and how progress is being made towards solving the problem. It includes any steps taken by the problem solver to ensure he remains on course to successfully solve the problem.
- **Evaluation** involves establishing if the problem has been solved well and if not, to identify any other improvements which can be made to solve the problem better, such as consideration of alternative solutions.

Both approaches describe the problem solving processes well. But since the second approach emphasizes justification more than the first one, it is more suited to the description of ill-structured problem solving processes. This is the case because for this type of problems, there is no clear definition of the problem, no clear guidelines and even no clear presentation of the goal of solving the problem.

2.3 Problem Solving in Relation to Cognition and Metacognition

2.3.1 Components of Ill-structured Problem Solving

The successful solving of ill-structured problems requires two important types of knowledge:

- **Cognition:** This is the knowledge a student has about concepts and specific strategies for learning or solving problems in the subject domain. The knowledge is represented as cognitive schemata in the learner's memory. Students who are experts in a given subject domain apply the knowledge they have of the domain intentionally to solve ill-structured problems from the domain (Ge, 2001). Therefore, domain or subject knowledge is an important component during the process of solving ill-structured problems.
- **Metacognition:** This is the other type of knowledge and it involves thinking about how one thinks, with a view to managing one's thinking for effective problem solving. Ill-structured problems have no clearly defined goals and usually have a number of possible alternative solutions from which the problem solver has to select the best solution and develop justification for it. The student is called upon to apply self-regulation skills in the selection and implementation of the solution.

The components of ill-structured problem solving are discussed in detail next.

2.3.2 Cognition

When students learn through direct experiences as with problem solving, they must be provided with the underpinning knowledge to guide the experience (Oliver & Herrington, 2001). They must be provided with the knowledge of and access to the information, principles and concepts they are to apply during problem solving. The knowledge of the domain which underlies the problem to be solved is known as cognitive knowledge.

Cognition is the source of knowledge, skills and strategies that are specific to the domain of the problem and which are to be applied by the problem solver while performing tasks or solving sub-problems related to the problem being solved (Schraw, 1998; Van Marrienoer et al., 2002). Expert problem solvers intentionally use the domain knowledge they have in solving ill-structured problems (Ge, 2001; Van Marrienoer et al., 2002). Its acquisition can be problematic, especially where learners have little or no support from other learners or an instructor, as is the case with online non-collaborative learning. In this case, the learners do not even have opportunities to discuss in groups the knowledge and skills they have acquired which are considered necessary for problem solving, participate in social negotiation and evaluate the viability of each other's individual understanding. This engagement with peers or the instructor leads to further evolution of the learner's knowledge.

Apart from the knowledge of domain-specific concepts, cognition involves knowledge of how different concepts forming domain knowledge are interrelated. This relationship is used to form the student's mental model of the domain, which contains the domain's concepts and domain specific strategies for problem solving (De Villiers, 2002; Van Marrienoer et al., 2002).

The mental model is used to facilitate efficient retrieval of knowledge which is to be applied during problem solving. During problem representation, an appropriate schema of domain knowledge guides selection of the factors/concepts relevant in the context of the problem. While constructing a problem space, experts also use structural information to have a well defined routine for searching critical information and have the ability to terminate immediately after collecting a sufficient amount of information.

The schemata contained in long term memory is also used to help organize novel information, which is being processed in working memory, through being added to existing and related schemata. This results in one complex element of schemata in long term memory instead of several separate elements that need to be processed

individually in working memory. This is how expertise increases. The higher the number of separate elements to be processed individually in working memory, the higher the chance of cognitive overload (Van Marrienoer & Sweller, 2005).

To reduce cognitive load and ease the process of acquisition of the domain's cognitive schemata in online non-collaborative learning, suitable supports can be introduced to guide the presentation and sequencing of subject content in a way that is favourable to the learner (Van Marrienoer et al., 2002).

2.3.3 Metacognition

Metacognition refers to the range of learning strategies and the capacity required to manage one's own learning (McLoughlin & Hollingworth, 2001). It is the ability to reflect upon, understand and control one's learning (Schraw & Dennison, 1994). Students who have metacognitive ability are capable of planning, sequencing and monitoring their problem solving (or learning), leading to improvement of their performance. Students who have metacognitive ability are also more strategic in their approach to problem solving and learning in general, and perform better than students who lack metacognitive ability (Schraw, 1998).

Solving ill-structured problems requires the student to not only learn procedures for solving problems such as identifying the problem, defining the problem, planning how to solve it and testing the solution but also knowing when to apply such strategies during problem solving. Metacognition is a student's knowledge of strategies and cognition, and the student's ability to monitor and regulate those processes. Regulation and control of the processes takes place through planning how to proceed, monitoring and self-orientation, and self-evaluation of one's performance to find out if the purpose which the student had in mind has been met i.e. whether the problem has been solved.

Cognition is domain-specific, whereas metacognition is general and is applicable to many, perhaps even unrelated domains/subjects (Schraw, 1998). Since domain knowledge such as intellectual ability and academic achievement does not guarantee

higher levels of metacognition, there is need to emphasize both cognition and metacognition during problem solving as both contribute to successful problem solving.

According to Brown's model of metacognition (1987), metacognitive knowledge is divided into two categories: (i) knowledge of cognition and (ii) knowledge of regulation of cognition.

(a) Knowledge of cognition

Knowledge of cognition is about what individuals know about their own cognition and includes three kinds of metacognitive knowledge:

- **Declarative knowledge** - Knowing about "things". This is knowledge about oneself as a learner and what factors influence one's performance, such as memory capacity limitation, rehearsals and distributed learning
- **Procedural knowledge** - Knowing how to do "things". Knowledge of how to do things is represented as heuristics and strategies. Individuals with more procedural knowledge perform tasks more automatically, have a higher range of strategies to use, sequence strategies more effectively and put strategies to better use
- **Conditional knowledge** - Knowing when and why to do "things". This is knowledge of when and why to use declarative and procedural knowledge. It includes knowledge of when and what information to rehearse, how to selectively allocate resources and use strategies more effectively and to adjust to changing situational demands of each learning task/problem being solved.

(b) Regulation of cognition

Regulation of cognition is made up of abilities and related activities which enable students to control how they solve problems and also how they learn. Metacognitive regulation is important for problem solving and learning. It improves performance by leading to better use of attentional resources that enable students to focus on what is of higher priority or most relevant to the current stage of problem solving , better use of

existing strategies, and a better understanding of breakdowns in the problem solving process (Schraw, 1998). When students are given instruction on how to use regulation of cognition, even young, novice learners acquire these skills (Schraw, 1998). The skills are independent of age (Gama, 2004). Research has also shown that improvement in one aspect of regulation of cognition, such as planning, may lead to improvement in others, such as monitoring.

What happens in the absence of such instruction designed to impart regulatory metacognition skills? There is no such instruction in distributed learning environments where learners are away from their instructors and their more knowledgeable peers. Research has shown that those who learn complex topics without support, especially in computer based learning environments, not only have poor self-regulation skills but also do not gain a conceptual understanding of the topic they are learning (Azevedo & Cromley, 2004; Greene & Land, 2000).

There are many ways of classifying metacognitive regulation skills in literature (Azevedo & Hadwin, 2005; Dabbagh & Kitsantas, 2004; Quintana et al., 2005; Woods, 1994) but three vital metacognitive regulation skills are common: (i) planning, (ii) monitoring and (iii) evaluation.

- **Planning**

Planning is done before commencing problem solving. It entails the setting of goals, selection of appropriate strategies and budgeting of resources that affect performance during problem solving such as allocating time to tasks. Planning also entails identifying other resources needed such as skills and knowledge, and allocating resources to appropriate stages of problem solving. Strategy sequencing is also part of planning. Poor or less experienced or young (read novice) problem solvers have less planning ability than more experienced or older problem solvers (Schraw, 1998).

- **Monitoring**

Monitoring involves deliberately checking and testing to be aware of one's progress in performing problem solving tasks. It also involves selecting appropriate corrective strategies when strategies selected originally are not working, such as revising or re-scheduling tasks (Gama, 2004). An example of monitoring is keeping daily records of one's efforts and successes in performing a task or engaging in periodic self-testing. This assists the problem solver in determining how to make adjustments so as to attain the set goals. Research has shown that monitoring ability develops slowly and is quite poor in children and even in a big proportion of adults (Pressley & Ghatala, 1990). Nonetheless, monitoring ability improves with training and practice (Shraw, 1998).

- **Evaluation**

Evaluation means comparing outcomes of performance, such as the product of problem solving or the efficiency of one's learning, with a standard or goal (Schraw, 1998; Zimmerman, 2000). A number of studies have shown that metacognitive knowledge and regulatory skills such as planning and monitoring are related to evaluation (Baker, 1989). Those who in general are poor problem solvers will also not be good at evaluating their performance in terms of processes applied and how they were applied, as well as the quality of the product of problem solving, if there is any. For such problem solvers, the emphasis should be on the problem solving processes, and not evaluation specifically.

Five Component Version of Regulation of Cognition:

Schraw and Dennison (1994) identified five components of regulation of cognition that facilitate control of learning:

- **Planning** - This involves goal setting, identification of strategies, knowledge and skills and other resources needed
- **Information management** – This involves formation of skills and strategy sequences

- **Monitoring** – This involves assessment of progress on one’s problem solving or learning process
- **Debugging** - This includes strategies of correcting one’s performances to ensure attainment of set goals
- **Evaluation** – This is assessment of one’s learning performance, product quality and strategy effectiveness at the end of problem solving.

The sequencing of skills and strategies (i.e. information management) together with strategies for correcting one’s performance to ensure staying on course to achieving set goals (i.e. debugging) are means of ensuring effective implementation of metacognitive strategies (Schraw & Dennison, 1994). Therefore, the five categories can also be collapsed into the three categories of planning, monitoring and regulation, and evaluation (Quintana et al., 2005; Schraw, 1998).

Metacognition is indispensable. It can be deliberately improved in a student through training. Therefore, there is need to find ways of improving a students’ (i.e. problem solvers’) metacognitive skills (Schraw, 1998). It is also necessary to find ways of facilitating the learner to apply metacognitive skills, especially when learning by solving complex, ill-structured problems, away from a supportive instructor or more able peer. The process of providing such supports to make it possible and easier for a learner to apply metacognitive skills during ill-structured problem solving is called scaffolding. Scaffolding is discussed in detail next.

2.4 Scaffolding

2.4.1 Introduction

Scaffolding describes the special form of learner support used extensively in modern learning settings to support the learning of domain knowledge as well as solving of ill-structured problems. This is especially vital when the student has to take greater

responsibility for his own learning, as is the case with constructivist, web-based learning environments.

Scaffolding involves controlling those elements of learning that are initially beyond a student's ability, therefore allowing him to concentrate on and complete only those elements that are within his range of capability (Oliver & Herrington, 2001). This allows the student to solve problems or perform tasks that would otherwise be out of reach (Becvar, 2005). This is in line with Vygotsky's notion of Zone of Proximal Development (ZPD), which describes the range between what tasks the student can accomplish alone and what he could accomplish with assistance (Becvar, 2005). Therefore, scaffolding is used to bridge the gap between a student's observed ability and the same student's potential abilities (Chng, 2001; Oliver & Herrington, 2001).

Scaffolding can be provided by the instructor or a more knowledgeable peer. It can also be provided by the learning system through the way instruction elements are designed and sequenced and the way feedback is provided through use of adaptive learning features. Tools which offer learners opportunities to engage in reflection, self-explanation and self-regulation can also be used (Azevedo & Hadwin, 2005).

2.4.2 Fading

The support reduces as learning progresses since the student takes charge of his learning more and more, due to improvement with time spent learning and experience gained. When the support is no longer needed, it is withdrawn. This process is called fading. Learning should go on after the fading of the scaffold.

2.4.3 Scaffolding in Online Learning Environments

Online learning environments can be designed to provide scaffolding; that is provide enough support to enable students to succeed in more complex tasks, thereby extending the range of experiences from which they can learn (McLoughlin & Hollingworth, 2001). Online learning through solving complex, ill-structured problems

can cause the students high levels of cognitive load. This can make them feel frustrated, lose motivation and end up feeling like failures if they are not supported by the system in their learning or problem solving tasks, especially since they are mostly on their own. Learning by solving complex, ill-structured problems requires learners to make more input into their learning and also apply metacognitive skills (Oliver & Herrington, 2001). Scaffolding reduces these negative effects due to cognitive load and enables learners to achieve higher levels of success than was otherwise possible.

2.4.4 Scaffolding Cognition

Scaffolding strategies work by improving the students' cognition through the activation of their cognitive schema, retrieval of knowledge and enhancement of comprehension and schema construction. Techniques such as simple to complex sequencing of instruction and provision of complete examples followed by incomplete examples and lastly conventional problems can be used (Van Marrienoer et al., 2002; Van Marrienoer & Sweller, 2005). These approaches reduce element interaction at the beginning, then provide progressively complex information as the learner's constructed schemata in LTM becomes more complex. These constructed schemata assist in organizing information in WM during processing. The techniques reduce undesirable intrinsic cognitive load and extraneous cognitive load effects (Kirschner, Sweller, & Clark, 2006).

2.4.5 Scaffolding Metacognition

Scaffolding strategies also improve metacognition by making the students' thinking explicit. The strategies also guide the students in planning their problem solving, organizing the information necessary for solving the problem, monitoring their progress, revising their strategies to improve performance and evaluating their performance (Ge, 2001). The strategies also enable the learner to take control of his learning and problem solving, engage in reflection of progress in terms of steps completed in problem solving, self-explanation and justification of the reasons for this. This can be done in many ways.

2.5 Cognitive Load Theory

2.5.1 Introduction

Cognitive load is the demand placed on the learner's memory, in particular, working memory (WM), as a result of processing novel information coming in through sensory memory, which the learner tries to make sense of during learning (Van Marriënboer & Sweller, 2005). WM holds the information the learner is focusing on at any moment, say, during learning or task handling. WM is therefore important during conscious processing of information by the learner. In general, if cognitive load is less, then WM has more capacity for handling the actual learning task. Cognitive load theory can be used to guide instructional design to achieve increased ease of information processing in WM.

2.5.2 The Human Cognitive Architecture and Cognitive Load

Cognitive load is related to the human cognitive architecture. The human cognitive architecture is made up of two parts which are crucial for learning: working memory (WM) and long term memory (LTM).

WM holds the information the learner is processing consciously at the present time. LTM holds the knowledge the learner has learned successfully, also called prior knowledge, organized in the form of schemata.

WM has two key characteristics: (a) WM is limited in terms of capacity and (b) in terms of the duration it can hold an element of information. WM has a capacity of only 5 to 7 independent chunks of novel information, which reduces to only 2 or 3 chunks when there is interaction among the elements (Paas, Renkl, & Sweller, 2003; Van Marriënboer & Sweller, 2005). Both complex and simple information elements are counted as single elements. WM loses all its information within 20 seconds. When compared to the number of interacting elements involved in most substantive human intellectual activities, WM's limitations can only allow trivial human activities (Paas et al., 2003).

On the other hand, LTM is effectively unlimited in terms of capacity and the duration knowledge elements can be retained in it (Paas et al., 2003; Van Marrienoer et al., 2002). The attributes of WM and LTM are summarized in Table 2-1.

Table 2-1: Summary of key features of WM and LTM

Attribute	Working Memory (WM)	Long Term Memory (LTM)
Kind of information	Information the learner is processing consciously at the present time	Knowledge the learner has learned successfully
Memory Capacity	Limited to 5-7 independent elements of novel information or 2-3 interacting elements	Unlimited
Volatility	Loses all information within 20 seconds	More long term storage

The knowledge held in LTM is organized in the form of structures such as mental models and cognitive strategies. A mental model is a declarative representation of the way the world is organized, such as the subject domain being studied by the learner. It guides reasoning about the domain. The elements forming the mental models have non-arbitrary relationships among them. Cognitive strategies are domain specific and are interpreted to guide the problem solving process. They reflect the way problem solving may effectively be approached in the domain, including rules of thumb or heuristics on how to progress from phase to phase (Van Marrienoer et al., 2002).

When handling information retrieved from LTM, WM does not experience the capacity limitation. This is because LTM organizes many elements into a single complex element with a specific function. This makes it possible for WM to process a small number of elements, which might be containing a large number of lower level interacting elements (Van Marrienoer & Sweller, 2005). Therefore, a large number of elements which could exceed the WM capacity if processed individually are processed as a small number of (perhaps complex) elements (Kirschner et al., 2006). In this sense, LTM alters the characteristics of WM by organizing the information to be processed in WM into a number of elements that is within the capacity range of WM. This reduces cognitive

load, and therefore improves learning (Van Marrienoer & Sweller, 2005). Hence, the knowledge a learner has in LTM, which is organized as schemata of varying complexity, is the source of human expertise. It makes it possible for WM to process information currently being focused on by the learner.

When the learner has no prior knowledge (or if prior knowledge is not activated), the learner engages in arbitrary reasoning with many elements which have not been stored in schemata in LTM. The information is organized randomly in WM (e.g. random generation of problem solving moves), after which the organized information elements (e.g. the moves) are tested for effectiveness since without prior knowledge, no other mechanism is available.

As the number of elements to be organized increases in a linear manner, the number of possible combinations to be tested increases exponentially. WM capacity cannot handle this, leading to an overload of cognitive resources (Van Marrienoer & Sweller, 2005). Learning becomes inhibited.

The same happens when learners are required to solve complex, ill-structured problems without relevant schemata in LTM i.e. without prior knowledge. Hence, prior knowledge is considered the most important cognitive characteristic of the learner in influencing learning processes and cognitive performance (Kalyuga, 2009).

2.5.3 Types of Cognitive Load

There are three types of cognitive load: (a) extraneous, (b) intrinsic and (c) germane.

(a) Extraneous Cognitive Load

Extraneous cognitive load wastes cognitive resources and is due to factors which are not core to the instructional material. These factors include poor presentation design of the materials, poor selection and sequencing of tasks to be performed by the learners or using unguided problem solving approach for novice learners without learning support (Kalyuga, 2009; Van Marrienoer & Sweller, 2005).

Among others, the methods may split the attention of the learner due to multiple related sources of information being presented in a such a way that they are separated in location or time. The same information may also be duplicated in more than one modality such as visual and audio presentation. The methods may also lead to the learner being presented with several physically disintegrated sources of information (Kalyuga, 2009). As a result, valuable WM cognitive resources are used up and therefore, the available limited resources become overloaded, hence slowing down learning and problem solving.

(b) Intrinsic Cognitive Load

Intrinsic cognitive load is the cognitive demand placed on the learner's WM due to the element interactivity in the learning material, and which varies according to the expertise of the learner. If the level of interactivity is high, then the number of elements to be processed simultaneously in WM is high. Such material may be difficult for the learner to process unless appropriate cognitive schemata incorporating the interacting elements exist in LTM (Van Marrienoer & Sweller, 2005). A larger number of interacting elements for one person (e.g. novice learner) might be considered as a single element in WM for another person who is more experienced, which reduces the demand for cognitive resources in WM. Without LTM schemata, random processing of presented information elements and random solution searches are applied, which use up the limited WM capacity. This increases cognitive load, and learning and problem solving become inhibited as a result (Kalyuga, 2009).

Both extraneous cognitive load and intrinsic cognitive load are concerned with how information is "presented" to the learners (Valcke, 2002). Both can be influenced through instructional design.

(c) Germane Cognitive Load

Germane cognitive load enhances learning. It motivates learners to invest mental effort in their learning tasks so as to make use of any extra cognitive resources made available

due to reduction of the total of extraneous and intrinsic cognitive load. Mindful engagement in schema construction initially increases cognitive load, called germane cognitive load, which increases the learner's motivation i.e. construct new schemata or elaborate the ones existing (Van Marrienoer & Sweller, 2005). The constructed schemata further reduces intrinsic and extraneous cognitive load, further freeing up more cognitive resources, which the learner can use with newly learned information to construct more advanced schemata. This continues over many cycles.

The view of germane cognitive load presented above builds on the cognitive perspective of learning. There is another view of germane cognitive load that builds on metacognition, which is also related to constructivism (Valcke, 2002). This is the type of germane cognitive load that is generated to enable the learner to cope with metacognitive processes such as planning, monitoring, information management, debugging and evaluation during schema construction as well as application of the knowledge during problem solving.

Constructivism meets with germane cognitive load in the way both encourage the active construction of external representations of the learners' internal representations (i.e. schemata), to enable learners express themselves. Externalization gives the learners a chance to interrogate their own knowledge, which forces them into self-explanation in order to justify the positions taken while constructing the external representations (Chng, 2001; Sorden, 2005; Van Marrienoer & Sweller, 2005). An explicit monitoring activity is also invoked, which increases germane cognitive load. Engaging in externalization of internal processes, self-explanation and reflection on the external representation, which spurs further self-explanation, results in further elaboration and better organization of the learners' schemata (Valcke, 2002).

Therefore, germane cognitive load motivates learning and it has two components: cognitive load related to the construction and storage of cognitive schemata and

cognitive load related to metacognitive monitoring and control (self-regulation) of the problem solving processes (Valcke, 2002).

2.5.4 Expertise Reversal Effect and Adaptive Learning

Expertise reversal effect is experienced in long training programs. It is as a result of the interaction of various effects associated with extraneous cognitive load, and the learner's level of expertise.

Modality effect: This is due to novice learners being provided information in one mode, audio or visual, yet they may prefer both audio and visual sources of information presented together. However, when their expertise becomes advanced, they prefer the visual mode only and therefore, the auditory source becomes redundant (Van Marrienoer & Sweller, 2005).

Split attention and redundancy effects: Novice learners may prefer several sources of information, such as text and diagrams, to be physically integrated, in order to avoid splitting a learner's attention. If the sources of information are separate, the learners are required to integrate the information mentally, which causes extraneous cognitive load. When the learners become experts over time, they may not need some of the information in the integrated resource e.g. the text, which becomes redundant. But the text cannot be ignored because it is already integrated with diagrams. Where split attention effect was being avoided, **redundancy effect** emerges. The learners spend extra cognitive resources reconciling the redundant information with what they know (Kalyuga, 2009).

Worked example effect: Novice learners do well learning with worked examples because the examples focus the learners' attention on problem states and useful solution steps. When the learners become experts, they prefer conventional problems, with no guidance or suggested solution steps. If they are still forced to use worked examples, they have to keep reconciling the solution steps provided with what they already know or how they would prefer to solve the problem.

All these extraneous cognitive load effects may reduce the learning outcomes of advanced learners. Hence, it is called expertise reversal effect.

Adaptive e-learning

The way to approach the problem of expertise reversal is to use adaptive e-learning, altering the format of presentation of learning materials, the level of difficulty of materials or even amount of problem solving support provided, according to the learner's level of knowledge or expertise.

2.5.5 Influence of Cognitive Load Theory on Instructional Design

Intrinsic Cognitive Load

Simple to complex sequencing of instructional information can be used to reduce intrinsic cognitive load (Van Marrienoer, Kirschner, & Kester, 2003; Van Marrienoer & Sweller, 2005). Novice learners without fundamental schemata are provided with the simplest learning material whose focus is the fundamental elements of information. Even if the presented material is simplified, it represents the "whole" learning task, and it is not a sequence of discrete part-tasks. Simplifying factors such as the type and number of examples to be included, number and kind of exercises and level of difficulty of the material are considered.

As the learner's level of expertise increases, the complexity of the material should increase. Information presented in each subsequent level should be in addition to or an elaboration of the previous level's information, to enable the learners complete things they could not before (Van Marrienoer et al., 2002). Full understanding of the material can only be reached after the learning material has been presented in its full complexity i.e. all elements and their interactions typical in the domain are presented.

During problem solving, learners can also be required to plan their own problem solving and are given hints to remind them to follow their plan from phase to phase, possibly handling one phase at a time. The plan acts as a performance constraint (Van Marrienoer et al., 2002; Van Marrienoer & Sweller, 2005), requiring the learner to

complete some phase or phases, before proceeding to others. The learners are required to identify all their problem solving phases at the planning stage (but they are allowed to update them later as their understanding improves). The problem solving tasks within each phase should also be ordered during planning. Resources such as time, cognitive knowledge and skills required to complete each phase should also be identified.

Germane Cognitive Load

There is need to include approaches in the instructional context, which are informed by cognitive load theory, to enhance awareness and application of the elements of metacognition (Valcke, 2002). There is also need for tools which prompt the students to apply the problem solving processes. The students need opportunities, created by use of prompts or hints, during which they are compelled to engage in explicit reflection on their learning and problem solving strategies. Such prompts or hints puzzle the students, forcing them to find a new way or level of doing things, in terms of learning and problem solving.

The students should be offered metacognitive tools to aid them take control of their problem solving, engage in self-explanation and make their steps explicit. When learners have control over their learning and problem solving processes, they are able to carry out their monitoring activities better. Control places more responsibility on them, which increases germane cognitive load. The learners are forced to engage in reflection and justification of steps taken. This results in better schema construction, problem solving and transfer of learning (Van Marrienoer & Sweller, 2005).

Germane cognitive load can be accommodated through a number of ways. Firstly, provide a description of the problem solving process to the student before commencement of problem solving. Secondly, remind the student of the problem solving process steps and current progress. The students should be shown visually how they have progressed on various sub-tasks and the resources spent so far, together with the remaining sub-tasks, to stimulate reflection. This acts as feedback to facilitate self-

regulation and learner control of the problem solving process. Thirdly, provide chances to record intermediate results of one's problem solving, externalize one's problem solving process and reflect on both application of the problem solving process and progress in ill-structured problem solving (Valcke, 2002).

The time when the supports are presented to the student should also be managed well. Some of the supports should be provided earlier to enable the students learn how they work, before others are presented. This reduces undesirable cognitive load associated with the way of presenting tools for fostering germane cognitive load.

Extraneous Cognitive Load and Expertise Reversal Effect:

Adaptive learning support should be provided to reduce expertise reversal effect (Kalyuga, 2009; Van Marrienboer & Sweller, 2005). This can be realized through the way the learning material is presented, suitable hints to guide navigation through the course as well as the features to enable a student ignore parts of learning material which have become redundant due to increase in the learner's expertise. For example, learning materials can be presented using the simple to complex sequencing, full-example to partial example to conventional problem sequencing of material (Van Marrienboer et al., 2002). Top-down presentation of material can also be used, with general, high level information presented first, and the student drilling down for more detailed information (Hoffman, 1997). The system proposes an appropriate level of material presentation according to the student's attributes.

Expertise reversal effect can also be reduced by giving students control over what to learn, which guidance hints to follow or ignore, and which presentations of information to select for studying (Benyon, 1993a; Langley, 1997). The students can be afforded control by leaving all the learning material links clickable, whichever way they have been annotated. In a system where some sections are adaptively hidden or disabled, the students can also be provided an alternative, longer access route for accessing the hidden or disabled parts (Papanikolaou et al., 2003).

2.6 Tools for Supporting Metacognition During Ill-Structured Problem Solving in Online Learning

2.6.1 Introduction

Designers of online learning have to provide scaffolds for metacognition for the solving of ill-structured problems (McLoughlin & Hollingworth, 2001). There is enough evidence that there are technological techniques and tools to support such scaffolding needs (Azevedo, Cromley, & Seibert, 2004; Azevedo & Hadwin, 2005; Oliver & Herrington, 2001). According to Lin, Hmelo, Kinzer and Secules (1999), some of the factors to consider during the design of online learning for ill-structured problem solving with support for metacognition include:

- (a) Prompting learners to investigate their own thinking while solving problems
- (b) Providing visual displays of the processes students have utilized as they solve problems and progress on the various tasks, shown in graph form
- (c) Developing a strong sense of self as student and problem solver, by enabling the student to set goals of problem solving, associated to the sub-tasks he is to carry out in order to solve the problem.

Two tools for scaffolding metacognition during ill-structured problem solving, which accommodate these factors, namely question prompts and task schedule are discussed next.

2.6.2 Use of Task Schedule

Introduction

The task schedule is used as a personal task management scaffold for cognitive learning. It contributes to the achievement of self-organized learning when using problem solving approaches such as project management (Chng, 2001). The task schedule functions as a knowledge elicitation tool since it gives personal voice to prior knowledge, building

experiential links between past and present learning, therefore increasing meaning to a greater depth of personal relevance (Coombs, 2001).

The task schedule helps the student to personally identify with the abstract concepts and strategies to be used during problem solving since each student designs his learning plan personally. Therefore, the students are able to model personal knowledge from the experiential event. They are able to exercise control over the problem solving processes for the task. As a result, the students become more aware of the skills necessary for independent learning and problem solving. They also develop a positive attitude towards critical thinking - purposeful, reflective decision on what to believe or do according to one's observations and experiences.

The ill-structured problem to be solved guides the entire problem solving process. The designed task schedule acts as a personal learning contract that the student has with himself. The student's problem solving and the resulting learning is to be appraised against the task schedule (Jonassen, 1999). Therefore, with the task schedule, student centered scaffolding is achieved.

The students create their own task schedules by doing a summary of the sub-tasks of the problem and sequence the sub-tasks. They also identify resources needed for the sub-tasks such as knowledge and skills to be applied, and why and when to use them. They define the time to be spent on each sub-task in hours.

For adults, it is better if the students can author their own learning plans in response to a problem-solving project (Chng, 2001; Oliver & Herrington, 2001). Therefore, using a task schedule to scaffold learners is suitable for non-collaborative web-based learning through problem solving, where the students are on their own and have no guidance or feedback from peers or the instructor.

Related Tools

The schedule is similar to **Strategy Evaluation Matrix (SEM)** as described by Schraw (1998). In the SEM, the students were required to list strategies they intended to use for problem solving, how each strategy was to be used, when it was to be used and why it was to be used.

Oliver and Herrington (2001) proposed the introduction of a **learning schedule** that learners can use to pace themselves as they proceed with problem solving. The learning schedule was to provide them with a possible weekly timetable and task list to assist them plan and undertake learning tasks to enable them overcome hurdles that might otherwise hold them back and prevent their progress.

The task schedule is also similar to the **learning plan** used by Chng and Coombs (2001) to promote critical thinking in students as they solved ill-structured problems. The ill-structured problems required the students to analyze and summarize news articles using economic concepts and principles in a project based learning framework.

Comparison with related tools

(i) Strategy Evaluation Matrix

SEMs significantly improve learning by improving strategy use. They promote explicit metacognitive awareness since the students are asked to fill the matrices individually for their problem solving tasks and update these strategies as their understanding improves. SEMs encourage students to actively participate in knowledge construction. They also force the students to reflect on their progress as they continue updating the matrices throughout the problem solving process. The shortcoming of SEMs is that they concentrate on knowledge of cognition, such as declarative, procedural and conditional knowledge, but leave out knowledge of regulation of cognition, which is the other part of metacognition. Knowledge of regulation of cognition is also critical for successful solving of problems, especially ill-structured ones.

(ii) Learning Schedule

The learning schedule has a short scope, usually covering the activities for a week yet solving an ill-structured problem can take longer. The learning schedule also concentrates on declarative knowledge which is one of the elements of knowledge of cognition. It pays attention to learning activities that should be undertaken, information to be collected, people to be contacted, list of tasks to be planned, and list of tasks to be finalized, which are all declarative knowledge. The learning schedule does not include knowledge of regulation which is essential for ill-structured problem solving.

(iii) Learning Plan

Learning plans (Chng, 2001), cover all the stages of problem solving. A learning plan has the following sections:

(a) **Objectives** of solving the problem

(b) **Activities to activate prior learning** i.e. knowledge and skills already acquired in a subject, which are to be applied during ill-structured problem solving. They establish a link between the task and relevant prior knowledge

(c) **The process of problem solving.** This involves identification of authentic task based activities in terms of purpose of the task, strategy i.e. designed learning tasks linking both prior knowledge and new knowledge, outcome as a result of assessing individual task/activity progress and review of the problem solving procedures in terms of reflection on the activity/task outcomes and making improvements where necessary

(d) **A description of how final evaluation will take place.** This is through formal assessment tasks such as examining the log of student activities, oral presentation of the product, writing reports, and so on.

Learning plans also support all the aspects of knowledge of cognition and regulation of cognition (Schraw, 1998). This is explained below:

- The purpose of a problem solving task corresponds to the goal of learning

- The knowledge and skills to be learned to aid problem solving correspond to declarative knowledge
- The why and how of problem solving/learning objectives correspond to procedural and conditional knowledge
- The strategy, which incorporates prior and new knowledge in the design of the learning tasks, together with learning outcomes, corresponds to planning.
- The linking of prior and new knowledge and identification of additional information that will be needed and how it will be obtained in order to aid in completing the task corresponds to information management
- The assessing of task progress, including specified minimum standards for the outcomes and evidences to show the standards have been met corresponds to monitoring as it gives feedback on progress in the problem solving tasks
- Review corresponds to debugging since the learner reflects on the outcome and plans how to improve where necessary.
- The final review corresponds to evaluation of the learning that has taken place as well as the end product of the problem solving process.

2.6.3 Question Prompts

Introduction

Question prompts refer to sets of static or adaptive questions or hints, which are designed to puzzle students at different stages of learning in order for them to reflect on the status of their learning and the activities they are engaging in (Oliver & Herrington, 2001). Question prompts are valuable for domain knowledge acquisition, the scaffolding of complex, ill-structured problem solving tasks (metacognitive functions) as well as guiding students through all the stages of problem solving (Ge, 2001). They make the students to examine what they have achieved vis-à-vis their learning goals and available time and other resources set aside for completing their work.

Question prompts make it necessary for the student to search for the knowledge required for problem solving from within the self. If it is not found, then the student searches for it from external sources. This makes it possible for prior knowledge to be activated and linked to new information, leading to better knowledge construction.

Question prompts also make the student to reflect on his problem solving successes so far and what should be done in order to realize the goals set at the beginning of problem solving.

The students are expected to answer questions prompts when faced with them. An example of a question prompt is, *"How are you feeling right now?"* Question prompts stir up the student into a conversation with oneself and in the process, the student examines and re-examines one's knowledge, skills and problem solving process status. This makes the process of problem solving to have more personal relevance (Coombs, 2001), therefore making student-centered scaffolding possible.

Question prompts, therefore, work by causing within student conflict or puzzlement, which the student responds to by reflecting on the current status of things so as to achieve a better understanding. Then the student does what he has to do in order to achieve another point of equilibrium that will make him more comfortable when answering the question. According to Piaget's Cognitive Development theory (Piaget, 1985), questions make cognitive demands on the individual, which create cognitive discrepancies and provide motivation for resolving them (the questions). A question generates tension while creating discrepancy, which in turn causes disequilibrium, then the student strives to solve these discrepancies through mental activity. Hence, questions challenge one to change a cognitive structure so as to make sense of the environment, and think about alternative solutions and consider various perspectives.

Types of Question Prompts

Questions may be classified as generic question stems and generic questions. Examples of generic question stems include “How is this related to ...”, “What is the main idea of ...” and “What are the strengths and weaknesses of ...”. Examples of generic questions include “How does this passage or chapter relate to what I already know about the topic?” and “What is the main idea of this chapter or passage?”

Related tools

Question prompts were used and popularized by King (1991). King used a **prompt card** having both full question and question stubs. Schraw (1998) used a similar scaffold called a **regulatory card** (RC), with three main sections i.e. planning, monitoring and evaluating as shown in Figure 2-1. The RC was suitable for scaffolding regulation of cognition (Schraw, 1998). The RC had explicit prompts in the form of checklists to help students to be more strategic and systematic. Students who were prompted to reflect on their goals outperformed the other students on a subsequent transfer problem that did not include any prompting (Lin, Schwartz, & Hatano, 2005). Schoenfeld (1985) found that asking and answering metacognitive questions helped students to focus on the process of problem solving and consequently improved their performance.

A Regulatory Checklist

Planning

- (a) *What is the nature of the task?*
- (b) *What is my goal?*
- (c) *What kind of information and strategies do I need?*
- (d) *How much time and resources will I need?*

Monitoring

1. *Do I have a clear understanding of what I am doing?*
2. *Does the task make sense?*

3. *Am I reaching my goals?*
4. *Do I need to make changes?*

Evaluating

1. *Have I reached my goal?*
2. *What worked?*
3. *What didn't work?*
4. *Would I do things differently next time?*

Figure 2-1: Example checklist

Summary of the Role of Question Prompts

As a summary, question prompts encourage the following important functions during problem solving: focusing attention, stimulating prior knowledge, enhancing comprehension, monitoring thinking and learning processes, and facilitating problem-solving processes. They also elicit thoughtful responses such as explanations and inferences (King & Rosenshine, 1993).

2.7 Machine Learning and Adaptive Cognitive Learning Support

2.7.1 Introduction

In learner-centred web-based learning environments, domain content can be customized to meet the individual student characteristics such as goals, needs, and abilities, among others. This is adaptive learning support for cognitive learning to facilitate learning of knowledge of the domain of the problem (Dabbagh & Kitsantas, 2004). The user interface adapts to the individual student (Langley, 1999). It is expected to respond effectively to individual student preferences, cognitive levels including knowledge level and errors committed during learning, and the student's prior experience, among others. At the same time, adaptation should minimize the frustration experienced by the user, the time and learning effort the user has to put in and any knowledge transfer errors. Generally, adaptation should prevent users from

having an unrewarding experience due to being misunderstood and difficulties arising from the inflexibility of systems (Benyon, 1993a).

Adaptive user interfaces achieve adaptive learning support by building pragmatic abstractions of relevant student characteristics (**learner model**) and domain/subject knowledge (**domain model**). **Machine Learning algorithms** are used to make **inferences** about values of the learner's attributes. These inferred values are used to update the learner model. The way interaction will take place between the learner model and the domain model so as to achieve adaptation (**the interaction model**) is also designed (Benyon, 1993b). The interaction model consists of rules used to adapt some aspect of the interface to suit individual learner attribute values. The system designer introduces multiple representations or abstractions to match the types of learners expected to use the system according to attribute values in the learner model (Benyon, 1993b; Langley, 1999).

2.7.2 Suitability of Adaptive User Interfaces for Web-based Learning

Users of web-based learning systems are usually dispersed in terms of geographical location, time and place of studying and therefore, their requirements are not the same. There are also individual differences among learners, such as the learner's background, entry knowledge level, current domain knowledge levels, and other cognitive traits, among others.

Due to the need to satisfy varying user requirements, web-based learning environments are well-suited for adaptive approaches to solve usability problems. Personalization is needed so that the learner can connect with the learning system in a way that he feels his distinct learning needs have been supported to a good extent. This is to enable comfort, satisfaction, effectiveness and efficiency in learning.

2.7.3 Techniques for Adaptive Learning Support for Hypermedia

Over the recent past, the field of adaptive hypermedia has undergone a lot of development in terms of technologies which can be used. These technologies have

matured over time and they can therefore adequately serve as the technological basis for the adaptive web (Brusilovisky, Karagiannidis, & Sampson, 2004).

Some of the technologies adapt the way information intended for the user is presented. They are called **adaptive presentation technologies**. Such technologies change aspects of the presented information such as the structure, the level of details provided, the amount of information provided or the knowledge modules combined to form a lesson page. They include **adaptive text presentation** and **adaptive multimedia presentation** (Brusilovsky, 1999). Adaptive text presentation involves presenting different combinations of text-based learning objects, based on some conditions from the learner, domain and interaction models.

Other technologies are used to influence the way the user can navigate through information on the website. They make it easier for the user to determine a suitable navigation path to follow through the course and are called **adaptive navigation technologies**. They include **link annotation**, **link disabling** and **link hiding**.

There are also other technologies that can provide adaptive support through the sequencing of the curriculum units in case of a learning system. This is to give a learner a pre-determined sequence of units to study, which are deemed suited to the learner's attributes such as level of knowledge. These technologies are called **adaptive curriculum sequencing technologies** (Brusilovisky, 1996, 1999, 2004).

The three are some of the commonly used techniques to provide adaptive support in hypermedia-based learning systems.

Adaptive technologies are used with page design to make sure that the amount of content displayed per web page is manageable for the learner. Adaptive technologies are also used with instructional design to reduce to **expertise reversal effect**, which is as a result of providing the same type of support to learners who are at different levels of domain knowledge (Kalyuga, 2009). They can also be used with instructional design to

ensure that the information is sequenced in such a way that it is suited to the learner's level of knowledge. This can be through simple to complex sequencing or through complete example to partial example and to conventional problem sequencing. To form a lesson page, various knowledge modules are combined and sequenced in different ways to prevent the learner from suffering too much extraneous cognitive load and intrinsic cognitive load (Papanikolaou et al., 2003).

2.7.4 Machine Learning Techniques

(a) Introduction

Machine Learning is an area of Artificial Intelligence which is concerned with the development of techniques, mostly called algorithms, which can be used to enable a system to "learn" from what it "experiences". The experience is represented in the form of some data sets. The system then uses this learned knowledge to perform better in solving similar problems, as it acquires more experience i.e. as it is exposed to more data sets. Some measure of performance is used to determine this improvement (Mitchell, 1997). The computer system uses Machine Learning algorithms to acquire this knowledge from the data sets. This type of Machine Learning that relies on available data sets is called **inductive** or **empirical learning**. The hypotheses generated about concepts underlying the data are generalized not only over the data but also to unseen instances drawn from the same domain.

(b) Supervised and Unsupervised Inductive Learning

Supervised learning is inductive learning that develops hypotheses from labeled data. The labels mostly represent classification for the instance of data. This type of learning is also called learning from examples or concept acquisition. On the other hand, **unsupervised learning** is inductive learning that develops hypotheses about concepts underlying data when the data is not labeled.

(c) Supervised Inductive Learning Techniques

Let us assume that a training example in supervised inductive learning is represented as $(x, f(x))$ where x is the instance itself and $f(x)$ is the target attribute value i.e. the label.

Then, the learning problem can be described as computing a function f' which approximates f . f in turn is the function that approximates the target concept.

f' can be represented in different ways. The way it is represented is related to the different inductive learning paradigms existing as well as the specific learning technique.

Table 2-2 provides details of common inductive Machine Learning techniques, including the paradigm, specific technique, representation schemes for learned concepts when using the technique and examples of algorithms using the technique.

Table 2-2 : Classification of supervised inductive machine learning techniques

Paradigm	Technique	Representation Schemes	Example algorithms
Symbolic Learning	Rule/Tree learning	Rules, Decision Trees	C4.5, ID3
	Probabilistic learning	Prior probability and conditional probabilities of the attributes given the	Naive Bayes Classifier, Bayesian Belief Networks
	Instance based learning and case based learning	Class is represented simply as a set of instances or episodes	K-Nearest Neighbour, Case Based Reasoning
Neural Learning	Artificial Neural Networks	Networks composed of nodes called units, connected by weighted, directed links	Feed forward with backward propagation, Adaptive Resonance Theory, Kohonen networks
Genetic or evolutionary learning	Genetic algorithms	Genetic patterns, with each pattern denoting an individual represented as a	

NOTE: Bayesian Belief Networks or simply Bayesian networks are defined in terms of nodes and arcs which link the nodes. Each node represents some random variable and is conditionally independent of its ancestors given its parents. When all nodes are known and all the variables are observable, then learning for these networks is the estimation of conditional probabilities of the links of the network. Otherwise, the learning process is similar to that of the feed forward Artificial Neural Networks (Sison & Shimura, 1998).

(d) Unsupervised Inductive Learning Techniques

It is more difficult to learn from unlabelled data than from labeled data. Hence, supervised learning techniques are not understood as much as techniques for supervised learning. With unsupervised learning, it is so clear how to define the goals of learning or even the criteria for evaluation.

The main technique of inductive unsupervised learning is conceptual clustering algorithms. Clustering applies to Symbolic (e.g. K-Means clustering algorithm), neural learning (e.g. Kohonen networks or self-organizing maps) and even evolutionary learning paradigms.

(e) Machine Learning Classification and Learner Modeling

Most of the problems which are solved by Machine Learning algorithms are **classification problems**. The learned target concept is used to predict whether a given instance is of one type or another, from a set of possible types, categories or classes. Such an approach can be used for learner modeling since the algorithm can infer the learner's category from the provided data. This category is added to the learner model, such as current level of knowledge or preference to use additional learning material.

(f) Challenges of Using Machine Learning Algorithms for Adaptive User Interfaces

In order to use these algorithms, some of the **challenges of using inductive Machine Learning algorithms for adaptive user interfaces** need to be addressed. These challenges arise because adaptive user interfaces, unlike other applications of Machine

Learning, are intended to aid humans while using an interface to perform some task (Benyon, 1993b; Langley, 1999). These challenges include:

- The need to select algorithms which **can learn quickly** because most user modeling tasks need to be done in real time, as the user is taking part in an activity (Langley, 1999; Stern, Beck, & Woolf, 1999)
- The need to select algorithms which **do not need a lot of data** to learn but still achieve high levels of accuracy since the user model needs to be used as soon as the user begins interacting with the system (Stern et al., 1999). For most Machine Learning techniques, there is need for large data sets in order to build accurate models. For such techniques, small data sets lead to overfitting. The effect of small data sets is reduced through initialization of a proper model of the learner (Burak & Mustafa, 2007). It is also reduced by selecting algorithms which do not need a lot of data before beginning to function properly
- The need for the **problem being solved**, e.g. classification of the learner in terms of some attributes, **to be represented** in such a way that it can be **accessed directly** by the inductive Machine Learning algorithms. Define the task (T), the measure of performance (P) and the experience (E) i.e. training examples for the algorithm (Langley, 1999; Mitchell, 1997)
- The need to identify a suitable way of **representing the training examples** so that they can be easily used by the inductive Machine Learning algorithms. This is either as a bag of words as is mostly the case for text-based documents or as a scheme of attribute value pairs or some other way
- The need to define a **way of collecting user data** so that interaction between the user and the interface is not obstructed in any way (Benyon, 1993b; Langley, 1999).

Other challenges include:

- **Overfitting:** Overfitting leads to models which perform well on training data but poorly on classification data. This is as a result of developing a model that fits the training data closely due to overtraining or inadequate data. Overtraining is a result of lack of clear heuristics for determining when to stop training. There is need for automated detection of overfitting when building models
- **Erosion of model predictive power:** The predictive power of a model reduces due to differences between the training data and classification data. The differences in data could be due to passage of time, economic factors or even the dynamic nature of the attributes of the people who the data is about. The model needs to adjust fast or be reviewed every so often to make sure it reflects the current status. The problem may also be called concept drift due to changes in the users' interests and profile information (Sison & Shimura, 1998)
- **Selection of attributes for models:** Which features would represent the situation being modeled well? Mostly, the attributes are selected manually. How can it be automated? There are a lot of details about a learner which can be made available when using a web-based system for learning and therefore, modeling learners is a complex process
- **Model validation:** This is also a challenge because there are no standardized ways of validating models. It is unclear whether the focus should be on improvement in classification accuracy, computational complexity or efficiency of classification. Should it be based on data obtained directly from the users of the model or on data obtained indirectly, such as the user's log record?

A description of how adaptive user interfaces and Machine Learning algorithms were used to achieve adaptation in this research is presented in Chapter 4: Methodology - Machine Learning and Adaptive Learning Support Design.

2.8 Theories of Learning and Instructional Design

2.8.1 Introduction

Instruction needs to be designed deliberately to allow successful learning through maximizing the appeal, effectiveness and efficiency of learning experiences offered by the instructional system. The learner's current status and learning needs are established and the goal of instruction is defined. Then, instructional interventions are designed to enable the learner to progress from current status towards the goal state and, in the process, meet the learner's learning needs.

Effective design of an instructional system is guided by learning theories. These theories attempt to provide knowledge of the way people learn and this knowledge is used to guide the design and development of learning (Nam & Smith-Jackson, 2007). The kind of learning theory considered, such as behaviorism, cognitivism or constructivism, determines the specific instructional design model to be used. As a result, learning theories help in defining and shaping the outcome to be expected of instructional materials and learning experiences for learners during acquisition of domain content and participation in problem solving.

2.8.2 Learning Theories

(a) Behaviorism

Behaviorism considers learning as a change in behavior due to selective reinforcement of learner actions. It does not pay attention to the contribution of and changes to mental processes of the learner as a result of learning (De Villiers, 2002).

(b) Cognitivism

Cognitivism considers learning more as an active process of learning rather than passive receiving of knowledge. The instructor activates learning and learners learn depending on their individuality factors such as expectations, mental processing and aptitude (De Villiers, 2002).

(c) Constructivism

Constructivism is of the view that knowledge is not a pre-existing entity to be passed to and to be assimilated by the learner but must rather be actively “constructed” by learners (Savery and Duffy, 1995). Knowledge construction is based on the learners’ prior knowledge and views of the real-life contexts in which they learn.

2.8.3 Instructional Design Models

The three theories of learning lead to three instructional design models. These are the **objectivist**, **constructivist** and **mixed instructional design models** (Nam & Smith-Jackson, 2007; Hadjerrouit, 2006). These design models are discussed briefly below:

(a) Objectivist Learning Design Model

The objectivist instructional design model assumes that there exists objective knowledge of the domain being learned. According to Vrasidas (2000), there is an objective world, a shared reality which shapes people’s experiences and also defines the constraints for interpretation of experience and the meanings attached. Every learner needs to learn the same knowledge.

This model is used to guide the construction of cognitive schemata of domain knowledge within the learner’s memory (Van Marrienboer et al., 2002; Van Marrienboer & Sweller, 2005).

The objectivist learning design model is guided by behaviorism and cognitivism.

(b) Constructivist Learning Design Model

The constructivist learning design model assumes that knowledge is constructed by the learners as individuals or in groups, using information obtained from their context of learning, together with their prior knowledge and experience. This process of learning by knowledge construction requires metacognitive skills such as reflection, self-regulation and self-evaluation because the instructor does not transfer knowledge to the learners, but only guides and facilitates them.

The learners are required to solve problems which are authentic, and which provide them with a challenge that is meaningful and as close as possible to the real world of practice which they (learners) are being prepared for (Savery and Duffy, 1995). Ill-structured problems satisfy this requirement (Jonassen, 1999). An example of an ill-structured problem is programming problems (Boyle, 2003; Wiedenbeck, 2005).

It also important to entrench the assessment of learning into the learning process and this must focus on individual learners' orientation (Hadjerrouit, 2006).

(c) Mixed Learning Design Model

The mixed or eclectic learning design model is a mix of the objectivist and constructivist learning design approaches, as shown in Figure 2-2.

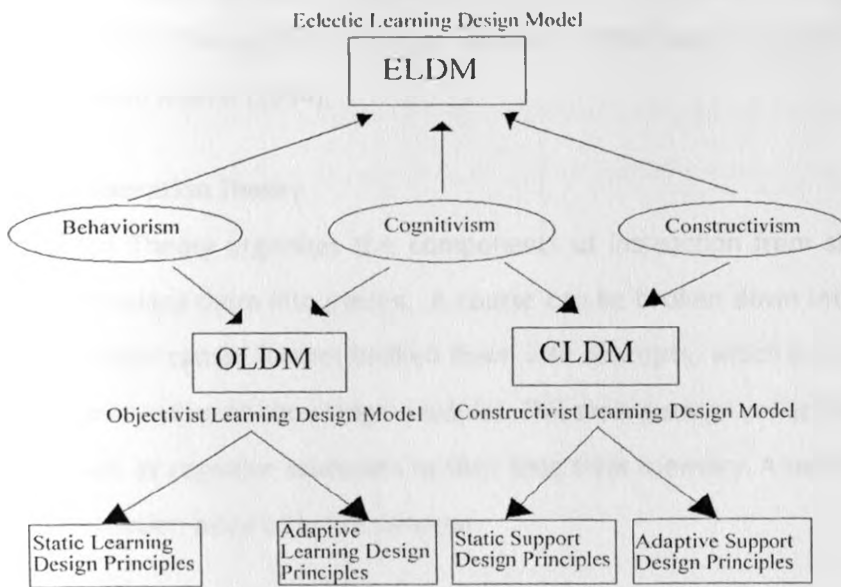


Figure 2-2 : Learning Theories and Instructional Design Principles in the Eclectic Design Model

As shown in the figure, the eclectic learning design model borrows from behaviorism, cognitivism and constructivism and constitutes both static and adaptive design principles.

The mixed model is guided by the viewpoint that novice learners need to learn the largely objective domain knowledge first, facilitated by the stability and certainty of knowledge acquisition and learning outcomes expected of the learners (Nam & Smith-Jackson, 2007). Advanced learners need to learn more advanced cognitive and metacognitive skills such as problem solving, analytical thinking and self-regulation of their learning, and these are better guided by constructivist learning design principles.

2.8.4 Instructional Design Theories

There are two theories of instructional design which are particularly appropriate for guiding the design of course components for online learning (De Villiers, 2002). These are Elaboration Theory (ET) of Charles Raigeluth (1998) and Component Display Theory (CDT) of David Merrill (1994).

(a) Elaboration Theory

Elaboration Theory organizes the components of instruction from simple to complex without breaking them into pieces. A course can be broken down into a small number of goals. Goals can be further broken down into concepts, which in turn can be broken down into constituent knowledge modules. This arrangement reflects how people store information as cognitive schemata in their long term memory. A new concept is stored under a broader, more inclusive concept.

A goal within a subject area is an epitome or organizing concept. This is the first concept to teach, and then what follows is the elaboration of the epitome, in this case the goal. Each component of the elaboration also has its own epitome (Hoffman, 1997). The elaboration of an epitome could include **concepts**, which answer the question "What?", **procedures**, which answer the question "How?", and **theories**, which answer the

question "Why?". Further elaboration of these could include **definition, examples** and **practice** such as exercises, assignments and projects, among other forms of practice.

The kind of structuring as suggested by Elaboration Theory is **well suited to hypermedia**, since with hypermedia, the general ideas can be displayed first and if there is need, a link can be followed to see more details. This can be done recursively. Hence, Elaboration Theory is suited to online learning environments (Hoffman, 1997).

(b) Component Display Theory

Component Display Theory (CDT) is used at the **level of learning objects** and is based on the assumption that there are **different kinds of learning objectives**, which need **different kinds of conditions to be achieved** (Gagné, 1985). Each objective is related to the required content and to the desired performance outcome.

There are 3 ways in which the learner can perform in the process of learning: (i) learning the facts and concepts of the knowledge domain (**remember level**), (ii) applying these facts and concepts in a procedure to carry out a task or part of a task (**use level**) or (iii) determining new principles or relations or generalizations from the learning experience (**find level**).

- **Remember and Use Levels of Performance**

Each level of performance is a different combination of **primary presentation forms**, which provide the basic building blocks for instruction (Merrill, 1994). Merrill proposed 4 different primary presentation forms, which can be classified along **two dimensions**: (a) content mode and presentation mode. In terms of content mode, a primary presentation form can be a *generality* which can be a statement, definition, principle or steps in a procedure, or an *instance* which can be a *specific* object, illustration of an object, a symbol, event, process or a procedure. In terms of presentation mode, a primary presentation form can be an *expository*, which means to present or to show, or an *inquisitory*, which means to question or to require practice.

From these 2 dimensions, Merrill suggested the **four primary presentation forms** as:

- *Expository Generality*, which is applied at remember level, e.g. theory presentation
- *Inquisitory Generality*, which is applied at remember level, e.g. an opening question followed by theory presentation
- *Expository Instance*, which is applied at use level e.g. application example for the learner to follow
- *Inquisitory Instance*, which is applied at use level, e.g. exercise for the learner to carry out.

- **Find Level of Performance**

The find level of performance involves tasks requiring **higher order performance** from the learners, beyond application level according to Bloom's taxonomy. It is an exception to the matching of primary presentation forms according to CDT.

Therefore, CDT operates at the level of domain content or knowledge modules and it guides the **matching of content to the expected performance outcomes**. Hence, CDT guides how to support specific instructional outcomes with the right learning material.

2.9 Summary

2.9.1 Summary of Relevant Concepts and Their Relationships

Cognitive load theory has evolved over its short life span to become an instructional design model. Literature has shown that in a bid to ensure the best use of Working Memory (WM), cognitive load theory is used to guide the design and sequencing of the domain content. This is with a view to reducing the mental effort the learner spends on the learning process, due to intrinsic and extraneous cognitive load, and at the same time make sure schema construction is achieved (Van Marrienoer et al., 2002; Van Marrienoer & Sweller, 2005). There is also need to provide for activation of prior knowledge so as to avoid learners solving problems by random means, which generates a lot of extraneous and intrinsic cognitive load.

To complement instructional design, adaptive features can also be added to a system, especially online learning systems which are highly interactive and are used by diverse users. This is partly to reduce the expertise reversal effect which arises due to subjecting expert learners to instructional approaches suited for novice learners (Kalyuga, 2009). Adaptive features in general aid the system in providing learning support that is appropriate to an individual learner's attributes, thus reducing undesirable cognitive load effects. For web-based learning, adaptive hypermedia techniques such as adaptive presentation and adaptive navigation design are handy (Brusilovsky, 1999). As part of reducing undesirable cognitive load, learners also need to have control over their learning so that they can choose to ignore the system's suggestions if they are not appropriate. This way undesirable cognitive load effects such as redundancy and split attention effects can be avoided.

In order to encourage constructivist learning, especially in training programs involving authentic, complex, ill-structured tasks, learners need to be motivated to utilize any cognitive resources which become available due to reduction in extraneous and intrinsic cognitive load. Encouraging learners to apply metacognitive monitoring of their learning processes through increasing metacognitive awareness is important. The learners need to be aided to engage in planning, management of task related information for the problem solving process, monitoring, debugging and evaluation, reflection, self-explanation and external representation of their internal knowledge (Valcke, 2002). This can be done through prompting learners while they are engaged in ill-structured problem solving, providing them with space where they can express themselves and allowing them to have control over their learning and problem solving processes, among others.

In the instructional, adaptive user interface and constructive learning sub-systems of a web-based learning system for ill-structured problem solving, feedback is very vital. Various scaffolds included in the system provide feedback to the student, according to the student's actions. For cognitive learning, feedback could be provided directly in the

form of integrated assessment test results at the end of every task or concept. Feedback for cognitive learning can also be provided indirectly in the form of adaptive system suggestions or adjustments to the adaptive user interface as a result of a test taken or some other user actions. Besides feedback on performance in a test, the student needs to be informed of questions not answered correctly and the likely sources of error, such as concepts not well understood. This is corrective feedback.

Feedback could also be in the form of quality of performance, especially for non-recurrent aspects in constructivist learning. This is to stimulate learner reflection and self-explanation. This is cognitive feedback. The feedback could be in the form of information on tasks running late, tasks completed, and tasks not started beyond when they should have commenced, among others. The interface for the constructivist learning sub-system can also change adaptively to pass a message to the student and prompt action.

2.9.2 Conceptual Framework

Following the literature review, a conceptual framework has been adopted to guide the rest of the study. Both cognitive and metacognitive knowledge have been considered. Tasks related to the construction of schemata during the acquisition of domain knowledge, as well as tasks related to the solving of rich, ill-structured problems have also been considered. An instructional design model has been described to accommodate both objectivist and constructivist learning with both static and adaptive interfaces. Learning support has also been considered in terms of three sub-systems namely instructional system, adaptive interface system and constructivist learning system with self-regulation for ill-structured problem solving. Supports have been included for the activation of prior knowledge during the learning of domain knowledge and problem solving. Adaptive learning techniques have been included, with the focus being personalization according to the learner's characteristics, such as level of knowledge. The overriding theme is the use of different types of supports to reduce cognitive load on the learner. This is to reduce the cognitive effort spent during

cognitive learning and ill-structured problem solving. This is expected to lead to deep learning, which results in better transfer of knowledge to both similar or dissimilar real life contexts.

A number of constructs have been adopted for the study, to guide the design of the learning environment and the research methodology. The constructs are based on the foregoing literature review. These constructs are guided by types of knowledge, related Cognitive Load Theory components and the problem solving processes for ill-structured problems (project based learning design has been adopted). The constructs are also guided by the types of supports adopted to reduce cognitive load and encourage the acquisition of domain knowledge and application of different types of knowledge during problem solving such as cognition and metacognition. The constructs and their relationships are shown in the conceptual framework in Figure 2-3:

Conceptual Framework

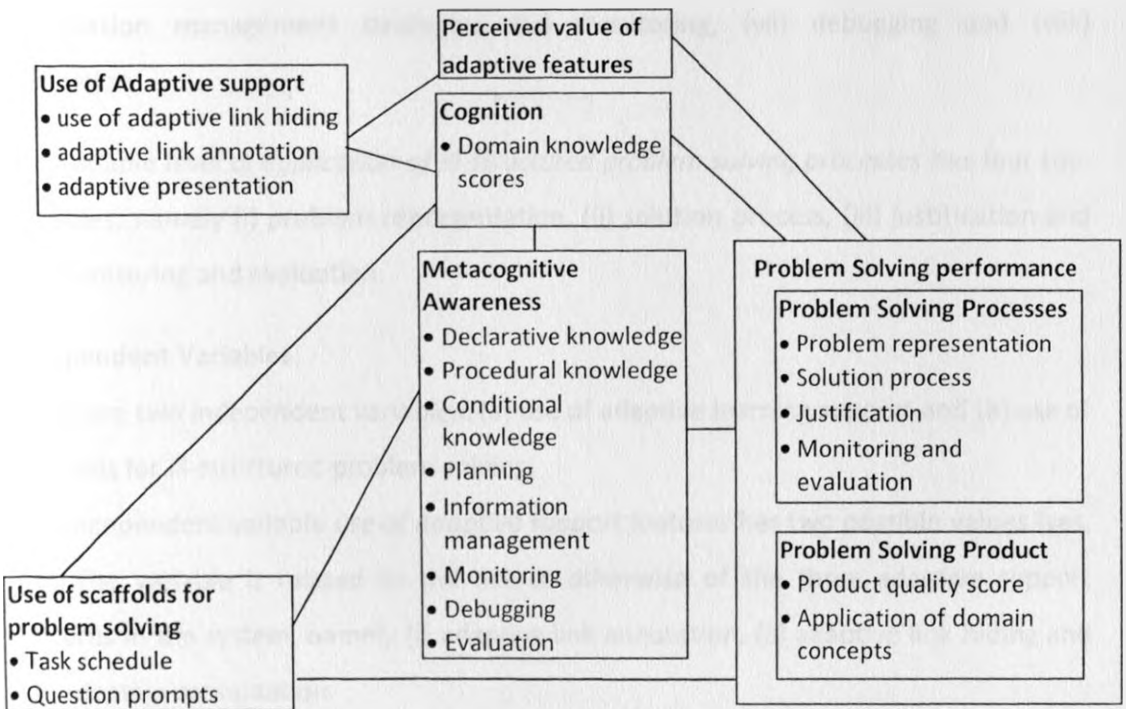


Figure 2-3 : Conceptual Framework: Variables for the research and their relationships

The Variables

Dependent Variables

As shown in the conceptual framework in Figure 2-3, there are five dependent variables: (a) cognitive learning score, (b) perceived value of the adaptive features of the system, (c) level of metacognitive awareness, (d) level of application of ill-structured problem solving processes and (e) the evaluation score of the product of problem solving.

Level of metacognitive awareness is expected to have a positive relationship with the *level of application of problem solving processes*. The positive effect of *metacognitive awareness* on the *level of application of problem solving processes* is expected to improve with *utilization of the scaffolds for self-regulation* during ill-structured problem solving.

The variable *level of metacognitive awareness* has 8 sub-variables, namely (i) declarative knowledge, (ii) procedural knowledge, (iii) conditional knowledge, (iv) planning, (v) information management strategies, (vi) monitoring, (vii) debugging and (viii) evaluation.

The variable *level of application of ill-structured problem solving processes* has four sub-variables, namely (i) problem representation, (ii) solution process, (iii) justification and (iv) monitoring and evaluation.

Independent Variables

There are two independent variables: (a) use of adaptive learning support and (b) use of scaffolds for ill-structured problem solving.

The independent variable *use of adaptive support features* has two possible values {yes, no}. The variable is related to the use or otherwise of the three adaptive support features in the system, namely (i) adaptive link annotation, (ii) adaptive link hiding and (iii) adaptive presentation.

Chapter 2 : Literature Review

The independent variable *use of scaffolds for ill-structured problem solving* has three possible values {Task Schedule, Question Prompts and no scaffold}.

Methodology for this research is presented in three chapters:

- Which deals with the design of the learning support system
- Which deals with the application of Machine Learning algorithms in the design of adaptive learning support
- Which deals with experimental design

These chapters are presented next.

Chapter 3 : Methodology – Design of Learning Support System

3.0 Introduction to Integrated Design Process

The Integrated Design Process (IDP) is a framework for the development of web-based learning systems, which was proposed and used by Nam and Smith-Jackson (2007). It is a theory-based approach which integrates instructional design with human computer interface design. Theories of learning inform the design goals for the system and the features to be included in the system. The IDP framework has four phases: needs analysis, conceptual design, development and formative evaluation. This framework was used to guide the development of the web-based system used in this research. The developed system had adaptive features for supporting the acquisition of domain knowledge. It also had features for supporting self-regulation.

The IDP as implemented by Nam and Smith-Jackson (see Figure 3-1) supported instructional features mainly based on cognitivism, for the learning of domain concepts, but it did not support constructivist learning. The human computer interface defined by the framework was static and not personalized to the individual online learner, yet learners are not the same, and therefore, one page does not fit all (Brusilovsky, 2001). Navigation and page design in IDP was therefore for the general learner using the web-based learning system and could not fit the multiple audience groups expected, such as novice, intermediate and advanced learners.

Nonetheless, IDP emphasized evolutionary development, which suits the development of web-based interfaces because of the dynamic nature of the purposes supported by such systems, the users and the content presented (Hadjerrouit, 2007). To facilitate evolutionary development, IDP paid adequate attention to formative evaluation (Nam & Smith-Jackson, 2007). Moreover, unlike traditional software development approaches used for the development of learning systems (Hadjerrouit, 2007; Katia & Granger, 2000), IDP paid a fair amount of attention to the instructional system, and not just the delivery of content without consideration of principles of instruction.

As shown in Figure 3-1, the IDP in its adopted form had the instructional sub-system and human computer interface sub-system. The area of intersection of the two ellipses represents the desired features of the system combining the two sub-systems. The original IDP was also informed by the characteristics of the domain whose knowledge was being learned, the representative characteristics of a typical learner using the system and theories of learning, mainly cognitivism, and instructional design principles informed by these learning theories.

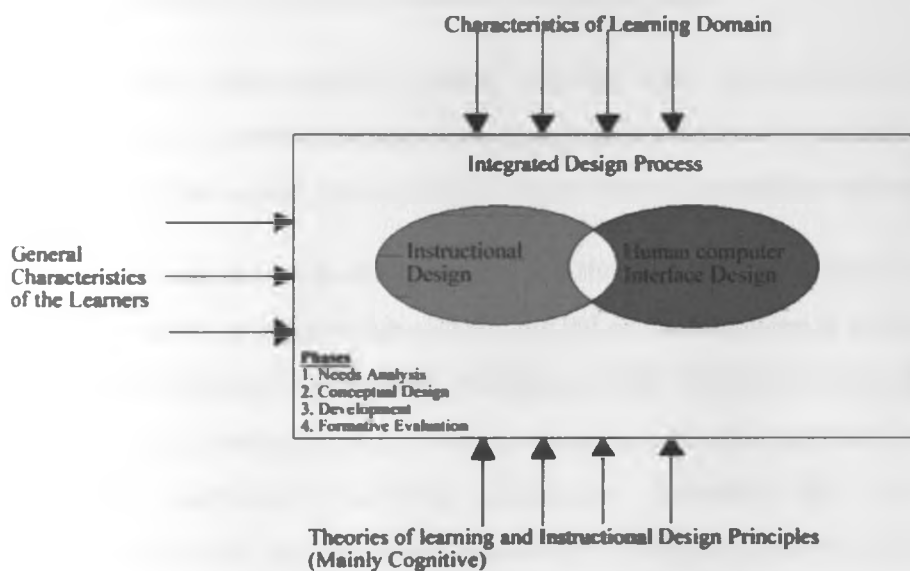


Figure 3-1 : Integrated Design Process without Adaptive Support and Problem Solving Support features, adapted from Nam & Smith-Jackson (2007)

For purposes of this study, some features were added to IDP:

- **Audience-driven website design:** Due to the need to make the human computer interface adaptive, it became necessary to incorporate audience-centred website design features into IDP. These features were adopted from audience-driven design for websites in the Website System Development Methodology (WSDM) by Olga De Troyer (2001). Navigation design

and page design features suitable for different audiences were included in order to realize adaptive navigation and presentation support respectively

- **Adaptive features:** The human computer interface specified by IDP was enhanced through the addition of adaptive features into IDP in order to realize an adaptive user interface. The adaptive features were used to provide learning support for the learners as they acquired domain knowledge i.e. OOP concepts. These technologies are used for adaptation in Adaptive Educational Hypermedia (Brusilovisky, 1996, 2003). The system's interface adapted itself to the different audience classes of users
- **Support for metacognitive control:** Features were also added to IDP for supporting constructivist learning through solving complex problems. An online task schedule was included in the system to support self-regulation during problem solving.

The adaptive web-based learning system had three sub-systems: (i) instructional sub-system, (ii) adaptive learning support sub-system and (iii) problem solving or metacognitive control and self-regulation support sub-system. The extended IDP is shown in Figure 3-2. The 4 areas of intersection of the three ellipses showing the sub-systems represent the desired features of the system that combines these three subsystems. Extended IDP is also informed by the characteristics of the domain whose knowledge is being learned, the individual characteristics of learners who are using the system and theories of learning, mainly cognitivism and constructivism, and instructional design principles informed by these learning theories. Extended IDP has additional processes, which are shown inside Figure 3-2, such as audience-driven navigation and page design as well as designing for the learner's metacognitive control.

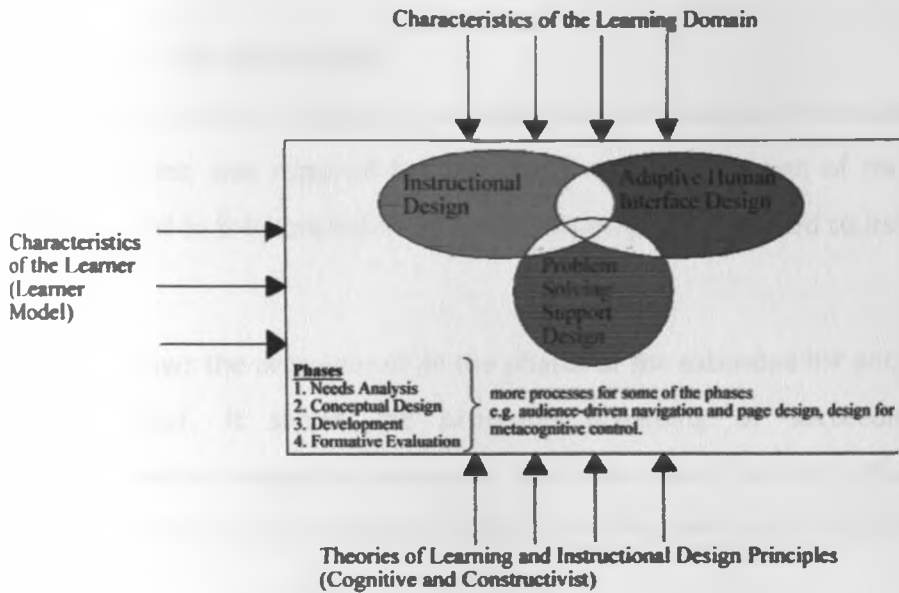


Figure 3-2 : Extended IDP with Adaptive Support and Problem Solving Support features

This chapter is structured as follows: Section 3.1 describes the needs analysis phase. The phase includes three processes: requirements specification, component and feature specification, and goal design. Section 3.2 presents information on how the conceptual design phase was carried out, including how the phase’s processes were performed: development of design scenarios and use cases, information design, navigation design, page design, structure design, presentation design and data design. Section 3.3 describes the system development phase, including its processes, namely development of the course and related tools and items, development of the low fidelity prototype, design walkthrough and development of the high fidelity prototype. Section 3.4 presents the formative evaluation phase. The phase was organized in terms of the three sub-systems and hence had three processes: evaluation of the instructional sub-system, evaluation of the adaptive user interface sub-system and lastly,

evaluation of the problem solving support sub-system. The section also describes the technical abilities of the experts who participated in the formative evaluation of the system.

3.1 Phase 1: Needs Analysis

Needs analysis involved gathering, analyzing and summarizing information about the course. This information was required for the design and construction of the web-based learning prototype used in this research. Needs analysis may also be called course analysis for learning systems.

Figure 3-3 shows the processes of all the phases of the extended IDP and how they are related to each other. It shows the processes preceding or succeeding others and the interdependencies among the processes. The 3 sub-systems are also shown in the figure, along the border of the system, to indicate that they are the ones that constitute the learning system.

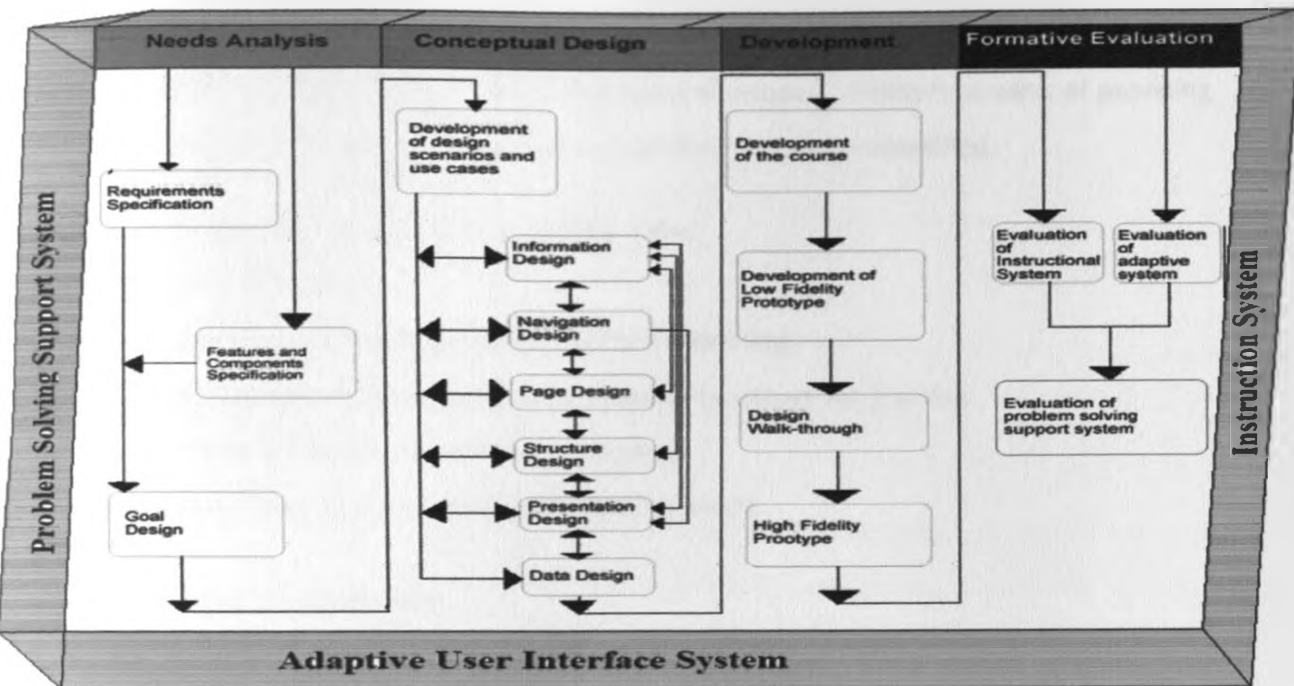


Figure 3-3 : The Phases of the Extended Integrated Design Process and their processes

The needs analysis phase involved three processes:

- (i) Requirements specification
- (ii) Component and feature specification
- (iii) Goal design

3.1.1 Process 1: Requirements Specification

Requirements specification involved various design activities intended to capture abstract high-level goals as well as more specific requirements for the development of the system prototype. The prototype was intended to be adaptive to the individual learner characteristics so as to offer personalized support when learning OOP concepts. It was also intended to provide learning support to the learner while engaged in solving ill-structured problems in a non-collaborative web-based learning set up.

An understanding was reached of the targeted user groups and the tasks they were expected to carry out. Then user and system requirements were developed. Pointers to ways of providing learning support to the learners as they carried out their tasks were identified.

Requirements specification involved five design steps:

- (a) Gathering of information
- (b) Analysis of instructional subsystem for objectivist learning
- (c) Analysis of problem solving support subsystem for constructivist learning
- (d) Analysis of the adaptive user interface subsystem
- (e) Preparation of the requirements specification summary

(a) Gathering of Information

To facilitate the development of the web-based learning system, different types of information was collected about:

- Learners such as learner characteristics, which may affect the learners' success in acquiring skills and knowledge of OOP programming
- Content to be included in the instructional system, such as depth and scope
- How best to structure, organize and sequence the content for successful learning
- Prior knowledge for the course as well as prior knowledge for individual concepts
- Learning objects to include in a lesson in the course
- Which learner characteristics to consider as the basis for adaptive interaction between the learner and the system
- Which abstractions to consider when forming multiple representations of the course for the different types of learners expected
- How inferences and adaptation can be made in the adaptive system given the prevailing conditions such as the number of participating learners, the learners' prior knowledge and background

- The characteristics of an ill-structured problem and how to scaffold metacognition and problem solving processes
- Various designs for web-based learning using ill-structured problem solving, such as project based, problem based, inquiry based and case based learning
- Which adaptive technologies are suitable for the course that is to be offered

(b) Analysis of the Instructional Subsystem for Objectivist Learning

The instructional sub-system was concerned with content suitable for the OOP course, designed in a way that suits the specific types of learners expected to use the system, and delivered according to selected suitable pedagogical, particularly objectivist learning approaches. Objective reality needs to be represented in terms of models, symbols and suitable structuring of content to reduce cognitive load and enable the passing of the content onto the learner's mind i.e. into the learner's cognitive schemata. The learner manipulates the symbols and models through his thought processes, until the mind mirrors the objective reality.

In order to achieve delivery of the objective knowledge to the learner, there was need to do a course analysis (Montiliva, Sandia & Barrios, 2002. Course analysis focused on two aspects:

- (i) course domain and
- (ii) course components

(i) Analysis of the Course Domain

The program of study to which the OOP course belonged was examined. The course belonged to a Bachelor of Information Technology program. This program required students to develop practical information technology skills such as solving programming tasks.

The OOP course had a pre-requisite course which had already been taught called Introduction to Programming using the C language.

The concepts learned in the OOP course were also to be used during an Object Oriented Analysis course later on.

(ii) Analysis of the Course Components

Analysis was done according to the main components which are involved in a course. The following analyses were conducted:

- a) Course content analysis
- b) Instructional goals analysis
- c) Course objectives analysis
- d) Learning task analysis
- e) Learner characteristics analysis
- f) Analysis of the learners' environment
- g) Analysis of strategies for promoting interaction and learner control
- h) Analysis of required assessment and evaluation
- i) Feedback analysis

(c) Analysis of the Problem Solving Support Subsystem for Constructivist Learning

With respect to the problem solving support subsystem, the following analyses were carried out:

- a) Goal analysis
- b) Content analysis
- c) Analysis of the learning context
- d) Analysis of the learner's prior knowledge and how to activate it
- e) Learner analysis
- f) Analysis for appropriate learning support
- g) Analysis of strategies for promoting learner control
- h) Analysis for appropriate and useful evaluation and assessment strategies

(d) Analysis of the Adaptive User Interface Subsystem

Analysis for the adaptive user interface subsystem was guided by the following questions:

- a) What are the **reasons** for making the user interface system adaptive?
- b) How much **control** is the **learner** going to have in the system?
- c) Which features of the system are to be adaptive?
- d) Which learner characteristics will the system adapt to?
- e) Which multi-representations should be created so as to make the system flexible enough to support different types of learners during adaptation?
- f) How will the adaptations be realized through interactive design?
- g) How are the values of the learner characteristics going to be acquired? Is it directly (obtrusively) by asking the learners or implicitly (non-obtrusively) from the record of interaction between the learner and the system?
- h) Out of the information about the interaction between the user and system, what characteristics should be included in the interaction record to be used for adaptation?
- i) Should the learner model be an overlay over the entire domain model or over part of the concepts contained in the domain model?
- j) Which rules should be used to guide the inferences and adaptations made by the system?
- k) Evaluation – which usability metrics could be used during formative and summative evaluation of the adaptive system?

(e) Summary of Requirements Specification

I. Functional Requirements

This part of requirements specification involved developing a high level abstraction of possible user activities when using the web-based system. These activities were identified along two dimensions: according to the learners' knowledge level i.e. novice, intermediate

or advanced and according to the type of learning design i.e. objectivist or constructivist design.

II. Data Requirements

Data requirements for the system were defined in terms of the **course/domain model**, the **learner model** and the **interaction model**. Consideration was also given to both the objectivist and constructivist learning designs.

a) The Course Model/Domain Model

Objectivist Course Model

- ✓ **Task level: Four learning goals** were identified for the course
- ✓ Identification of **what needs to be done** and **components** to be included in order to achieve the purpose of the system: 14 **course concepts** and their **learning activities**, as well as **other objects** to included for every concept to ensure effective learning of the concept's content
- ✓ **Levels of learner actions: Three levels** of performance during learning were identified i.e. **remember level**, **use level** and **find level**.
 - i. **Remember level** involves learning facts, concepts, principles and procedures of the domain.
 - ii. **Use level** involves applying the knowledge from remember level to carry out a task.
 - iii. **Find level** involves determining new principles or relations or generalizations from the learning experience
- ✓ **Presentation styles of components:** There was need to use different presentation styles for the components to make the actions possible
 - i. **Expository generality**, which involves presentation of general information on facts, concepts, principles and steps in a procedure, through the use of objects for presenting theory of the course.

- ii. **Expository instance**, which involves instantiation of the concepts, principles and procedures, through provision of examples with code and real world examples to demonstrate OOP concepts, principles and procedures.
- iii. **Inquisitory generality**, which is achieved through an opening question to pose a challenge at the start, followed by presentation of related theory.
- iv. **Inquisitory instance**, which includes things for the learner to do such as regular short exercises and problem solving tasks

A more detailed description of levels of performance and presentation styles for knowledge components are in section 2.8.4.

Constructivist Course Model

The knowledge and skills needed for solving a problem may be obtained from multiple sources.

- ✓ There was need to specify the **domain** from which concepts to be applied for problem solving were to be drawn i.e. the OOP domain
- ✓ There was need to provide the learners with **basic stable content** for problem solving. The learners were provided with content of the OOP course in order to give them some basic stable source of knowledge and skills before commencement of problem solving
- ✓ There was need to provide the learners with **ill-structured problems** whose context they could easily identify with, so as to reduce the effort spent in understanding the context of the problem before embarking on actual problem solving. The problems they were provided with were all from such contexts
- ✓ The **problems** were supposed to be of a **size** that could be solved within the available time and **relevant** to the OOP course

b) Learner Model

The learner model was for individual learners and not groups of learners.

Learner Model for the Objectivist System

The learner model constitutes knowledge about the learner that the learning system uses to influence its interaction with the learner.

In this study, the top level of the learner model had the **user profile model and other general information**, as shown in Table 3-1. The items of information are not presented in any specific order.

Table 3-1 : Information contained at the top/profile level of the learner model

✓ Names of the user	✓ Date of last login	✓ Number of self initiated programming languages already done
✓ Username	✓ Overall knowledge level	✓ Grade in pre-requisite course
✓ Password	✓ Whether the learner is using adaptive support or not	✓ Number of courses involving mandatory use of computer applications
✓ Gender	✓ Number of programming languages mastered	✓ Level of anxiety when engaging in programming
✓ Program of study	✓ Number of programs already written	✓ Self report of interest in programming
✓ Date of first login	✓ Average length of programs	✓ Perceived level of difficulty of OOP

✓ **Goal level of the student model.** The information held at this level included

- Performance level for each goal (test marks)
- Overall knowledge level for the goal i.e. novice, intermediate or advanced

✓ **Concept level of the student model.** This level contained the following information:

- Prior knowledge level for each concept
- Current level of knowledge
- Level of performance in the concept i.e. marks
- Preference for additional learning material

The learner model for the objectivist part of the course is shown in Figure 3-4. It was an overlay model, with the learner attribute values closely mapped over the domain model’s goals and concepts. For each goal $LG_1 \dots LG_n$ and concept $C_1 \dots C_n$, the learner had a value in the model.

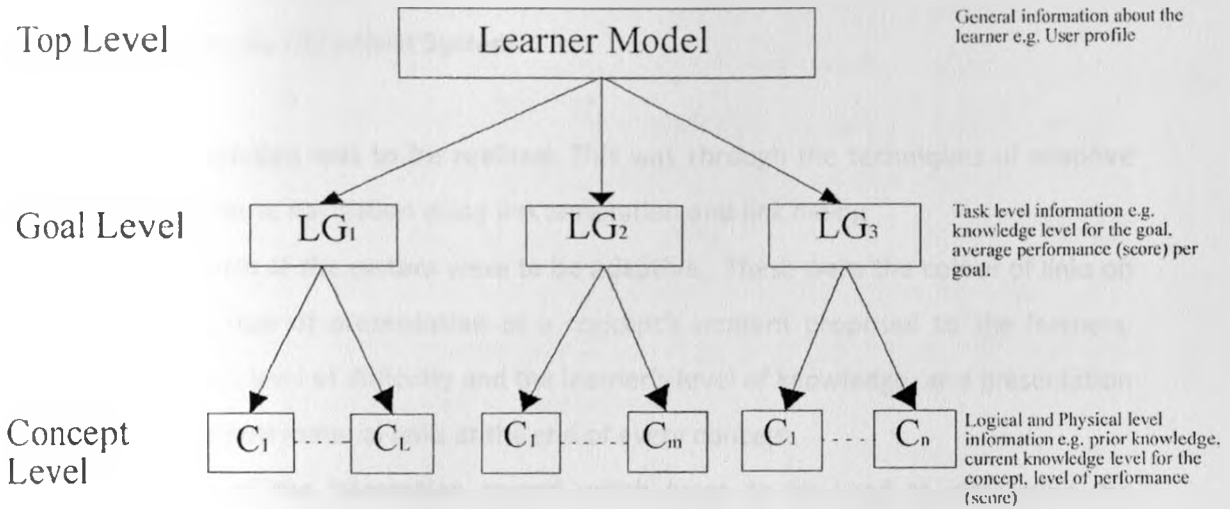


Figure 3-4 : Learner model showing the top, goal and concept levels

Learner Model for the Constructivist System

✓ **Learner task schedule information.** This is information contained in the task schedule that was needed for the adaptive display of feedback about the progress of the learner in different tasks which were part of the problem to be solved. The learner model information included:

- Percentage completion per task
- Start date for a task
- Duration of time elapsed since the start date

This information would be useful in deciding the colour of concept names in the graphic that was to be used to provide feedback on how the learner was progressing with their tasks.

c) The Adaptive Interaction Model

Interaction Model for the Objectivist System

There was need to:

- ✓ Determine **how adaptation was to be realized**. This was through the techniques of adaptive presentation and adaptive navigation using link annotation and link hiding
- ✓ Establish which **features of the system were to be adaptive**. These were the colour of links on concept names, the type of presentation of a concept's content proposed to the learners, based on the content's level of difficulty and the learner's level of knowledge, and presentation or otherwise of additional material links at the end of every concept
- ✓ Determine **contents of the interaction record** which were to be used to determine the interaction suitable for a particular learner. The information that was logged to constitute the dialogue record includes results for pre-test and concept evaluation tests and whether the learner had used additional learning materials for a concept or not
- ✓ Establish a way of **updating the learner model** to reflect progress in terms of knowledge acquisition. This was provided through heuristics
- ✓ Establish which **rules to be used to guide the inferences made by the system**. The Heterogeneous Value Difference Metric (HVDM) was used to determine the initial knowledge level of the learner. The Naïve Bayes classifier (NBC) was used to classify the learner as likely to use additional materials at the end of a concept or not
- ✓ Establish rules to guide **adaptations** on the system's interface. This was done in the form of suitable heuristics
- ✓ Provide learners with **control over what they studied**. This was achieved by providing them with means of ignoring the system's suggestions. Control was provided by leaving all annotated concept links and tabs for all content presentation types clickable as well as providing a way for the learner to access hidden additional material links

The interaction model for the adaptive user interface system and its parts are shown in Figure 3-5. The model had a dialogue control and presentation components. The dialogue control

Component constituted the rules determining presentation of information to the learner, the intelligence i.e. Machine Learning algorithms for inferring learner model attribute values and a dialogue record of the learner's activities in the system. The presentation component constituted the different types of presentation of information to the learner, such as general, static presentation or adaptive user interface presentation in terms of text presentation, link annotation or link hiding. The interaction model relied on the learner model for learner attribute values used during adaptation, such as knowledge level. The interaction model also relied on specific abstractions of the domain model's content for adaptive user interfaces and the general domain content presentation for the static user interface.

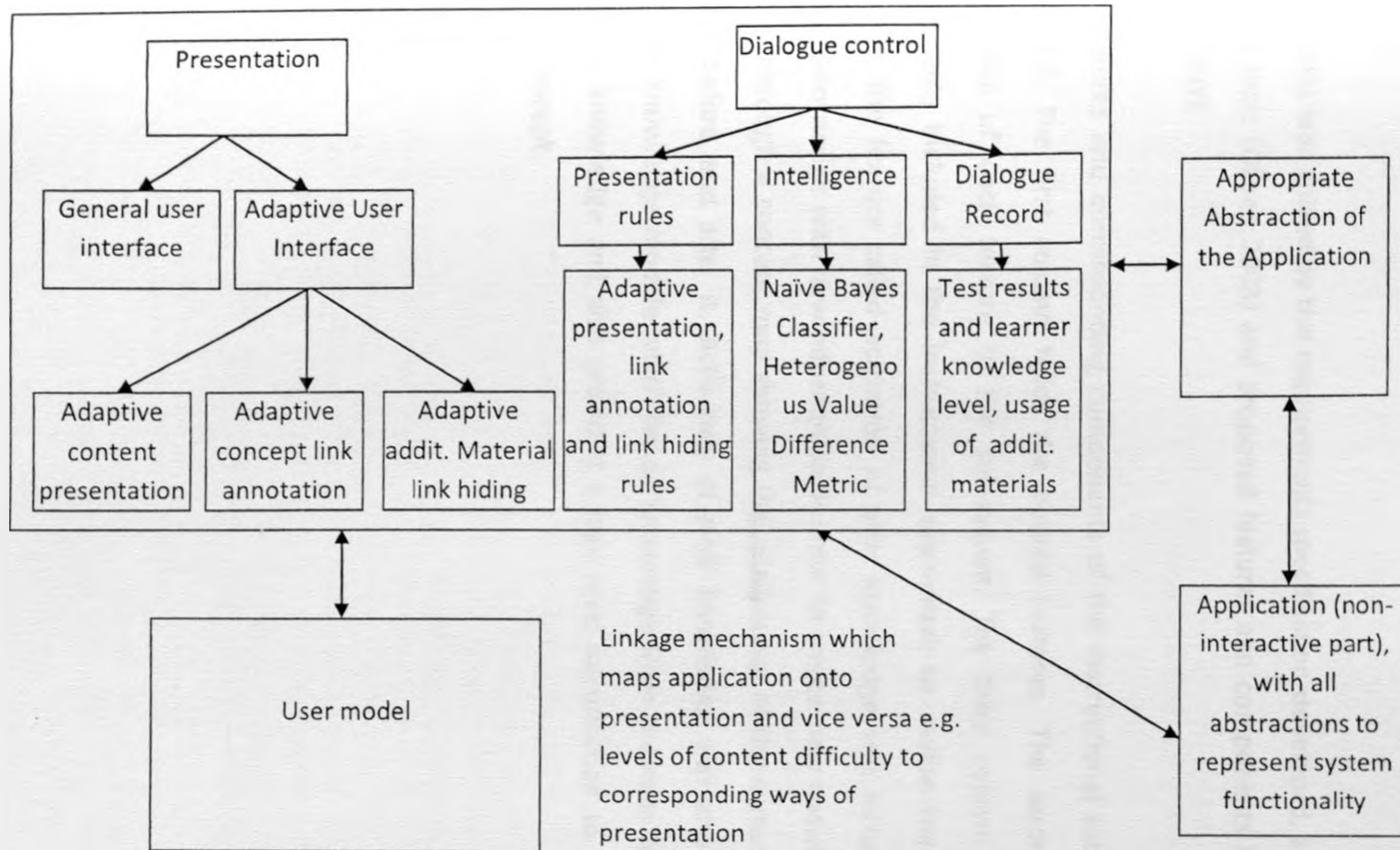


Figure 3-5 : Interaction model for the adaptive user interface system

3.1.2 Process 2: Component and Feature Specification

The purpose of this process was to come up with key features of the web-based learning system and the corresponding components to satisfy the requirements specified in the previous process.

The process was guided by the requirements specification developed, a proposed list of online learning tools (Oliver, 2003) and proposed features and components for web-based learning (Khan, 1997).

The Features and corresponding components of the instructional sub-system are shown in Table 3-2. The first column holds the desired features. The second column shows the importance of each feature to the sub-system. The third column presents the specific components included in the instructional sub-system to realize the desired features. For example, the feature called activation of prior knowledge was included to link previously learned knowledge with new information in order to create new knowledge. The feature was realized through a concept map showing the relationship of the current concept to concepts coming before and after it. Activation of prior knowledge was also achieved through an overview knowledge module which linked knowledge from previous concepts to the current concept's knowledge and also provided a high level introduction to the knowledge of the current concept.

Table 3-2 : Features and components of the instructional system

Feature	Relationship with system	Components
Simple to complex wholesome sequencing of domain content	<ul style="list-style-type: none"> To enable the learner to study domain content at a level of difficulty that suits the level of cognitive load that he can manage 	<ul style="list-style-type: none"> Knowledge modules such as theory, different types of examples and different types of exercises Dynamic generation of the lesson, from a combination of knowledge modules at three levels of performance i.e. remember, use and find, to suit novice, intermediate and advanced levels of knowledge Sequencing structure over the entire course, with domain concepts presented before problem solving The arrangement of goals, concepts, sub-concepts and their objects in the default order in which they should be studied (with the learner having the freedom to choose any order in which to study them)
Top-down structuring of content	<ul style="list-style-type: none"> To allow the students to access the knowledge under a goal in manageable chunks To provide the relationship between the various main concepts, sub-concepts and pre-requisite concepts 	<ul style="list-style-type: none"> A listing of goals with links for drilling down to the concepts contained in each goal and a link from each concept to its contents (knowledge modules) An expandable and collapsible map showing the concepts of a selected goal and how they are ordered The structuring of the course into pre-test, objectives, overview, pre-requisites, content, exercise, find level task, summary, concept evaluation and additional materials
Feedback	<ul style="list-style-type: none"> To provide feedback that suits individual learners and their actions: performance in a test or the students' knowledge level 	<ul style="list-style-type: none"> Various types of tests with answers e.g. pre-test and concept evaluation Progress graph showing percentage score per concept and average score for a goal Summary table of concepts to be studied for remediation per question after concept evaluation Countdown of time left to the end of a test
Integrated assessment	<ul style="list-style-type: none"> Integrated assessment provides formative feedback 	<ul style="list-style-type: none"> Online tests providing instant feedback Online practice exercises
Learner control	<ul style="list-style-type: none"> To enable learners to be in charge of how they study course content 	<ul style="list-style-type: none"> Specification of pre-requisite concepts to enable the student increase one's entry level for the concept Online system available 24 hours The system was tested on common browsers
Activation of prior knowledge	<ul style="list-style-type: none"> To link previously acquired knowledge to what is being learned currently 	<ul style="list-style-type: none"> A concept map showing the relationship between previously studied concepts and the current concept Overview at the beginning of each concept Pre-requisite concepts for each main concept

For the adaptive user interface sub-system, the features which were identified, the value of each feature to the sub-system and the components of each feature are shown in Table 3-3.

Table 3-3 : Features and components of the adaptive user interface system

Feature	Relationship with system	Components
Learner control	<ul style="list-style-type: none"> • To enable learners to be in charge of how they study course content (which goals, concepts or level of difficulty) • To ensure the system does not disrupt learners through inflexible suggestions 	<ul style="list-style-type: none"> • Links of concept names to remain clickable, in spite of adaptive navigation suggestions • A map of concept components provided on the left hand side to enable selection of additional materials even when adaptively excluded from a concept's content • Clickable tabs for adaptive presentation, for all levels of difficulty, after the pre-test
Adaptive presentation support	<ul style="list-style-type: none"> • To enable selection of appropriate content to study 	<ul style="list-style-type: none"> • A tab on the page for each of the three levels of difficulty of course content
Adaptive navigation support	<ul style="list-style-type: none"> • To enable the selection of an appropriate concept to study • To reduce cognitive load by hiding or otherwise of additional materials according to learners predicted needs 	<ul style="list-style-type: none"> • Link hiding (of additional material links) to remove unnecessary additional materials • Link annotation (of concept names) to guide concept selection
Learner modeling	<ul style="list-style-type: none"> • To form a representation of the student's characteristics which are used to guide adaptation e.g. knowledge level and tendency to use additional materials 	<ul style="list-style-type: none"> • An overlay model with knowledge and usage of additional material links for every concept • A Machine Learning feature for making inferences • A component for collecting learner information as part of the learning process, without distracting learning
Multiple representation	<ul style="list-style-type: none"> • To have more than one abstraction to facilitate adaptation 	<ul style="list-style-type: none"> • Concept names annotated with different colours according to learner's readiness to study a concept • Concept content packaged dynamically at different levels of difficulty, according to the learner's knowledge level • Links to additional materials included as part of a lesson or not, according to usage tendency from previous concepts

For the problem solving support sub-system, the identified features, the importance of each feature to the sub-system and the components of each feature are shown in Table 3-4:

Table 3-4 : Features and components of the problem solving support system

Feature	Relationship with system	Components
Scaffolding self-regulation during problem solving	<ul style="list-style-type: none"> Support learners to be in control of their problem solving processes and metacognitive knowledge to ensure learning and project completion 	<ul style="list-style-type: none"> A task schedule authored by the learners, with tasks to be done and the required knowledge elements, skills and other resources for each sub-task A graphical display of task progress, with the expected task duration and time which has already passed, to remind learners of the expected levels of progress according to time already spent and to make them to think about appropriate adjustments in order to remain on track in problem solving Question prompts for each problem solving process expressed in terms of OOP, covering various aspects of metacognition; it was used to cause the learner to have an internal conversation on what should be done and how far one should have progressed in problem solving, among others A description of the problem solving process expressed in OOP terms; it should be explained before a copy is provided to them at the start of problem solving
Feedback	<ul style="list-style-type: none"> To provide feedback that suits individual learners progress in their problem solving tasks 	<ul style="list-style-type: none"> A graphic providing feedback on the problem solving progress, such as tasks running late or task completion percentage, using different coded colours.
Learner control	<ul style="list-style-type: none"> To enable learners to regulate themselves during problem solving To support problem solving with the necessary resources by allowing access to content on the system and additional materials from the Web 	<ul style="list-style-type: none"> Online system available 24 hours. Access to the Web provided A general guide for problem solving to start the learners off A guide on how to use task schedule A dynamic graphic providing current information on the student's progress on various sub-tasks
Activation of prior knowledge	<ul style="list-style-type: none"> To enable the linking of previously acquired knowledge to what is being learned currently 	<ul style="list-style-type: none"> A task schedule requiring identification of knowledge and skills for each sub-task at the beginning of problem solving

3.1.3 Process 3: Goal Design

Goal design was the process of determining goals and principles to guide all the remaining processes and steps in the development of the web-based learning system. The design goals and principles for the system were grouped according to the sub-systems:

- a) instructional system
- b) adaptive user interface
- c) constructivist learning design goals and principles

The inputs to this process were the requirements specification, and system features and their corresponding components as developed in previous processes.

The following general goals were identified:

- To understand the learning process in terms of learning theories and instructional design models and use this understanding to guide the development of the learning system
- To facilitate the management of learning and problem solving processes instead of considering technological affordances alone
- To consider the need for personalization of web-based learning
- To consider the system's look and feel as an important part of the user's experience with the web-based learning system
- To facilitate participation of learners in active, learner-centered learning through problem solving

The following design steps were applied:

- 1) Defining design goals of the instructional system (Table 3-5), guided by the work of Dick and Carey (2005) on systematic design of instruction and the requirements specification from a previous process
- 2) Defining design goals of the general interface system (Table 3-6) and the adaptive user interface system (Table 3-7), guided by the ISO 9241-11 standard for usability evaluation, as well as the work of Benyon (1992a) and Langley (1997)

3) Defining design goals for the constructivist learning system (Table 3-8, guided by the work of Azevedo and Hadwin (2005), Jonassen (1999), McLoughlin and Hollingworth (2001), Savery and Duffy (1995), and Brickell and Herrington (2006)

For each of Table 3-5, Table 3-6, Table 3-7 and Table 3-8, the first column provides the design goal e.g. to motivate learners. The second column provides a description of the meaning of the goal e.g. to give learners meaningful feedback to ensure meaningful and continued learning or problem solving. The last column provides sources of the literature which informed the goals.

Table 3-5 : Design goals for instructional system

Design goal	Instructional System	
	Description	Reference
Clarity	<ul style="list-style-type: none"> To design learning materials that are clear and easy to use for individual learners 	Dick, Carey & Carey (2005)
Impact	<ul style="list-style-type: none"> To design a learning system that increases an individual learner's attitude about the system 	Dick, Carey & Carey (2005)
Suitability to learner learning needs (leading to reduced cognitive load)	<ul style="list-style-type: none"> Ease of access through course formatting /structuring To ensure that the course content is organized in a flexible way through having goals, various types of concepts and their knowledge modules which can be combined in different ways The content has multiple representations e.g. levels of difficulty/performance to suit the needs of different types of learners, making simple to complex sequencing possible The content of each concept has different types of objects e.g. theory, different types of exercises, different types of examples and other activities Pedagogical considerations leading to different types of learning objects and their sequencing Learner control to ensure user experience and usability of the instructional system which suits the students, through the possibility of selecting what to study 	Martinez-Torres et al (2008), Wixom and Todd (2005), (Van Marrienboer & Sweller, 2005)
Motivate learning (integrated assessment)	<ul style="list-style-type: none"> To give learners meaningful feedback to ensure meaningful and continued learning 	Martinez-Torres et al. (2008)
Activation of prior knowledge	<ul style="list-style-type: none"> Encourage the integration of old and new knowledge by showing relations using pre-requisite concepts and a concept map showing positional relationship 	Brickell & Herrington (2006)

Table 3-6 : Design goals for the general user interface system

User interface system		
Design goal	Description	Reference
Effectiveness	<ul style="list-style-type: none"> To increase the accuracy and completeness of the course content including tests and their answers 	ISO 9241-11
Efficiency	<ul style="list-style-type: none"> To reduce the resources expended e.g. cognitive resources spent identifying appropriate navigation paths and content 	Martinez-Torres et al. (2008), ISO 9241-11
Satisfaction	<ul style="list-style-type: none"> To ensure users find the system comfortable and acceptable to use, through various design effects 	Martinez-Torres et al. (2008), ISO 9241-11
Ease of navigation	<ul style="list-style-type: none"> Through predictable, comprehensive and consistent location of various parts of the course 	Sia et al. (2007)

Table 3-7 : Design goals for the adaptive user interface system

Adaptive user interface system		
Design goal	Description	Reference
Controllability	<ul style="list-style-type: none"> To allow the learner to control his learning by choosing what content to study To let the learner keep the option of not following the system's suggestions by choosing an alternative navigation path 	(Papanikolaou et al., 2003), Martinez-Torres (2008)
Unobtrusiveness	<ul style="list-style-type: none"> To collect learner information without interfering with the learning process, such as information from clicks To ensure consistency of adaptation to avoid confusing and frustrating learners, hence reduce learning effort 	Sia et al. (2007)
Flexibility through multiple representations	<ul style="list-style-type: none"> To provide different abstractions of the course in the system, according to complexity, to correspond exhaustively to the expected types of learners 	Papanikolaou et al. (2003), Benyon (1992a)
Dynamic learner models	<ul style="list-style-type: none"> The system keeps improving by acquiring more knowledge about the learners e.g. through mining new knowledge from recorded learner activities 	Langley (1999), Benyon (1992a)
Personalized Supports	<ul style="list-style-type: none"> The system makes suggestions based on its interaction with the learner 	Virvou and Tsiriga (2004), Brusilovskiy (1996, 1999)

Table 3-8 : Design goals for the problem solving support system

	Self-regulation supporting system	
Design goal	Description	Reference
Learner control	<ul style="list-style-type: none"> Allow the learner to be in charge of the problem solving process through the possibility of taking various actions, monitoring, remediation and evaluation 	(Azevedo & Hadwin, 2005; Chng, 2001; Oliver & Herrington, 2001)
Authentic and meaningful learner experiences	<ul style="list-style-type: none"> Provide opportunities for authentic experiences, which together with prior knowledge, are used to build knowledge. 	(Vrasidas, 2000)
Task management support structure for metacognition	<ul style="list-style-type: none"> The interactive system should enable learners to manage their skills by providing them with some structure e.g. structure of the problem solving process 	(Vrasidas, 2000)
Activation of prior knowledge	<ul style="list-style-type: none"> Prior knowledge needs to be activated e.g. retrieval of relevant information related to a sub-task during sub-task planning, through enforced reflection 	(Brickell & Herrington, 2006)
Self-reflection	<ul style="list-style-type: none"> Allow the scaffold to invoke self-reflection according to an individual's reality as well as potential during problem solving 	Chng (2001), Azevedo and Hadwin (2005)
Personal meaning	<ul style="list-style-type: none"> To provide supports tailored to learner specific needs e.g. supporting individual plans, which catalyze the answers to come from within the individual learner 	Koohang, Riley, Smith & Schreurs (2009)

3.2 Phase 2: Conceptual Design

Conceptual design is a detailed design phase involving an explicit building of concepts about what the learning system is, what it can do, and how it is to be used (Nam & Smith-Jackson, 2007). It translates user requirements into a conceptual user interface, instructional design, adaptive learning support system design and design of a system for supporting problem solving.

Conceptual design is separated from actual implementation design. This makes it independent of implementation technology (De Troyer, 2001).

The processes involved in conceptual design are:

- (i) design scenarios and use case development

(ii) information design

(iii) navigation design

3.2.1 Process 1: Development of Design Scenarios and Use Cases

A use case is a sequence of steps showing how a user interacts with a system. It constitutes related scenarios which are tied together because they relate to a common user goal (Fowler & Scott, 2003).

The process of developing design scenarios involved four steps: (i) identification of main tasks, (ii) specification of user interface objects, (iii) identification of key user roles, and (iv) creation of design scenarios

1) Identification of the Main Tasks

The main tasks to be supported by the system in order to ensure meeting of user requirements were identified. They are related to learning and instructing. They are presented in Table 3-9. The first column contains the main tasks e.g. learning. The second column contains specific sub-tasks of the main task e.g. taking a pre-test or selecting a link for additional learning material.

Table 3-9 : Main tasks for system users

Main Task	Subtask
Learning	• Selecting goal and concept to study
	• Taking pre-requisites (if necessary)
	• Taking concept pre-test
	• Selecting the appropriate presentation (appropriate level of difficulty- novice ,intermediate or advanced) to study
	• Selecting additional learning material
	• Taking concept test
	• Managing the problem solving process using task schedule or question prompts
Instructing	• Learner administration
	• Course content administration

2) Specifying User Interface Objects Involved with the Tasks

User interface objects were the components identified during the process of identifying features and components for the learning system. These included concept map/menu, annotated hyperlinks, tabs for different types of presentation of domain content, various knowledge components, the task schedule and the graphic associated with the task schedule, among others

3) Identifying the Key User Roles Involved

The roles which people play with respect to the system, called actors, are important in the identification of use cases (Fowler & Scott, 2003). For this system, the roles were learner, instructor and web administrator. Where there were sub-roles, the sub-roles were considered as different actors, and the associated use cases were also identified (Fowler & Scott, 2003). Table 3-10 shows the roles identified for this system, their sub-roles and the criteria for breaking down the role into its sub-roles. For example, the sub-roles of a learner are Novice learner, Intermediate learner and Advanced learner, based on the learner's level of knowledge

Table 3-10 : Identification of key user roles and their sub-roles for the learning system

Key Role	Sub role	Criteria
Learner	• Novice learner	• Knowledge level
	• Intermediate learner	• Knowledge level
	• Advanced learner	• Knowledge level
	• Use additional materials	• Preference for additional learning materials
	• No use of additional materials	• Preference for additional learning materials
	• Problem solver	• Learning by solving ill-structured problem
Instructor	• Content administrator	• Primary roles for instructor
	• Learner administrator	• Primary roles for instructor

4) Creating Design Scenarios

Design scenarios were created, taking into consideration the main tasks, the key user roles and the user interface objects identified in previous steps. Table 3-11 shows a sample of the use cases and design scenarios created for each user role. The use cases and scenarios guided the design of this system.

Table 3-11 : Examples of use cases and design scenarios for the different roles in the system

Role	Use Case	Design Scenarios
Learner	• Learning domain content	• Main scenario for the use case
		• Logging in the first time
		• Learning activities for novice learners
		• Learning activities for advanced learners
		• Attempting a pre-test
		• Updating personal information
		• Referring to additional materials
		• Attempting concept evaluation
		• Solving the complex
		• Preparing for problem solving
Content administrator	• Course management	• Carrying out problem solving
		• Self-regulation during problem solving
		• Register personal information
		• Learner administration

Example design scenarios for the use case learning domain content

- **Main scenario for logging into the system the first time**
 - Login
 - Register personal information (*personal information page opens automatically*)
The system offers the student the goals for the course (list of goals which are clickable URLs to lead to the concepts of the goals)
 - Select a goal to follow.
The system offers concepts under the goal

- Select concept to study

The system offers the concept objects

- The student attempts the pre-test

Then the system proposes the correct level of difficulty the student is to follow for the current concept i.e. novice, intermediate or advanced (using colour gray for the proposed level of difficulty while the others remain black)

- The student selects an appropriate level of difficulty by either following the proposed level or just selecting another level that he feels is appropriate for him
- The student then does the activities contained in the selected level of difficulty
- After that the student reads the summary of the concept
- The student chooses to do further reading by following additional material links or not
- The student takes the concept evaluation test

– **Learning activities for novice learners scenario**

- Read the theory
- Read the real life examples

– **Learning activities for intermediate learners scenario**

- Read the theory
- Read the real life examples
- Read the examples with code
- Attempt exercises

– **Learning activities for advanced learners scenario**

- Read the theory
- Read the real life examples
- Read the examples with code
- Attempt exercises
- Attempt the find level task

3.2.2 Process 2: Information Design

Information design is the process of describing the conceptual design of the content for the instructional and user interface systems, and the supports for metacognitive control for the problem solving support system.

The result of the process is identification and outlining of the content required for the course, and the role of supports during problem solving.

During information design, learning theories and related instructional design principles were used, together with user requirements identified in previous processes.

The steps followed in information design include:

1) Auditing of Existing Learning Content

This step entailed reviewing lecture notes from two lecturers as well as a course book used by one of the lecturers. The intention was to select content suitable for the purposes of the experiment i.e. teaching OOP concepts which were to be applied during problem solving and that were also suitable for the kind of learners expected for the course.

2) Outlining of Learning Content

Theories of learning i.e. behaviorism, cognitivism and constructivism, and related objectivist and constructivist instructional design principles were applied. Cognitive load theory was also used to inform the conceptual design of content in order to meet the requirements of learners. The output of this step was content for the instructional system (see Table 3-12).

3) Outlining of Interface Content

The interface content to be included in the system was designed. Design scenarios, cognitive load theory as well as functional and navigational requirements were considered during the design of this content. The output was an outline of content for a general user interface (see Table 3-15) and an outline of content for an adaptive user interface (see Table 3-13).

4) Outlining of the Role of Problem Solving Supports

Theories of learning and related instructional design principles, cognitive load theory and theory of metacognition were used together with user requirements from previous processes, to guide the conceptual design of the supports for metacognitive control. The output of this step was an outline of content for the problem solving support system (see Table 3-14).

Some of the information was designed to be provided by the system to the learners. Some of the other information needed for problem solving was to come out of the learners during active participation in ill-structured problem solving, through the use prompts in the learning scaffolds for self-regulation.

For Table 3-12, Table 3-13, Table 3-14 and Table 3-15, the first column holds the requirements which were identified during the requirements specification process. For example, for the instructional sub-system in Table 3-12, one of the requirements is to provide clear learning outcomes to guide learning.

The second column identifies a suitable design principle to guide what to do in order to attain the corresponding requirement. For example, in providing clear learning outcomes, it is necessary to ensure that the outcomes defined are observable, measurable and specific.

The third column provides the learning theory that informs the design principle. For example, the principle “observable, measurable and specific learning outcome” is based on behaviorism, which emphasizes observable behavior in the learners as a result of an instructional intervention. Cognitivism links this behavior to change in the learner’s memory knowledge structures. Bloom’s Taxonomy guides the development of desirable learning outcomes at various levels of difficulty.

The fourth column presents the components which are included or supported in the system in order to attain the requirement. For example, for clear learning outcomes, learning objectives

are defined under the guidance of Bloom’s taxonomy of (cognitive) objectives, with the desired learner behavior and level of performance also specified in the objective

Table 3-12: Content for the instructional system

Requirement	Design Principle	Learning Theory	Learning Content
<ul style="list-style-type: none"> • Provide clear learning outcomes to guide learning 	<ul style="list-style-type: none"> • Emphasis on observable and measurable, specific learning outcomes 	<ul style="list-style-type: none"> • Behaviorism • Cognitivism and • Bloom’s taxonomy 	<ul style="list-style-type: none"> • Learning objectives according to Bloom’s taxonomy of objectives for the cognitive domain
<ul style="list-style-type: none"> • Provide learners with an appropriate abstraction of domain content 	<ul style="list-style-type: none"> • Sufficient content to learn, in a suitable presentation style and at an appropriate level of difficulty, in order to achieve the cognitive objectives 	<ul style="list-style-type: none"> • Behaviorism • Cognitivism • Component display theory • Elaboration theory • Cognitive load theory 	<ul style="list-style-type: none"> • The theory, example and exercise objects of the course both in the expository and inquisitory modes, in multiple presentation styles
<ul style="list-style-type: none"> • Provide learning activities 	<ul style="list-style-type: none"> • Provide a variety of learning activities to suit different learners 	<ul style="list-style-type: none"> • Cognitivism 	<ul style="list-style-type: none"> • Different types of tests and exercises
<ul style="list-style-type: none"> • Provide informative feedback 	<ul style="list-style-type: none"> • Provide learners with different types of messages to tell them how they are doing 	<ul style="list-style-type: none"> • Cognitivism 	<ul style="list-style-type: none"> • Provide test score per concept, average score per goal as a graph and a percentage; list of concepts for remediation per question in a test; display of countdown of time for a test
<ul style="list-style-type: none"> • Reduce the learners cognitive load 	<ul style="list-style-type: none"> • Provide system features which make it easy for the learner to manage course information 	<ul style="list-style-type: none"> • Cognitive load theory 	<ul style="list-style-type: none"> • Global structure of content (concept map), in the default logical order of concepts and the learning objects within a concept; chunking content into manageable portions; • Simple to complex sequencing of concepts and content within a concept i.e. real life → theory → examples with code → regular exercise → complex, conventional task; • Overview and summary for activation of prior knowledge

Table 3-13 : Content for the adaptive interface system

Requirement	Design Principle	Learning Theory	Learning Content
<ul style="list-style-type: none"> • Enable learners to be in charge of their learning 	<ul style="list-style-type: none"> • Provide the learners with opportunities for active involvement in their learning 	<ul style="list-style-type: none"> • Cognitivism • Cognitive load theory 	<ul style="list-style-type: none"> • Provide learners with an alternative route to enable them decide on what to study in case they are not satisfied with the systems' adaptive suggestions
<ul style="list-style-type: none"> • Support navigation through learner-centered suggestions 	<ul style="list-style-type: none"> • Adaptive navigation support through annotating concept name links using different colours (to suggest concept to study) or hiding of links to additional materials which are not needed 	<ul style="list-style-type: none"> • Cognitivism • Cognitive load theory 	<ul style="list-style-type: none"> • Using traffic light metaphor for multi-representations of concept names, according to the learner's knowledge level; • Dynamic hiding of additional material links, based on previous usage of additional materials
<ul style="list-style-type: none"> • Support the learning of content at an appropriate level of difficulty for each learner 	<ul style="list-style-type: none"> • Adaptive presentation support, suggesting the appropriate level of difficulty and presentation style of a concept's content for a learner 	<ul style="list-style-type: none"> • Cognitive load theory 	<ul style="list-style-type: none"> • Multi-representations of the content, through suitable combination of re-usable learning objects, for each learner's level of knowledge
<ul style="list-style-type: none"> • Support interaction with the system without obstructing the learner 	<ul style="list-style-type: none"> • Support the non-obstructive collection of learner model information 	<ul style="list-style-type: none"> • Cognitive load theory 	<ul style="list-style-type: none"> • Non-disruptive collection of learner model information, as the learner carries out regular activities, such as during registration at the beginning, taking a test or selection of content to study • Consistent way of carrying out adaptation to avoid learner frustration

Table 3-14 : Content for the problem solving support system

Requirement	Design Principle	Theory	Learning Content
<ul style="list-style-type: none"> • Enable learners to be in charge of their learning 	<ul style="list-style-type: none"> • Provide the learners with opportunities for active involvement in their problem solving through decision making, by providing various supports 	<ul style="list-style-type: none"> • Cognitivism • Constructivism 	<ul style="list-style-type: none"> • Avoid telling learners how to solve problems • But provide structures to scaffold them e.g. providing OOP concepts before the learners engage in problem solving, general guides such as the problem solving process, task schedule and question prompts
<ul style="list-style-type: none"> • Provide informative feedback 	<ul style="list-style-type: none"> • Provide learners with different types of messages to tell them how they are doing 	<ul style="list-style-type: none"> • Constructivism 	<ul style="list-style-type: none"> • Sub-tasks in problem solving shown in different colours to remind the student about the status of different sub-tasks (using a color key)
<ul style="list-style-type: none"> • Reduce the learners cognitive load 	<ul style="list-style-type: none"> • Provide system features which make it easy for the learner to manage task information and control problem solving 	<ul style="list-style-type: none"> • Cognitive load theory • Constructivism 	<ul style="list-style-type: none"> • An online journal (task schedule) and question prompts to provide a structure for thinking about problem solving processes and controlling problem solving (through application of elements of metacognition)
<ul style="list-style-type: none"> • Provide the possibility for construction of new knowledge and elaboration of existing knowledge 	<ul style="list-style-type: none"> • Activation of prior knowledge and encouraging linking of new information to pre-existing knowledge 	<ul style="list-style-type: none"> • Constructivism • Cognitivism 	<ul style="list-style-type: none"> • Learners required to identify knowledge and skills required for each sub-task during the planning phase • Learners required to explain their solutions (justifications)
<ul style="list-style-type: none"> • Provide authentic challenges to the learners 	<ul style="list-style-type: none"> • Give the learners authentic problems to do to enable them think and act like experts in their field of practice 	<ul style="list-style-type: none"> • Constructivism - Authentic learning • Cognitive load theory 	<ul style="list-style-type: none"> • Ill-structured problems set in real-life contexts
<ul style="list-style-type: none"> • Supports tailored to learner needs 	<ul style="list-style-type: none"> • Support provided according to the learner's traits and stage of problem solving to make support meaningful 	<ul style="list-style-type: none"> • Constructivism • Cognitive load theory 	<ul style="list-style-type: none"> • Question prompts designed for each stage of problem solving and made relevant to OOP • The interaction between the learner and the task schedule depended on progress and metacognitive traits

Table 3-15 : Content for the general interface system

Requirement	Design Principle	Learning Theory	Learning Content
<ul style="list-style-type: none"> Ease of access of important information on a lesson web-page 	<ul style="list-style-type: none"> Important information to be accessed straight away, may be in a hurry, and perhaps repeatedly, should be placed at the top of the page 	<ul style="list-style-type: none"> Minimalist theory Cognitive load theory - sequencing 	<ul style="list-style-type: none"> Place important information near the top: task schedule; links to login and logout pages; access to goals; tabs for selecting a suitable level of abstraction (difficulty) of content; average score per goal and per concept
<ul style="list-style-type: none"> Keeping the pages easy to focus on 	<ul style="list-style-type: none"> Keeping the pages uncluttered by avoiding unnecessary elements 	<ul style="list-style-type: none"> Minimalist theory Cognitive load theory 	<ul style="list-style-type: none"> Using top down design, with drilling down allowed from a goal to its concepts and from a concept to its learning objects Indentation and different formatting styles for different types of learning objects e.g. examples, exercises and theory presentation. Consistently using different font sizes to show different types of objects on a lesson page
<ul style="list-style-type: none"> The display should not compete with or obscure the central message 	<ul style="list-style-type: none"> Keep frames simple and observe consistency in design of text, graphics and sound to limit cognitive overload. 	<ul style="list-style-type: none"> Minimalist theory Cognitive load theory 	<ul style="list-style-type: none"> The page frames were kept simple and consistent using Cake PHP framework. The display of the messages was distinct through choice of format and presentation style
<ul style="list-style-type: none"> Control of other sites linked to the course 	<ul style="list-style-type: none"> Ensure that links were only established to sites which had useful content and which were well maintained, stable and available 	<ul style="list-style-type: none"> Minimalist theory 	<ul style="list-style-type: none"> Links to additional materials were selected carefully to supplement the content of a particular concept so as to avoid unnecessary web browsing The links were regularly checked to ensure they were not broken

The process of information design had three outputs:

- Design of the domain model of the system from instructional design
- Design of the interaction model, to enable adaptive and general user interface
- Scaffolding model for ill-structured problem solving.

Together, these three models constitute the information design of the website. These three models were used as input for the next process i.e. navigation design.

3.2.3 Process 3: Navigation Design

During navigation design, the conceptual structure of the website is described. Navigation design also models how users belonging to the different audience classes i.e. user roles and sub-roles can navigate through the website (De Troyer, 2001).

A navigation track is created for each class of users. Together, the different navigation tracks constitute the navigation model of the website.

The inputs for navigation design were the outputs from scenario design i.e. system functionality and information design i.e. the system's information.

Components, Links and Navigation Tracks

In describing a navigation model, three important elements are used: (i) the components, (ii) links and (iii) navigation tracks.

- (i) **Components** - A **component** represents a unit of information or functionality.
- (ii) **Links** - Links are used to join different components to form a navigation model. Models are critical since they not only show the structure of the website but also indicate the needed navigation paths (De Troyer, 2001).
- (iii) **Navigation tracks** - A navigation track is created for each of the audience class. The tracks are linked according to the sub-link structure of the audience class hierarchy, shown in Figure 3-6. The audience class hierarchy is developed based on the functional and information requirements of the different categories of users. The child classes down the hierarchy have requirements and attributes found in the classes above them, but also contain more details which make the members of these classes different and more specialized than members of the classes higher in the hierarchy. Functional requirements were specified during scenario design and information requirements during information design.

Different **components** of an audience class are joined using **navigation links** to form a navigation track for that class. The **navigation tracks** for the audience classes in the navigation hierarchy are linked to form the **navigation model** for the system.

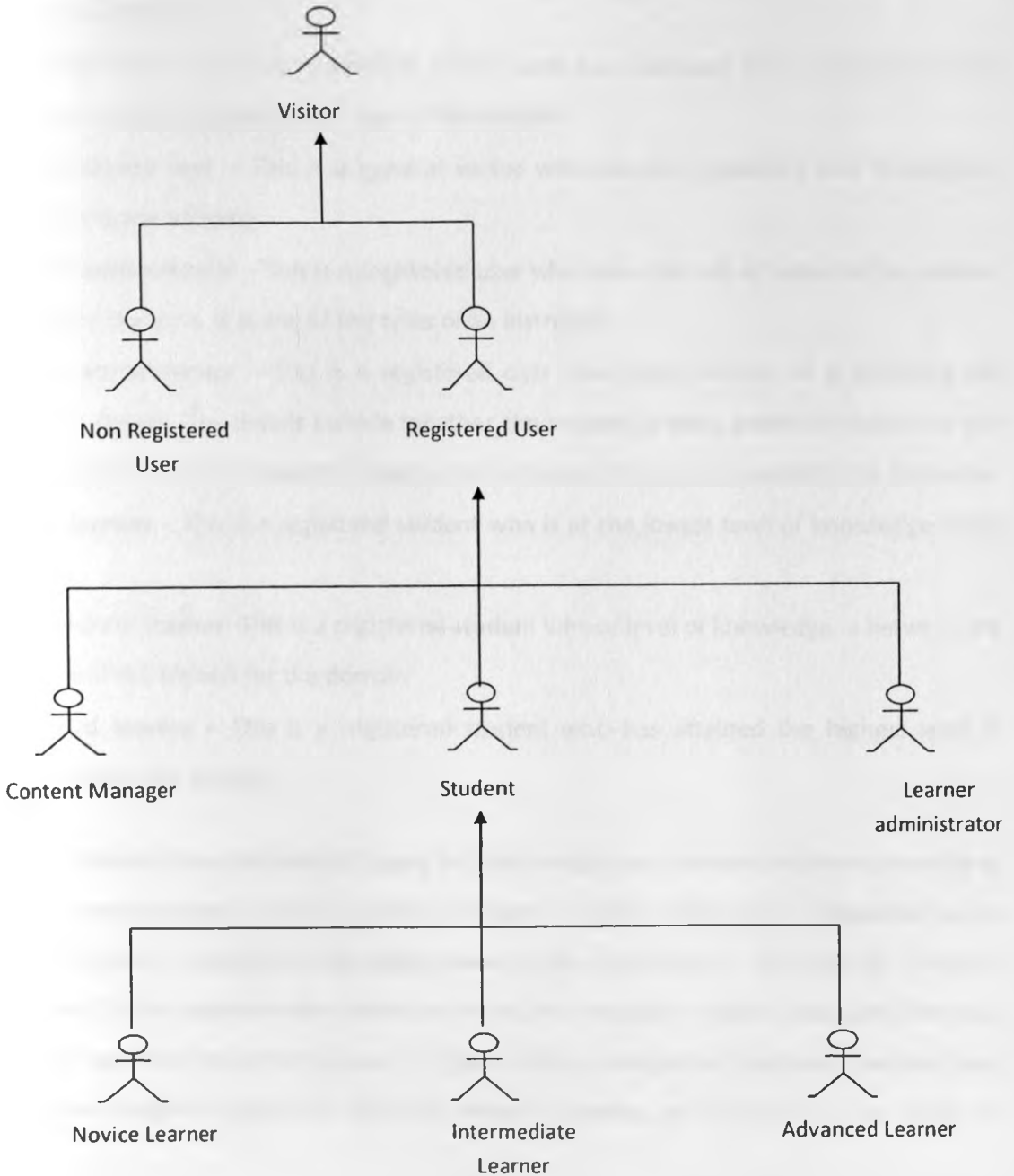


Figure 3-6 : General audience class hierarchy for the learning system

The audience classes in the navigation design shown in Figure 3-6 are as follows:

- **Visitor** – This refers to general user of the website whose requirements represent the common requirements for all visitors to the website. This is the user who pays a passing visit to the website
- **Registered user** – This is a general visitor who has provided some personal profile information to be registered as a user of the website
- **Non-registered user** – This is a general visitor who has not registered and therefore is unknown to the website
- **Content administrator** – This is a registered user who plays the role of updating the content studied by students. It is one of the roles of an instructor
- **Learner administrator** – This is a registered user who plays the role of maintaining the student’s details. The details include whether the student is using adaptive features or not and whether or not the student is using a task schedule. This role is played by the instructor.
- **Novice learner** – This is a registered student who is at the lowest level of knowledge of the domain
- **Intermediate learner**- This is a registered student whose level of knowledge is between the lowest and the highest for the domain
- **Advanced learner** – This is a registered student who has attained the highest level of knowledge of the domain

From the audience class hierarchy in Figure 3-6, the navigation structure for the main tracks of the system was designed, and it is shown in Figure 3-7. Since part of the navigation for an audience sub-class is modeled at the higher level of the superclass, a “hierarchical” structure that is similar to the audience class hierarchy among the navigation tracks is designed. For each class in the audience hierarchy (shown in Figure 3-6), a navigation track was designed and linked to the navigation tracks of the other classes according to the hierarchy (as shown in Figure 3-7).

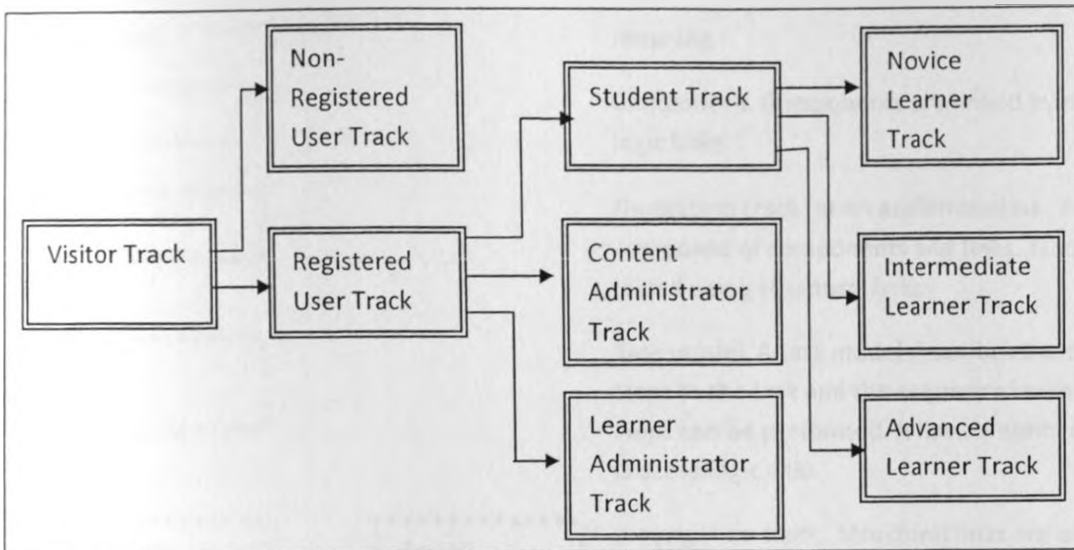


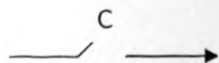
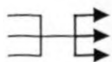
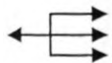
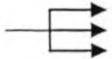
Figure 3-7 : Main track structure based on the general audience hierarchy

Steps of Navigation Design

1. Make task model for each task, using components and process logic links
2. Make audience track for each audience class, by combining the task models of an audience class using structural links
3. Compose the audience tracks into a conceptual structural model, using structural links, according to the audience class hierarchy
4. Make audience chunk model
5. Add navigational aid links if necessary

The symbols used for navigation design and their meanings are shown in Figure 3-8.

Notation



Meaning

Component. Components are linked by process logic links

Navigation track for an audience class. A track is composed of components and links. Tracks are joined using structural links

Task model. A task model describes the different steps in the task and the sequence in which the steps can be performed. It has components and process logic links

A navigation track. Structural links are used to link task models

One -to-one unidirectional link

One-to-one bidirectional link

One-to-many unidirectional link

One-to-many bidirectional link

Many-to-one unidirectional link

Many-to-one bidirectional link

Many-to-many unidirectional link

Conditional link

External component

Figure 3-8 : The notation used for navigation design

Step 1 of 5: From scenario design, make a model for each task using components and process logic links

Navigation models were designed for the following user tasks:

- a) Task: Logging into the system
- b) Task: Selecting concept to study
- c) Task: Taking pre-requisites and concept evaluation test
- d) Task: Studying a concept by a novice learner
- e) Task: Studying a concept by an intermediate learner
- f) Task: Studying a concept by an advanced learner
- g) Task: Use of additional learning materials
- h) Task: Taking concept evaluation test
- i) Task: Preparation for problem solving
- j) Task: Actual problem solving
- k) Task: Evaluating the solution

Sample navigation models for the tasks “selecting a concept to study”, “studying a concept by an advanced learner” and “actual problem solving” are provided in Figure 3-9, Figure 3-10 and Figure 3-11 respectively. The meanings of the components used in the models are as defined in Figure 3-8.

(i) Task: Selecting a concept to study

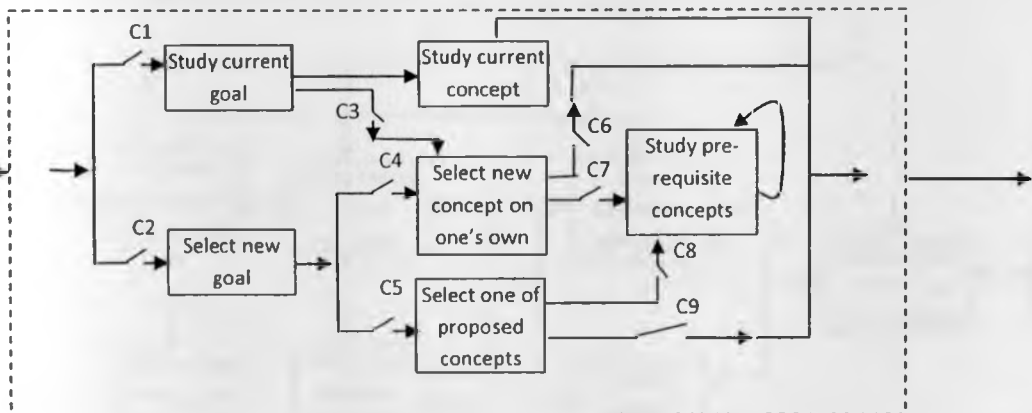


Figure 3-9 : Task model for selecting concept to study

Conditional links

C1 – The learner decides to continue studying the current goal

C2 – The learner does not want to continue studying the current goal

C3 – The learner decides to select a new concept from the current goal

C4 – The learner decides to select a new concept on his own from the new goal, ignoring the system’s suggestions

C5 – The learner decides to select one of the concepts of the new goal suggested by the adaptive navigation feature

C6 – After selecting a new concept on his own, the learner decides to go on studying without studying pre-requisites first. This is perhaps because the colour of the concept is orange i.e. the learner is ready for it or green i.e. the learner is expected to do very well in it

C7 – After selecting a concept on his own, the learner decides to study pre-requisites for the concept, perhaps because the colour of the concept is red i.e. the learner is not ready for it

C8 – After deciding to study one of the proposed concepts, the learner decides to study pre-requisites. This is perhaps to make sure he is ready for the current concept

C9 – After deciding to study one of the proposed concepts, the learner decides not study pre-requisites first, perhaps trusting the system’s judgment of his entry level knowledge for the concept

(ii) Task: Studying a concept by an advanced learner

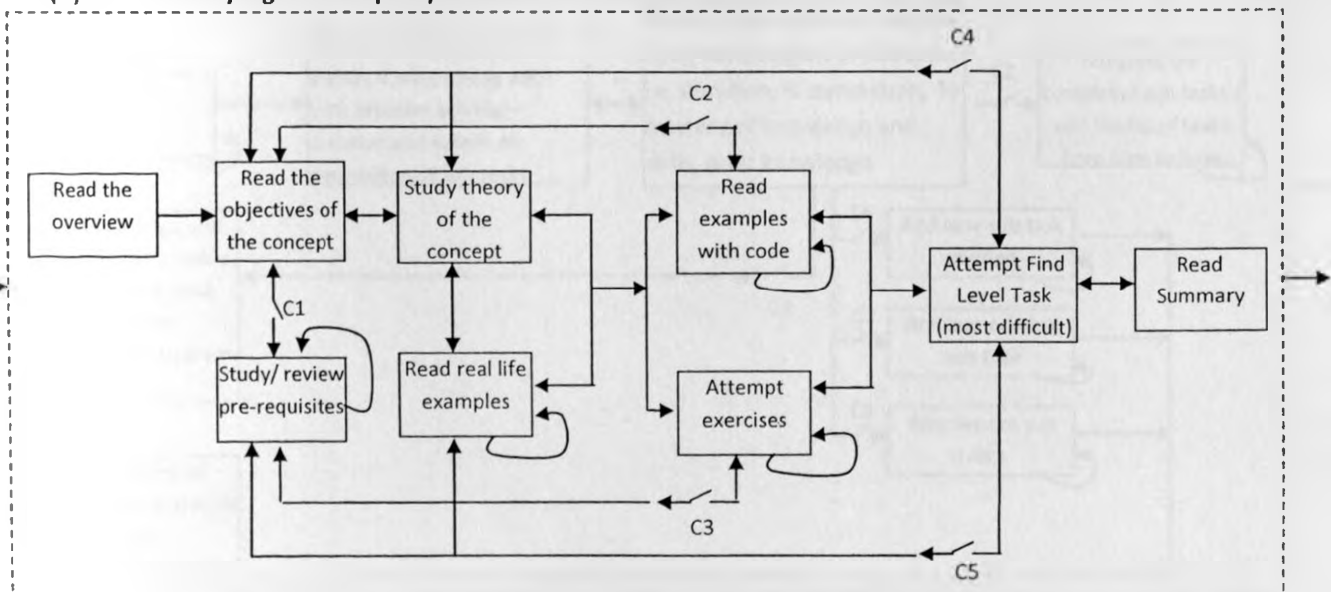


Figure 3-10 : Task model for studying a concept by an Advanced learner

Conditional links

C1 – The learner decides to study pre-requisites or review them (pre-requisites) if he has already studied them but is not feeling confident before studying the content of the concept

C2 – The learner decides to go back from reading examples with code to review the learning objectives of the concept.

C3 – The learner decides to go back from attempting exercises to review pre-requisites for the concept.

C4 – The learner decides to go back from attempting the find level task to review the theory for the concept and/or the learning objectives for the concept

C5 – The learner decides to go back from attempting the find level task to review the real life examples and/or to review the pre-requisite concepts

(iii) Task: Actual problem solving

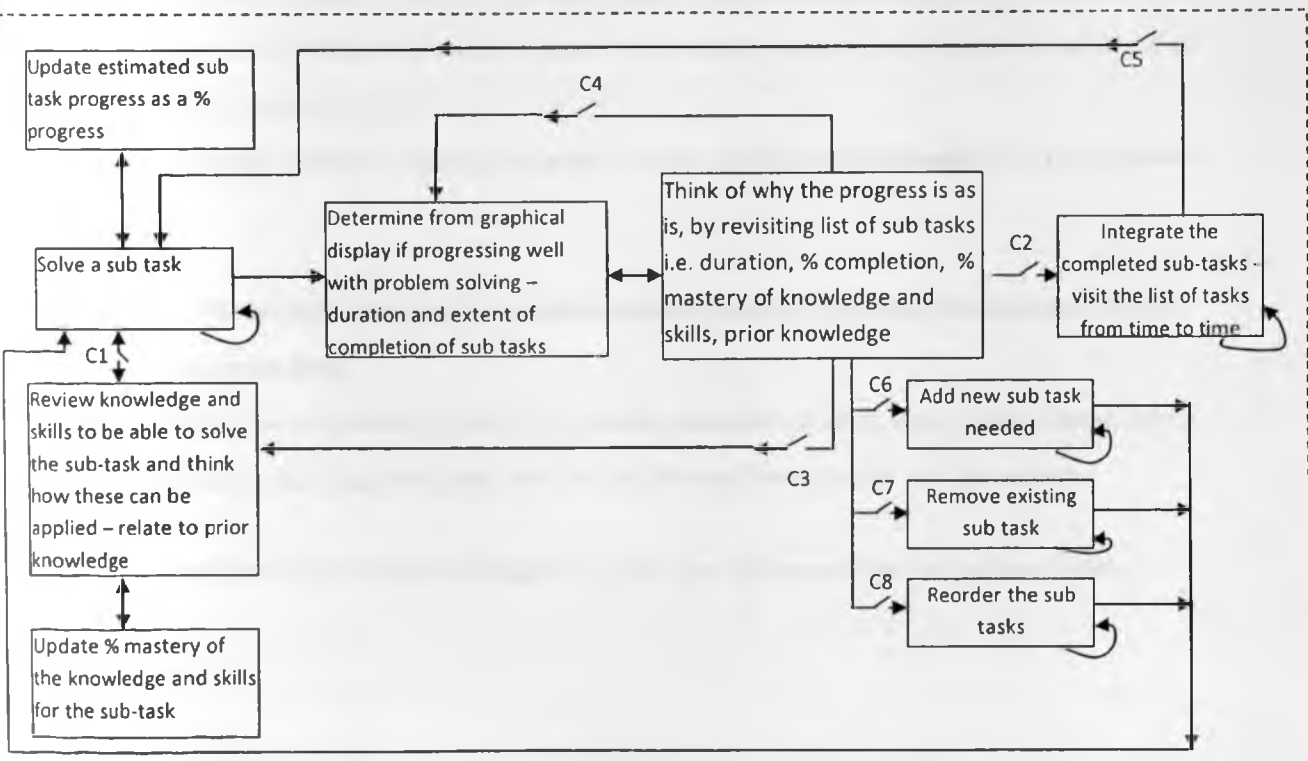


Figure 3-11 : Task model for actual problem solving

Conditional links

C1 – The learner feels the need to improve his knowledge and skills i.e. improve extent of mastery or acquire new ones, in order to be able to fully solve the sub-task

C2 – The learner feels that progress in problem solving is good

C3 – The learner feels that progress in problem solving is not good and he needs to improve his mastery of relevant knowledge and skills

C4 – The learner feels that progress in problem solving is not good and probably he needs to improve his pace and time spent on the task. The graphic shows rate of progress

C5 – The learner decides to log into the task schedule from time to time to remind oneself of progress in the different sub tasks, in terms of time spent and percentage progress achieved, as a way of monitoring oneself.

C6 – The learner feels the need to add new sub tasks which are necessary for successful solving of the problem

C7 – The learner feels the need to remove some tasks which were initially introduced but which need not be solved in order to solve the overall problem, or are subsumed in other tasks whose progress is good

C8 – The learner feels the need to reorder the sub tasks as understanding of the problem increases

Step 2 of 5: Make audience track for each audience class by combining task models of the class using structural links

Navigation tracks formed in step 1 for the tasks of an audience class were consolidated into a navigation track for the class. This was done for all the audience classes for the system.

A sample navigation track is shown in Figure 3-12 for the Advanced Learner audience class.

(a) Audience Track for the Advanced learner class

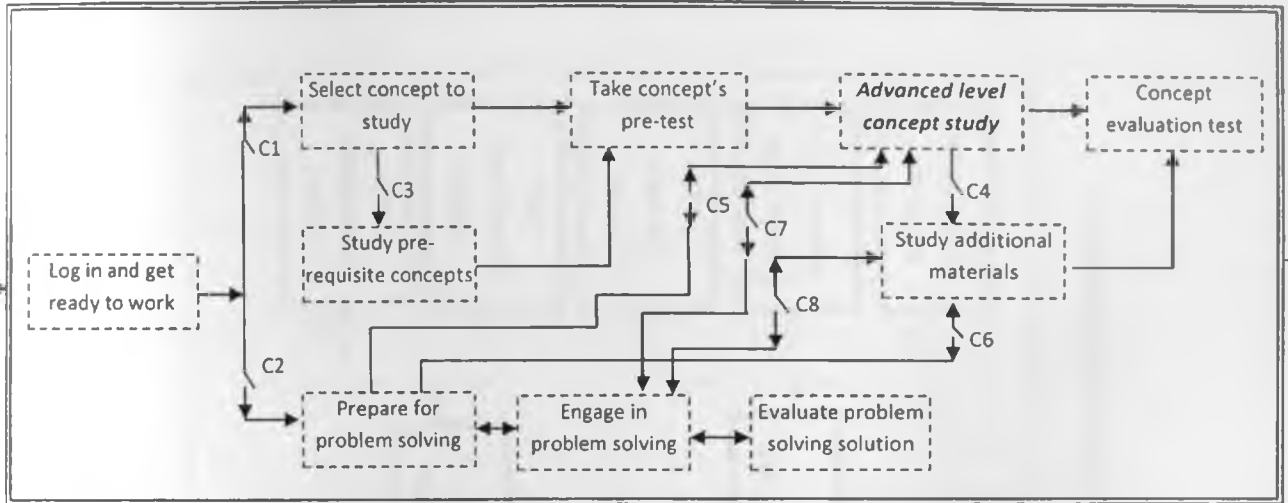


Figure 3-12 : Advanced Learner Audience class track

Conditional links

C1 – The learner decides to study domain knowledge first

C2 – The learner decides to commence problem solving straight away

C3 – The learner feels the need for studying pre-requisite concepts first

C4 – The learner decides to consult additional study materials while studying OOP concepts

C5 – While in the early stages of problem solving, the learner feels the need to consult OOP concepts for skills and knowledge

C6 – The learner decides to consult additional study materials while in the early stages of problem solving

C7 – While problem solving, the learner feels the need to consult OOP concepts for skills and knowledge

C8 – The learner decides to consult additional study materials when well into problem solving

The same approach was used for the other audience classes. It should however be noted that for the web administrator, other tools were used for work (PHPMYAdmin) and therefore, there was no structure design done for it.

Step 3 of 5: Compose audience tracks into conceptual structural model using structural links following the audience class hierarchy

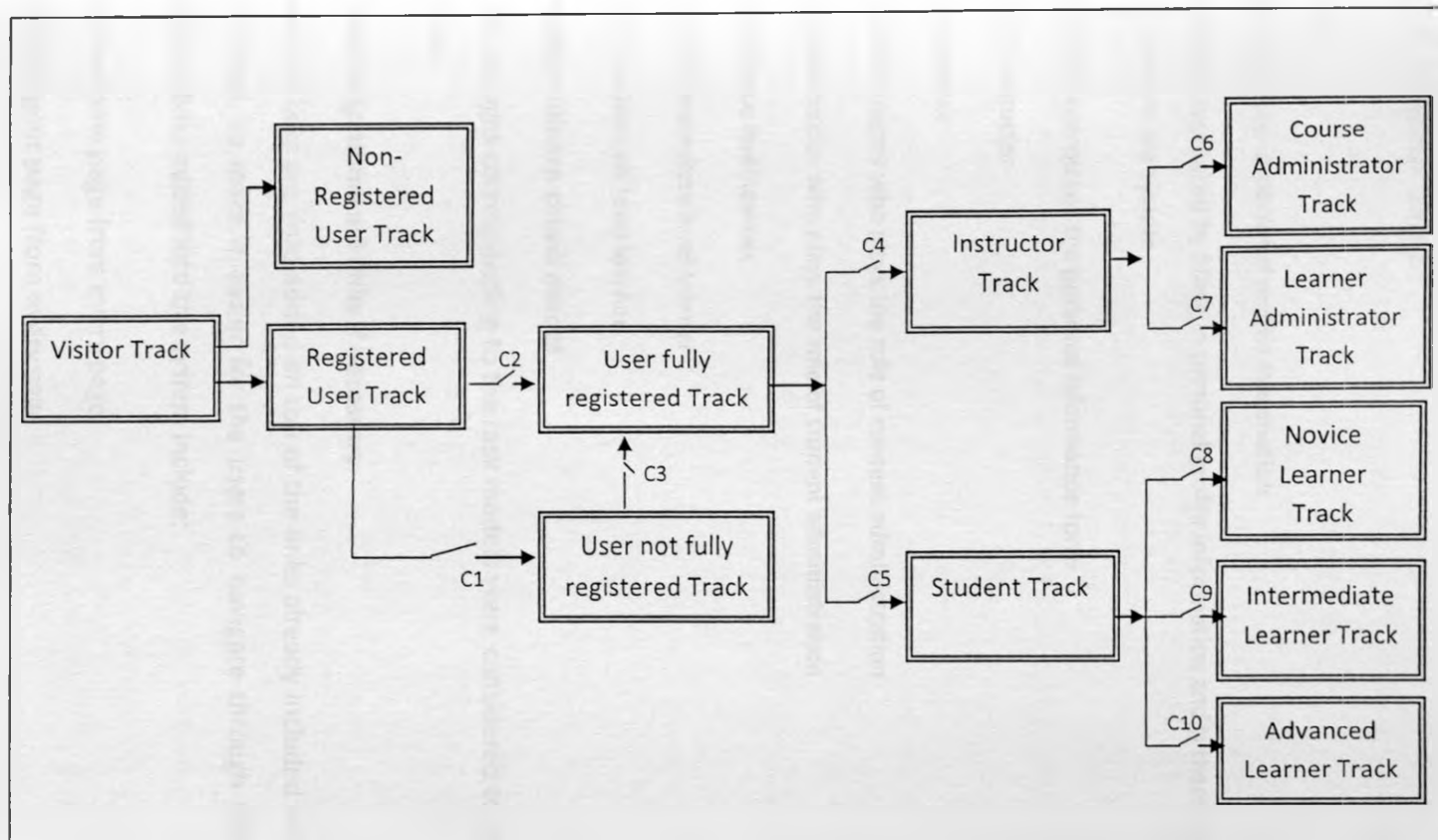


Figure 3-13 : Conceptual structural model using structural links following the audience class hierarchy

As shown in Figure 3-13, the navigation tracks for all the user roles, which have also been defined as audience classes, are combined in order to realize the conceptual navigational structure of the entire system for all users. The links C_1 to C_{10} are all structural links joining the various audience navigation tracks.

Conditional links

C1 – The user has not filled in personal profile information

C2 – The user has fully registered by filling in personal profile information and is therefore ready to carry out expected activities in the system

C3 – The user has fully completed the personal information form

C4 – The user is an instructor

C5 – The user is a learner

C6 – The user is an instructor who plays the role of content administration

C7 – The user is an instructor who plays the role of student administration

C8 – The user is a novice level learner

C9 – The user is an intermediate level learner

C10 – The user is an advanced level learner

Step 4 of 5: Make audience chunk model

The scenario designs corresponding to the task models were considered to be the audience chunk models.

Step 5 of 5: Add navigational aid links if necessary

Navigational aid links are links added on top of the links already included in the conceptual structural model, to make it easier for the users to navigate through the system. The navigational aid links added into the system include:

- Link to the home page from every page
- Link to the logout page from every page

- Link to My Task Schedule from every page of the student tracks
- Link to Personal Information from every page. The personal information page also contained a link for logging out of the system and changing the user password
- Link to goals page from every page of the student tracks to make it easy and shorter for the students to be reminded of and to select concepts to study

Other considerations during navigation design:

- **Clear labels indicating the purpose of the links** - This was to enable the users to scan the links and quickly identify those which they needed to follow.
- **Provision of feedback to tell the users where they are in the site.** Some of the ways through which this was achieved include:
 - i. Displaying the name of the current goal and the current concept (being studied) at the top of the page
 - ii. The colour of the link on the name of concept being studied also changed from blue to purple
 - iii. In the navigation menu for a concept, the currently selected item (object) was marked with a red tag to indicate the current position of the user
- **Consistent use of navigation elements.** Users get to expect to again see links they have seen previously looking the same in terms of colour, font and font size, in the same location, and functioning in the same way. The purpose of this is to ease the process of link recognition by the users.

Consistency contributes to the reduction of cognitive load while using the system and contributes to the users completing their tasks in good time.
- **Provision of persistent links to the home page and other high-level page groupings.** These persistent links when included in every page enable the users to easily navigate from one part of the site to the other. These links should be those which are accessed

frequently, such as home page, logout, add users, link to the goals to be studied, or links to other pages supporting frequently executed functions.

End of conceptual design and beginning of implementation design

So far, a high conceptual level design has been realized for

- i. the functions to be performed with the system (scenario design)
- ii. the content to be handled by the system (information design)
- iii. how users are to navigate through the system (navigation design)

The next challenge is to specify the system in a way that leads to the realization of the desired

- i. Website-pages (page design)
- ii. Navigation and relationship among pages of the website (structure design)
- iii. Presentation of content on pages (presentation design)
- iv. Database or some other structure that is to be used to hold data (data design).

3.2.4 Process 4: Page Design

Page design involves the partitioning of information into chunks of the right size. A page must have at least one component and a number of links. The links may be to other parts of the same page or to other pages. The links can be set from text, images or even using image maps.

Page design is achieved by dividing the components identified during navigation design into logical groups of related components, together with the process logic links joining them.

The input for page design was the navigation design model.

The following steps were followed:

- 1) **Review of user paths.** The focus was on the different audiences of users and their navigation. The user navigation paths, as identified and designed during navigation design, were reviewed.

- 2) **Review of the information requirements.** The information requirements of the users were reviewed according to the different stages of navigation and the functions the users would be carrying out at the time.
- 3) **Formation of the pages.** This was achieved by breaking down the components, links and information into groups of logically related components, process links and information items. The pages were either for a component i.e. knowledge module or for a lesson, combining a number of knowledge modules. Examples of component pages include pages for overview, objectives, pre-requisites and summary. The lesson pages contained all components needed for a complete lesson for the different categories of learners. A sample lesson page is shown in Figure 3-14. At the top, it shows the current goal and current concept. All concepts for the current goal are shown on the left hand side. For the currently selected concept, the figure shows overview, objectives, pre-requisites, summary and links to all its other knowledge modules.

Other components of information were included, such as pop-up windows in the test pages to provide information on number of attempts and duration of time left for a test.

Links were provided on the left hand side of the page to take the learners to other components. At the bottom of the lesson page, there were links to additional material pages.

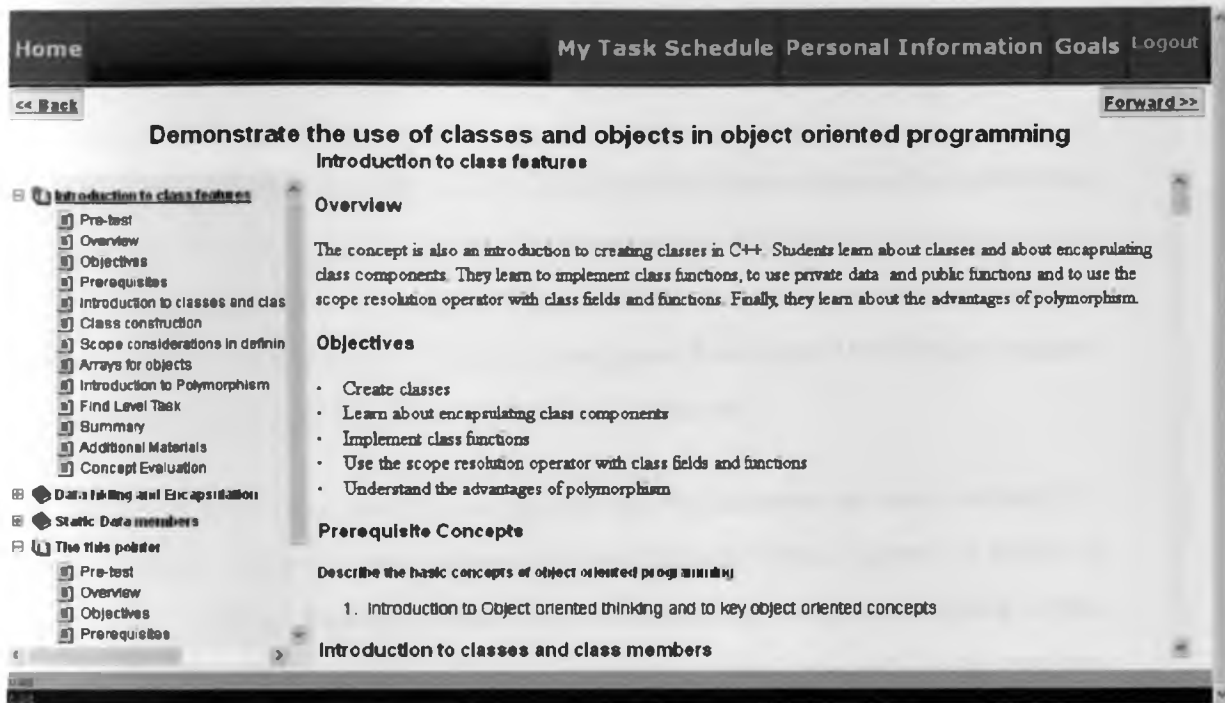


Figure 3-14 : Components of a lesson page

3.2.5 Process 5: Structure Design

It is important to come up with a structure for information in the learning environment to enable the learners to access the information they need or to perform tasks which are needed as part of their learning experience. Structure design involves specification of the presentation and storage designs of the learning environment.

The inputs for this process included the navigation and page design for the different classes of users and the scenarios (functional design) and information design from previous processes.

The following steps were followed:

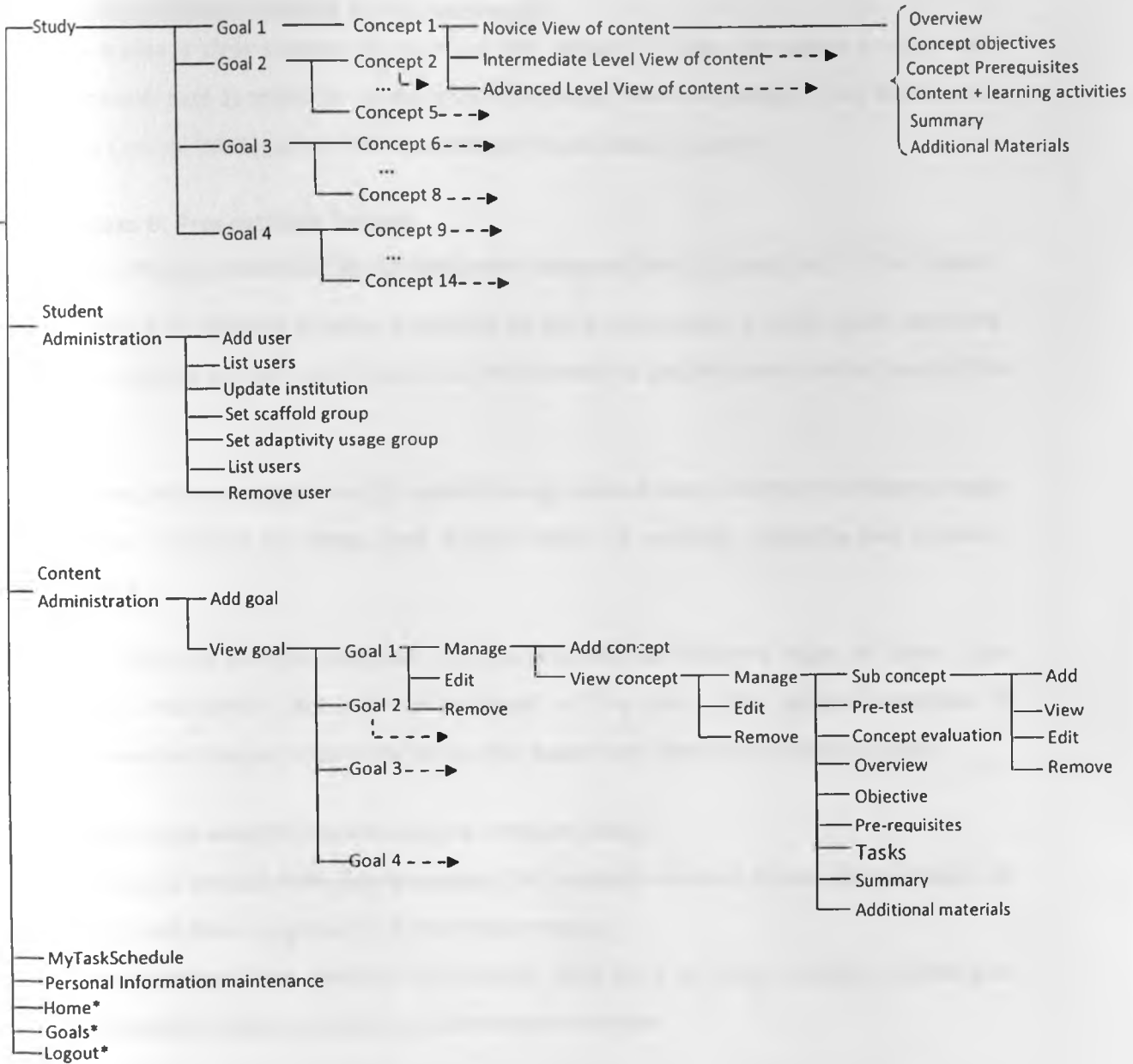
- 1) **Defining key user paths** through the system for the different classes of users. This was mainly done from the navigation design and scenario design. The purpose was to organize information in ways which make sense to the users to enable them find different types of information and to complete certain specific tasks. For each of the audience classes, it was considered necessary to provide top-down access to content

and other aspects of the instructional system. This was naturally supported by the use of hypermedia (Hoffman, 1997).

- 2) **Specifying presentation structure.** The presentation structure used was the hierarchical presentation structure, used alongside the hypermedia structure. For each audience class, the top most links were displayed first, and the lower links could be obtained by drilling down from the top-level links e.g. from goals→concepts→knowledge modules (overview, learning objectives, theory, examples, and so on).

The design of the course content was guided by the outcomes of user analysis and scenario design. The design was presented in the form of a flow diagram as shown in Figure 3-15. The flow diagram defined the structure of the website, identified all the pages of the site, and showed the links to each page. It was very useful in guiding the development of the website. To the left hand side of the diagram are higher order links for the website. The home page is the link to the extreme left of the diagram. The links to the right in the diagram are sub-ordinate links which lead to more detailed information.

Some pages were added at the top level page, such as content administration and student administration used by the instructor, a link to the home page, the log out page, the page for viewing and adjusting personal information, the page for the student's task schedule and the page with the course goals.



KEY * Represents Navigation aids
 ---> Paths to pages which are not shown in the diagram

Figure 3-15 : Hierarchical structure of the system

3) Defining storage structure for the web pages

To create a clear storage structure of the website's pages, the pages created were generally held as controller or model or view pages. This is according to the Model, View and Control (MVC) pattern for the creation of web-based systems.

3.2.6 Process 6: Presentation Design

Presentation design consisted of layout design and design of the "Look and Feel" of the system.

Layout design was intended to make it possible for the users to easily and efficiently locate the information needed (content) and the links to be followed to lead the user to other parts of the system.

Look and Feel included aspects such as use of the right colour code, contrast and balanced page layout design, inclusion of menus, text design, extent of scrolling, simplicity and courtesy, among others.

The inputs into this process included the characteristics of different types of users, user requirements, navigation and information needs of the users. The general principles of designing websites that are usable and which give a good user experience were also used.

The following steps were followed during presentation design:

- 1) **Review of outputs from previous steps.** The outputs reviewed include characteristics of users, and their navigation and information needs.
- 2) **Wire framing of the content.** Wire frames were used to design multiple, distinct and independent viewing areas within the browser window.

The use of the wireframe was also intended to maintain a consistent page layout throughout the system.

There was a frame for navigation that was used to select what the user wanted to view. There was also a presentation frame which was the main central frame that was used for viewing the dynamic instructional content selected in the navigation frame.

The wireframe that was used for the main pages of the system is shown in Figure 3-16.

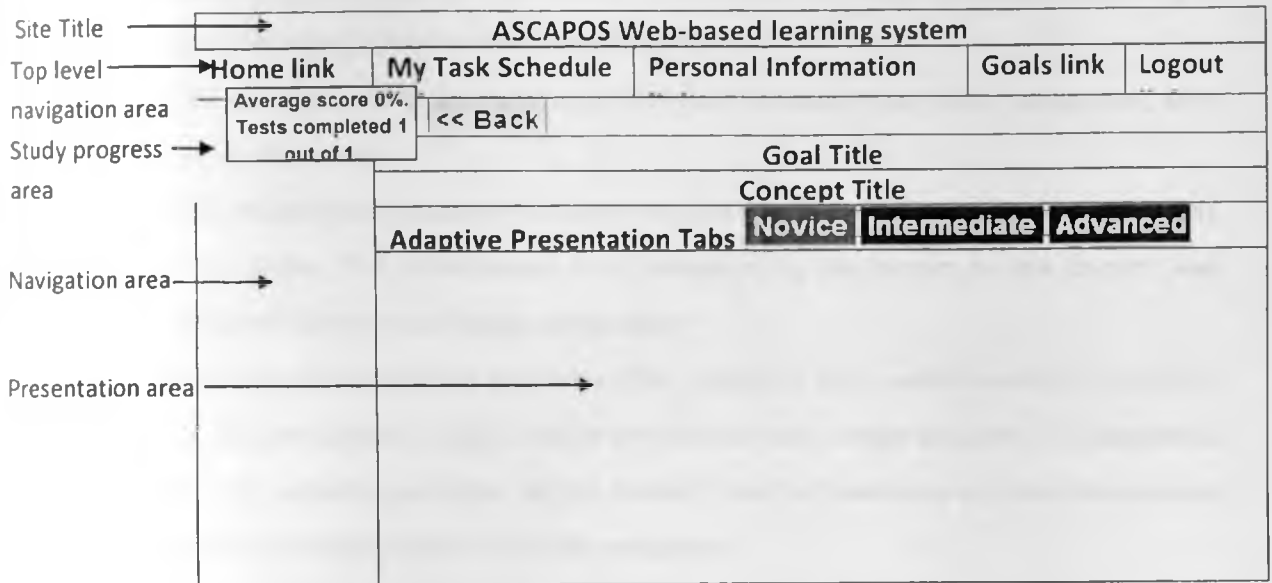


Figure 3-16 : Wireframe used for page layout design

The sections shown in Figure 3-16 were used as follows:

- The **top level navigation area** was maintained throughout the system
- The **navigation area** was used throughout the system to show the menu of the goals and concepts selected
- The **presentation area** was used to show the content of the selected learning object or combination of learning objects
- The titles of the selected goal and concept were displayed at the **same position** throughout the course
- Every **other element** of a page such as the content, link messages, display of student progress information had a place on the page

3) Design the look and feel of the system

Some design decisions were made with regard to the look and feel of the system:

- (i) Considering that this website system was for learning, it was decided to keep things simple and courteous. For example, it was decided to keep the background plain and use black font colour for the presentation area
- (ii) The top level navigation area was designed to have a dark blue background, with white font colour
- (iii) The adaptive presentation tabs were designed to have black background with white font colour. The presentation level proposed by the system to the student was designed to have gray background colour
- (iv) The adaptive navigation support in the navigation area, which involved annotation of links on concept names, was to be coloured red, orange or green. This depended on the system's prediction of the learner's level of readiness to study the concept and was guided by the traffic light metaphor
- (v) The goal title was designed to have font size of 18 (Arial) whereas the concept title was to have a font size of 14 (Arial). The body text in the presentation area was to have font size 11 (Calibri)
- (vi) Information cues, using different colours, were to be used in the display of the tasks scheduled by the learner during problem solving. A key was displayed at the bottom of the page, showing the meanings of the colours i.e. status of the task
- (vii) Navigation cues included the use of the traffic light metaphor to show the learner whether the system considered him ready to study a particular concept or not
- (viii) The links for additional materials were to be separated from the narrative blocks in the presentation area. They were to be displayed at the end of the lesson page to avoid disrupting the flow of studying and understanding by the learner
- (ix) The text for examples and exercises was flushed from the left (i.e. indented), to make the examples and exercises easier to locate
- (x) The areas to the left and top of the page, where users look first on a web page, were to be used for navigation and identification of what the learner was currently studying. Generally, users are comfortable and familiar with this design

3.2.7 Process 7: Data Design

This involved defining the storage of persistent data of the system. The database was designed using MySQL relational database. It was used to hold information about the course i.e. the domain model, the learner i.e. the learner model, and the interaction between the learner and the domain. It also contained information about the use of the task schedule to scaffold self-regulation during problem solving.

3.3 Phase 3: Development

The development phase may also be called the implementation phase of the system (Hadjerrouit, 2007). This was the stage when the prototype of the system was developed.

The phase consisted of the following processes:

- (i) development of the course
- (ii) development of low-fidelity prototype
- (iii) design walk-through for the low-fidelity prototype
- (iv) development of high-fidelity prototype

3.3.1 Process 1: Development of the Course and Related Tools and Items

There were four steps involved in this process: (i) production of course elements, (ii) production of the integrated course and (iii) validation of the course elements and the integrated course.

Step 1) Production of course elements

The elements constituting the course were developed, such as the text material and additional material links. The text material was summarized from a text book recommended by one of the lecturers of the course while also taking into account the scope of the course, arrived at during course analysis. The book was “Object Oriented Programming using C++” by Joyce Farrell (2006). This step also involved identification of links to relevant and appropriate study material from the Web, which could serve as additional material for each of the concepts.

a) The Design of content for the learning system

The course Introduction to OOP was designed through the guidance of Elaboration Theory and Component Display Theory. The course had goals, concepts and knowledge modules.

The content was structured hierarchically, with one goal having several concepts and one concept being presented through several knowledge modules. This is shown in Figure 3-17.

The goals and concepts were generally ordered from simple to complex so as to make it possible for students to start learning more concrete concepts first (Callear, 2000). This contributes to building learner confidence early (Yau, 2004), through the creation of basic cognitive schemata, which reduces cognitive load during learning (Van Marrienoer & Sweller, 2005).

The concepts constituting a goal were also divided into layers, from the easiest to the most difficult. The concepts were to be presented to the student in that order for easier assimilation of the content into their cognitive structures in long term memory (Anderson & Elloumi, 2004; De Villiers, 2002).

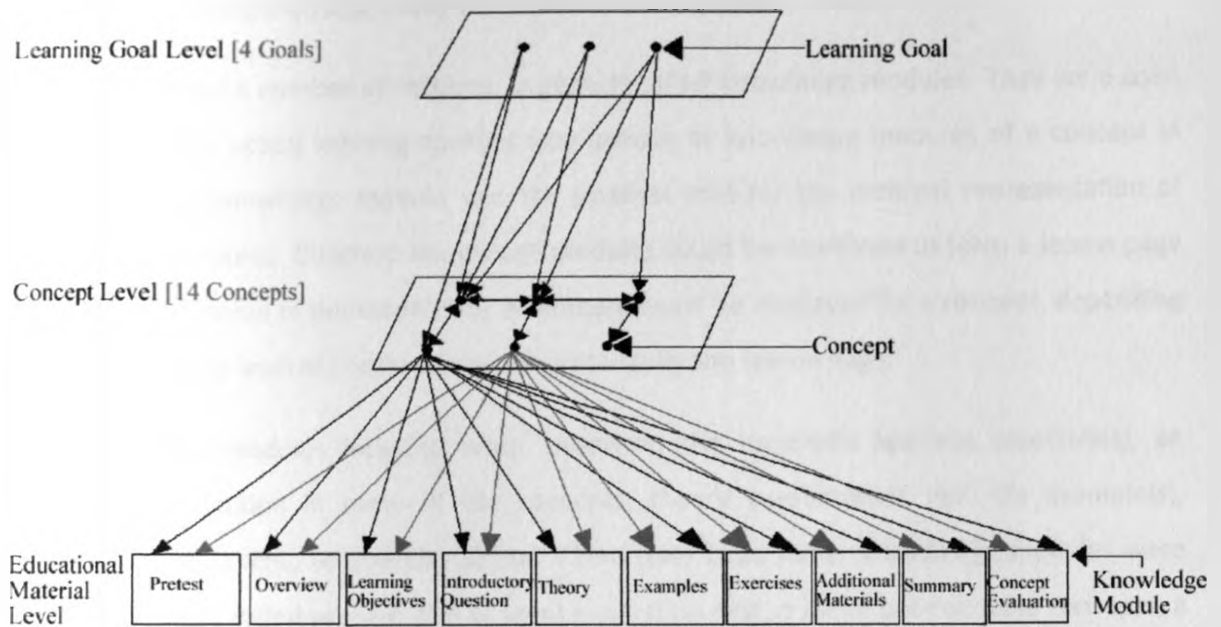


Figure 3-17 : Hierarchical organization of the course into goals, concepts and knowledge modules

Goals, Concepts and Knowledge Modules

The course was organized into 4 learning goals. Each goal had a number of concepts to be learned so as to achieve it. There were 14 concepts in the course, with the first 8 belonging to the Class Design part and the last 6 to the Inheritance part. The goals and their concepts may be seen in Appendix B.

The goals were ordered according to the common textbook order. This order was found to be consistent with the order preferred by specialized instructors of the course “Introduction to Object Oriented Programming” (Yau, 2004).

The concepts for a goal were organized into three layers, with the first layer having the simplest concepts and the third layer having the most complex concepts. Layering according to levels of

difficulty is significantly important in an adaptive learning environment that uses simple to complex sequencing (Kalyuga, 2009; Van Marrienoer & Sweller, 2005).

Each concept had a number of sections, in general called knowledge modules. They were used to represent the actual learning content (see sample of knowledge modules of a concept in Appendix C). A knowledge module was the smallest unit for the external representation of content in this course. Different knowledge modules could be combined to form a lesson page for the learner. Some of the modules or all of them could be displayed for a concept, depending on the knowledge level of the learner who was to study the lesson page.

The knowledge modules included were: overview, the concept's learning objective(s), an introductory question in some of the concepts, theory presentation, real life example(s), application example(s), an exercise(s) and a find level task. Other knowledge modules were assessment tests called pre-test and concept evaluation test, a list of pre-requisite concepts, a summary and additional materials for those interested in more course content. Appendix D shows a sample lesson page for a concept, with its knowledge modules.

b) The design of Tests

I. Pretest

The purpose of the pretest was to evaluate the level of the learner's knowledge and to determine the scope of their learning needs, before the beginning of learning. There was a pretest for each concept.

Each pretest had 6 questions. The first 2 questions were at the novice level of difficulty i.e. the easiest questions, the next 2 at the intermediate level and the last 2 at the advanced level i.e. the most difficult questions. Figure 3-18 shows the first 2 questions of a pre-test for the concept called "Introduction to object oriented programming and to key object oriented concepts".

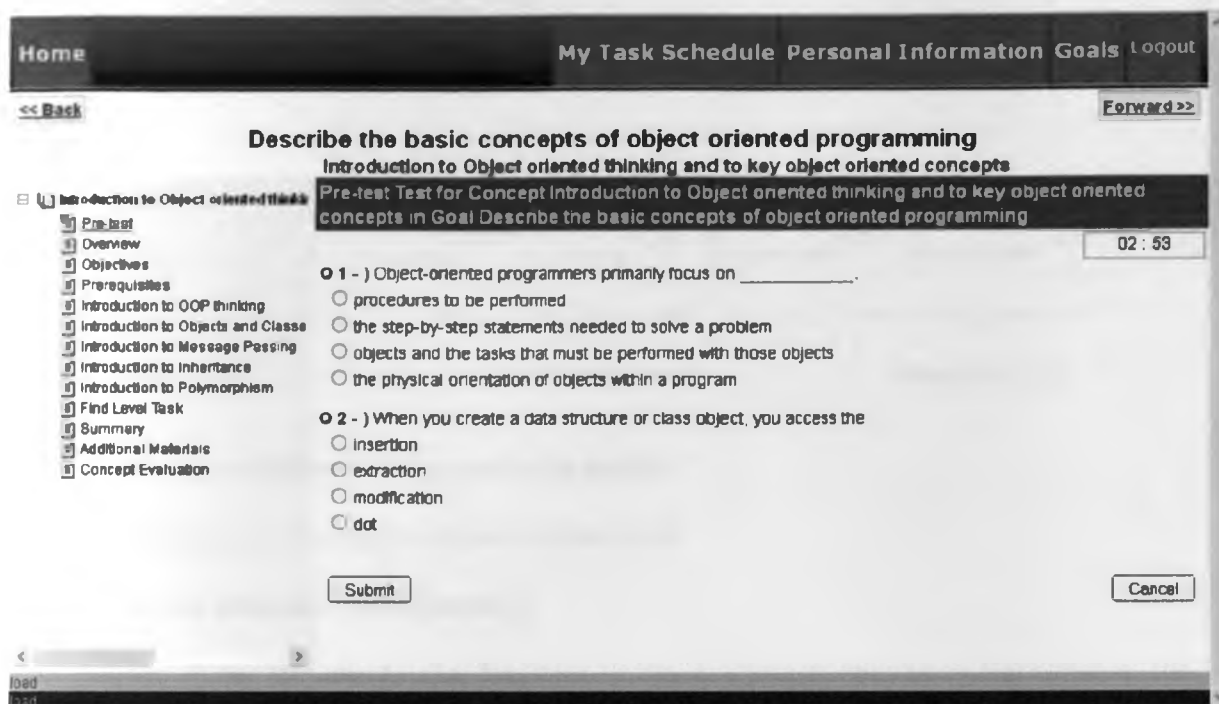


Figure 3-18 : Two questions for the first level of the pre-test for the concept “Introduction to object oriented programming and to key concepts”

Each level of knowledge had two questions in order to take care of situations where a student’s answer to a question is a guess or as a result of a slip, which can reduce the accuracy of the learner model (Virvou & Tsigira, 2004). Only the questions of one knowledge level were displayed at any given time.

The guideline was not to give students, especially novices, too many difficult questions when their level of performance i.e. test scores could be approximated well by less difficult questions (Gouli, Papanikolaou, & Grigoriadou, 2002; Yau, 2004).

The way of presenting the questions is summarized in Figure 3-19. Question 1 and Question 2, which were at the novice level of knowledge, were presented together. If both questions were answered correctly, then Question 3 and Question 4, which were at the intermediate level of knowledge, were presented together. Also, if both questions were answered correctly, then

Question 5 and Question 6, which were at the advanced level of knowledge, were presented together.

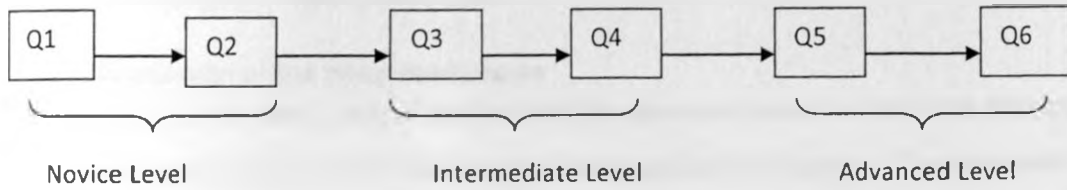


Figure 3-19 : Presentation plan for pre-test questions

A sample pretest for a concept is shown in Appendix E.

II. Concept evaluation test/post-test

The purpose of the test was to give feedback to the students at the end of the concept. The student’s performance in this test was also used to update the learner’s knowledge level in the learner model.

It was intended that the pretest questions be re-used in the concept evaluation, with some variations. The variations include providing the learners with all the questions at once since at this point in the course, they were expected to be more confident. The learners were also allowed one attempt per question instead of two attempts as was done for the pretest, since chances of slipping were expected to reduce after studying the concept. The scores per question were determined according to the scheme in Table 3-16.

Table 3-16: Scheme for awarding marks per question

Type of Question	Marks
Novice	4
Intermediate	5
Advanced	6

Adopted from (Yau, 2004)

This feature is not in the pretest, in which the level of knowledge i.e. novice, intermediate or advanced is determined, and not the level of performance i.e. the actual marks.

The reason for re-using the test this way is that only a few of the students were expected to go beyond the novice level during the pretest because OOP was new to them and is said to be generally challenging for students (Boyle, 2003; Schmidt, 2005; Yau, 2004).

Step 2) Production of the integrated course

The different elements developed or adopted for re-use were combined into the final course, complete with goals, concepts, learning objects and activities for students. The materials were sequenced in a logical and commonly used order to make the course easy to follow.

The ill-structured problems, supports for metacognitive control and the problem solving processes were also put together i.e. the task schedule and question prompts. Details about their design and validation are in section 5.2.5 under instrument and treatment material development.

Step 3) Validation of the course elements and the integrated course

The course content together with the two assessment tests to be done after concept 8 and concept 14 were validated by four lecturers. Two lecturers were also asked to validate the ill-structured problems to ensure that they were relevant to the course and that they were indeed ill-structured.

3.3.2 Process 2: Development of the Low-fidelity Prototype

- **Paper based prototypes** - At first, sketches were made on paper, also called paper based prototypes, to show how the system would look. The sketches not only assisted in the realization of ideas in the mind, but also enabled discussions among the people involved in development
- **System prototype** - The paper-based prototype was made into a software prototype. This step involved the actual development of the prototype for the system using Personal Home Page (PHP), Javascript, Ajax and MySql

Some of the steps of this process were done in parallel while others were done in iteration so as to include new information. The steps involved were:

- 1) Review of the requirements specification
- 2) Review of features and components
- 3) Integration of data design
- 4) Integration of information content
- 5) Integration of presentation design and page design
- 6) Integration of navigation design and structure design
- 7) Testing

One of the weaknesses of prototyping is lack of a well tested system. The system was tested with the view of discovering and correcting errors, such as typographical errors, formatting mistakes and inconsistencies and grammatical mistakes. Testing also aimed at discovering errors in the course content, errors in graphical representations, broken links or links leading to the wrong destination (navigation errors), and so on.

The testing strategy used for this system was adopted from the work of Hadjerrouit (2006). It involved:

- (i) Review of the content of the system to uncover errors
- (ii) Review of the design model to uncover navigation errors
- (iii) Unit testing of selected components
- (iv) Integration testing
- (v) Testing the system for overall functionality
- (vi) Testing of the system in a variety of environment configurations
- (vii) Testing the system by a controlled and monitored population of end users

3.3.3 Process 3: Design Walkthrough

The aim of the walkthrough was to establish any obvious shortfalls in the system which could affect the usability and user experience of users while using the system.

The focus was on the way different requirements identified during requirement specification and different aspects of conceptual design had been implemented in the low-fidelity prototype.

The inputs to the process were the requirements specification, conceptual design and the low-fidelity prototype. The output was a list of changes which needed to be introduced into the low-fidelity prototype.

The design walkthrough was done by a lecturer of OOP, who also taught the design of web-based applications using Internet technologies and had also been trained on pedagogy. She was assisted by the researcher and the DBA.

The participants handed in and discussed their feedback with the researcher who summarized them down for further discussion with the developer. The summary of feedback from the DBA may be found in Appendix A.

3.3.4 Process 4: Towards High-fidelity Prototype

The following steps were taken to improve the low-fidelity prototype:

- a) **Discussion of the feedback** from the walkthrough with the developer with a view to agreeing on necessary changes to be incorporated into the system.
- b) **Review of the requirements specification and conceptual design** in terms of the feedback to ensure that any changes to be made were in line with the goals of developing the system.
- c) **Improving the low-fidelity prototype** by introducing the necessary changes. This was followed by testing of the system.

3.4 Phase 4: Formative Evaluation

3.4.0 Introduction

For a web-based learning system, formative evaluation involves finding out, in the course of system development, whether the goals and objectives of the system are being achieved instead of waiting until the system is presented to the target audience.

3.4.1 Formative Evaluation of the Sub-systems

Formative evaluation had 3 processes organized according to the system's 3 sub-systems:

- (a) The **instructional system** – It involved evaluation of both the instructional material and the instructional experience of the learner
- (b) The **adaptive user interface system** – It was carried out to ensure that the system provided accurate predictions, that it did not inhibit the learner while using the system, that it was easy to locate what the learner might be looking for and generally that the system is easy to use
- (c) The **problem solving support system** – It involved evaluation of the effectiveness of the various supports for problem solving processes and self-regulation by checking the effectiveness of the interface, accuracy and sufficiency of information provided, and non-interference with learner’s problem solving process.

3.4.2 Those who Participated in the Formative Evaluation of the System

At different stages, formative evaluation involved:

- **Content experts**, also called **subject matter experts**, with experience in teaching OOP using languages such as Java and C++. There were 7 of them
- **Instructional design experts** with experience in design of instruction for e-learning systems. There were 3 of them
- **Content-specific education specialists** trained in pedagogy besides being experienced in teaching OOP. There was 1 such expert
- **Experts in adaptive learning** with experience in the design, implementation and evaluation as well as teaching of Machine Learning algorithms. There were 2 of them.
- **Learners**. There were 37 from station 1 and 22 from station 2

3.4.3 Process 1: Evaluation of the Instructional System

The inputs to the formative evaluation of the instructional system included relevant parts of the requirements specification, features and components, course goals, conceptual design and the prototype.

Static analysis of the instructional system:

a) Structure of the course

i. Aspects considered:

- The structure of the course as organized into goals, concepts and knowledge modules
- The different types of knowledge modules such as pre-test, overview, objectives, pre-requisites, theory presentation, different types of examples, different types of exercises, summary and concept evaluation
- The inclusion of different types of knowledge components such as regular exercise, find level task, real-life example and example with code to suit the needs of different types of learners according to knowledge level
- The breakdown of the course into goals. Was it acceptable to the expert?
- The breakdown of the goals into concepts. Was it acceptable to the course expert?
- The breakdown of concepts into knowledge modules. Was it acceptable to the course expert?
- The objectives. Did they represent the outcomes expected of the students after learning the course?
- The pre-requisites for each of the concepts. Were they provided? Were there relationships among the concepts?
- The content covered. Could it achieve the goal of the course, which was to equip the learner with the subject matter needed for problem solving in the OOP domain?
- The relative difficulty of each concept, as shown by the layer in which the concept is placed i.e. layer 1 (simplest), layer 2 (moderate) and layer 3 (advanced). Was it acceptable?
- The concept map provided on the left hand side of the page. Could it be used to show the way different concepts were related?
- The number of internal and external links per page.

ii. Expert Review

Four subject matter experts who had been trained in pedagogy examined the structure of the course. The complete list of experts involved in the research is shown in Appendix F.

Three other experts were provided with accounts in the system and reviewed, among other things, the structure of the content. One of them was a subject matter expert, the second one a content specific education specialist with expertise in instructional design for e-learning and the third one an expert in designing instruction for e-learning.

Suggestions

- To remove Friends part from the course because it was not part of pure OOP concepts. It only applied to C++
- The objectives needed to be re-stated in clear terms to enable the learners understand the performance outcomes expected of them at the end of the concept
- Rename the goals into topics to make them resemble the terminology normally used in text books, as well as reducing the effort and time the learners take to get used to the environment

iii. First Field Trial

Expert review was followed by a field trial at SCI between February and March 2008. It involved second year students studying the OOP course. 37 students participated and at the end, they completed a survey.

Suggestions

Out of 37 participants, 17 of the students liked the organization of the course. 12 of them liked the features included. Nonetheless, they raised the need for:

- More material in some of the concepts such as the “this pointer” concept
- They raised the need for more examples, especially examples with code
- They raised the need for more additional material links
- They raised the need for more detailed, authentic examples

b) Integrated assessment

i. Aspects considered

- Pre-test and concept evaluation for each concept
- Class Design and Inheritance paper based tests
- Regular and Find level exercises
- The assessments were reviewed in terms of variety of the questions they contained i.e. whether they covered all the levels of Bloom’s taxonomy of cognitive objectives
- The depth of the questions in terms of simple, more advanced and advanced questions
- Coverage in terms of whether the critical parts of the concept were captured in the test
- Sufficiency in terms of whether the questions were enough
- Whether the test questions corresponded to the content covered.

ii. Expert Review

Four subject matter experts with training on pedagogy participated in the review of the pre-test, concept evaluation test and the two post-tests for Class Design and Inheritance sections of the course.

Four other experts also participated in the review of assessment integrated into the course. One was a subject matter expert, the second one both a content specific education expert and an instructional designer for e-learning systems, the third one was also an expert in instructional design for e-learning systems and the fourth one an expert in adaptive learning with experience in the design of learning systems.

Suggestions

- The questions did not cover all the levels of cognitive objectives according to Bloom's taxonomy. Therefore, the tests were improved by removing the True/False section and replacing it with a section requiring analysis, synthesis and evaluation skills (see Appendix H for the Class Design test).

- The feedback provided was not enough and there was need to inform the learners of the questions in which they did not do well and the concepts in which they may need remediation
- Some of the questions were designed with words which gave a clue of the answer to the learners, especially for the pre-test and concept evaluation tests, which had multiple choice questions. These questions were identified and rephrased.
- There was also need to show the average score per goal for each learner. It was decided to show it as a percentage and also as a graph that is filled proportionately according to the score.

iii. Field Trial

This was part of the field trial at SCI between February and March, 2008. The Second year BSc Computer Science students of OOP course participated.

Suggestions

- The number of times a test could be attempted (one) was not enough. Even then, the number of attempts was not changed because the tests were merely supposed to evaluate current status of knowledge. More feedback was provided on concepts underlying incorrectly answered questions, which needed to be studied again
- The duration of time allowed for the test was short. Even with this, the duration was not altered as it was considered to be sufficient. However, the subsequent learners were advised to attempt the tests only when they felt they were fully ready, so that they take less time per question
- Answers were not provided for incorrectly answered questions. This prompted provision of comprehensive feedback per question, to guide remediation, instead of just providing the correct answer to the learner
- Remove the adaptive pre-test evaluation feature. This feature could not be removed because it was important for the initialization of learner models. To reduce the discomfort of taking a pre-test for a concept whose content had not been studied, the test questions

were displayed one level (two questions) at a time. If the learner did not answer both questions for the level correctly after two attempts, the test was terminated without displaying questions for the next level, in order to reduce discomfort

c) Informative feedback

i. Aspects considered:

- Can the feedback enable the learners to know how they are doing?
- Is the feedback provided after the pre-test and the concept evaluation test effective and sufficient?
- Is the timing of the feedback appropriate?
- Is the feedback clear?
- Are the messages showing countdown of the duration for the pre-test and concept evaluation test useful and well presented so as to be effective?
- Are the pop-up windows reminding the learners about the number of attempts for each test useful and well presented so that they are effective?
- Is the feedback on knowledge level through suitable annotation of concept names with one of the colours of the traffic light system effective?

ii. Field Trial

- There was need for a facility for providing immediate feedback to students on questions attempted. This was provided in terms of percentage score and average score for all the concepts of a goal

If a question was answered incorrectly, feedback was also provided in terms of concepts related to the question which needed to be studied again

Suggestions

- The learners indicated that they were happy with the questions at the beginning of each concept, which estimated their entry knowledge level for the concept

- They indicated that they were also happy with the questions at the end of each concept as a source of feedback on their progress

d) Learner control

i. Aspects considered:

Does the learner have control over which concept to study and the type of presentation of course content to use?

ii. Expert Review

The experts in instructional design for e-learning systems were involved.

Feedback

- The experts appreciated the control the learners had while studying the course since the learners were provided with some guidance so as not to go astray while studying the concepts. The system also allowed the learners the choice in terms of whether to follow the suggestions from the system or make their own decisions on what to study.

iii. Field Trial

- The learners indicated that they were happy with the flexibility of the times when they could study
- They also indicated that the interface was simple to use

e) Reduce the learners cognitive load when accessing information

i. Aspects considered:

- Are there any knowledge modules introduced to make it easier for the learner to focus on the main points first before embarking on details?
- Does the chunking of content into small sizes, such as concepts, make it easier for the learner to learn the course content?

- Does the sequencing of the content from simple to complex make it easier for the learner to study a concept's content?
- Does the arrangement of concepts into layer 1 (simple) up to layer 3 (most complex) make it easier for the learners to study the concepts?
- Instead of the learner combing the Web for relevant additional materials, the course provided carefully selected additional material links from the Web and associated them to the appropriate concepts

Did the feature make it easier for the learners to find additional materials? Did the students indicate a liking for this feature in the closed or open items of the questionnaire administered immediately after the Class Design and Inheritance sections of the course?

- Global structure of the content i.e. concept map provided on the left hand side, which laid out concepts and learning objects within a concept in the commonly used logical order

Did this make it easier for the learners to study the course content?

ii. Field Trial

The learners indicated that they liked:

- The structure of the course, which went a long way in reducing their cognitive load
- The overview, summary and the step by step way in which the content of the concepts was presented.
- The relevant examples provided. They asked for more detailed examples
- The system's interface was simple to use, though two of them indicated that it had a dull appearance. The intention was to keep it simple according to the principle of moderation
- The additional material links provided, though some said that it was not always available. It means they did not see the alternative way of accessing the materials through the menu, for the cases where the material was hidden adaptively yet the student wanted to access it. An attempt was made to improve the situation by training the students on the system before actually using it for learning.

f) Support the linking of prior and current knowledge

i. Aspects considered:

- The concept map showing the relationship between previously studied concepts and the current concept
- The provision of pre-requisite concepts for every concept
- The example running through the OOP course content, in which previously learned knowledge was inevitably linked to current knowledge.
- Provision of the overview at the beginning of each concept

ii. Expert Review

- One of the experts in instructional design indicated the need to include clear pre-requisites for each concept. Therefore, pre-requisite concepts were provided for each concept, in the default order.
- The expert also recommended that learners needed to be brought to the same entry level before commencement of the course, by being reminded of relevant concepts from the pre-requisite course. In as much all the learners had taken the pre-requisite course on programming methodology using the C language, the researcher decided to revise the concepts of programming but using the C++ language.

3.4.4 Process 2: Evaluation of the Adaptive Learning Support System

The evaluation of adaptive learning systems is an important issue in the development of such systems because it is used to establish whether a particular technique which has been implemented works or not. It also establishes the level of benefit the system can produce in exchange for the complexity which comes with the introduction of adaptive features. This is an important contribution to the efforts being made towards making adaptive technologies mainstream.

The evaluation of adaptive learning support systems can be at two levels

- (i) Global or system level

(ii) Level of system components

a) Global level evaluation

Aspects considered:

- The performance of the students in two tests i.e. overall user performance
- Responses to a survey (subjective satisfaction) by learners learning with adaptive features enabled

Approach used

During field trial, the learners responded to a survey after studying the course with adaptive features enabled. The items on the questionnaire used were on a 7-point Likert with points 1 (Strongly disagree) to 7 (Strongly agree). The responses to both closed ended and open ended items were analyzed. The results of the analysis of the closed ended questions are shown in Table 3-17. The mean, minimum and maximum values have been used to describe the results.

Suggestions/feedback:

Table 3-17 : Results of analysis of the trial questionnaire on the perceived value of the adaptive features

		var1	var2	var3	var4	var5	var6	var7	var8	var9	var10	var11	var12	var13	var14
N -Valid	34	33	33	33	33	33	33	33	33	33	33	33	33	33	33
- Missing	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Mean		5.91	5.30	6.06	5.61	5.64	4.39	4.42	5.42	4.85	5.79	5.61	5.15	5.30	5.27
Std. Deviation		1.10 0	.984	1.08 8	1.22 3	1.34 2	1.51 9	1.78 6	1.30 0	1.43 9	1.317	.998	1.202	1.185	1.232
Minimum		3	4	3	3	3	1	1	3	1	2	3	1	2	2
Maximum		7	7	7	7	7	7	7	7	7	7	7	7	7	7

The mean responses for the 14 variables shown in Table 3-17 are described below:

Variable 1 - The **course's organization** into goals, concepts, pre-requisite concepts and knowledge modules (5.91). The students agreed that it was useful, with a mean of 5.91 compared to the highest possible mean of 7 (strongly agree). The lowest score was 3, meaning

that none of the students strongly disagreed. The highest score was 7, meaning that some of the students strongly agreed with the usefulness of the feature.

Variable 2 - The structure of the educational material into different levels of performance (5.3). The students agreed that the presentation of course content in different levels of performance, from simple to complex, was useful. The lowest score was 4, meaning that none of the students disagreed with the usefulness of the feature. The highest score was 7, meaning that some students strongly agreed with the value of the feature to their learning.

Variable 3 - The presentation of the content of each concept using different modules (i.e. theory, exercises, examples and activities, among others) contributed to learning (6.06). The mean was high, tending to strongly agree and therefore, the students felt that the use of different modules to present content had a positive contribution to their learning. There was no student who strongly disagreed with the feature since the lowest score was 3 but there were those who strongly agreed with the value of the feature, with the highest score being 7.

Variable 4 - The presentation of individual sub-sections of a concept's reading material (i.e. theory, sub-concepts, additional reading materials, etc) through links which are seen when you expand a concept by clicking on + next to the concept name (5.61). This feature showed the relationship between goals, concepts and knowledge modules in a top-down view. The mean score was high, indicating that the students agreed that the feature contributed to their learning. The lowest score was 3, indicating that no student strongly disagreed with the feature's usefulness. There were those who strongly felt that the feature contributed to their learning with a score of 7.

Variable 5 - The provision of feedback (i.e. test scores and concepts which should be studied further after a test) was seen by the students as having a positive contribution to their learning because of the high mean score (5.64). The lowest score was 3, indicating that no student strongly disagreed with the contribution of the feature to their learning. Some students actually made a score of 7, indicating strong agreement with the value of the feature to their learning.

Variable 6 - The adaptively proposed level of difficulty for a concept, at which the student was to start studying the concept (Novice, Intermediate or Advanced) (4.39). The average score was slightly above neutral, indicating weak agreement to the contribution of the feature

that none of the students strongly disagreed. The highest score was 7, meaning that some of the students strongly agreed with the usefulness of the feature.

Variable 2 - The **structure of the educational material into different levels of performance** (5.3). The students agreed that the presentation of course content in different levels of performance, from simple to complex, was useful. The lowest score was 4, meaning that none of the students disagreed with the usefulness of the feature. The highest score was 7, meaning that some students strongly agreed with the value of the feature to their learning.

Variable 3 - The **presentation of the content of each concept using different modules** (i.e. theory, exercises, examples and activities, among others) contributed to learning (6.06). The mean was high, tending to strongly agree and therefore, the students felt that the use of different modules to present content had a positive contribution to their learning. There was no student who strongly disagreed with the feature since the lowest score was 3 but there were those who strongly agreed with the value of the feature, with the highest score being 7.

Variable 4 - The **presentation of individual sub-sections of a concept's reading material** (i.e. theory, sub-concepts, additional reading materials, etc) **through links which are seen when you expand a concept** by clicking on + next to the concept name (5.61). This feature showed the relationship between goals, concepts and knowledge modules in a top-down view. The mean score was high, indicating that the students agreed that the feature contributed to their learning. The lowest score was 3, indicating that no student strongly disagreed with the feature's usefulness. There were those who strongly felt that the feature contributed to their learning with a score of 7.

Variable 5 - The **provision of feedback** (i.e. test scores and concepts which should be studied further after a test) was seen by the students as having a positive contribution to their learning because of the high mean score (5.64). The lowest score was 3, indicating that no student strongly disagreed with the contribution of the feature to their learning. Some students actually made a score of 7, indicating strong agreement with the value of the feature to their learning.

Variable 6 - The **adaptively proposed level of difficulty for a concept, at which the student was to start studying the concept** (Novice, Intermediate or Advanced) (4.39). The average score was slightly above neutral, indicating weak agreement to the contribution of the feature

towards the learning of the course content. Some students strongly agreed, making a highest score of 7 and others strongly disagreed with the value of the feature, making a lowest score of 1.

Variable 7 - The **traffic light metaphor** (Green, Orange and Green colours) **on concept names** showing the extent to which a student was ready to study different concepts (4.42). The average score was slightly above neutral, indicating weak agreement with the contribution of adaptive link annotation, through the traffic light metaphor, to their learning of OOP concepts. Some students strongly agreed with the value of the feature, making a highest score of 7. Others strongly disagreed with the value of the feature to their learning of course content.

Variable 8 - The **way the links of additional materials at the bottom of the lesson page were adaptively presented** if the student had been using additional materials or were **adaptively hidden** if the student had not been using additional materials (5.42). The students agreed that the adaptive hiding of additional material links was helpful to their learning, with the lowest score being 3 (nearer neutral than disagree) and the highest score being 7, indicating that they strongly agreed with the value of the feature.

Variable 9 - The **option for the student to select the material he preferred to study next** i.e. learner control of their learning (5.42). The students felt that the control they had over their learning facilitated learning. There were some students who strongly disagreed, making a lowest score of 1 and those who strongly agreed, making a highest score of 7.

Variable 10 - The **different features adaptively proposed by the system** (concept to study through different colours, level of difficulty to follow, and adaptive hiding of links to additional materials) **were not disruptive** (5.79). The students did not find the adaptive features disruptive, with the lowest score of 2 indicating that some felt it was disruptive to some extent and the highest score of 7 indicating that the adaptive features were not disruptive at all to others.

Variable 14 - **Overall, interaction with the system was very satisfying** (5.27). The students indicated that their overall interaction with the system was satisfying, with the lowest score being 2, indicating disagreement to some extent and the highest score being 7, indicating that some students strongly agreed that the overall interaction with the system was satisfying.

Decisions for improvement

Overall, it was felt that there was need to sensitize the students further about how the adaptive features worked. This was made necessary by their responses. For example, they liked the presentation of content in three levels of performance i.e. remember, use and find (5.3), yet their mean response to the importance of the feature which suggested the level of difficulty to follow, based on the three levels of performance, was lower (4.39). There was also need to test the system further to establish whether the adaptive features were working as expected.

b) Component based layered evaluation

The layered evaluation approach **compliments the global evaluation**. When the system is successful, all is well but when the system is deemed unsuccessful, the components need to be investigated in order to establish the parts of the system which are not working well.

With layered evaluation, **part of the system can be evaluated early** enough to influence the design and development process of the rest of the system. Ways of improving the system can be identified early (Brusilovisky et al., 2004).

Adaptive systems have three basic models: (i) a **user model**, (ii) **domain model** and (iii) an **interaction model** (Benyon & Murray, 1993c; Brusilovisky, 1996). The interaction model is made of the **inference process**, which predicts learner model attribute values, and the actual **adaptation** of the system (Langley, 1999).

The layered evaluation concerned itself with the **learner model**, and the **inference** and **adaptation** components of the interaction model.

1) Evaluation of the learner model

The learner modeling process was aimed at producing and updating the learner model, mainly represented as an overlay over the domain model. A vector of feature values was used to represent the estimated learner knowledge and tendency to use or not to use additional learning materials, among other characteristics.

The purpose of the evaluation of the learner modeling process was to establish how well the system predicted the values of the two main attributes of the learner in this system i.e. learner concept knowledge level and tendency to use or not to use additional materials.

Low-level information was collected and used to estimate the successfulness of learner modeling.

(a) Guiding questions

Main Question

Are the characteristics of the learners being successfully established by the system and stored in the learner model?

Sub-questions:

- (i) Is the learner's knowledge level being successfully captured by the system?
- (ii) Are the learner's preferences for additional learning materials successfully determined, captured and represented in the system?
- (iii) Are the learner's characteristics continuously changing as interaction with the system continues?
- (iv) Is the knowledge level classification of learners into novice, intermediate and advanced categories sufficient or should they be reduced or increased?

(b) Evaluation activities

- (i) The scores of the learners in the concept evaluation tests were examined to see if they were in two, three, four or more clusters. A scatter graph was plotted for a sample of the concept evaluation test scores and the written tests.
- (ii) To find out if there was a relationship between the learners' initialized concept scores and the student's actual performance after studying the content, expressed as concept

evaluation test score. A relationship would imply that the initialized/predicted level of knowledge, which was used to determine learner readiness, is correct.

- (iii) To find out if there was a relationship between the learner's level of readiness to study a concept, based on the learner's knowledge level, and the relative time spent on a concept.

The investigation was not conclusive. A decision was made to train the students more on how to use the system's adaptive features as part of preparations for the actual study.

2) Evaluation of inference by Machine Learning algorithms

The inferences made were evaluated to see if the Machine Learning algorithms were accurate. Two algorithms were used: (a) the Heterogeneous Value Difference Metric (HVDM) and (b) Naive Bayes Classifier (NBC).

(a) Guiding questions:

- (i) Are the **predictions by the HVDM** Machine Learning algorithm of the learner's initial knowledge level accurate?
- (ii) Are the **predictions by the NBC** Machine Learning algorithm of the learner's preference for additional learning materials accurate?

(b) Evaluation activities:

- (i) The **inferences by HVDM** were evaluated by comparing the score predicted at the beginning against the actual concept evaluation test score at the end of studying the concept. The number of accurate predictions was divided by the total number of predictions. The accuracy achieved by HVDM was 60 % (Oboko, Wagacha, Omwenga, & Libotton, 2009a).
- (ii) The **inferences by NBC** were evaluated by comparing the prediction of the algorithm with the actual usage of additional materials. The level of accuracy obtained was 72 % (Oboko et al., 2009c).

3) Evaluation of the adaptation effect

The learner model was used to generate the adaptation effect. The aim of evaluation was to see if the adaptations made on the system's interface suited the learners. Evaluation was made to find out whether adaptation led to the building of a more effective navigation path through the knowledge space, selection of an appropriate level of performance by the learner and achievement of better learning outcomes.

The **sources of information** for the evaluation of this component were test results (Class Design and Inheritance tests), concept evaluation tests and navigation logs.

(a) Guiding questions:

Main Question

Are the adaptation decisions valid and meaningful, according to the state of the learner model?

Sub-questions:

- (i) Does the applied adaptive presentation support technique suit the learner, according to the learner's knowledge level, by improving interaction?
- (ii) Does the applied adaptive navigation support technique, using annotation of concept names, suit the learner's knowledge level, by improving interaction?
- (iii) Does the applied adaptive navigation support technique, using link hiding, suit the learner's preference to use or not to use additional learning materials, by improving interaction?

(b) Evaluation activities:

- (i) Comparing the **order of accessing the concepts** when using the adaptive features and when not using them. This was to see whether the use of adaptive features influenced the navigation path followed by learners. Was there any **disorientation**? Some learners skipped some concepts completely, suggesting their disorientation. However, it was not possible to link disorientation to use or otherwise of adaptive features

- (ii) Analysis of the **changes in performance (test scores)** of the learners from when using adaptive features to when not using adaptive features or conversely from when not using adaptive features to when using adaptive features

A strong conclusion could not be drawn on the differences in performance due to use of AI.

- (iii) Comparing the number of **cancelled concept evaluation tests i.e. concepts visited without the student attempting the concept evaluation test** when using AI to when using NO AI (evaluation of adaptive presentation). This could indicate **early desertion** due to frustration with the way the content is presented

The evaluation was inconclusive.

- (iv) Comparing the **number of tests in which the learners failed** when using adaptive features to when learning without adaptive features, as an indication of effectiveness or lack of it, for adaptive presentation support. The results were not conclusive, because of the small number of records available.

A decision was made to provide more explanation to the students on how to use the adaptive features, especially adaptive presentation. This was to be done before commencement of the actual research. This was to avoid frustration due to lack of knowledge on how to use the system's adaptive features.

- (v) Analysis of navigation logs of **additional material usage** by learners using adaptive support to establish if there was a **pattern of usage** i.e. tendency towards not using or using additional materials. The analysis also aimed to establish whether or not there was **irregular access** of the additional materials. Irregular access would probably indicate that the adaptive support provided was not working or the learner had ignored the system's suggestions and used the alternative route for accessing the materials, which was provided to give learners control over their learning process.

For those who used additional materials, the pattern was regular. For example, for the group starting with NO AI followed by use of AI:

- one of the students used additional materials in the first 3 concepts and did not use them after that (Student 19)
- one student used the materials for one concept but did not use them before and after this concept (Student 38)
- one student used them in the first 6 then did not after that (Student 123)
- one student used the material in the first 2 and did not use them thereafter (125)
- one student used the materials in 12 out of 14 concepts but did not use them in the last 2, which were possibly studied in a hurry (no 134).

Therefore, no action was planned, except ensuring that the links to additional materials were not broken.

3.4.5 Process 3: Evaluation of the Problem Solving Support System

The aim of this part of formative evaluation was to collect information about problem solving which could be used to establish the presence of constructivist approaches (Lake & Tessmer, 1997). The evaluation was also intended to establish the effectiveness of the scaffolds for problem solving (Brickell & Herrington, 2006).

Some of the constructivist features looked for in the course include (i) generative learning i.e. knowledge construction, (ii) self-reflection, (iii) active learner control, (iv) authentic and meaningful learning experiences, (v) informative feedback, (vii) personal meaning (Azevedo & Hadwin, 2005; Brickell & Herrington, 2006; Chng, 2001; Koohang, Riley, Smith, & Schreurs, 2009; Lake & Tessmer, 1997; Schraw, 1998; Vrasidas, 2000).

These features are presented in Table 3-18. The first column shows the feature, for example, providing opportunities for self-reflection. The second column shows the meaning of the feature and its implication for the sub-system. For example, in providing for learner self-reflection, the system should provide opportunities for the learners to analyze and be critical of their own problem solving. The third column provides evidence that these features were

included in the constructivist learning support sub-system. For example, opportunities to think about and explain their progress during problem solving, prompted by the task progress shown by the graphic on the task schedule.

Table 3-18 : Features considered during formative evaluation of the constructivist learning support system

Indicator	Meaning/Implication	Evidence
<ul style="list-style-type: none"> Generative learning 	<ul style="list-style-type: none"> Did the learners construct new knowledge (by linking new information to prior knowledge, and also linking new ideas among themselves) 	Learners were required to <ul style="list-style-type: none"> build an application which involved the synthesis of both old and new ideas develop explanations or justifications for their decisions, which involved using any previous knowledge and experience to explain the current situation identify their own knowledge and skills, from previous learning and experience, which would be required for problem solving stop and think for a while about the question prompts and the progress shown on the task schedule graphic
<ul style="list-style-type: none"> Learner's self-reflection 	<ul style="list-style-type: none"> Did the learners have opportunities for analyzing and being positively critical of their own learning through rigorous self-questioning and deep probing of what and how one was learning? 	<ul style="list-style-type: none"> Learners were presented with opportunities to think about and explain why their individual progress in problem solving, as shown on a task schedule graphic, was going as it was and what could be done to improve. Learners were required to find within themselves answers to the question prompts designed to be suitable for solving an ill-structured programming problem. The prompts were made relevant to the different stages of the problem solving process and also relevant to the domain of OOP
<ul style="list-style-type: none"> Active learner control 	<ul style="list-style-type: none"> Did the learners have control over their progress through problem solving? 	<ul style="list-style-type: none"> The learners were allowed make their own decisions during problem solving. They also made their own problem solving plans They were allowed to define the problem, break down the problem into subtasks, define their own task objectives, problem solving activities, and identify the resources they needed for problem solving, including knowledge and skills. They were required to monitor their progress aided by the scaffolds, to take corrective action to remain on course, and manage information, say through grouping it according to relevance to the different sub-tasks and typing it into the online task schedule. They were also required to evaluate their performance

Indicator	Meaning/Implication	Evidence
<ul style="list-style-type: none"> • Task management support structure (scaffolds) 	<ul style="list-style-type: none"> • Were any supports provided to enable learners to work beyond their normal levels, towards their potential? Most learners get into problem solving without adequate task management skills 	<ul style="list-style-type: none"> • The task schedule, the phases of problem solving and suitable question prompts provided a structure for the learners to follow in planning their activities, organizing their information (knowledge and skills), and ordering of the tasks. • This spurred them into self-reflection and a review of their progress, leading to actions considered appropriate to improve below par performances and even evaluation of their processes and the resulting product.
<ul style="list-style-type: none"> • Authentic and meaningful learning experiences 	<ul style="list-style-type: none"> • Were the learners provided with authentic problems to solve, which could make the learners think and act like experts in their field 	<ul style="list-style-type: none"> • The learners were provided with ill-structured problems to solve, set in real-life contexts which they could easily identify with, such as bus terminus control, calculation of tax, the voting system and lift control system
<ul style="list-style-type: none"> • Feedback 	<ul style="list-style-type: none"> • Were there cues to provide learners with meaningful information about their progress in order to motivate them or to get them to change their approach so as to successfully complete their tasks 	<ul style="list-style-type: none"> • The learners were reminded of how they were progressing, according to their current phase when using question prompts • The percentage completion of each sub-task was shown on the graphic of the task schedule.
<ul style="list-style-type: none"> • Personal meaning 	<ul style="list-style-type: none"> • Were the supports provided tailored to suit the needs of individual learners, the stage of problem solving at which the learners were and the specific activities they were carrying out 	<ul style="list-style-type: none"> • Question prompts were designed for each problem solving stage and were made relevant to OOP • The learner was required to answer the questions according to his understanding. The answers were expected to spur further self-reflection and self-regulation • The interaction between the learner and the task schedule was personal to the learner, depending on his knowledge, skills, and problem solving plan. The initiative came from the learner, supported by the structure provided by the task schedule

3.5 Summary

The chapter presented the extended Integrated Design Process (IDP), a framework for the development of web-based learning systems. It is based on learning theories and integrates instructional design with human computer interface design. The initial IDP was extended to make the user interfaces adaptive and also make the development of the system user-centred

i.e. audience-driven. Navigation and page design were also added to the original IDP, as well as metacognitive control. Formative evaluation was also given adequate attention, because it facilitates evolutionary development, which is suitable for the development of web-based system interfaces.

IDP has four phases, namely needs analysis, conceptual design, development and formative evaluation. The chapter explained how each of the phases was applied during the design and development of the learning environment. Each phase was implemented through a number of processes. For example, in this research, the conceptual design phase had the following processes: development of design scenarios and use cases, information design, navigation design, page design, structure design, presentation design and data design.

*A man may conquer a million men in battle but one who conquers himself is,
indeed, the greatest of conquerors.*

- Buddha

Chapter 4 : Methodology - Machine Learning and Adaptive Learning Support Design

4.0 Introduction

Adaptive user interface systems rely on models of learners, models of the domain and inference rules in order to make adjustments to their interfaces to suit individual learners. Inference rules were used to update the learner model. Adaptation rules were used to determine how to adjust the interface.

For this research, Machine Learning algorithms were used to learn some of the attributes of the learners. Specifically, they were used to learn the initial knowledge level of the learner and whether the learner would choose to use additional learning materials while studying a concept or not.

4.1 Specification of the Adaptive User Interface Components

Six adaptive user interface components were considered: learner model, domain model, data collection techniques, multiple representations, inference mechanisms and adaptation rules.

4.1.1 The Learner Model Components

The learner model had domain dependent attributes, which mostly related to prior programming experience, non-programming computer experience, level of comfort and the learner's attitude to programming (Bergin & Reilly, 2005; Ventura, 2005; Wiedenbeck, 2005). Some of these attributes were: number of programming courses taken by the learner in university, number of high level programming languages studied while in university, approximation of number of programs written while in university and approximation of average length of programs written while in university. Other domain dependent attributes were number of programming languages studied out of choice, approximation of pre-course level of C++ programming, the pre-requisite course (programming methodology) grade and courses studied in university requiring use of computers. The rest of the attributes were level of anxiety experienced while programming, level of interest in programming, perceived level of difficulty of OOP, and level of preference to learn with hypermedia.

The other domain dependent information included **concept level knowledge** and **tendency to use additional learning materials**, which were an overlay over the domain model. These two attributes were used during adaptation while the other domain dependent attributes were used to infer these two.

The domain independent attribute included was gender (Bergin & Reilly, 2005).

Figure 4-1 shows the learner model used for adaptation, with its' domain independent and domain dependent attributes. The attributes are shown at the global level, goal level and concept level.

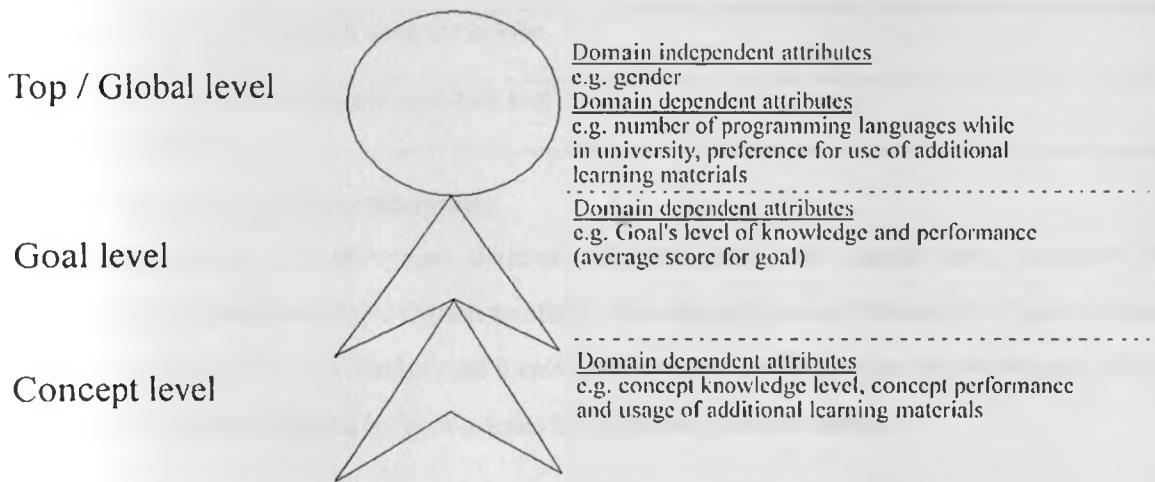


Figure 4-1 : Learner model showing domain independent and domain dependent attributes

4.1.2 Data Collection Techniques

Data was collected **obtrusively** and **non-obtrusively**, as shown in Table 4-1. Obtrusive collection of data is direct whereas non-obtrusive collection is implicit. In the table, data collected

obtrusively is shown in the first column whereas data collected non-obtrusively is shown in the second column.

Table 4-1: Showing how different elements of data were collected

Obtrusively collected data	Non-obtrusively collected data
• Personal information (e.g. gender)	• The marks in pre-tests and concept tests
• Survey information on the learners' experience	• The sequence of studying the concepts
• The sub-tasks defined by the learner during problem solving	• The usage (used or did not use) of additional study materials for a concept
• A percentage figure expressing the extent of task completion – task progress	• How long it took the learner to study a concept
• The date of commencement of a sub-task	• The concepts whose study was aborted (opened but no answers were submitted for the test)
• The time elapsed since commencement of a sub-task	
• The written tests at the end of each of the two sections of the course	
• The answers to the self-evaluation pre-test and concept evaluation	

4.1.3 The Domain Model Components

The domain model i.e. the top-down structure of the course had 4 **goals** with a total of 14 concepts which the learner could choose to study. The concepts were divided into 2 parts, Class Design and Inheritance. Class design had 8 **concepts** whereas Inheritance had 6 concepts. Each concept had a number of knowledge modules for delivering course content.

4.1.4 Multiple Representations:

(a) Presentation of Course Content

The level of difficulty of the concepts was considered during the design of multi-representations of the course. Three levels of knowledge were considered i.e. novice, intermediate and advanced. The **presentation of the course content** was organized into 3 abstractions to suit learners at the 3 different levels of knowledge.

(b) Presentation of Concept Names

Multiple ways of presenting **concept names** were also designed, using different colours, according to the level of difficulty of a concept and the overall level of knowledge of the learner. This was done according to the traffic light metaphor, according to the learner's readiness to study the concepts. Some concept names were shown in red (not ready), others in orange (ready) and others in green (ready and was expected to do well).

(c) Additional Learning Materials

For **additional materials**, there were 2 representations designed: to show or not to show the additional materials at the end of the lesson page.

4.1.5 Inference

Machine Learning algorithms were used to predict the initial level of knowledge of the learner and the likelihood of the learner using additional learning materials for each concept.

- (a) A flavor of K-Nearest Neighbour (k-NN) algorithm called Heterogeneous Value Difference Metric (HVDM) was used to determine those learners who were similar to the current learner. The knowledge levels of these learners were used to initialize the current learner's knowledge level, on commencing learning using adaptive support (Oboko et al., 2009a).
- (b) The Naive Bayes Classifier (NBC) was used to estimate the likelihood that the learner would need to use additional materials for the current concept. This was based on the additional material click history of the current learner and the other learners who had studied the course up to the current concept (Oboko et al., 2009c).

4.1.6 Adaptation of the User Interface

Adaptation rules were used to determine the appropriate adjustments to be made to the system's interface.

- (a) The presentation of the appropriate abstraction of course content, in terms of appropriate level of difficulty, was determined by the level of knowledge of the learner. The three levels of knowledge were matched to three types of course content presentation
- (b) The annotation of the names of the concepts was in terms of three colours which were matched to the three levels of knowledge
- (c) The decision to present or not to present additional materials was made based on whether the likelihood of usage of additional materials was inferred as Yes or No.

Figure 4-2 and Figure 4-3 show these 3 adaptive features on a lesson page.

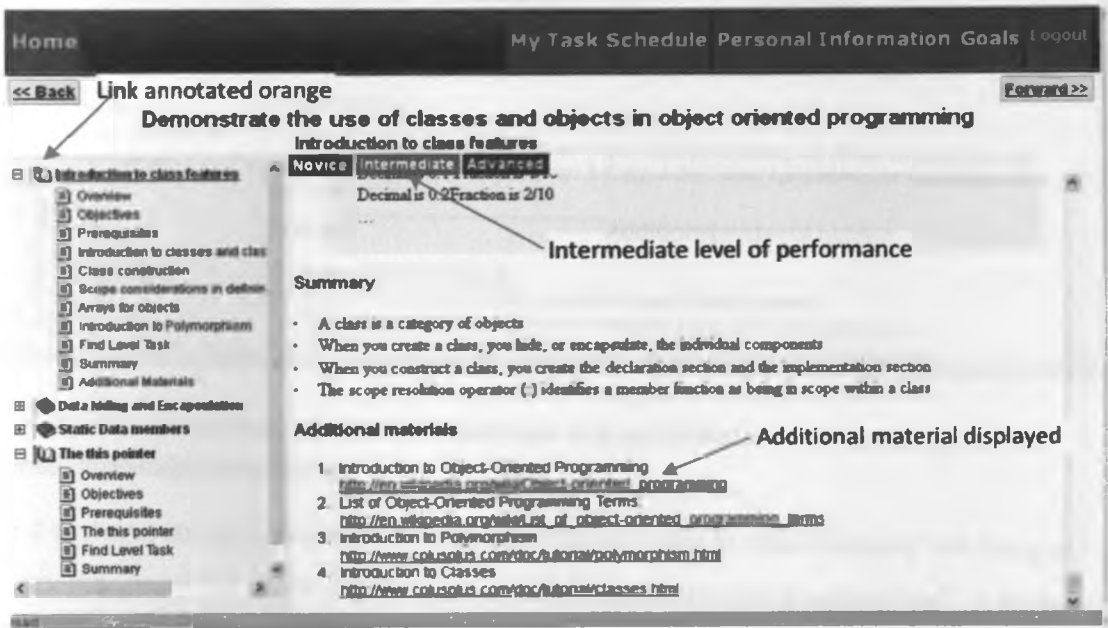


Figure 4-2 : Link annotation, adaptive presentation and additional material presentation for intermediate knowledge level learner

Figure 4-2 shows course content of the concept “Introduction to Class Features” of the goal called “Demonstrate the use of classes and objects in object oriented programming”. It shows that for a learner at intermediate level of knowledge, the concept has been annotated with orange color. This colour indicates that the learner is ready to study the concept. The tab for

intermediate level of performance of content is also proposed. Figure 4-2 also shows that links for additional materials have been displayed, indicating that the learner and other similar learners tended to use the materials in previously studied concepts.

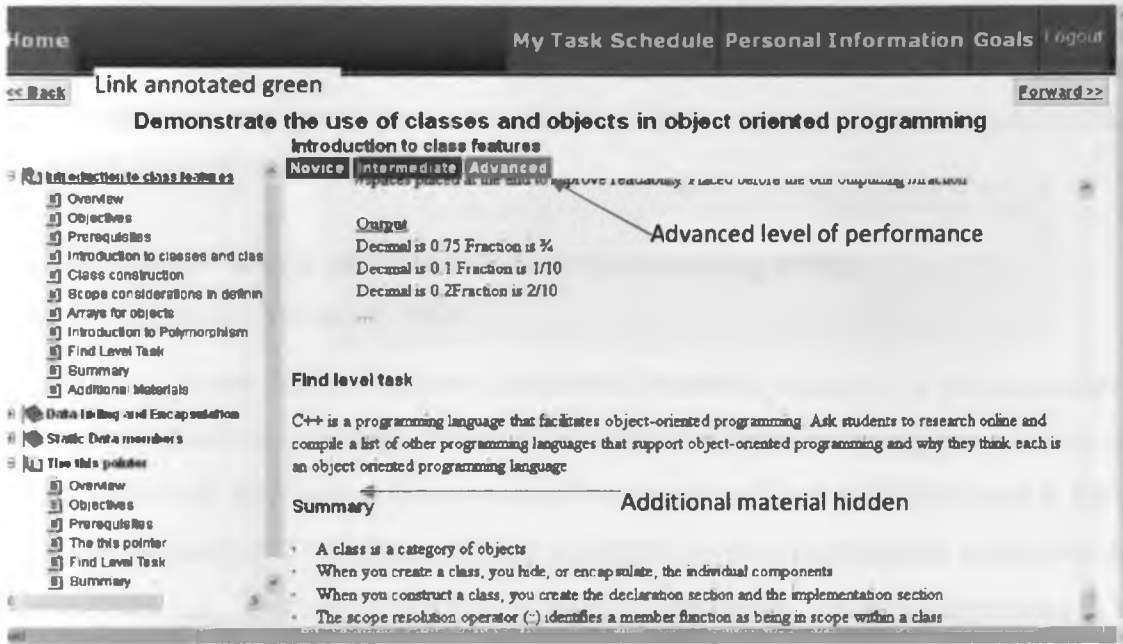


Figure 4-3 : Link annotation, adaptive presentation and additional material presentation for advanced knowledge level learner

Figure 4-3 shows course content of the concept “Introduction to Class Features” of the goal called “Demonstrate the use of classes and objects in object oriented programming”. It shows that for a learner at advanced level of knowledge, the concept has been annotated with green color. The colour indicates that the learner is expected do well in this concept. The tab for advanced level of performance of content is also proposed. Figure 4-3 also shows that links for additional materials have not been displayed, indicating that the learner and other similar learners tended not to use the materials in previously studied concepts.

4.2 Application of Machine Learning Algorithms

4.2.0 Introduction

This section presents justification of choice of each of the algorithms. It also describes how the challenges of using Machine Learning algorithms for adaptive user interfaces were addressed for each of the algorithms. The section also explains in detail how each one of the algorithms was used for making inferences. The section also presents information on how the inferred learner model attribute values were used for adaptation as well as how the algorithms were evaluated.

4.2.1 Initialization of Learner Model Concept Scores Using HVDM

(a) Background to the use of HVDM

Initialization of learner models has been considered important because it is not reasonable to assume that all learners have the same level of domain knowledge at the beginning or that they have no knowledge at all about the domain at the beginning (De Bra, 2000; Virvou & Tsiriga, 2004). If the system uses these assumptions about the level of performance of the learners, then there is a high likelihood of the system giving unreasonable advice or suggestions to the learners, which may end up disrupting instead of supporting the learner's learning (De Bra, 2000).

The estimation of the initial level of performance of a learner was based on the current levels of **performance of the other learners** already learning OOP using the system. The assumption was that if the learners were similar in terms of profile and background attributes, then the level of performance of those already in the system could be used to estimate the initial level of performance of the current learner, who was joining the course (Virvou & Tsiriga, 2004).

Since the learners' attributes were heterogeneous, with some being nominal e.g. *level of anxiety while engaged in programming* and others numeric e.g. *number of programming languages studied*, a combination of methods was used to determine the total distance between two students.

- The **Euclidean Distance** measure was used to determine the distance between two values of a numeric attribute.
- For nominal attributes, the method which has been used the most has been the **overlap measure**. The **overlap measure** merely compares two values to see if they are the same. If they are the same, distance = 0 and if they are not the same, distance = 1. This measure loses some information since it does not consider how far the two values are from each other (Virvou & Tsiriga, 2004; Wilson & Martinez, 1997). Therefore another measure called **Value Difference Metric (VDM)** was used. VDM takes into consideration the **similarity of the values** of a nominal attribute and the **proportions of the training examples** with the values which are being compared. It also considers the class or target function value in determining the distance between two values (Wilson & Martinez, 1997).

See equation (1) below for the VDM formula. The formula is used for calculating the distance between two values x and y of a nominal attribute a :

$$VDM_a(x, y) = \sum_{c=1}^C |(N_{a,x,c} / N_{a,x}) - (N_{a,y,c} / N_{a,y})|^q \quad (1)$$

where

$N_{a,x}$ is the number of instances in a training set T which have x as the value of attribute a

$N_{a,x,c}$ is the number of instances in a training set T that have x as the value of attribute a and output class c

$N_{a,y}$ is the number of instances in a training set T which have y as the value of attribute a

$N_{a,y,c}$ is the number of instances in a training set T that have y as the value of attribute a and output class c

C is the number of output classes in the problem domain

q is a constant, usually 1 (Domingos, 1995) or 2 (Wilson & Martinez, 1997). $Q=2$ has been found to do better over some datasets

(b) Considering the challenges in using Machine Learning for adaptive interfaces

- **Learning time**

The learning times for HVDM were found to be generally acceptable (Wilson & Martinez, 1997). In fact, the times for training are very fast, which suits the purpose of estimating the initial knowledge level of the learner so as to start adaptation soon after the commencement of learning (Virvou & Tsiriga, 2004).

- **Quantity of data needed**

When provided with a sufficiently large set of training data, the basic k-NN algorithm is quite effective (Mitchell, 1997). Since the learners were to learn in a laboratory situation and the course was to be studied by the students generally at the same time, the data to be used for training HVDM was expected to accumulate fast.

- **Problem Formulation**

The problem was represented in terms of the task (T), experience (E) and performance measure (P).

T was to learn the initial knowledge of a new student joining the system to learn concepts of OOP.

E was the combination of the values of various attributes in the learner model and the corresponding knowledge level of individual learners. All the learners already learning using the system were considered as instances for training the algorithm.

P was the number of times the algorithm inferred the correct performance level (test scores) of the learner, out of all the inferences made by the algorithm. The predicted score was compared to the actual score of the student in the concept evaluation test.

- **Representation of the instances and training examples**

The instances were represented as vectors of attribute-value pairs, having the format *<attribute 1 value, attribute 2 value ... attribute n value>* where *n* is the number of attributes forming the

vector (Oboko et al., 2009a). The attribute - value pairs were formed from attributes such as those about the profile and experience of the learner as contained in the learner model, and their corresponding values. For example, the attribute-value pair (*learner's interest in programming, moderate*).

Two vectors were used. The **first vector** consisted of the attribute- value pairs of profile and prior experience information for each student. It was used to determine the students who were similar to the current student, using the distances calculated between each of the students and the current student, using HVDM. A **second vector** was used to estimate the initial knowledge level per concept for the current student. It constituted concepts already studied by each student and their actual concept evaluation scores (Oboko et al., 2009a).

- **Collection of the data used by the algorithm**

Data on the student's personal profile and prior experience was collected directly using an opening questionnaire. Data about concepts studied and scores in tests was acquired indirectly as the learners participated in the learning process.

(c) **How HVDM was used**

- **Filling in of personal information**

The process began with the collection of personal information from the learner immediately after logging in for the first time. The information collected was related to the **profile of the learner and the learner's prior experience** relevant to OOP learning. These attributes were used to form the initial learner model.

- **The pre-test**

The pre-test scores of the learner were used to estimate the learner's initial knowledge level. The pre-test questions were designed in such a way that there were two questions at the novice level of difficulty, two questions at the intermediate level of difficulty and two more questions at the advanced level of difficulty.

If both questions had been answered incorrectly, then the learner was assigned the level of knowledge preceding the current knowledge level of the questions. If one question was answered incorrectly after two attempts, the learner was assigned the current level of knowledge of the questions. If the level was the first one and both answers were incorrect, then the learner was still assigned the first level i.e. novice. If the level was the last one and the learner answered both questions correctly, then he was still assigned the last level i.e. advanced.

- **The feature vectors for the learners and the calculation of the initial scores**

- (i) **Vector 1**

The first feature vector had the following attributes:

<StudentId, knowledge level, gender, learning_with_adaptive_support, number of programming courses taken by the learner in university, number of high level programming languages studied while in university, approximation of number of programs written while in university, approximation of average length of programs written while in university, number of programming languages studied out of choice, approximation of pre-course level of C++ programming, the pre-requisite course grade, courses studied in university requiring use of computers, the level of anxiety experienced while programming, level of interest in programming, perceived level of difficulty of object oriented programming, level of preference to learn with hypermedia>

The pre-requisite course grade was attained by the student in the programming methodology course studied in the previous semester.

The attributes which covered ranges of values were coded as follows:

- ❖ Approximation of number of programs written while in university: A → 1-10 programs, B → 10-20, C → 20-50, D → Over 50 programs
- ❖ Average length of programs: A → up to 20 lines, B → 20- 100 lines, C → 100-250 lines, D → 250 lines and over

(ii) **Determination of the nearest neighbours**

1. From vector 1, the system determined if there were any other students at the same level of knowledge level as the current learner.
2. If there was **no other student** of the same knowledge level in the system, the student's concept scores were initialized using one's own pre-test values.

This was considered a better way than assuming no initial values or assigning the student values determined by some other criteria. This is because the **pre-test scores were real student scores**.

3. The distance of each of the students from the new student, also called **query student**, was determined using HVDM.
4. If there was **only one other** student at the same knowledge level as the new student, the new student was assigned **that student's concept scores**
5. The distances between the nominal values were determined. They were probabilities, in the range $0 \leq \text{distance} \leq 1$.
6. The distances between the numeric attributes were also determined. They were normalized in order to be in the same range as distances between nominal attributes i.e. $0 \leq \text{distance} \leq 1$.

This was to give **each attribute a similar influence** in the function for calculating overall distance between two instances. It was also to reduce the **effect of spurious values**.

To convert the distances into the range of 0..1, the numeric attribute distances were divided by 4 standard deviations. The calculation of standard deviation was based on the values of the numeric attributes being considered (Wilson & Martinez, 1997).

7. If a **value for an attribute was unknown** in either of the two instances being compared, then the maximum distance of 1 between the attribute values was assumed.

8. After the normalization of distances, the distances were summed.
9. Then the total distance of each of the students from the new student was considered.
10. The highest odd number of nearest students was selected for use in the estimation of concept scores. This was in order to avoid reducing the number of training examples. This was also considered necessary because with three levels of knowledge, there were chances of only a small number of neighbours belonging to the same knowledge level as the new student.

(iii) Vector 2

The second vector had the following information:

<userid, concept1_eval_score, concept2_eval_score, concept3_eval_score, concept4_eval_score, concept5_eval_score, concept6_eval_score, concept7_eval_score, concept8_eval_score, concept9_eval_score, concept10_eval_score, concept11_eval_score, concept12_eval_score, concept13_eval_score, concept14_eval_score>

(iv) Estimation of the initial scores for the learner

1. The distances of the nearest neighbours from the new student were used to determine the extent to which each of the other student's concept score would influence the calculation of the initial concept score for the new student (Virvou & Tsiriga, 2004).

This was the weight, and it was obtained by the formula:

$$\text{Weight} = 1/\text{distance}^2$$

The estimation of the concept scores was the **classification process**. It was done concept by concept, using vector 2.

2. **Initial Score** = sum of the product of the weight and concept score, taken over all the nearest neighbours, divided by the total of the weights for all the neighbours.
3. In case the new student had **exactly the same feature values for vector 1 as one of the nearest neighbours**, then

distance = 0 ,

weight =1/distance²,

weight =∞.

In this case, the weight was assigned a maximum value of 1.

- Use of the initial scores for adaptation

The initialized scores were used to estimate the learner’s level of readiness for the different concepts. The level of readiness was used to suggest concepts which the learner was expected to do well in and a suitable level of performance of concept content for the learner. These two features are called adaptive navigation support with link annotation and adaptive text presentation support respectively, and are shown in Figure 4-4.

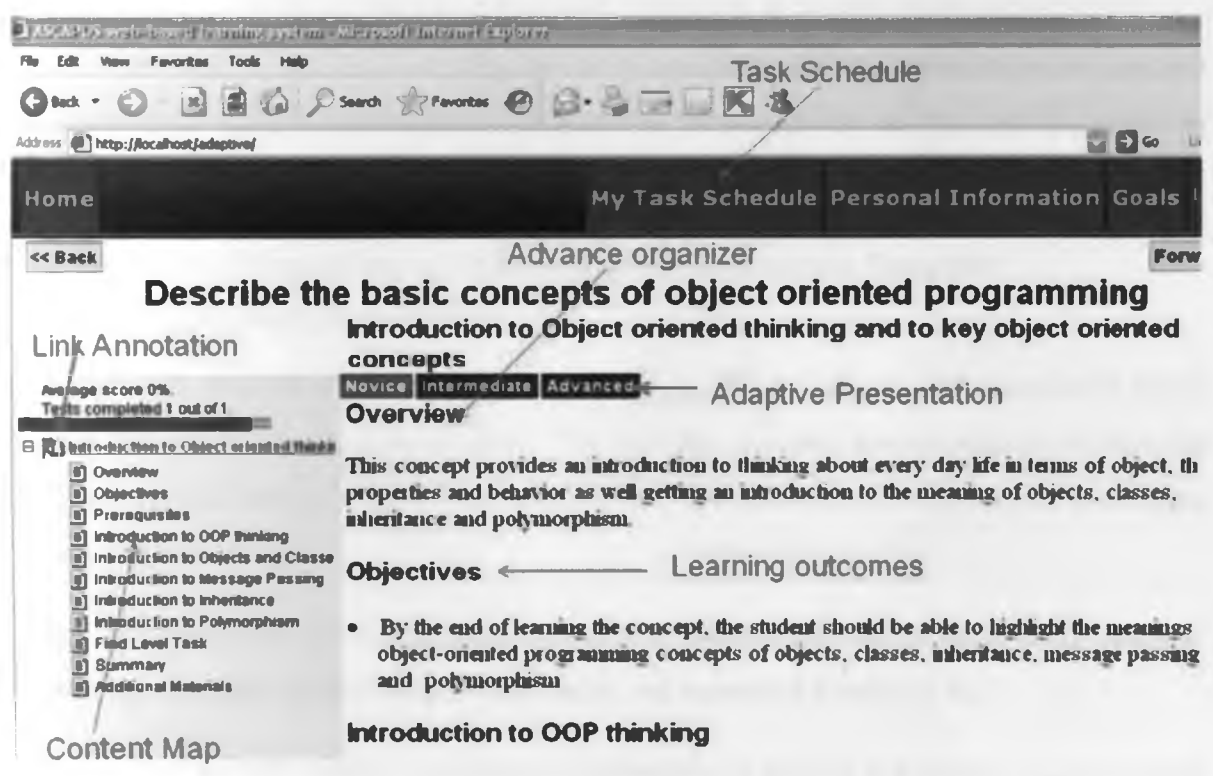


Figure 4-4 : Adaptive link annotation and adaptive text presentation using initialized scores

If the scores suggested a **lower level of readiness than the pre-test**, then the level of readiness was followed. If the scores suggested a **higher level of readiness than the pre-test**, then the pre-test level of readiness was followed. This was intended as a double check in order to avoid giving students suggestions based only on the pre-test level, which could possibly be due to guess work or a slip while selecting the answers (Virvou & Tsiriga, 2004).

- **Evaluation of HVDM**

1. The predicted scores were automatically saved in two locations in the database.
2. One set of the scores was updated when the student actually took the concept evaluation test. The scores were also used to update the students' overall level of knowledge.
3. Another set of concept evaluation scores was kept separate for comparison with the spare set of initial scores.
4. The percentage accuracy was determined by comparing corresponding initial and concept evaluation scores. The number of **tallying scores was divided by the overall number of initialized scores**.

4.2.2 Adaptive Presentation of Additional Learning Materials Using NBC

(a) Background and the use of NBC

NBC has one class variable and a number of other variables, also called attributes or features, which are used to describe an instance. The instance variables are assumed to be mutually independent i.e. the occurrence of one variable does not determine the occurrence of another, given the class variable.

Class variable, which can also be expressed as $f(x)$, takes values from a finite set of values V , which are normally distinct categories or classes. An example is the set $\{Y, N\}$.

The formula for NBC used to calculate the probability of each of the classes, V_{BN} , is as given below (Mitchell, 1997):

$$V_{NB} = P(v_j) \prod_{i=1}^n P(X_i/v_j)$$

Where $X_i \in X$, $X = (X_1, \dots, X_n)$ is the set of instance variables.

$v_j \in V$, $V = \{\text{Yes, No}\}$ is the finite set of possible classes

The calculation of the probabilities of the classes i.e. V_{NB} involves:

1. calculation of the prior probabilities of each of the classes i.e. $P(v_j)$ and
2. calculation of conditional probabilities of each attribute value, given the class v_j i.e. $P(X_i/v_j)$
3. calculation of the product of the various $P(X_i/v_j)$ values when $V = v_j$ together with $P(v_j)$

The class with the highest probability is assigned to the query instance $X_q = (X_1, \dots, X_n)$.

(b) Why select NBC?

NBC was considered suitable for online learning, which requires real time adaptation as the learner continues with learning (Stern et al., 1999). Adaptive learning support should be provided to a learner right from when he starts to learn (Virvou & Tsiriga, 2004) and NBC was selected because it needs very little data to do classification (Stern et al., 1999). Moreover, besides NBC being a very intuitive technique, it also has very few parameters to be set, unlike other approaches such as Artificial Neural Networks (Mitchell, 1997).

(c) Clicking on Links indicates Intention to Use Materials

The links to additional materials were well labeled with descriptive text to indicate what the nodes underlying the links were about. Therefore, clicks on the links were assumed to indicate intention to read the materials (Sia, Zhu, Chi, Hino, & Tseng, 2007). The click history of the current student and the other students was used to calculate the probability of the student wanting to access the additional materials (Oboko et al., 2009c).

(d) Considering the challenges in using Machine Learning for adaptive interfaces

- **Learning time**

NBC can make quick computations for decision making, and it is therefore suitable for supporting online learning with adaptive navigation support (Stern et al., 1999).

- **Quantity of data needed**

NBC can start making adaptations after a few interactions as it requires very little data to start making predictions (Stern et al., 1999; Wolf, 2004).

- **Problem Formulation**

The problem was represented in terms of the task (T), experience (E) and performance measure (P).

T was to learn the category of the current learner for the current concept in terms of preference to use additional learning materials (Yes or No).

E consisted of the feature vector values of how the other students had accessed the additional learning materials (Y or N) for all the concepts preceding the current concept being studied by the current student.

The students selected for this purpose were those who had studied up to the current concept being studied by the current learner.

P was the percentage accuracy of the predictions made by NBC. Predictions were compared to the actual usage or not of the additional materials for each of the concepts. The percentage accuracy was determined.

- **Representation of the instances and training examples**

Training examples were represented as vectors of attribute-value pairs, having the format *<concept 1 value, concept 2 value ... concept n-1 value> <Concept n>*. Value was Y or N. *Concept n* was the current concept being studied by the current student for which a classification value of Y or N was being sought. The value of *concept n* for all the other learners was the **class** of the

training examples and the values of *concept 1* to *concept n-1* were the **instances** of the training examples.

Student n = <concept₁, concept₂, ..., concept_{n-1}><concept_n>

If n=10 then the vector could be as shown below for one student:

Student 1 = <concept₁=N, concept₂=Y, concept₃=N, concept₄=Y, concept₅=Y, concept₆=Y, concept₇=Y, concept₈=Y, concept₉=Y ><concept₁₀=Y >

A query instance which represents information of the current learner could be represented as:

Student Q = <concept₁=N, concept₂=Y, concept₃=Y, concept₄=Y, concept₅=N, concept₆=Y, concept₇=N, concept₈=Y, concept₉=Y ><Concept₁₀=?>

The query instance does not have the classification part of the vector.

(e) Collection of the data used by the algorithm

The data used by this algorithm was collected non-obtrusively. As the learners clicked on additional learning material links, a log was made. If the learner submitted a concept evaluation test without having clicked on the additional learning materials, then access = N was added to the log for the student for this concept.

(f) Evaluation of NBC

1. Determination of the number of times the predicted use of additional materials was the same as the actual usage.
2. This number was divided by the number of all the predictions made to determine the percentage accuracy of the algorithm.

4.3 Summary

The chapter described how Machine Learning algorithms were applied to adaptive user interfaces. Adaptation was intended to provide hints to students to assist them in their navigation through the course. The hints were also intended to assist the students in the selection of the appropriate presentation of a concepts' knowledge which they were to study,

based on a student's current level of knowledge. The chapter described how adaptive user interface components were specified. These components needed to be specified as a pre-condition for successful application of Machine Learning algorithms to adaptive user interfaces. The application of the Naïve Bayes Classifier (NBC) and the Heterogeneous Value Difference Metric (HVDM) algorithms in this research was also presented in detail.

I would rather lose in a cause that will some day win, than win in a cause that will some day lose!

-Woodrow Wilson

Chapter 5 : Methodology – Experimental Design

5.0 Introduction

The purpose of this research was to investigate the effects of (a) using adaptive learning support features and (b) techniques for scaffolding metacognitive control, on (i) students learning performance as measured by test scores, (ii) students' perceived value of the adaptive features of the system, (iii) level of application of problem solving processes, (iv) level of metacognitive awareness and (v) quality of the product of problem solving. The research was carried out in the context of a web-based learning environment.

Two experimental research studies were carried out. The first study was on the effect of adaptive learning support on learner performance and perceived value of the adaptive learning environment. The second study was on the effect of techniques for scaffolding metacognitive control, in particular, use of question prompts and task schedule, on the application of problem solving processes, metacognitive awareness and quality of the product of problem solving.

5.1 Participants

The students who participated in this study were first year students of BSc Information Technology from two campuses of one of the public universities in Nairobi. The students were studying the course called Introduction to Object Oriented Programming. They had already taken a pre-requisite course Programming Methodology but they had never formally been taught or used the problem solving approach.

At the beginning, there were 92 students. Their ages approximately ranged from 18-20 years. 22 % were female while 78 % were male. 75 % were from the main campus whereas 25 % were from the Nairobi Campus.

Judgmental non-random sampling was used to select 2 out of the 8 campuses of the university. The campuses had a large population of students taking the desired course. The goal was to involve at least 90 students because at some point during the research, 3 treatment groups

were to be used. In order to assume normality, it is good practice to have groups with at least 30 members.

Out of the initial 92, only 89 actually registered in the system at the beginning of Study 1. Not all of those who participated in the study took the tests and the perception evaluation survey.

The students who had studied domain concepts in study 1 are the ones who participated in this study. There were 66 participants in both pretest and post test of both metacognitive awareness and problem solving processes. Out of these, 15 students were picked to participate in the in-depth qualitative comparative case study.

The evolution of the sample size through the various stages of the research is shown in Table 5-1. The table shows the number of participants at different stages during the research, from the beginning of study 1 up to the qualitative study. For each stage, the table shows the specific events that took place, the number of participants involved, the percentage reduction in number of participants and the possible reasons for this reduction. For example, the students who completed the post survey in study 1 are fewer than those who took Test 1 and Test 2 because it was voluntary though its importance had been emphasized.

Table 5-1 : Evolution of the Research sample Size during the Research

Study	Research Level	Level Item	Sample size	Percentage Reduction	Remarks
Start	Beginning	Original sample	92	0%	• Total sample size at the beginning
Study1 (Week 1)	Online Registration	Registered users	89	3.3	• Drop out due to lack of fees and change of program of study
(Week 5)	Post Tests	Test 1	84	8.7	• It took some students time to adjust to learning online
(Week 8)		Test 2	85	7.6	• It took some students time to adjust to learning online
(Week 5, 8)	Perception evaluation	Post Survey	81	11.0	• Many missing values. • It was made clear that it was optional and did not contribute to the semester grade so some opted out
Study 2 (Week 9)	Metacognitive knowledge	Pre-test	80 (66)	13.0	• Reduced to 66 due to missing values and • Effort to match pre-test and post test for both problem solving processes and metacognitive knowledge. Some did one test and not the other
(Week 12)		Pos -test	77 (66)	16.3	• Same as pre-test
(Week 9)	Problem solving processes	Pre-test	80 (66)	13.0	• Reduced to 66 due to missing values and • Effort to match pre-test and post test for both problem solving processes and metacognitive knowledge
(Week 12)		Post-test	78 (66)	15.2	• Same as pre-test
(Week 12)	Product Evaluation	Score from score sheet	74	19.4	• Some opted not to appear because it was a challenging exercise due to the nature of ill-structured problem solving
Qualitative cross-case comparative study	Individual think aloud session	Code analysis	15	-	• A small number was selected through judgmental sampling due to the intensity of the process, time taken and huge amount of data generated per individual participant.

5.2 Overall Research Design

The overall research design was mixed with two experimental studies and a qualitative comparative case study as shown in Table 5-2.

Table 5-2 : Overall Research Design

Research Question	Data Collection Approach	Materials and Instruments	Scale Used	Data Sources	Data Analysis
RQ1	Experimental study with control group, randomization and post test. Had 2 groups	<ul style="list-style-type: none"> • Short written exam • Scoring rubric for the written exam • Online course content with adaptive support 	Ratio scale	2 written test scores	t-test
RQ2	Experimental study with control group, randomization and post test. Had 2 groups	Self-reported post survey	7 point scale Likert scale on ordinal scale	Post survey	Mann-Whitney test
RQ3	Experimental study with control group, randomization, pre-test and post-test. Had 3 groups	<ul style="list-style-type: none"> • Complex problem solving tasks • Online task schedule • Description of problem solving processes • Task Schedule scaffold • Complex problem solving skills survey 	7 point Likert scale on ordinal scale	Pre-survey and post survey ; Coded and analyzed think aloud protocols	Kruskal-Wallis test
RQ4-A	Experimental study with control group, randomization, pre-test and post-test. Had 3 groups	<ul style="list-style-type: none"> • Complex problem solving tasks • Question prompts • Description of problem solving processes • Complex problem solving skills survey 	5 point Likert scale on ordinal scale	Pre-survey and post survey ; Coded and analyzed think aloud protocols	Kruskal-Wallis test
RQ4-B	Experimental study with control group, randomization and post-test	<ul style="list-style-type: none"> • Ill-structured problem-solving project task • Scoring rubric for the problem-solving product 	Ratio scale	Problem solving project score	ANOVA

Experimental study was used because of trying to influence the dependent variables through interventions to find out the effect of design factors under study. Study 1 was a quasi-experimental design because participants were placed in 2 groups randomly and they took a post test, but there was no pre-test. This was to avoid examining students at the beginning on knowledge of OOP which they had not been exposed to, and therefore, this could cause them a lot of cognitive load. This is in addition to the fact that OOP is said to be a challenging domain (Boyle, 2003; Burkhardt, Tienne, & Wiedenbeck, 2002). The study had 2 groups: a treatment group and a control group. Both groups were exposed to adaptive features treatment at different points in time. The course was divided into 2 equivalent parts so that one group could use adaptive features when learning the first part and the second group could use the features when learning the second part.

Study 2 was a true experimental design with pre-test, post test, 2 treatment groups and a control group. The participants were randomly assigned to the groups. The number of groups was determined by the 2 treatments (task schedule and question prompts) and need for a control group. It was possible to do a pre-test because the questions were stated in a non-technical way and they were based on metacognitive knowledge and problem solving, which people apply to some extent in everyday life (Schraw, 1998).

The qualitative study which followed immediately after study 2 was an intensive case-based qualitative study for observing student thought and decision making during problem solving. It was for validating the results of the quantitative study 2 and also gave the students the chance to apply both recall (study 2) and task performance (qualitative study) as part of evaluation of learner cognitive processes.

As shown in Table 5-2, research Question 1 and 2 were answered by study 1 whereas Question 3 and 4 were answered by study 2 and the comparative case study. The results of study 2 that was quantitative were triangulated with the results of the comparative study that was qualitative. For each research question, the table shows the relationship between the research questions, the corresponding study design for answering the questions, materials and

instruments for the study, the scale used for data collection, sources of data and data analysis techniques used.

5.3 Study 1 – Learning OOP Concepts with Adaptive Support

5.3.0 Introduction to Study 1

The students were introduced to the study and were given an explanation on how they were expected to participate. They were given assurance that their privacy would be assured throughout the experiment and that the responses they gave, especially in surveys, were not going to affect their end of semester course grade.

The students were provided with programming tasks. All students were randomly assigned to one of the five ill-structured problems. Each one of them was provided with a copy of the problem which he was expected to solve in the subsequent study (Study 2). It was provided in advance to focus the students' attention. Therefore the students were not only made to learn the OOP course concepts available through the system but were also given an early opportunity to think of how to apply the concepts to solve the problem at the end by (Burkhardt et al., 2002).

To ensure an ethical study, a number of measures were instituted. Some of these are: (i) all students used the adaptive features of the system, but at different stages of learning domain concepts; (ii) each student had his own online space that was password protected to assure privacy and confidentiality; (iii) the students were given freedom not to participate in the perception survey which was purely for research, though they were informed that participation was important; (iv) all students were introduced to C++ programming for 2 weeks, without object orientation concepts, and were also given a demo on how to use the system's features for learning, to ensure that all were at the same entry level and ready to participate in study 1; (v) the students were informed that the purpose of the study was purely academic research.

5.3.1 Design of the Experiment

The experiment was designed to answer research questions 1 and 2. The students were randomly divided into two groups. The duration of the treatment was 8 weeks. The design is shown in Table 5-3.

Table 5-3 : Experimental Design for study 1 showing groups, treatments and observations

	Class Design	Posttest1	Inheritance	Posttest2
Group 1	X – Learning system with adaptive support	O ₃	Learning system without adaptive support	O ₅
Group 2	Learning system without adaptive support	O ₄	X – Learning system with adaptive support	O ₆

For the class design part as shown in the table:

Group 1 with Treatment 1: Students learned using adaptive learning support

Group 2 with Treatment 2: Students learned without using adaptive learning support

A posttest i.e. class design test was taken immediately after treatment to capture the treatment effect. A brief post survey was applied on group1 immediately after the posttest to collect information on student perception of the system.

For the inheritance part:

Group 1 with Treatment 2: Learning with the learning system, without adaptive support

Group 2 with Treatment 1: Learning with adaptive support

A posttest i.e. inheritance test was taken immediately after treatment to capture the treatment effect. A brief post survey was applied on group 2 immediately after the posttest.

All the students in both groups and treatments were provided with the same assessment questions.

5.3.2 Treatments

Adaptive support was provided in the form of adaptive navigation support and adaptive presentation support.

Adaptive navigation support through link annotation was used for suggesting the concepts of a goal which the student was adjudged to be ready to learn, depending on his current knowledge level.

Adaptive navigation through link hiding was provided through dynamic display or hiding of additional learning materials for a concept.

Adaptive presentation support was provided through the system annotating the level of content difficulty, also called level of performance for a concept's content, which was considered suitable for the student, based on his level of knowledge.

Without adaptive support, the system provided the student with all the learning goals, presented in the default order, from simple to complex. When a goal was selected, all the concepts in all the layers of the goal were displayed. Also when a concept was selected, all the knowledge modules for the three levels of performance were presented, together with their pre-requisite concepts and the additional learning material.

5.3.3 Measures to Assure Validity of Results

- i. Random allocation of students to the two groups
- ii. Students from the two groups were allowed to use the adaptive features of the system, though at different points in time.
- iii. Students were given prior explanation on how to use the system, a demonstration of how to use the different parts of the course, and a brief manual with steps on using it.
- iv. The students were informed that the tests were timed. Therefore, they were asked to attempt tests only when they were ready. Figure 5-1 shows a message to the learner about number of attempts allowed (1 attempt) and maximum time per level of the pre-test (3 minutes per level). The message is displayed at the beginning of the pre-test and

if the learner is ready, he selects “ok” and takes the test. If he is not ready, he can select “cancel” in order to take the test later on.

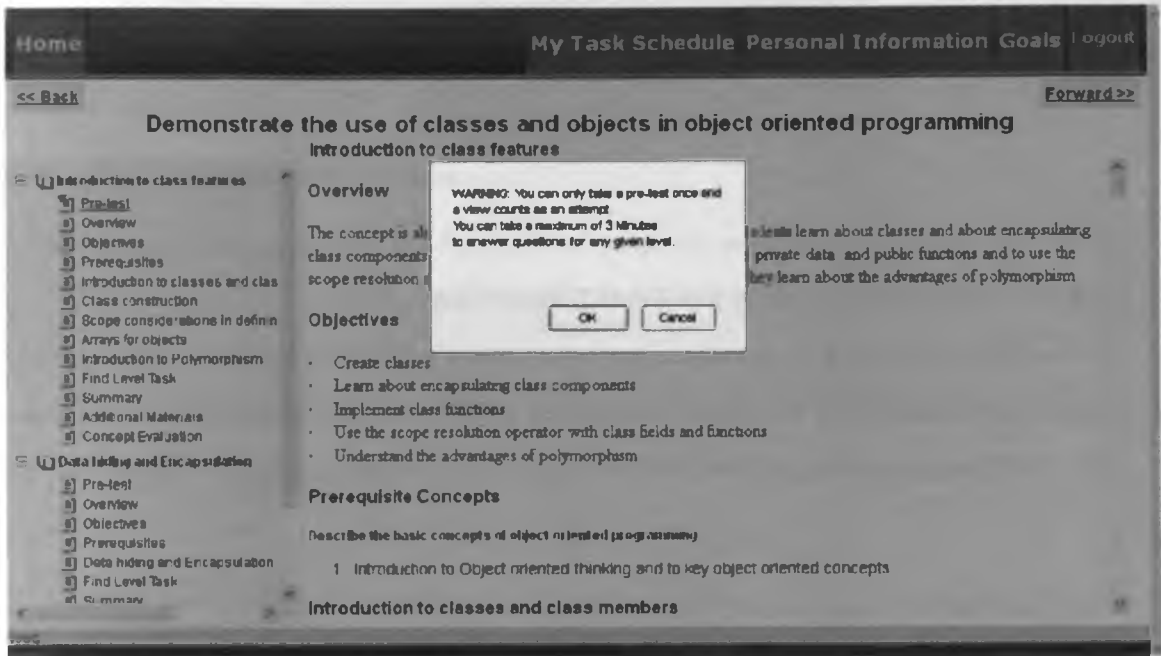


Figure 5-1 : Warning message about number of attempts and time per level of the pre-test

- v. The students were allowed one attempt per test.
- vi. The students were given two weeks of introduction to programming using C++, without object oriented concepts, to ensure the entry level was high and uniform.
- vii. The students from the campus that did not have adequate computing resources were provided with a dedicated computer laboratory available throughout the day for accessing the course.

5.3.4 Instruments

The instruments used for Study 1 were the posttest after class design, also called posttest1, the posttest after the inheritance part, also called posttest2 and a post survey.

(a) The posttest1 (Class design test) and posttest2 (Inheritance test)

The two posttests i.e. posttest1 and posttest2 were designed to assess the students' level of mastery of the different concepts of the OOP domain.

Details on the construction and validation of the posttests are provided under course development in section 3.3.1.

Validation of the experiment's test scores

The students were asked to go over the marked test scripts to see if there was a problem with tallying or with marking. The students could check if there was erroneous award of marks per question or erroneous tallying. They were also asked to compare each other's scripts to see if there was erroneous marking. They raised the issues they found and corrections were made. Then, they returned the marked test scripts. This formed the first stage of validation of the test scores.

After that, a lecturer of OOP from a different university, where the experiment actually took place, validated the marking. The lecturer took a sample of 20 student scripts out of 89 (22.5%) and remarked them, checking for issues such as erroneous tallying, awarding the wrong number of marks, erroneous marking i.e. the student's answer was marked as correct yet it was not or the answer was marked as wrong yet it was correct.

For the class design test, out of the 20 test scripts the validating lecturer selected randomly, 9 (45%) did not need re-marking for section A. Out of section B questions, Questions 5 out of 7 were to be remarked. Out of section C questions, 2 out of 6 were to be remarked. For the inheritance test, out of 20 test scripts selected randomly for validation, 13 (65%) did not need remarking for section A. For section C, questions 4 out of 6 questions were to be remarked.

The lecturer also observed that the researcher had been generous in marking the tests and gave pointers to that effect.

The issues raised were summarized and used to guide remarking of the two tests for all the students. The new scores were considered final scores for the tests.

(b) The Post Survey

Instrument Construction

In coming up with the post survey, a previously used survey on the students' attitude towards features for providing adaptive learning support in a web based learning system used for teaching a course in Computer Science was consulted (Papanikolaou et al., 2003). Most of the questions used for collecting data about students' attitude towards the system's adaptive features, organization and other system features were adopted but tailored to this research. A question was added on how the students felt about the feedback after concept evaluation for each concept. Two questions for collecting data about general satisfaction with the system were adopted from the survey on user satisfaction and technology acceptance (Wixom & Todd, 2005).

For the open-ended questions, an existing post survey on attitude towards an online learning environment that was used to teach OOP with Java was consulted (Schmidt, 2005). The post survey of Schmidt tapped information on the things the students liked or disliked about the learning environment. Two questions were added specifically on what the students liked and disliked about the interface of the system, which included adaptive navigation and adaptive presentation support features. The questionnaire is shown in Appendix M.

Validation and reliability

The questionnaire was validated by an expert in the design of questionnaires and other instruments for research at the School of Computing and Informatics. Then the questionnaire was administered on 33 students of an OOP class at the School of Computing and Informatics. The responses had a Cronbach Alpha of 0.77.

5.4 Study 2 – Problem Solving with Scaffolding for Self-regulation

5.4.0 Introduction

This study was used to answer research questions 3 and 4. The research design was mixed; it had a **quantitative experimental pretest and post test with control group** part and a **qualitative comparative case study** afterwards.

The **experimental study** was to measure the effect of scaffolding on the students' problem solving skills and metacognitive awareness. The student's problem skills were defined in terms of level of application of the processes and quality of the product of problem solving. The **comparative case study** was to provide more insights into students' problem solving processes and their metacognitive awareness, as influenced by the use of question prompts and the task schedule.

The pretest was a self-report survey to estimate the students' problem solving skills and problem solving strategies at the start of problem solving, together with the pre-experimental level of metacognitive awareness, for all the participating students. The posttest was a post-experimental application of the metacognitive awareness inventory and a survey on the level of application of problem solving skills and strategies. An oral examination followed in which students were required to describe, explain and justify the process they followed when solving the problem. There was also an evaluation of the developed system, demonstrated by the students. The data collected was used to further determine the extent of success of the students in problem solving, in addition to the responses of the students in the survey.

The qualitative part of the study constituted the use of think aloud protocols on a small group of students. The students were selected from among the students in the different treatments of the study using judgmental non-random sampling.

- 1) At the beginning of the study, the students were provided with an explanation of the purpose of the study, procedure to be followed, how they were expected to conduct themselves, how the lab sessions were scheduled, what was expected of them at the end of

problem solving and how long the experiment was to take. They were also provided with a fresh copy of the programming task (see Appendix G for a sample task) and a brief description of the problem solving process (see Appendix K). The students who were members of question prompts group were also given a copy of question prompts to be used (See Appendix J). The other students who were members of task schedule group were given a copy of the scaffold for the preparation of the task schedule (See Appendix I) and access to the online schedule. The rest received neither a copy of the question prompts nor a copy of the scaffold for preparing a task schedule.

To ensure an ethical study, all students were introduced to ill-structured problem solving processes. They were also given a demo on how they could solve a real-life problem. So, even members of the control group had an idea of how to carry out problem solving. The treatment effect was measured over and above the effect of having knowledge of problem solving. It was related to the effect of prompts or hints provided to students while solving the assigned problems. All students were provided with both a soft and a hard copy form of the problem solving processes for reference as may be necessary. Students were also provided with clear instructions for the study and were briefed as one group on what was expected of them during the study. At the beginning of the qualitative study, each student was given an explanation and demo of how to perform think aloud.

5.4.1 Design of the Experiment

The participants were randomly divided into 3 treatment conditions: using question prompts, using a task schedule and not using any of question prompts and task schedule.

At the beginning, all the students completed self-report questionnaires on their level of problem solving processes and metacognitive awareness.

The experiment was run for 4 weeks, during which they engaged in solving the ill-structured programming task, without intervention from the researcher.

At the end of the 4 weeks, the students attended an oral examination, demonstrated the systems they had developed and also completed a metacognitive awareness inventory and a problem solving skills survey.

The experimental design showing the groups, treatments and observations for study 2 is shown in Table 5-4.

Table 5-4 : Experimental Design showing groups, treatments and observations for study 2

Group	Pretest	Treatment	Posttest
Experimental group1	O ₁	X ₁ - Use of Task Schedule	O ₂
Experimental group2	O ₃	X ₂ - Use of Question prompts	O ₄
Control group	O ₅		O ₆

5.4.2 Treatments

This study had 3 treatments. The difference among them was the scaffolding technique used by the students within each group during problem solving.

Treatment 1: As shown in Table 5-4, the members of this group used the task schedule as a source of scaffolding. The task schedule was an online space for each student in which one could key in the various aspects of their task management, such as the sub-tasks, with information on resources needed to solve each one of them, including duration of time in hours, and OOP knowledge and skills required.

Training: The members of the task schedule treatment group were given a demo on how to use the task schedule i.e. how to create and update a task schedule, reorder tasks, and how to interpret the graphic, especially the meaning of the different colours.

Creating own Project: Each member of this group was asked to create his own project and a task schedule for the project. The students were asked to break down the problem into sub-tasks and enter each sub-task into the schedule, in the order in which the sub-tasks were to be carried out. Each sub-task in the schedule had resources such as time required in hours, the

knowledge and skills needed, and an estimate of the student’s mastery of the knowledge and skills at the beginning, expressed as a percentage. The students were asked to set the initial percentage completion status for each sub-task to 0 %. Figure 5-2 shows the page used for capturing details of a new sub-task in the task schedule. These details include name of sub-task, whether it is a major sub-task or not i.e. a milestone, start date and duration of time for the sub-task in hours.

Figure 5-2 : Screen for capturing sub-tasks in the task schedule

Viewing task progress: Whenever a student in this group logged into the system, he could see his schedule of tasks. The schedule displayed details such as the name of the task, expected start date, expected end date, duration of time for each task in hours, preceding task, whether complete or not, current progress estimate (as a percentage), elapsed duration in hours, graphic in different colours to show current status, and date of last update.

Update task of schedule: The students were reminded from time to time of the need to update the task schedule with actual progress, expressed as a percentage of completion of each sub-task. Every time some work was done on the sub-tasks, the students were expected to update the field showing percentage completion of the sub-task.

Comparing performance with the plan: The students were asked to compare the actual progress against the plan of the task-based activities, whenever they were logged into the task schedule. If the progress was not according to plan, the concerned student was encouraged to ask himself why this was so and what could be done to improve. The students were asked to implement what they thought could improve their progress. At the end, the students were asked to pass judgment on their products.

The features of the task schedule scaffold are shown in Figure 5-3 . It shows sub-tasks, a section for managing sub-tasks, a section for managing knowledge and skills for each sub-task and a graphic for visual display of progress per task, among other features.

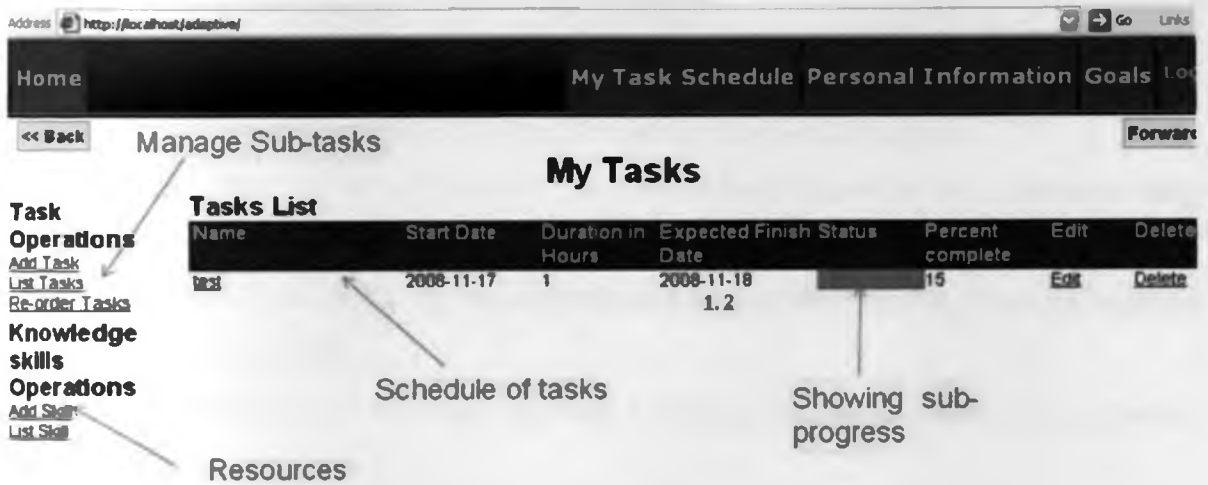


Figure 5-3 : Features of the Task Schedule Scaffold

Treatment 2: As shown in Table 5-4, the members of this group used question prompts as a source of scaffolding. The students were provided with a hard copy set of questions expressed using OOP terms, for each phase of problem solving. The students were asked to think about the questions/ask themselves the questions as they carried out the different processes of problem solving.

Each member of the group was asked to use answers to the questions, which were generated from within oneself, to decide if he was progressing well or not. If necessary, the students were

asked to make decisions on what actions to take to make sure there was improvement in their progress, so as to remain on course to solving the problem successfully within the time allowed.

From time to time, the students were reminded that they needed to have the question prompts in front of them as they solved the problem, and to think about the questions for the phase being done at the time.

NOTE: Detailed description of the construction of the task schedule and question prompts is in section 5.4.4

Treatment 3: This was the control group. As shown in Table 5-4, the students in this group were not required to use questions prompts or task schedule.

5.4.3 Measures to Assure Validity of Results

- i. All students were provided with a copy of the problem solving process
- ii. The students were given a demonstration at the beginning of the study on how to solve an ill-structured problem
- iii. The students were given an explanation and demonstration on how to use the scaffolds which were provided
- iv. The students were reminded regularly to keep using the scaffolds during problem solving
- v. The students were asked to attend every laboratory session
- vi. The students were given clear instructions on how to use the data collection instruments, including the think aloud method
- vii. The students were randomly assigned to the treatment groups. Students from the two campuses were put in one list and randomly assigned to the 3 groups.
- viii. At the beginning of the study, the students were made aware of how they were to conduct themselves and all the deliverables expected of them at the end of the study

5.4.4 Treatment Materials and Instruments for the Experimental Study

(i) The Ill-structured Programming Tasks

The programming tasks were ill-structured problems. Each of the tasks described a scenario but did not provide enough information on how the problem could be solved. The students were only asked to solve their problems as individuals, using OOP concepts and that there was no correct solution, as long as the solution generated was logical. Five problems were developed.

Instrument validation

Two lecturers were asked to look at the problems and indicate if they were ill-structured. They were also asked to indicate whether the problems were at the right level of difficulty to evaluate application of the concepts of the course. The lecturers were also asked to indicate whether the problems could provide the students with an opportunity to apply most of the concepts learned.

(ii) Brief summary of Problem Solving Process

All the students were expected to apply the problem solving approach to their ill-structured tasks. In order to ensure that students were ready for problem solving, a summary of the process of problem solving was provided to all the students in all the treatment groups. The researcher also explained the problem solving process to the students. This was after the pre-test for the study and just before the students began solving the problem.

(iii) Question Prompts

Construction of the question prompts

The researcher referred to 10 general question prompts generated by experts for a course called Introduction to Information Science and Technology. The prompts were used in a study involving ill-structured problem solving (Ge, 2001) to guide the design and implementation of the prototype of a proposed system. It was felt that the questions could also apply to this study because they were general enough, and were related to the development of a system.

The questions were organized according to the 4 problem solving processes. They had been constructed to support cognitive thinking and metacognitive skills in the students.

The researcher used the 10 general questions as reference when generating 18 specific questions related to OOP tasks. These specific questions were also organized according to the four problem solving processes.

Validation of question prompts

The questions were validated by two subject matter experts, who were also course lecturers, to find out whether they supported the programming task. The experts were asked whether they could ask themselves such questions if they were solving such a problem. Their comments were used to improve the questions.

(iv) Task Schedule

The task schedule was used by students in one of the treatment groups for the planning and management of their processes for solving the ill-structured problem. The task schedule also reminded the students of their current progress in each of the sub-tasks.

Instrument construction and validation

The researcher reviewed a number of project planning and management tools in order to identify features which could be useful in assisting students to manage their problem solving processes. In particular, Microsoft Project from Microsoft Corporation and Dot Project which was open source were reviewed.

Some of the features identified include:

- Each student should have one project
- Each project should have a number of tasks to be carried out in order to accomplish the project i.e. problem to be solved
- Each project needs to have some information that is needed in order to start solving the problem i.e. initial knowledge and skills

- From time to time, each student should update the project with progress, based on success achieved for each individual tasks i.e. sub-task progress
- From time to time, each student needed to be reminded of one's current progress on each individual sub-task, in the form of a report, using different colours on a task progress chart or prompts based on task-progress
- Each student needed to act on the feedback provided in order to ensure that progress in problem solving was good and that the solution was of good quality, achieved using the least amount of resources, including time
- Each student's activities, especially task level activities, needed to be logged into the task schedule, such as updates on percentage progress, task re-ordering and task description

The interface components, database components and the interaction between the student and the task schedule were designed and implemented.

A guide was developed on how to prepare and use the task schedule, including the meaning of the different features of the schedule used for showing task status. It was intended to scaffold the students' usage of the schedule by making the process more obvious and tractable to the students (Quintana et al., 2004).

(v) The Metacognitive Awareness Inventory

Instrument construction

Schraw and Dennison (1994) developed a Metacognitive Awareness Inventory (MAI) to be used to assess the student's level of awareness of their metacognition. It had 52 items covering the eight elements of metacognition, measured on a 7-point Likert scale. This MAI was adopted for this research. The MAI is shown in Appendix N.

Validation and reliability

The developers reported that the MAI had a Cronbach Alpha reliability level of between 0.88 and 0.93 for the 8 factors and 0.90 for the entire instrument, when tested with 197 college students of an introduction to educational psychology course. For this study, a Cronbach Alpha reliability for the entire instrument was calculated as 0.931 for pre-test and 0.958 for the post-test.

(vi) Problem solving processes self-report

Instrument construction

The questionnaire was a modified version of the questionnaire developed by Ge Xun (2001), based on the work of Alan Schoenfeld on understanding and teaching Mathematics as problem solving. This work was referenced by Hong (1998) in developing a questionnaire for assessing problem solving skills of students engaging in the development of a prototype system using the problem solving approach.

The questionnaire had two parts, with the first part asking for background information and the second part requiring students to report on their problem solving skills.

It contained 20 questions grouped as follows: questions 11-15 tapped interpretation and problem representation skills; questions 16-20 focused on skills for developing solutions and monitoring the solution processes; question 21-25 on making justifications and evaluating of the students' problem solving processes; questions 26-30 on the students' specific strategies for problem solving.

The first part on background information was left out since this information had already been collected from the students using other instruments and was therefore available. The 5-point Likert scale was retained. The instructions were re-written to suit the context of the study.

Validity and reliability

During formative evaluation of the system, the questionnaire was filled in by 33 students of an OOP class at SCI, who used the prototype to study the course. The responses had a Cronbach Alpha reliability of 0.867.

(vii)The product evaluation score sheet

Instrument construction

The programmed system was the end product of the programming project, which all the students did as individuals. The system was marked out of 20.

Three lecturers, including the researcher, sat down and borrowed ideas from the process normally used to evaluate student projects at SCI, to aid them in identification of factors to consider when grading student projects.

The two major aspects: After looking at various possible factors and discussing them, the lecturers decided that marks were to be awarded for the **demonstration of the product** by the student (10 marks) and the **application of OOP concepts** in the solution's written programs (10 marks). For each aspect, a number of supporting factors were considered during evaluation, as shown in Table 5-5. For example, in evaluating the system demonstrated by a student, if the student could show that it was running and already had some data, the score was 1. If the system accepted sample data selected by the student during the demonstration, the score was 2. If the system also accepted ad hoc data as required by the evaluators during the demonstration, then the score was 3.

Table 5-5 : Aspects of product evaluation and their corresponding factors

Aspect of Evaluation	Factors considered	Marks
The demonstrated product	• Did the system run when data from different scenarios were tried on it?	3
	• Was the system usable i.e. how easy was it to use the system ?	3
	• To what extent was the system complete i.e. how far had the student progressed in solving the problem ?	2
	• How was the user interface for the system?	2
Application of concepts	• Were the programs clean and simple i.e. could one follow or figure out the code – (i) how the segments followed each other, (ii) how the main{} function was used to create objects and call the methods and (iii) attributes of different classes	2
	• Was there proper use of classes i.e. was there proper use of constructors, variables, objects, polymorphism, message passing, memory management (proper use of destructors)?	2
	• Was there proper use of inheritance – (i) use of the access specifiers to bring about the intended access in the child classes, (ii) declaration of a child class, including access specifier and listing of parent classes, (iii) proper access within the child classes of the methods and attributes of the parent classes	2
	• How was general concept implementation done – (i) how indenting was done, (ii) how variable names were formed, (iii) how were comments included, (iv) whether the system code was organized into header and implementation files with header and implementation sections	4

Validity and reliability

- Validation of the product evaluation instrument

The instrument was valid since it was done through consultative work involving 3 lecturers of the OOP course and was also guided by the practice of evaluating student projects at SCI.

- Validation of test scores

Two raters used the product evaluation score sheet to evaluate the programmed system presented by the students. At the end, the scores of the students were entered in a spreadsheet with the student's registration number and the name of the rater. The two records

were presented together in consecutive rows in the spreadsheet, with a third row next to them containing an agreed on score for the student.

For the data collected using this instrument, the Kappa inter-rater coefficient was calculated as 0.38. As a result, the differences between the ratings were discussed between the raters and the final student score was assigned using a procedure agreed upon between the two raters.

5.3 The Qualitative Study

5.3.0 Introduction

The qualitative study was conducted soon after the completion of the experimental study.

The study used the think-aloud method. Through this method, the researcher examined the students' thoughts and decision-making during the problem solving process (Ge, 2001). This way, the researcher was able to gain more information on how the scaffolding features provided affected the students' metacognitive awareness and problem solving processes during ill-structured problem solving.

With this rich data available, the researcher was also able to interpret the experimental study results better (Johnstone, Bottsford-Miller, & Thompson, 2006).

5.3.1 Participants in the Qualitative Study

15 individuals from the 3 treatment conditions participated in this study. Five students were selected from each group using judgmental non-random sampling (Johnstone et al., 2006), based on their performance in the domain concepts and consistency in attending laboratory sessions.

Since think aloud is a labour-intensive method, with a single subject getting engaged for up to an hour or even longer, the sample size of the subjects is usually small. Even then, small sample sizes can still be a source of valid information (Johnstone et al., 2006). Jakob Nielsen (1994) did an experiment to estimate the number of subjects needed to participate in a think aloud

experiment, and found out that as few as 5 subjects can yield adequate information about how people solve problems.

5.3.2 Data Collection Techniques

(a) Think-aloud protocols

Think-aloud protocols are **verbalizations of one's thinking process** (Ericsson & Simon, 1993) as one engages in problem solving. The protocols provide **evidence of the student's cognitive processes** such as thinking, reasoning, decision making, monitoring and evaluation processes. Since all the cognitive processes pass through **working memory**, the thoughts which the subject is developing to aid in problem solving can be captured and reported at the time they are formed (Johnstone et al., 2006).

The method was applied to all the 3 treatment conditions i.e. with question prompts, with task schedule and with neither question prompts nor task schedule.

The think aloud methods for collecting real-time data can also put a **heavy cognitive load** on the subjects because they not only have to solve the problem but also verbalize their thoughts, and this can be tough for some of the students.

To reduce the cognitive load, the problems the students were to solve during the think aloud sessions were drawn from **familiar contexts** such as a system for the library or the management of student registration information for a college.

Since the students were already **familiar with problem solving** and they were also **familiar with the contexts of the problems** to be solved during think aloud sessions, a kind of **retrospective effect** was achieved. This was expected to **reduce the student's cognitive load**, leading to the production of richer and more understandable protocols (Branch, 2000). The questions were also designed not to be too challenging or too simple to increase chances of collecting meaningful data.

5.3.3 Data Collection

(a) The protocols used

Two problems were prepared for use in eliciting the cognitive processes in the working memory of the subjects:

- 1) “You have been asked to develop a system for a small library using OOP concepts and apply the problem solving process”
- 2) “You are required to develop a system to manage student records for a school/small college such as your campus, using OOP. You should be able to apply the problem solving process”

(b) Administration of the Think-aloud Protocols

1) Time schedule

The selected students were notified of the time they were scheduled to participate in the study.

2) Purpose of the study

When a student arrived and was made comfortable, the purpose of the study was explained to him. The students were told that there was no specific solution that was best and that the process to be followed in solving the problem was important.

3) Instructions for think aloud

The student was asked to think of how he would solve the problem and verbalize the thoughts going through his mind while solving the problem.

4) Demonstration of the think aloud process

The researcher demonstrated how to think aloud while solving a problem. The student was also provided with a sample problem and a typed example of someone’s thought processes while solving the problem. The researcher ensured that the student understood how to do the think-aloud procedure before turning on the video recorder.

5) Videotaping

The student was informed that the proceedings were going to be videotaped so as to avoid losing any information.

6) During problem solving

As the think aloud session went on, the participant was reminded on a continued basis to continue talking or raise the voice.

5.3.4 Analysis of Qualitative Data

(a) Development of the codes for analysis

The data collected was used to answer research questions 3 and 4.

For purposes of question 3, a **coding scheme** was developed by following the classification of the elements of metacognition according to the model of Schraw and Dennison (1994). Each of the 8 elements was defined in operational terms i.e. each element had **operational constructs**. Each **construct** further had **example pointers**, which were specific things to be **looked for in the transcribed text**, to indicate the presence of the construct in the protocols.

Each construct was assigned a carefully designed **code**. The codes were designed such that someone who knew the coding scheme could readily associate the construct to the correct element of metacognition.

The codes were important for **tagging the text** during analysis in order to easily identify the elements of metacognition present in a student's thought processes during problem solving. The codes were also used when **aggregating** the frequencies of the occurrence of certain constructs **from bottom to top**, in order to conclude the extent to which metacognitive elements manifested themselves in the protocols.

The researcher iterated through the development of the codes, each time checking (i) if the pointers were **clear** enough to be used by someone who did not participate in their design, (ii) whether the constructs were **exclusive** and (iii) whether the constructs were **sufficient** to

identify the presence of a specific element of metacognition in the transcribed thought processes of a student engaged in problem solving. **Constructs** which seemed to refer to the same element were **combined**.

For research question 4, codes were also developed using the models of Voss and Post (1988) and Sternberg (2009). The codes were applied in a similar way for all the 4 **problem solving processes**. Each of the processes had **constructs** and **pointers**.

(b) Validation of the codes

Two raters were asked to use the codes separately to identify and tag metacognition and problem solving process constructs present in the protocols of each student, using software for qualitative analysis.

The codes identified by the two raters for each line of transcribed text were compared to see how the two raters coded the protocols, to establish the extent to which they had a **common understanding of the codes** developed.

One of the raters was the researcher. The other rater knew OOP well, and this was important since OOP was the domain in which problem solving processes were being applied.

He was briefly **introduced to the constructs**, their corresponding **codes** and the ways in which the codes could possibly manifest themselves in a protocol (the **pointers**). The rater had the qualitative analysis software installed on his computer and therefore he tagged the lines of text with appropriate codes independently.

In the process of introducing the other rater to the constructs and the codes, it was realized that there were some codes whose pointers were **not clear** or their explanations or pointers were **not clearly distinguished**. This was corrected. **Extra codes** were also introduced as a result of **splitting some of the constructs** which were already there, inspired by a better understanding while explaining the constructs to the other rater. The other rater also made suggestions for improvement.

After this, the researcher tagged one protocol using the **refined codes** as an illustration to the other rater. Further discussions followed, with some more changes being introduced into the instrument, especially changes to distinguish codes clearly.

The next step involved each of the raters **coding three protocols independently**. The tagged protocols were compared line by line for inter-rater reliability, using the Kappa coefficient. The agreement among the raters was low.

There were further discussions among the raters on what could be improved. Some of the constructs which were similar and which were consistently confusing were either combined or one of them was dropped. Those pointers which were not clear enough to bring out what the designer of the instrument had in mind were further clarified.

(c) Analysis of the protocols

The frequencies of the constructs in the tagged text were identified for students in the different treatment groups. The construct frequencies were further aggregated to find out how each of the elements of metacognition or problem solving process was manifested in the thought processes of the students in each of the 3 treatments.

The element frequencies were compared to the results obtained using the questionnaires in the post-tests for metacognitive awareness (8 elements) and problem solving processes (4 elements).

5.4 Summary

The research was carried out in two studies. Study 1 was a post test only experimental study with control group. The study assisted in answering research questions 1 and 2 and was based on the effects of adaptive learning support technologies on the students' cognition: student scores and perceived value of the features of the online learning environment. The experimental group used adaptive features, and the control group did not. Adaptive navigation support with link annotation was used to provide suggestions on the concepts the student was ready to study and those that he was not ready to study. Adaptive navigation support with link

hiding was used to dynamically hide or display additional material links according to the preference of the students. Adaptive presentation was used to guide the selection of an appropriate presentation of domain content to the student: the simplest or least challenging presentation was offered to those students with the lowest level of knowledge and the most complex presentation to the students with the highest level of knowledge.

Study 2 was mixed and involved a quantitative pre-test and post-test experimental study with control group, and a qualitative comparative case study. The study was carried out to answer research question 3 and 4. It was based on the effects of using scaffolding techniques for self-regulation on the student's metacognitive awareness and application of the ill-structured problem solving processes. The experimental groups used question prompts and also created and used a task schedule. The control group used neither of the two. The qualitative study was carried out soon after the experimental study on a few cases (15) selected using judgmental sampling from the three treatment groups. It was possible to study these cases using the think aloud method.

The whole secret of life is to be interested in one thing profoundly and in a thousand things well.

- Horace Walpole

Chapter 6 : Results and Discussion

6.0 Introduction

This chapter presents the results of data analysis from the experimental studies and the cross case comparative study. The results of study 1, which was intended to answer research questions 1 and 2, are presented and discussed first. The results of study 2 are presented next. Study 2 was intended to answer research questions 3 and 4 and their sub-questions. Thereafter, the findings obtained from the qualitative comparative case study using the think aloud protocol method are summarized and discussed. The basis is the four problem solving processes and eight elements of metacognition, across the three treatment groups of experimental Study 2. Lastly, an overall discussion of the results of the research is presented. SPSS 17.0 was used in statistical analysis of quantitative data. Qualitative data from the cross case comparative study was analyzed using AtlasTi Version 5, a spreadsheet and SPSS 17.0.

6.1 The Experimental Study 1 Results

6.1.1 Introduction

Study 1 investigated the effect of adaptive learning support on (a) the learner's cognitive knowledge, expressed as learner performance in a test and (b) the learner's perceived value of the online learning environment. The study had two treatment groups i.e. those using the adaptive version and those using the non-adaptive version of the system while studying OOP concepts.

From Study 1, the means of the student scores in Continuous Assessment Tests (CATs) for the two experimental groups were compared using Student's *t*-test (equal variance). This test was used to answer research question 1. The Mann Whitney test was used to compare the responses in a post-survey of the students in the adaptive and non-adaptive groups. The survey responses were on a 7-point Likert scale. The overall Cronbach Alpha reliability for the survey

questionnaire was 0.84, which was higher than the recommended minimum of 0.7 (Sekaran, 2004). The data collected from the survey was used to answer research question 2.

6.1.2 Statistical Data Analysis for Study 1

a) Comparing Level of Performance for the Adaptive and Non-adaptive groups

The students took two CATS, CAT 1 after studying the first part of the OOP course i.e. Class Design and CAT 2 after studying the second part of the course i.e. Inheritance.

The means and distribution of student scores for the adaptive and non-adaptive groups for CAT 1 is shown graphically in the 2 box-plots in Figure 6-1.

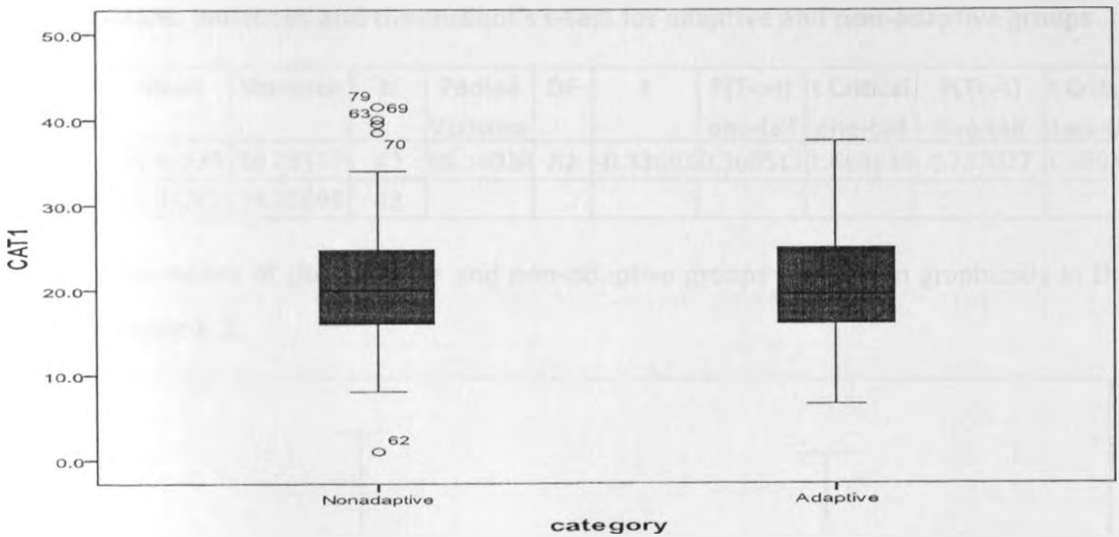


Figure 6-1: Box-plots showing comparison of means and distribution of student scores in CAT 1 for the adaptive and non-adaptive groups

The mean scores for students in the two treatment conditions are nearly the same, with the non-adaptive group having a slightly higher mean score of 21.04762 compared to 20.45238 for the adaptive group. The results are shown in Table 6-1. The CAT 1 scores for the adaptive group ranged from 6.5 to 37.5 and for the non-adaptive group, they ranged from 1.0 to 41.5. The variance ratio for the two groups was not significant ($F=0.763569$, $df=41$, $P>0.05$). There were some outliers in the non-adaptive group scores as shown in Figure 6-1.

Table 6-1 : Means, variances and test for equality of variances for CAT 1

	Mean	Variance	Observations	df	F	P(F<=f) one-tail	F Critical one-tail
Adaptive	20.45238	56.75377	42	41	0.763569	0.195698	0.594656
Non Adaptive	21.04762	74.32695	42	41			

Since the variances were not significantly different (hence the two groups were homogenous) and the two groups were also independent, and the CAT 1 scores were on the ratio scale, the student's t-test was used to find out if the means for the two groups of students for CAT 1 were significantly different or not. As shown in Table 6-2, the means for the adaptive and non-adaptive groups were found not to be significantly different ($t = -0.33$, $p > 0.05$).

Table 6-2 : Means, variances and the student's t-test for adaptive and non-adaptive groups

	Mean	Variance	N	Pooled Variance	DF	t	P(T<=t) one-tail	t Critical one-tail	P(T<=t) two-tail	t Critical two-tail
Adaptive	20.45238	56.75377	42	65.54036	82	-0.33693	0.368513	1.663649	0.737027	1.989319
Non Adaptive	21.04762	74.32695	42							

For CAT 2, the means of the adaptive and non-adaptive groups are shown graphically in the box-plots in Figure 6-2.

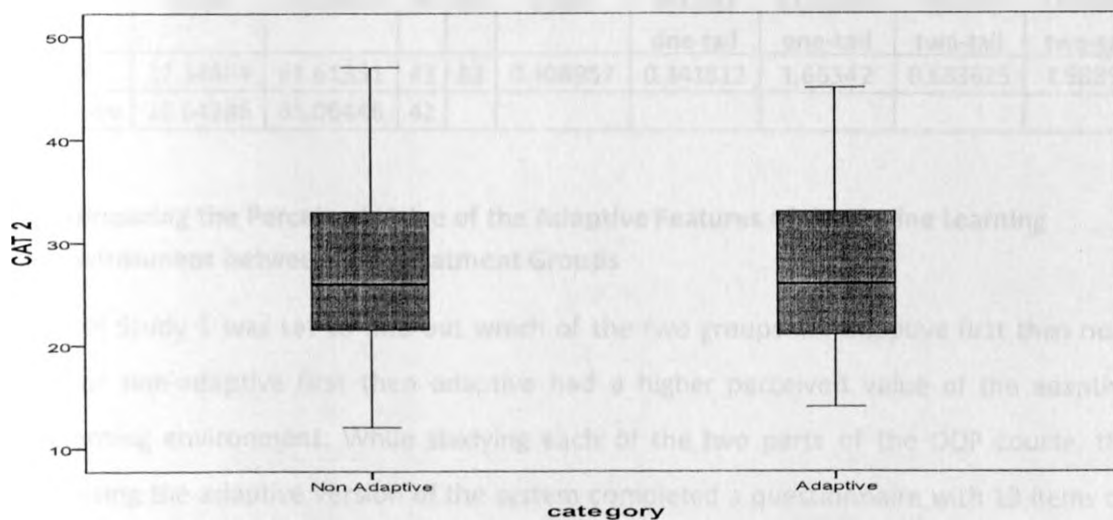


Figure 6-2 : Box-plots showing comparison of means and distribution of student scores in CAT 2 for the adaptive and non-adaptive treatment conditions

As can be seen in the box-plots in Figure 6-2, the mean scores for the adaptive group, which was non-adaptive in the first part of the course, and non-adaptive group, which was adaptive in the first part of the course for CAT 2, were nearly the same. The adaptive group had a higher mean test score of 27.34884 while the non-adaptive group's mean test score was 26.64286. The scores for CAT 2 ranged from 14 to 45 for the adaptive group and 12 to 47 for the non-adaptive group. As shown in Table 6-3, the variance ratio for the two groups was not significant ($F=0.94696$, $df =42$, $P>0.05$).

Table 6-3 : Means, variances and test for equality of variance for CAT 2

	Mean	Variance	Observations	DF	F	P(F<=f) one-tail	F Critical one-tail
Adaptive	27.34884	61.61351	43	42	0.946961	0.430253	0.597101
Non adaptive	26.64286	65.06446	42	41			

No significant statistical difference was found between CAT 2 test scores for the adaptive and non-adaptive groups, using the student's t-test ($t = 0.408957$, $p > 0.05$). The results are shown in Table 6-4.

Table 6-4 : Test two means, standard deviation of adaptive and non-Adaptive groups

	Mean	Variance	N	DF	t Stat	P(T<=t) one-tail	t Critical one-tail	P(T<=t) two-tail	t Critical two-tail
Adaptive	27.34884	61.61351	43	83	0.408957	0.341812	1.66342	0.683625	1.98896
Non Adaptive	26.64286	65.06446	42						

b) Comparing the Perceived Value of the Adaptive Features of the Online Learning Environment between the Treatment Groups

This part of Study 1 was set to find out which of the two groups i.e. adaptive first then non-adaptive or non-adaptive first then adaptive had a higher perceived value of the adaptive online learning environment. While studying each of the two parts of the OOP course, the students using the adaptive version of the system completed a questionnaire with 13 items on a 7-point Likert scale. The questionnaire measured their perception of the value of the various adaptive features of the learning system. The Mann Whitney test was used for comparison and the results are shown in Table 6-5.

Table 6-5 : Mann Whitney U test of the difference in perceived value of the system’s adaptive features between adaptive and non-adaptive groups

Ranks				Test Statistics				
Category	N	Mean Rank	Sum of Ranks	Mann-Whitney U	Wilcoxon W	Z	P-value	
Cats Total	Adaptive	39	31.69	1236	456	1236	-3.434	0.001
	Non-adaptive	42	49.64	2085				
	Total	81						

The results of the non-adaptive group (mean ranks = 49.64) appear to be significantly better ($p = 0.001$) than those of adaptive group (mean ranks = 31.69). The students in the non-adaptive group did not use adaptive features for part one of the course i.e. Class Design but they used the features for the second part i.e. Inheritance. On the other hand, students in the adaptive group used the adaptive features when studying the first part of the course but did not use them when studying the second part.

The scores for the perceived value of the system’s adaptive features for the 2 groups were also found to be negatively skewed as shown by the histograms in Figure 6-3 and Figure 6-4.

Histogram of the average perceived value score for the non-adaptive first group

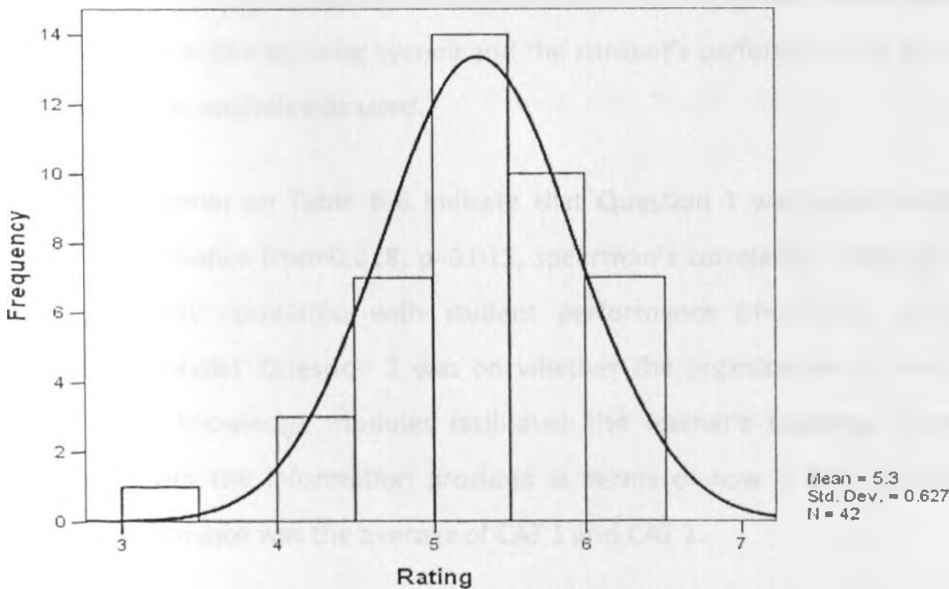


Figure 6-3 : Histogram of the perceived value score for the non-adaptive first group

Histogram of the average perceived value score for the adaptive first group

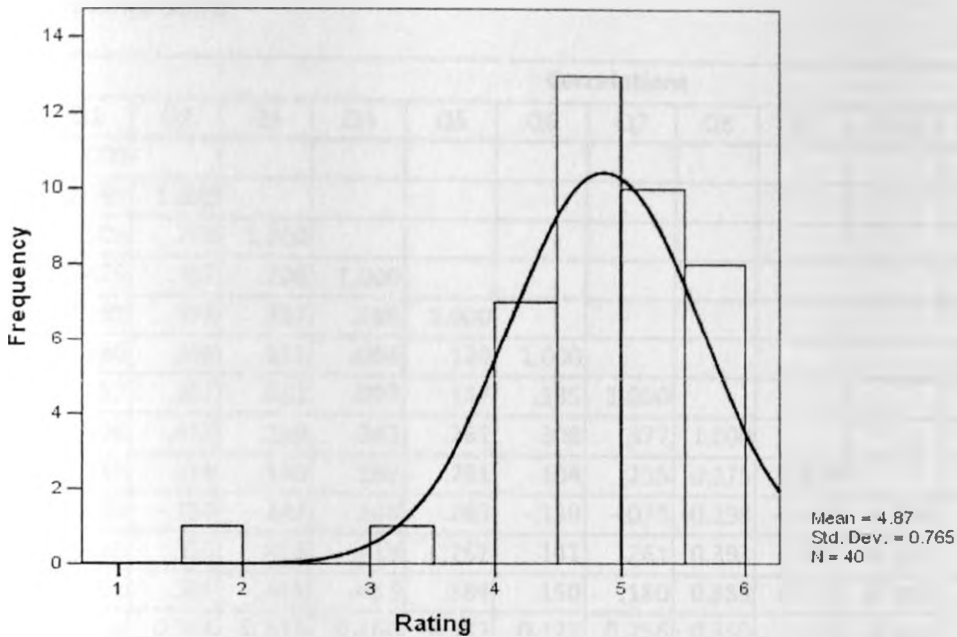


Figure 6-4 : Histogram of the average perceived value score for the adaptive first group

c) Investigating the Relationship between Perceived Value of Adaptive Features of the system and Learner Performance Score

To find out if there was a significant relationship between the perceived value of the adaptive features of the online learning system and the student's performance in the tests, Spearman's Rank Correlation analysis was used.

The results, shown on Table 6-6 indicate that Question 1 was significantly correlated with student performance ($\rho=0.218$, $p=0.012$, spearman's correlation analysis). Question 11 was also significantly correlated with student performance ($\rho=0.224$, $p=0.045$, spearman's correlation analysis). Question 1 was on whether the organization of the course into goals, concepts and knowledge modules facilitated the learner's studying. Question 11 was on satisfaction with the information provided in terms of how it met the learner needs. The student performance was the average of CAT 1 and CAT 2.

Table 6-6 : Correlation of perceived value of adaptive features of online system with performance score

Correlations														
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	CAT
Q1	1.000													
Q2	0.495	1.000												
Q3	0.506	.270	1.000											
Q4	0.425	.387	.200	1.000										
Q5	0.381	.399	.337	.286	1.000									
Q6	0.180	.266	.131	.096	.120	1.000								
Q7	0.257	.307	.072	.097	.147	.335	1.000							
Q8	0.406	.418	.239	.343	.261	.308	.377	1.000						
Q9	0.332	.174	.180	.163	.291	.104	.255	0.275	1.000					
Q10	-0.403	-.152	-.247	-.148	-.285	-.159	-.073	-0.198	-0.184	1.000				
Q11	0.425	.355	.416	.419	.267	.141	.261	0.393	0.267	-0.141	1.000			
Q12	0.551	.501	.412	.453	.384	.150	.180	0.389	0.313	-0.194	.584	1.000		
Q13	0.518	0.314	0.513	0.260	0.337	0.172	0.256	0.350	.273	-0.210	.574	.713	1.000	
CAT	0.299	0.084	0.064	0.071	0.008	0.180	-0.003	0.082	0.162	-0.102	0.113	0.089	0.211	1.000

6.1.3 Summary of Findings for Research Questions 1-2

Question 1

Which group of learners (those learning online with adaptive support or those learning online without adaptive support) achieves higher test scores when learning object oriented programming concepts?

Overall, the use of adaptive features for learning OOP concepts does not seem to have a significant effect on the test scores obtained by the students. Neither of the two groups performed significantly better than the other. However, the group which started without adaptive features for Class Design part and then used adaptive features for the Inheritance part had marginally higher scores for both CAT 1 and CAT 2.

Question 2

Does the utilization of adaptive learning support lead to differences in the perceived value of the online learning environment?

Both adaptive and non-adaptive groups completed the perception questionnaire. The adaptive group responded after studying the Class Design part of the course during which they used the features. The non-adaptive group responded after studying Inheritance during which they used the features. Responses were negatively skewed for both groups, meaning the use of the features had a positive influence on the perceived value of the system. The adaptive features were perceived to be of high value to the students using the system. The non-adaptive first group had a more favourable perception of the system compared to the adaptive first group.

Table 6-7 provides a summary of the results from Study 1.

Table 6-7 : Summary of the results from Study 1

Purpose	Test	Result
Investigate effect of use of adaptive support features on learner performance in a test (Research Question 1)	Independent samples t-test	There was no significant difference between non-adaptive first group's performance and the adaptive first group's performance. However, the non-adaptive first group had slightly higher scores in both CAT 1 and CAT2
Investigate the effect of use of adaptive features on the students' perception of learning environment (Research Question 2)	Mann Whitney U test	The non-adaptive first group had a more favourable perception of the learning environment than the adaptive first group. Both groups had a strongly positive perception of the system, as shown by the negatively skewed responses
Investigate the relationship between the perceived value of the system and the student's test scores	Spearman Rank Order correlation	Out of 13 items in the questionnaire for perceived value of the system, only two had a significant positive relationship with the student's test scores

The table shows the purpose of the study, which is related to a research question, the test carried out and the result obtained.

6.1.4 Discussion of the Results for Study 1

From the results of the study, there was no significant difference between those who used adaptive features first and those who used them later. This could be because the course being studied was contributing to the semester grade for the students. Therefore, the students already had enough pressure to pass the tests i.e. had the favourable Germane Cognitive Load, which motivated them to perhaps invest extra cognitive effort to master the course content in order to pass the tests.

The non-adaptive first group had a slightly better score and also had a significantly more favourable perception of the learning environment. This could be because the students in this group were introduced to the full complexity of the system's features gradually (plain system with course content first, then system with course content and adaptive features) leading to better management of cognitive load. This was an implementation of simple to complex sequencing of information elements (Van Marrienoer & Sweller, 2005). The adaptive then non-adaptive group had to contend with full the complexity of the system, with all its features presented to them at once. These students had to learn the meaning of and how to use adaptive features, and at the same time concentrate on learning the concepts of OOP using an online environment. This probably caused them to suffer the unfavourable intrinsic cognitive load because of too many new things. They probably needed more time to cope with the learning environment.

The students in the non-adaptive first group, which performed slightly better and also had a significantly more favourable perception of the adaptive features, had enough time to just get used to learning OOP concepts online without adaptive features. When they had mastered learning with the system, they were provided with more elements of information (adaptive feature effects) to handle. They did not suffer as much intrinsic cognitive load because the elements of the adaptive online learning environment were introduced to them gradually.

In terms of significance of differences in CAT 1 and CAT 2 between the two treatment groups, there was no significant difference at $\alpha = 0.05$ level of significance. This can be explained in terms of the pressure the students usually have to pass examinations in order to proceed with their studies. The course studied by the students was to contribute to their end-of-semester grade, and therefore, those who might not have been supported well by adaptive features perhaps drew their motivation from the need to pass. This extra pressure increased their Germane Cognitive Load, which is the perturbation one experiences that forces him to invest more cognitive effort so as to resolve a situation in order to reach a new level of equilibrium, in this case passing the course. So, the scores might not have been significantly different at $\alpha=0.05$ level of significance, but perhaps one group spent more cognitive effort to reach its mean score than the other.

This is also supported by the lower number of students (82) who participated in the self-report section investigating the students' perceived value of adaptive features. The students' participation in this part of the study was not contributing to the semester grade, and therefore, they were not obliged to participate. This is in comparison to the higher number of students (89) participating in the CATs, which was contributing to the student's semester grade. The students felt obliged to participate in this activity.

In spite of only 2 out of 13 items of the perception questionnaire having a significant positive relationship with the student test scores as shown in Table 6-6, it should be noted that 11 out of 13 items of the questionnaire had a positive correlation with student test scores (see Table 6-6). Hence, it can be concluded from this study that adaptive features and other user interface features can have a positive influence on the performance of the students using an adaptive system to study a course.

6.2 The Experimental Study 2 Results

6.2.1 Introduction

Study 2 investigated the effect of use of (a) question prompts and (b) a task schedule as scaffolding techniques on the (i) application of problem solving processes, (ii) level of metacognitive awareness and (iii) the quality of the product of ill-structured problem solving i.e. quality of information system developed using OOP concepts. The 4 problem solving processes are (i) problem representation, (ii) the solution process, (iii) solution justification and (iv) monitoring and evaluation. The 8 elements of metacognition are (i) declarative knowledge, (ii) conditional knowledge, (iii) procedural knowledge, (iv) planning, (v) information management, (vi) monitoring, (vii) debugging and (viii) evaluation. The study had 3 treatment groups i.e. those using a task schedule, those using question prompts and those using neither task schedule nor question prompts (control group).

A Wilcoxon signed-ranks test was used to compare the levels of application of problem solving processes and metacognitive awareness in the learners before the experiment (pre-test) and after the experiment (post-test). The test was used to investigate any changes in the level of awareness or application as a result of participating in the study. A 7-point Likert scale was used to evaluate the elements of metacognitive awareness while a 5-point Likert scale was used for evaluating the students' awareness of their problem solving processes for ill-structured problems. The questionnaire used for assessing level of awareness of problem solving processes had a Cronbach Alpha reliability of 0.719 and 0.699 for the pre-test and post-test respectively. The Metacognitive Awareness Index (MAI) had a Cronbach Alpha reliability of 0.922 and 0.898 for the pre-test and post-test respectively.

Comparison of increase in the students' level of awareness among the three treatment conditions was done using Kruskal-Wallis test in order to decide the most effective among the treatments.

The rationale for using nonparametric tests i.e. Kruskal-Wallis test and Wilcoxon signed-rank test in some comparisons was the inadequate number (<30) of students in the treatment groups.

6.2.2 Descriptive Statistics for Study 2

(a) Descriptive Statistics for the Pre-test and Post-test of Metacognitive Awareness

Table 6-8 summarizes the descriptive statistics for the pre-test and post test of the 8 elements of metacognitive knowledge for all the students who participated in Study 2.

66 students participated in the pre-test whereas 64 students took part in the post-test. For the pre-test, the mean scores ranged between 3.2576 and 3.7386 and the standard deviation ranged between 0.68558 and 0.78998. For the post-test, the mean scores ranged between 4.2656 and 4.6594 and the standard deviation ranged between 0.72809 and 4.6992. This shows that there was a less uniform distribution for the post-test compared to the pre-test.

Table 6-8 : Descriptive statistics and reliability levels for the pre-test and post-test of the 8 elements of metacognitive awareness

	Pre-test $\alpha=0.922$			Post-test $\alpha=0.898$		
	Mean	Std. Deviation	N	Mean	Std. Deviation	N
Declarative Knowledge	3.7386	.70939	66	4.6133	.72945	64
Procedural Knowledge	3.7197	.78998	66	4.6992	.72809	64
Conditional Knowledge	3.6758	.77798	66	4.4844	.80774	64
Planning	3.2576	.78014	66	4.3036	.97772	64
Information Management System	3.6894	.73067	66	4.6594	1.13329	64
Monitoring	3.3680	.74423	66	4.3058	1.00625	64
Debugging Strategies	3.4273	.68558	66	4.2656	.92449	64
Evaluation	3.5530	.73389	66	4.5521	.91763	64

(b) Descriptive Statistics for the Pre-test and Post-test of the Problem Solving Processes

Table 6-9 summarizes the descriptive statistics and Cronbach Alpha reliability levels for the pre-test and post test of the 4 processes for solving ill-structured problems for all the students who participated in Study 2.

62 students participated in the pre-test whereas 60 students took part in the post-test. For the pre-test, the mean scores for the processes ranged between 2.3548 and 2.6581 and the standard deviation ranged between 0.48342 and 0.87173. For the post-test, the mean scores ranged between 1.9463 and 3.6000 and the standard deviation ranged between 0.34987 and 1.03252, showing a more spread out distribution for the post test compared to the pre-test.

Table 6-9 : Descriptive statistics and reliability levels for the pre-test and post-test of the 4 problem solving processes

	Pre test $\alpha=0.719$			Post Test $\alpha=0.699$		
	Mean	Std. Deviation	N	Mean	Std. Deviation	N
Problem representation	2.6581	.54126	62	3.5033	.62976	60
Solution process	2.4014	.48342	62	1.9463	.34987	60
Solution Justification	2.4516	.87173	62	3.6000	1.03252	60
Monitoring and	2.3548	.67531	62	3.1333	.77222	60

6.2.3 Main Statistical Data Analysis for Study 2

1. Comparison between Pre-test and Post-test scores for the 8 Elements of Metacognitive Awareness across all Treatments

To determine the specific differences between the pairs of pre-test and post-test scores for the 8 elements of metacognitive awareness, the Wilcoxon test was used. The test was also used to establish whether or not the differences were significant. The results are presented in Table 6-10.

The Wilcoxon Signed-Ranks Test uses rankings instead of means. However, means were supplied for ease of interpretation.

The Wilcoxon Signed-Ranks Test shows that awareness in all the 8 elements of metacognitive knowledge increased significantly, in that the post-test scores were significantly higher than the pre-test scores for all elements, across the three treatments. All elements for all treatment methods had $p \leq .05$ as shown in Table 6-10.

Table 6-10 : Showing means and Matched-Pairs Wilcoxon Signed-Ranks Test for the elements of Metacognitive Awareness

		Ranks				
		N	Mean Rank	Sum of Ranks	Z	P-value
Declarative Knowledge	Task schedule group	21	11.00	231.00	-4.031	0.000
	Question prompts group	21	11.00	231.00	-4.039	0.000
	Control group	24	12.50	300.00	-4.301	0.000
Procedural Knowledge	Task schedule group	21	11.00	231.00	-4.050	0.000
	Question prompts group	21	11.00	231.00	-4.075	0.000
	Control group	24	12.50	300.00	-4.381	0.000
Conditional Knowledge	Task schedule group	20	10.50	210.00	-3.965 ^a	0.000
	Question prompts group	18	9.50	171.00	-3.755	0.000
	Control group	23	12.00	276.00	-4.250	0.000
Planning	Task schedule group	21	11.00	231.00	-4.025	0.000
	Question prompts group	21	11.00	231.00	-4.047	0.000
	Control group	23	12.00	276.00	-4.241	0.000
Information Management	Task schedule group	21	11.00	231.00	-4.030	0.000
	Question prompts group	21	11.00	231.00	-4.062	0.000
	Control group	24	12.50	300.00	-4.296	0.000
Monitoring	Task schedule group	21	11.00	231.00	-4.034	0.000
	Question prompts group	21	11.00	231.00	-4.044	0.000
	Control group	22	13.36	294.00	-4.134	0.000
Debugging Strategies	Task schedule group	21	11.00	231.00	-4.040	0.0001
	Question prompts group	21	11.00	231.00	-4.032	0.0001
	Control group	21	11.07	232.50	-3.485	0.0005
Evaluation	Task schedule group	21	11.00	231.00	-4.137	0.0000
	Question prompts group	21	11.00	231.00	-4.072	0.0000
	Control group	24	12.50	300.00	-4.415	0.0000

2. Comparison between Pre-test and Post-test scores for the 4 Problem Solving Processes across all Treatments

To determine the specific differences in performance between the matched pre-test and post-test pairs and whether the differences were significant or not for the 4 processes of problem

solving among the three treatment conditions, the Wilcoxon test was used. The results are presented in Table 6-11.

The Wilcoxon Signed-Ranks Test shows that there were a significant difference between the pre-test and post-test for the 4 processes of problem solving, across the three treatments. All elements for all methods had $p \leq .05$.

Table 6-11 : Showing means and Matched-Pairs Wilcoxon Signed-Ranks Test for the Problem Solving Processes

Process	Category	Ranks			Z	P-value
		N	Mean Rank	Sum of Ranks		
Problem Representation	Task schedule group	20	10.5	210.00	-3.941	0.000
	Question prompts group	20	10.50	210.00	-3.929	0.000
	Control group	19	10.00	190.00	-3.880	0.000
Solution Process	Task schedule group	16	11.28	180.50	-2.826	0.005
	Question prompts group	18	10.64	191.50	-3.236	0.001
	Control group	20	11.35	227.00	-3.883	0.000
Solution Justification	Task schedule group	21	11.00	231.00	-4.110	0.000
	Question prompts group	20	11.40	228.00	-3.953	0.000
	Control group	20	11.28	225.50	-3.888	0.000
Monitoring and Evaluation	Task schedule group	19	10.00	190.00	-3.869	0.000
	Question prompts group	20	10.50	210.00	-3.943	0.000
	Control group	21	11.00	231.00	-4.064	0.000

3. Comparison of the effectiveness of the Scaffolding Methods on level of the Elements of Metacognitive Knowledge among the treatment groups

After establishing that there was a significant difference in the level of metacognitive awareness, further analysis was carried out to establish which one among the three treatments was more effective: use of task schedule, use of question prompts or use of no scaffold for the control group.

The Kruskal-Wallis test was used to compare gains in level of metacognitive awareness from pre-test to post-test, for the students in the three treatments. Which of the treatments had a statistically significant effect above and beyond any one of the other methods, in terms of improvements in the post-test scores over the pre-test scores? Table 6-12 shows the results of this analysis. Provided next is an element by element presentation of the comparison.

Declarative Knowledge: There was a statistically significant difference in the gain in the level of awareness from pre-test to post-test of Declarative Knowledge to be applied during ill-structured problem solving, among the scaffolding methods used by students ($\chi^2 = 8.52$, $df=2$, $p = 0.014$).

A comparison of their rankings among the three treatment groups indicates that the highest difference is in the task schedule group. This analysis therefore shows that the task schedule method is more effective than one or more of the other scaffolding methods in increasing the level of awareness of Declarative Knowledge among the three treatments, during ill-structured problem solving.

Procedural Knowledge: There was no statistically significant difference in the gain in the level of awareness from pre-test to post-test of the Procedural Knowledge among the scaffolding methods used ($\chi^2 = 3.618$, $df=2$, $p = 0.164$).

A comparison of the rankings among the three scaffolding methods indicates that the task schedule group registered the highest gain in level of awareness of Procedural Knowledge during ill-structured problem solving, although the difference is not great enough to be considered significant at the 0.05 level of significance.

Table 6-12 : Test of gains between pre-test and post-test in terms of level of awareness of the 8 elements of metacognitive knowledge among the three treatment groups

Ranks						
	Category	N	Mean Rank	χ^2	DF	P-value
Declarative Knowledge	Task schedule	21	43.40	8.520	2	0.014
	Question prompts	21	29.31			
	Control	24	28.50			
	Total	66				
Procedural Knowledge	Task schedule	21	38.45	3.618	2	0.164
	Question prompts	21	34.45			
	Control	24	28.33			
	Total	66				
Conditional Knowledge	Task schedule	21	29.29	2.430	2	0.297
	Question prompts	21	31.67			
	Control	23	37.61			
	Total	65				
Planning	Task schedule	21	52.05	35.478	2	0.000
	Question prompts	21	32.29			
	Control	24	18.33			
	Total	66				
Information Management	Task schedule	21	45.76	15.789	2	0.000
	Question prompts	21	32.90			
	Control group	24	23.29			
	Total	66				
Monitoring	Task schedule	21	53.60	37.233	2	0.000
	Question prompts	21	29.31			
	Control	24	19.58			
	Total	66				
Debugging Strategies	Task schedule	21	52.26	38.872	2	0.000
	Question prompts	21	30.98			
	Control	23	17.26			
	Total	65				
Evaluation	Task schedule	21	23.62	45.762	2	0.000
	Question prompts	21	55.90			
	Control	24	22.54			
	Total	66				

Conditional Knowledge: There was no statistically significant difference in the gain from pre-test to post-test in terms of the level of awareness of Conditional Knowledge among the scaffolding methods used by students ($\chi^2 = 2.430$, $df=2$, $p=0.297$).

A comparison of the rankings among the three scaffolding methods indicates that the control group registered the highest gain in level of awareness of Conditional Knowledge during ill-structured problem solving, although the difference is not great enough to be considered significant at the 0.05 level of significance.

Planning: There was a statistically significant difference in the gain from pre-test to post-test in terms of the level of awareness of the Planning strategies to apply during ill-structured problem solving, among the three scaffolding methods used by students ($\chi^2 = 35.478$, $df=2$, $p < 0.0001$).

A comparison of the rankings among the three groups indicates that the highest gain in the level of awareness of Planning is in the task schedule group. This analysis therefore shows that the task schedule method is more effective than one or more of the other scaffolding methods in terms of student awareness of Planning during ill-structured problem solving.

Information Management Strategies: There was a statistically significant increase from pre-test to post-test, among the scaffolding methods used by students, in terms of level of awareness of Information Management Strategies to apply during ill-structured problem solving ($\chi^2 = 15.789$, $df=2$, $p < 0.0001$).

A comparison of the rankings among the three groups indicates that the highest gain is in the Task schedule group. This analysis therefore shows that the task schedule method is more effective than one or more of the other scaffolding methods in terms of increase in the level of awareness of Information Management Strategies to apply during ill-structured problem solving.

Monitoring: There was a statistically significant increase in level of awareness of Monitoring strategies to apply during ill-structured problem solving, from pre-test to post-test, among the scaffolding methods used by the students ($\chi^2 = 37.233$, $df=2$, $p < 0.0001$).

A comparison of the rankings among the three groups indicates that the highest gain is in the task schedule group. This analysis therefore shows that the task schedule method is more effective than one or more of the other scaffolding methods in terms of increase in the

students' level of awareness of Monitoring strategies they can apply during ill-structured problem solving.

Debugging Strategies: There was a statistically significant difference in the increase of the level of awareness of Debugging strategies to apply during ill-structured problem solving, from pre-test to post-test, among the scaffolding methods used by students ($\chi^2 = 38.872$, $df=2$, $p < 0.0001$).

A comparison of the rankings among the three scaffolding methods indicates that the highest gain is in the task schedule group. This analysis therefore shows that the task schedule method is more effective than one or more of the other scaffolding methods in terms of increase in the student's level of awareness of Debugging strategies to apply during ill-structured problem solving.

Evaluation: Regarding Evaluation, there was a statistically significant gain in the level of awareness of the Evaluation strategies to apply during ill-structured problem solving, from pre-test to post-test, among the three scaffolding methods used by students ($\chi^2 = 45.762$, $df=2$, $p < 0.0001$).

A comparison of the rankings among the three scaffolding methods indicates that the highest increase from pre-test to post-test is in the question prompts group. This analysis, therefore, shows that the question prompts method is more effective than one or more of the other scaffolding methods in terms of increase in level of awareness of Evaluation strategies to apply during ill-structured problem solving.

4. Comparison of the effectiveness of the scaffolding methods on level of awareness of the 4 problem solving processes

After establishing that there was a significant difference in the level of awareness of the 4 problem solving processes from pre-test to post-test scores for all the treatments, further analysis was carried out to establish which one among the three treatments was more

effective: use of task schedule, use of question prompts or use of no scaffold (the control group). The Kruskal-Wallis test was used. Table 6-13 shows the results of this analysis.

Table 6-13 : Test of gains between pre-test and post-test in the level of awareness of the problem solving processes among the three treatment groups

Ranks						
	Category	N	Mean Rank	Chi-Square	DF	P-value
Problem Representation	Task Schedule	20	37.40	31.583	2	0.000
	Question Prompts	20	44.10			
	Control	22	14.68			
	Total	62				
Solution Process	Task schedule	21	29.86	1.166	2	0.558
	Question prompts	20	30.55			
	Control	22	35.36			
	Total	63				
Solution Justification	Task schedule	21	32.55	23.312	2	0.000
	Question prompts	21	47.12			
	Control	23	20.52			
	Total	65				
Monitoring and Evaluation	Task schedule	19	32.42	19.649	2	0.000
	Question prompts	20	42.78			
	Control	22	19.07			
	Total	61				

Provided next is a process by process presentation of the analysis.

Problem Representation: From the analysis, there was a statistically significant increase from pre-test to post-test in terms of level of awareness of the process of Problem Representation while solving an ill-structured problem, among the scaffolding methods used by students ($\chi^2 = 31.583$, $df=2$, $p < 0.0001$).

A comparison of the rankings among the three scaffolding treatments indicates that the highest increase is in the question prompts group. This analysis, therefore, shows that the question

prompts method is more effective than one or more of the other scaffolding methods in terms of increase in the level of awareness of Problem Representation as a process for solving ill-structured problems.

Solution Process: This analysis indicated that there was no statistically significant difference in the level of awareness of the Solution Process from pre-test to post-test, among the scaffolding methods used by students ($\chi^2 = 1.666$, $df=2$, $p= 0.558$). However, it should be noted that the ranking of the control group is somewhat higher than the other groups, although the difference is not great enough to be considered significant at the 0.05 level of significance.

Solution Justification: Regarding the Solution Justification process, there was a statistically significant increase in level of awareness from pre-test to post among the students while solving ill-structured problems, among the scaffolding methods ($\chi^2 = 20.523$, $df=2$, $p < 0.0001$).

A comparison of the rankings among the three scaffolding methods indicates that the highest increase in awareness is in the question prompts group. This analysis therefore shows that the question prompts method is more effective than one or more of the other scaffolding methods in terms of increase in the level of awareness of Solution Justification as one of the processes for solving an ill-structured problem.

Monitoring and Evaluation: There was a statistically significant gain from pre-test to post-test in terms of level of awareness of Monitoring and Evaluation as a process for ill-structured problem solving, among the scaffolding methods used by students ($\chi^2 = 19.649$, $df=2$, $p < 0.0001$).

A comparison of these rankings among the three scaffolding approaches indicates that the highest gain from pre-test to post-test is in the question prompts group. This analysis therefore shows that the question prompts method is more effective than one or more of the other scaffolding methods in terms of increase in the level of awareness of Monitoring and Evaluation as a process for solving an ill-structured problem.

5. Comparison of the effects of scaffolding methods on the quality of the product of the programming assignment

The purpose of this part of the study was to compare the influence of the three scaffolding treatments on the quality of the system developed by the students: task schedule, question prompts and neither task schedule nor question prompts (control). The variable “quality of the product” had two elements: demonstration of the system’s features and application of OOP concepts. Comparison of system quality scores was done using one-way ANOVA test. The means and standard deviations of each of the treatments are shown in Table 6-14.

Table 6-14 : Comparison of the effect of the scaffolds on the quality of the system developed using OOP concepts in terms of the three treatment groups

		N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Between-Component Variance
						Lower Bound	Upper Bound	
Product Quality	Task schedule	19	11.11	4.284	.983	9.04	13.17	
	Question prompts	28	11.96	4.480	.847	10.23	13.70	
	Control	27	10.30	5.371	1.034	8.17	12.42	
	Total	74	11.14	4.770	.554	10.03	12.24	
	Model	Fixed Effects			4.780	.556	10.03	12.24
	Random Effects				.556 ^a	8.74 ^a	13.53 ^a	-.153

From comparing the effects of the use of the scaffolds for problem solving on product quality scores, the average score for the question prompts group was the highest compared to other methods, though it was not significant ($p > 0.05$) as shown in Table 6-15.

In terms of Between Groups differences, Table 6-15 shows there was no significant difference in product quality scores among the three methods of scaffolding ($F = 0.837$, $p = 0.437$).

Table 6-15 : Comparison of product quality scores across the three scaffolding methods using ANOVA

		Sum of Squares	DF	Mean Square	F	Sig.
Product Quality scores	Between Groups	38.265	2	19.133	.837	0.437
	Within Groups	1622.383	71	22.850		
	Total	1660.649	73			

6.2.4 Summary of Findings for Research Questions 3-4

Question 3:

Does the use of metacognitive scaffolds (question prompts or task schedule) have an effect on the students' metacognitive awareness while solving an ill-structured programming task?

- a) *Does the use of question prompts influence students' level of metacognitive awareness while solving an ill-structured programming task?*
- b) *Does the use of a task schedule influence students' level of metacognitive awareness while solving an ill-structured programming task?*

As shown in Table 6-16, the Wilcoxon Signed Ranks test was used to compare the pre-test and post scores of the students' level of awareness of the elements of metacognitive knowledge at $\alpha=0.05$ level of significance. Both the task schedule and question prompts were found to have a significant positive effect of the students' level of awareness. However, the level of effect of the treatments among the three elements of Knowledge of Cognition i.e. Declarative Knowledge, Procedural Knowledge and Conditional Knowledge was found to be lower than the effect of the treatments on the elements of Knowledge of Regulation of Cognition. The effects were expressed in the form of mean ranks, as shown in Table 6-12.

In terms of which of the approaches had a greater effect, the Kruskal-Wallis test was used to compare gains in scores from pre-test to post-test, among the treatment conditions.

For the elements of Knowledge of Cognition, it was found out that the task schedule had a significantly greater effect on Declarative Knowledge compared to the other two treatments at $\alpha=0.05$ level of significance. But there was no significant difference in effect for Conditional Knowledge and Procedural Knowledge, among the three treatment groups.

For the elements of Knowledge of Regulation of Cognition, the task schedule had a significantly greater effect than the other treatments on 4 out of 5 elements at $\alpha=0.05$ level of significance. These elements were Planning, Information Management Strategies, Monitoring and Debugging Strategies. However, for the element named Evaluation, the question prompts group had a significantly greater effect than the task schedule and control groups.

Table 6-16 : Summary of the Results from Study 2 for Question 3

Purpose	Test	Result
Investigate the effect of the use of scaffolds (question prompts, task schedule) on the student's level of awareness of the 8 elements of metacognitive knowledge	Wilcoxon Signed Ranks test	Each of the three treatment conditions i.e. use of task schedule, question prompts and the control had a significant increase in the level of metacognitive awareness ($\alpha=0.05$), from pre-test to post-test, over all the 8 elements
Investigate the comparative effect of the scaffolds (question prompts, task schedule) on the student's level of metacognitive awareness	Kruskal- Wallis test	<p>(a) Declarative Knowledge – task schedule had a statistically significant higher effect on awareness ($\alpha=0.05$) than question prompts and control group</p> <p>(b) Procedural Knowledge – No statistically significant difference in effect among the treatments ($\alpha=0.05$) even though control group had the highest increase in awareness</p> <p>(c) Conditional Knowledge – No statistically significant difference in effect among the treatments on awareness ($\alpha=0.05$) even though Task Schedule group had highest increase</p> <p>(d) Planning, Information Management, Monitoring, and Debugging Strategies – task schedule had a statistically significant higher effect on awareness ($\alpha=0.05$) than question prompts and control groups</p> <p>(e) Evaluation – question prompts had a statistically higher effect on awareness ($\alpha=0.05$) than task schedule and control groups</p>

Question 4:

Does the use of metacognitive scaffolds (question prompts or task schedule) have an effect on student's level of application of problem solving processes when solving an ill-structured task?

a) *Does the use of question prompts have an effect on students' level of awareness of problem*

solving processes namely problem representation, the solution process, solution justification and monitoring and evaluation of solutions when solving an ill-structured programming task?

- b) *Does the use of a task schedule have an effect on students' level of awareness of problem solving processes namely problem representation, developing solutions, making justifications and monitoring and evaluation of solutions when solving an ill-structured programming task?*
- c) *Does the use of metacognitive scaffolds (question prompts or task schedule) have an effect on the quality of the product of solving an ill-structured the programming task?*

From Table 6-17, it can be seen that when using the Wilcoxon Signed Ranks test to compare pre-test and post rankings, there was a significant increase in awareness when students used the question prompts scaffold for 3 out of 4 problem solving processes at $\alpha=0.05$ level of significance. These processes were Problem Representation, Solution Justification and Monitoring and Evaluation of solutions. For the Solution Process, there was no significant change in the level of awareness.

When using the Wilcoxon Signed Ranks test to compare the pre-test and post-test scores , it was also found out that the task schedule had a significant effect on the students' level of application of 3 out of 4 problem solving processes at $\alpha=0.05$ level of significance. The Solution Process had no significant change.

In investigating which of the two treatments (question prompts or task schedule) had a higher influence on awareness of the 4 problem solving processes, the Kruskal-Wallis test was used to compare change in scores from pre-test to post-test, across the three treatments.

Table 6-17 : Summary of results from Study 2 for Question 4

Purpose	Test	Result
Investigate effect of use of scaffolds (question prompts, task schedule) on the student's application of the 4 problem solving processes	Wilcoxon Signed Ranks test	Each of the three treatment conditions i.e. use of task schedule, question prompts and the control had a significant increase ($\alpha=0.05$) in the application of 3 problem solving processes namely Problem Representation, Solution Justification, and Monitoring and Evaluation. One of them i.e. Solution Process was not affected significantly by the treatments
Investigate the comparative effect of the scaffolds (question prompts, task schedule) on the student's application of the 4 problem solving processes	Kruskal-Wallis Test	(a) Problem Representation, Solution Justification and Monitoring and Evaluation – There was a statistically significant difference in the effect of the treatments ($\alpha=0.05$) among the treatments and the question prompts group had the highest increase. (b) Solution Process – There was no statistically significant difference ($\alpha=0.05$) in effect among task schedule, question prompts and control groups. Control group had the highest increase in application of Solution Process strategies.
Investigate the comparative effect of the scaffolds (question prompts, task schedule) on the quality of the student's product of problem solving (the developed system)	One-way ANOVA	There was no statistically significant difference in the product evaluation scores among the treatment groups, though the question prompts group had the highest mean score.

Question prompts had a statistically significant higher effect on 3 of the 4 processes namely Problem Representation, Solution Justification and Monitoring and Evaluation at $\alpha=0.05$ level of significance. However, there was no statistically significant difference in effect among the three treatments for the process named Solution Process, at $\alpha=0.05$ level of significance.

In terms of effect of the treatments on the quality of the solution, the one way ANOVA test was used. At $\alpha=0.05$ level of significance, it was found out that there was no significant difference in the scores among the question prompts, task schedule and control groups. Therefore, it can be concluded that neither of the two treatments i.e. question prompts and task schedule had a significantly greater effect on the quality of the product of problem solving.

6.2.5 Discussion of the Results for Study 2

From the comparison of the pre-tests and post-tests, it can be concluded that in general the treatments were effective. This is because there was a significant gain from pre-test to post-test for both problem solving processes and elements of metacognitive knowledge.

From the comparison of gains in scores from pre-test to post-test across the three treatment groups for the elements Knowledge of Cognition, it was found out that the task schedule had a significantly higher gain than the other two treatments in one of the elements i.e. Declarative Knowledge. For the other two elements, there was no significant difference in gain in scores from pre-test to post-test. Moreover, the overall scores for the elements of Knowledge of Cognition were a lot lower when compared to the elements of Knowledge of Regulation of Cognition. This is because the Knowledge of Cognition is not well supported by the scaffolds used in this case i.e. task schedule and question prompts. It is also because knowledge of cognition tends to be declarative and therefore students are required to engage in abstract recall to retrieve information from the Long Term Memory in order to externalize it as responses to a self-report questionnaire (Somerén, Barnard, & Sandberg, 1994). This is a more difficult process for the students than applying Knowledge of Regulation of Cognition. This is more so because students are unconscious of the automated aspects of their knowledge, especially the knowledge which has been automated. The scores of Knowledge of Cognition were not good as they are mainly about the learners' awareness of Declarative, Procedural and Conditional knowledge.

For the Knowledge of Regulation of Cognition, the task schedule group significantly outperformed the question prompts in all elements except one, with question prompts having

a significantly higher effect on one element named Evaluation. This could be the case because the task schedule directly required the learner to plan his problem solving. The learner had to set sub-goals and define resources required to attain each of the sub-goals. The task schedule therefore directly enforced Planning.

The task schedule also directly enforced Monitoring because it required the student to keep determining percentage progress on each sub-task and also percentage progress on each of the OOP knowledge elements required for the sub-task. The student was required to update the online schedule with the current progress on a regular basis. And then, every time the student logged into the system, he was shown the current progress in terms of bars with annotated colours for different levels of progress. This directly enforced monitoring.

From the information obtained as a result of monitoring, if the progress was not good, the student could get perturbed and had to look for things to do so as to remain on course to solving the problem in time. Therefore, the student was directly predisposed to do debugging.

The student was also required to look for information such as OOP knowledge and skills required for problem solving and organize it under each sub-task. He was also required to keep evaluating his level of mastery of each of the elements of knowledge and skills required for each sub-task. If the mastery was not adequate, he was expected to go on searching for the required information and organize it under the sub-tasks. Therefore, the student was directly required to devise strategies for organizing his information. This is unlike the question prompts which were organized according to the processes or stages of problem solving.

The task schedule did not directly require the student to do anything in terms of evaluation, unlike the question prompts scaffold which had questions directly requiring the student to engage in monitoring and evaluation. Direct support reduces the extra cognitive load the student experiences due to the requirement to understand a tool and know how it works. Using a new unfamiliar scaffold can add cognitive load for the student because it is extra work to be

done concurrently with solving a learning task. The extent of cognitive load due to using the scaffold depends on the strategy used to introduce it to students during problem solving.

For the complex problem solving processes, question prompts had a significantly greater effect on the level of awareness of all processes except one, compared to the other treatment groups. The exception was the Solution Process in which there was no significant difference among the treatment groups.

The development of question prompts was guided by the processes of problem solving, and therefore, the students were able to apply them directly, which caused them less cognitive load and therefore, they could apply them well. This was unlike the task schedule which was not developed directly from the problem solving processes. For the Solution Process, each problem solver always has a way of solving a problem that he considers as suitable, with or without guidance. When this is the case, the students feel that being required to perform the task in another way i.e. answer question prompts while solving the problem (the Solution Process) slows them down. The students had to learn the prescribed way first and reconcile it with their prior knowledge of how to solve such problems, which slowed down their Solution Process. Hence, there was reduction in ratings from pre-test to post-test.

In terms of the quality of the product of problem solving, there was no significant difference among the three groups. This was because the students had to pass the course in order to progress in their studies, which caused them Germane Cognitive Load. Germane Cognitive Load increases one's motivation to commit extra cognitive resources in order to accomplish a task. It could be that the effect of the treatments (task schedule and question prompts) was not as big as the effect of the need of the students to pass a course that was to contribute to their end-of-semester grade. The influence of need to pass exams on effort invested by the students was confirmed by the fact that those who completed self-report questionnaires, which were not contributing to the semester grade, was 66. This is a lower number compared to the 74 who participated in the product evaluation oral examination, which was contributing to student's semester grade.

6.3 Results of the Comparative Case Study

6.3.1 Introduction

The qualitative study was carried out to gain more information about the students' problem solving, to assist in further understanding the results of experimental Study 2. Students were selected from each of the three treatments, i.e. those using task schedule, those using question prompts and those using neither of the two. The students participated in a think aloud session during which they explained how they would apply knowledge of OOP to solve an ill-structured problem. The manifestation of the four problem solving processes and the eight elements of metacognition in their protocols were tagged with appropriate codes, and their frequencies summarized for comparison over the three treatments. The development of codes for metacognition was guided by the model of metacognition developed by Schraw and Dennison (1994). The codes for problem solving processes were developed based on the model of Voss and Post (1988) and the scoring rubrics used by Ge (2001), which were improved during this research to become more relevant to the OOP domain.

The results were also compared with the quantitative results from experimental Study 2.

6.3.2 Presentation of the Results

a) Comparison of the effectiveness of Scaffolding methods on level of Awareness of the Elements of Metacognition

(i) Knowledge of Cognition

Knowledge of Cognition has three elements namely Declarative Knowledge, Conditional Knowledge and Procedural Knowledge.

Table 6-18 presents a summary of the frequencies of the codes related to Knowledge of Cognition, as found in the think aloud protocols, across the three treatments.

Declarative Knowledge: Declarative knowledge had 3 codes. The manifestation in the protocols of these codes was very low, with the codes mostly not present at all. For all the 3 codes, the control group had a total frequency of 1, the question prompts group had a total frequency of 0

and the task schedule group had a total frequency of 1. Two of the codes namely *awareness of own limitations* and *awareness of difficulties as they arise* were not manifested at all whereas the third code i.e. *awareness of own capacities* had a total frequency of 2. Clearly, there was no major difference in the total frequencies of the codes for Declarative Knowledge among the three treatment groups.

Table 6-18 : Summary of frequencies for the elements of Knowledge of Cognition across the three treatments as manifested in the think-aloud protocols

	Control	Question Prompts	Task schedule	Total
Declarative knowledge				
Awareness of own capacities	1	0	1	2
Awareness of own limitation	0	0	0	0
Awareness of difficulties as they arise	0	0	0	0
Conditional knowledge				
Awareness of when to use a learning procedure	0	6	4	10
Awareness of why use a learning procedure	5	11	8	24
Procedural knowledge				
Awareness of how to carry out tasks	11	12	17	40

Conditional Knowledge: Conditional Knowledge had 2 codes namely *awareness of when to use a learning procedure* and *awareness of why use a learning procedure*.

Awareness of when to use a learning procedure was not manifested among the control group's cases but had a frequency of 6 and 4 for the question prompts and task schedule groups respectively. There was a clear difference in manifestation in the protocols between the control group on the one hand and the question prompts and task schedule on the other, with the control group not manifesting the codes of conditional knowledge at all. There was no major difference between the latter two treatment groups i.e. question prompts and task schedule.

Awareness of why use a learning procedure was found in all three groups, with the question prompts group having the highest frequency of 11. The difference among the three groups was not major.

Overall, there was no major difference for Conditional Knowledge between question prompts and task schedule groups, but the control group's frequencies were very low.

Procedural Knowledge: Procedural Knowledge had 1 code i.e. *awareness of how to carry out tasks*. The frequencies were 11, 12 and 17 for control, question prompts and task schedule groups respectively. There was no major difference among the group frequencies.

For Knowledge of Cognition, overall, there was no major difference among the three treatments and the frequencies of the three elements of Knowledge of Cognition in the think aloud protocols were quite low.

(ii) Knowledge of Regulation of Cognition

Knowledge of Regulation of Cognition has 5 elements, namely Planning, Information Management Strategies, Monitoring, Debugging Strategies and Evaluation. Table 6-19 shows how the codes of these 5 elements manifested themselves in the think aloud protocols across the three treatments.

Planning: Planning had 5 codes. Of all the codes for Planning, only *allocation of resources for problem solving* and the *setting of goals of problem solving* had a major difference in frequencies among the three treatment groups. The task schedule group had the highest manifestation for the code *allocation of resources* (4) whereas the protocols of the other treatments had none (0). Even then, the overall frequencies for *allocation of resources* were low.

The question prompts group had the highest frequency for *setting of goals* (23) with the frequencies of the other groups also being fairly high at 14 and 12 for control and task schedule respectively.

Two of the other codes had high frequencies but with no major difference among the three treatment groups. These were *activation of prior knowledge* and *sequencing of actions to be taken to lead to the solving of the problem*.

The remaining 2 codes i.e. *paying attention to important information given* in the problem statement before commencing problem solving and *seeking of information not given* in the problem statement before commencing problem solving had low frequencies. There was no major difference in the frequencies of the codes among the three groups.

Information Management Strategies: Information Management Strategies had 1 code namely *processing of information for problem solving*. There were several examples of ways in which the code was expected to manifest itself in the think aloud protocols (See Appendix L). Frequencies for the code among the three treatments were fairly high but without a major difference among the groups as can be seen in Table 6-19.

Monitoring: Monitoring had 4 codes. The codes generally had low frequencies, though all had a major difference among the treatments. For three of the codes namely *checking for errors for a sub-task*, *checking solution integration* and *monitoring cognitive effort*, the control group had very low frequencies, with one code having a frequency of 0. The other two groups i.e. question prompts and task schedule registered clearly higher frequencies for these 3 codes but without a major difference among them. For the code named *checking number of complete tasks against time*, the control and question prompts groups had 0 and 1 frequencies respectively, with the task schedule group having a clearly higher frequency of 8.

Table 6-19 : Summary of frequencies for elements of Knowledge of Regulation of Cognition as manifested in the think-aloud protocols for the three treatments

	Control	Question Prompts	Task schedule	Total
Planning				
Activation of prior knowledge	14	16	11	41
Allocating resources for problem solving	0	0	4	4
Attention to important information given, before commencing	2	5	2	9
Seeking information not given, before commencing	6	5	6	17
Sequence of actions for problem solving	13	17	18	48
Setting of goals	14	23	12	49
Information management strategies				
Processing of information for problem solving	17	17	19	53
Monitoring				
Checking for errors for a sub-task	2	7	4	13
Checking number of complete tasks against time spent	0	1	8	9
Checking solution integration	1	5	4	10
Monitoring cognitive effort	2	6	3	11
Debugging strategies				
Strategies for ensuring adequate information during problem solving	0	0	1	1
Strategies for error reduction	4	7	12	23
Strategies to improve time utilization	2	0	5	7
Evaluation				
Effectiveness of the approach used	1	3	7	11
Justification for the choice of the solution	2	14	15	31
Quality of the entire solution	13	9	9	31
Was the time for the project adequate?	0	0	1	1
Were all the tasks completed?	0	0	1	1

Debugging Strategies: Debugging Strategies had 3 codes. In general, Debugging Strategies had low frequencies among its codes. The code named *strategies for ensuring adequate information during problem solving* had a total frequency of 1 for the three groups whereas the code named *strategies for error reduction* had the highest total frequency of 23. There was no major

difference among the three groups for the code named *strategies for ensuring adequate information during problem solving* because it was not manifested in the protocols, having a total frequency of 1 across the three groups. However, there was a major difference in frequency among the three groups for the other two codes, with the task schedule group having the highest frequency in both codes, and question prompts and control groups having comparatively low frequencies.

Evaluation: Evaluation had 5 codes. There was a major difference for two of the codes i.e. the *effectiveness of the approach used* and *provided justification for the choice of the solution*. In the case of the code *effectiveness of the approach used*, the question prompts had the highest frequency. For the code named *provided justification for the choice of solution*, questions prompts and task schedule had clearly high scores compared to the control group, but there was no major difference between the frequencies of the two. For the code named *quality of the entire solution*, the control group had the highest frequency but there was no major difference in frequency among the three groups. For the remaining two codes i.e. *finding out if the time for the project was adequate* and *checking whether all the tasks were completed*, very low frequencies were realized. There was no difference in frequency for the 2 codes among the three treatment groups, with the total frequency across the groups being 1 for each of them.

b) Comparison of the effectiveness of the Scaffolding methods on level of Awareness of the Problem Solving Processes

The problem solving processes adopted for the study were (a) Problem Representation, (b) Solution Process, (c) Solution Justification, and (d) Monitoring and Evaluation. Table 6-20 shows how the codes of these 4 processes manifested themselves in the think aloud protocols of the 15 cases across the three treatments.

Problem Representation: Problem Representation had 7 codes. The codes generally had a moderate to high frequency, with one of its codes named *identification of relevant information (knowledge and skills)* having the highest code frequency not only among the codes for problem representation, but among all the codes for the 4 problem solving processes. For the

code named *checking whether the problem is stated completely*, the lowest total frequency among the codes of Problem Representation across the 3 treatment groups was 12.

Table 6-20 : Summary of frequencies of the codes for the problem solving processes as manifested in the think-aloud protocols for the three treatments

	Control	Question Prompts	Task schedule	Total
Problem representation				
Statement of sub-goals	16	15	13	44
The problem is stated clearly	5	4	8	17
The problem is stated completely	4	5	3	12
Identification of relevant information components	18	10	15	43
Identification of relevant information context-setting, players	11	13	21	45
Identification of relevant information (knowledge and skills)	12	36	35	83
Seeking relevant information that was not provided	5	7	4	16
Solution generation				
Solution is identified in terms of components	18	32	31	81
Explicit explanation or thoughts of how the solution will work	7	10	11	28
How well does the solution include the required sub-tasks – functionally	12	15	21	48
How well does the solution include the required OOP components	0	5	6	11
Inclusion of user interface considerations	8	6	7	21
Solution justification				
Constructing argument	15	14	21	50
Providing evidence	3	8	8	19
Monitoring and Evaluation				
Alternative solution effectiveness	1	8	15	24
Passing judgment on the overall value of the solution	11	16	13	40
Solution strengths and weaknesses	1	4	5	10
Reasons for alternative solution choice	2	10	19	31
Were all solution sub goals completed	2	3	2	7
Checking for errors for a sub-task	0	1	8	9
Checking number of complete tasks against time spent	2	7	4	13
Checking solution integration	1	5	4	10
Monitoring cognitive effort	2	7	4	13

For 5 out of 7 codes, there was no major difference among the three treatments. These codes are *statement of sub-goals*, *checking whether the problem is stated clearly*, *checking whether the problem is stated completely*, *identification of relevant information components*, and *seeking relevant information that was not provided*. For the 2 codes with a major difference among the treatments, the control group had the lowest frequency. For one of these codes named *identification of relevant information context* such as the problem setting and players, the task schedule group had a clearly higher frequency compared to control and question prompts groups. For the code named *identification of relevant information (knowledge and skills)*, the task schedule and question prompts groups had frequencies which were clearly higher than that for the control group. However, there was no major difference between the frequencies of the question prompts and task schedule groups.

Solution Process: The Solution Process had 4 codes. Out of the 5, 3 had major differences among the treatment groups. The 2 codes with no major difference in frequency across the treatments, named *explicit explanation or thoughts of how the solution will work* and *inclusion of user interface considerations* had frequencies which were moderate but similar across the three treatment groups. Out of the 3 codes with major differences, the question prompts and task schedule groups had clearly higher frequencies than control group for the codes named *solution is identified in terms of components* and *how well does the solution include the required OOP components*. However, there was no major difference between the frequencies of the task schedule and question prompts groups. The task schedule group had clearly higher frequencies than question prompts in the third code named *how well does the solution include the required sub-tasks – functionally?*

Overall, in all codes of the Solution Process, the frequencies were moderate to high and question prompts and task schedule groups had higher frequencies than the control group.

Solution Justification: Solution Justification had 2 codes named *constructing arguments* and *providing supportive evidence*.

The code named *constructing arguments* generally had high frequencies across the three groups but the task schedule group had a clearly higher frequency than the other two groups.

The code named *providing evidence* had lower frequencies in general, with the control group having very low frequencies. Both question prompts and task schedule had a frequency of 8, which was clearly higher than the frequency of the control group.

Monitoring and Evaluation: Monitoring and Evaluation had 9 codes. Except for the code named *passing judgment on the overall value of the solution*, all the other codes generally had low frequencies, with the lowest group frequency being 0 and the highest 19. The control group had the lowest frequency in all the codes. Out of the 9 codes, 4 codes had frequencies which were not majorly different from each other. In 3 of the remaining 5 codes, task schedule group had clearly higher frequencies compared to the question prompts group. These codes are *alternative solution effectiveness*, *reasons for alternative solution choice* and *checking for errors for a sub-task*. In the remaining 2 out of 5, both task schedule and question prompts had significantly higher scores than the control group, but there was no major difference between the frequencies of the two groups. These two codes are *solution strengths and weaknesses* and *checking solution integration*.

In 6 out of 9 codes, there was no major difference between question prompts and task schedule. In 5 out of 9 codes, task schedule and question prompt groups had clearly higher scores compared to the control group.

6.3.3 Summary of Results for Research Question 3

Does the use of metacognitive scaffolds (question prompts or task schedule) have an effect on the students' metacognitive awareness while solving an ill-structured programming task?

(a) Does the use of question prompts influence students' level of metacognitive awareness while solving an ill-structured programming task?

(b) Does the use of a task schedule influence students' level of metacognitive awareness while solving an ill-structured programming task?

Table 6-21 provides a summary of the results of the 3 treatments on the level of metacognitive awareness, obtained using the think-aloud method. It can be seen that overall, there was a major difference between the frequencies of the control group on the one hand and question

Task schedule on the other.

Summary of the code frequencies for the 8 elements of metacognition as found in the protocols for cases from the three treatment groups

Knowledge Element	Control Group	Question Prompts	Task Schedule	Freq. Total	Major Difference?	Best Group
KNOWLEDGE	1	0	1	2	NO	ALL Close
KNOWLEDGE	5	17	12	34	YES	Question Prompts and Task Schedule
KNOWLEDGE	11	12	17	40	NO	ALL Close
KNOWLEDGE	49	66	53	168	NO	ALL Close
KNOWLEDGE	17	17	19	53	NO	ALL Close
KNOWLEDGE	5	19	19	43	YES	Question Prompts and Task Schedule
KNOWLEDGE	6	7	18	31	YES	Task Schedule
KNOWLEDGE	16	26	33	75	Yes	Question Prompts and Task Schedule
KNOWLEDGE	110	164	172	446	Yes	Question Prompts and Task Schedule

of the two areas of metacognition i.e. Knowledge of Cognition and Knowledge of Task Schedule, it is clear that knowledge of Cognition was manifested less. This is evident from the 3 lowest total code frequencies per element, with the lowest total code frequency being 2, compared to the highest of 168. In terms of the individual elements, there was a major difference between question prompts and task schedule in 3 out of 8 elements: Knowledge, Monitoring and Evaluation. The task schedule group had a higher total code frequency compared to question prompts for the element called Knowledge. In the remaining 4 elements, there was no major difference among the

three groups i.e. control, task schedule and question prompts.

Overall, both the question prompts and task schedule had a positive influence on the student's level of metacognitive awareness but there was no major difference in effect between the two scaffolds.

Discussion of Results for Question 3

In the quantitative results obtained using the Metacognitive Awareness Index (MAI), there was no significant difference between Conditional Knowledge and Procedural Knowledge for the 3 treatment groups. However, for Declarative Knowledge, there was a significant difference among the groups, with the task schedule group having the highest mean ranks. Overall, the three elements of Knowledge of Cognition had lower mean ranks than those of the five elements of Knowledge of Regulation of Cognition. This has been confirmed to be the case when using the qualitative think aloud protocols method because the elements of Knowledge of Cognition generally had the lowest frequencies among the 8 elements of metacognition. Moreover, for 2 of the elements, there was no significant difference among the three groups, this time the elements being Declarative Knowledge and Procedural Knowledge. The third element named Conditional Knowledge was slightly different, with the task schedule and question prompts groups having clearly higher frequencies compared to the control group, though there was no major difference between the frequencies of these two groups.

Knowledge of Cognition requires students to step out of themselves and use their own cognitive processes as objects when thinking and reflecting about problem solving processes. This is quite challenging for them.

For Knowledge of Regulation of Cognition, the students had higher mean ranks than Knowledge of Cognition when the self-report MAI was used. This was also the case when using the think aloud method. Students can apply cognitive processes which they are conscious of as well as those which are automated and therefore they are not conscious of. Hence, students register better results with Knowledge of Regulation of Cognition because they can apply knowledge

not even state (because of being automated), in regulating themselves during problem solving.

Qualitative results obtained using MAI, the task schedule group had significantly better than the question prompts group, in all elements of Knowledge of Regulation of Concept Evaluation. However, in the results obtained using think aloud protocols there was no difference in the effect of treatments on Knowledge of Regulation of Concept Evaluation between the task schedule and question prompts. Tobias and Everson (2002) stated that to get results close to classroom learning outcomes during the measurement of metacognitive awareness, it was necessary to use a two step-process: Ask students to state what they know and then ask them to use the knowledge in carrying out a task. The approach used in this research is similar. The students responded to an MAI on their metacognitive knowledge after which they were required to apply their metacognitive knowledge during problem solving with the think aloud method.

Therefore, the results obtained at the end of the think aloud process may be considered to be a more accurate picture of the effect of the two scaffolds i.e. question prompts and task schedule on metacognitive awareness. Hence, the two scaffolds were considered to have a significant effect on metacognitive awareness. Tobias and Everson (2002) called their method Monitoring Assessment and applied it to the element of Knowledge of Regulation of Concept Evaluation called monitoring.

4) explained that the self-report method could be inaccurate due to the students being required to engage in the difficult process of abstract recall of cognitive processes they experienced in the past. This may not be easy because that knowledge may not be in the Working Memory (WM) when completing the questionnaire but in the Long Term Memory (LTM). Therefore, the students have to undergo the process of retrieval or activation of knowledge from the LTM to WM because they can only use the knowledge that is in WM. Therefore, the results become inaccurate due to false memories (Someren, Barnard, & Sandberg,

The students may also be unaware of some of the automated processes which they carry out and therefore asking them to recall them may not yield accurate results (Gama, 2004; Someren et al., 1994). The students may also decide to make responses which please the experimenter (Tobias & Everson, 2002; Van Marrienoer & Sweller, 2005) or responses which do not reveal their weaknesses. Some of the recall questions may be difficult to answer, leading to false answers (Someren et al., 1994).

Therefore, for more accurate results when evaluating the effect of a treatment on cognitive processes, it is better to combine recall with application tasks. The commonly used self-report methods should be complimented with methods requiring students to apply their knowledge to a task.

6.3.4 Summary of Results for Research Question 4

Does the use of metacognitive scaffolds (question prompts or task schedule) have an effect on the student's level of application of problem solving processes when solving an ill-structured task?

- a) *Does the use of question prompts have an effect on students' level of awareness of problem solving processes solving namely problem representation, developing solutions, making justifications and monitoring and evaluation of solutions when solving an ill-structured programming task?*
- b) *Does the use of a task schedule have an effect on students' level of awareness of problem solving processes solving namely problem representation, developing solutions, making justifications and monitoring and evaluation of solutions when solving an ill-structured programming task?*

Table 6-22 provides a summary of the effects of the treatments on the level of awareness of problem solving processes, obtained using the think-aloud method. It can be seen that, overall, there was a major difference between the frequencies of the control group on the one hand and question prompts and task schedule on the other.

Table 6-22 : Summary of the code frequencies for the problem solving processes as found in the think aloud protocols from the three treatment groups

Problem Solving Process	Control	Question Prompts	Task Schedule	Total	Major Difference?	Best Group
PROBLEM REPRESENTATION	71	90	99	260	Yes	Question Prompts and Task Schedule
SOLUTION PROCESS	45	68	76	189	YES	Question Prompts and Task Schedule
SOLUTION JUSTIFICATION	18	22	29	69	YES	Task Schedule
MONITORING AND EVALUATION	22	61	74	157	YES	Question Prompts and Task Schedule
TOTAL	159	243	279	675	Yes	Question Prompts and Task Schedule

From this summary, it is clear that the control group had the lowest overall frequencies compared to the other two groups i.e. task schedule and question prompts. But between these latter two groups, there was no major difference between overall frequencies, except for the process named Solution Justification. For this process, the task schedule group had the highest frequency which was clearly greater than that of the control and question prompt groups.

It can be concluded that overall, the two treatments had a clear positive effect on the level of application of the four problem solving processes.

Discussion of Results for Question 4

From the results obtained using the self-report questionnaire for the problem solving processes, there was a significant difference among the mean ranks for the processes named Problem Representation, Solution Justification and Monitoring and Evaluation among the three treatments ($\alpha=0.05$). The question prompts had the highest mean ranks in each case. Only the Solution Process had mean ranks for which there was no significant difference among the three treatment groups ($\alpha=0.05$).

At the same time, from the analysis of think aloud protocols, it was clear that the question prompts and task schedule treatments had a major effect on the students' awareness of problem solving processes. The two groups had higher total code frequencies than the control group in all the 4 processes. However, unlike data from the self-report questionnaire, in which question prompts outperformed the task schedule in 3 out of 4 processes, from the think aloud protocols data, there was no major difference among the total frequencies of the two treatments. The only exception was the Solution Justification process in which the task schedule group clearly outperformed the question prompts group.

This difference between results from the self-report and think aloud methods can be explained in terms of weaknesses of the self-report method and hence the need to combine two methods when measuring cognitive processes. The self-report method requires the students to engage in abstract recall (Tobias & Everson, 2002), which does not work very well because the student has to first retrieve the information from LTM. This is especially the case if the student is required to respond to the questionnaire sometime after carrying out a cognitive task (Gama, 2004; Someren et al., 1994). Moreover, students cannot report automated processes because of not being conscious of them even though they can apply these processes (Gama, 2004; Someren et al., 1994). At the same time, students can give responses to please the researcher when the self-report method is used (Gama, 2004; Tobias & Everson, 2002; Van Marriënboer & Sweller, 2005).

To get results close to the reality in classroom learning, the students should first be asked what they know and then be required to apply the knowledge in carrying out a task. This is to enable the students to demonstrate how well they know cognitive processes and at the same time give them a chance to apply automated knowledge which they might be unconscious of. This method has been said to be natural, robust, with good reliability and internal validity (Gama, 2004) because it overcomes the difficulties of abstract recall and inaccurate responses in the self-report method (Tobias & Everson, 2002).

Therefore, according to results from the two methods i.e. self-report and think aloud, question prompts and the task schedule were effective as scaffolds for the students' awareness of problem solving processes. This is due to the big difference in mean ranks between the control group on the one hand and question prompt and task schedule groups on the other. However, none of question prompts and task schedule was better than the other in terms of increasing the student's awareness of the 4 problem solving processes.

Chapter 7 : Conclusions and Future work

7.1 Overview of the Findings

From study 1, it can be concluded that the use of adaptive learning features, achieved through the use of Machine Learning algorithms, was effective. This was clear from the students' responses to the questionnaire. However, care needs to be taken in terms of how the features are introduced to the students, especially for learning modes which have not been used extensively, such as web-based learning. Even if the features have the potential to influence learning positively, optimal results may not be guaranteed. The group starting with the plain system without adaptive features when studying Class Design followed by studying Inheritance with adaptive features activated did better in both CAT 1 and CAT 2 compared to the other group which started with adaptive features, then studied the second part of the course with adaptive features deactivated. A gradual introduction of new elements of information in the learning context avoids the students suffering undesirable intrinsic and extraneous cognitive load effects, which reduce the level at which the students achieve.

From Study 2 and the comparative case study at the end of Study 2, it can be concluded that the two treatments i.e. use of question prompts and use of task schedule are effective in fostering awareness among the students of their problem solving processes and metacognitive knowledge. This was confirmed from the comparison of pre-test and post test scores using the self-report method as well as from comparing total code frequencies across the treatment groups using the think aloud method. It was also found out that the effect of the scaffolds on Knowledge of Cognition was not major, using both the self-report and think aloud methods. This was possible because the scaffolds did not address these elements of metacognition. It was not conclusive whether this is because one scaffold cannot address all the elements adequately.

Overall, it can also be concluded that there was no major difference between the question prompts and task schedule scaffolds in terms of effectiveness in increasing the students'

awareness of their problem solving processes and metacognitive knowledge. This especially became clear from using the think aloud method, which required the students to apply these cognitive processes to an ill-structured task. Therefore, any of the scaffolding methods can be used but the challenge is to ensure that it covers all the processes of problem solving and all the elements of metacognitive knowledge. Otherwise, there should be another scaffold used concurrently to support the problem solving processes and elements of metacognitive knowledge not supported by the scaffold that is used. But the number of scaffolds needs to be limited to avoid introducing undesirable cognitive load effects to the learning context, due to the number of things the student needs to pay attention to at the same time, during learning.

7.2 Achievements

This section presents a summary of the achievements of the research. The achievements have been presented in terms of the objectives of the research.

The overall objective of the research was to develop a web-based learning system, informed by learning theory and technological considerations, such as incorporation of adaptive features to make the user interface adaptive. The system was to have inbuilt scaffolds for students to use while acquiring knowledge of the OOP domain as well as scaffolds for the students' awareness of metacognitive knowledge and problem solving processes, to be used as the students engage in ill-structured problem solving.

1. An Approach for designing and developing web-based learning systems with adaptive and metacognitive regulation support

An approach was developed for guiding the development of the web-based learning system used for the research. This is an important contribution to the engineering of e-learning systems. It not only emphasizes inclusion of supportive features for both cognitive and metacognitive learning but also provides a template for factoring into system design critical factors such as user understanding and profiling, automatic profile updating, provision of personalized user interfaces, domain or task modeling, and theory-based design of learning as opposed to mere delivery of content. This methodology will impact the field of development of

web-based systems whose users vary in many ways and therefore, making it necessary for the systems to respond to unique learner needs.

The approach was based on Nam and Smith-Jackson's (2007) Integrated Design Process, which used learning theory to inform goals of system design and the features to include in the system. Integrated Design Process integrated instructional design and human computer interface design. It supported online teaching of domain knowledge, informed by behaviorism and cognitivism. It did not support constructivist learning, such as solving complex problems while applying self-regulation and ill-structured problem solving processes. Integrated Design Process also supported static interaction between a typical (i.e. general) user and the system. It did not consider personal characteristics of learners, hence aspects of the system such as navigation design, presentation design and page design were general and could not fit the multiple audience groups of students using the system.

The Integrated Design Process was improved through incorporation of ideas from the Web System Development Methodology (WSDM) by Olga De Troyer (2001). The incorporated ideas include using audience-driven navigation, page and presentation design to support the needs of different types of users of the system. Effective navigation design is particularly important in web-based learning systems, where the students have to chart their way through the system (Hadjerrouit, 2006). The audience-driven approach was useful in realizing adaptive features in the user interface. Machine Learning algorithms were used to implement adaptive educational hypermedia technologies (Brusilovsky, 1999) such as adaptive navigation support with link annotation, adaptive navigation support with link hiding and adaptive presentation support. Other web-design concepts such as wire-framing were also incorporated. The new approach also included support for learners' metacognitive control of their problem solving.

The phases and processes through which the system was developed were defined taking into consideration the audience-driven classes and the three sub-systems namely instructional, adaptive user interface and problem solving support. New processes added include navigation design, presentation design and data design. Other new processes include a process for the

development of course materials and processes for the evaluation of the instructional, adaptive user interface and problem solving support sub-systems.

2. Supporting learning across the continuum of knowledge acquisition

A learner's prior knowledge, which is equivalent to the expertise he already has, is the most important factor for successful complex problem solving (Kalyuga, 2009). Prior knowledge reduces the cognitive load suffered by the learner at the beginning of problem solving since he can just retrieve the already successfully acquired knowledge and intentionally apply it to problem solving (Ge, 2001). This not only reduces the effort and time spent in problem solving but also leads to successful problem solving and better development of cognition. This research has contributed to knowledge by demonstrating the importance of cognition to complex problem solving and providing a strategy for incorporating it as one of the ways of providing learning support to reduce cognitive load during complex problem solving.

The system supported acquisition of domain knowledge and strategies as well as the subsequent application of this knowledge and other cognitive processes such as metacognitive control and problem solving processes during ill-structured problem solving. The system supported the student across the continuum of knowledge acquisition, from acquisition of domain knowledge by novice learners to complex problem solving by expert learners (Jonassen, McAleese, & Duffy, 1993). This was important since these aspects of learning are inter-related and are both critical as the student progresses from novice level to expert level in a given domain (Moallem, 2001). The system provided different types of support during the learning process, suited to the learner's current stage of learning and the learner's characteristics such as level of knowledge and tendency to use or not to use additional materials.

3. Incorporation of an evolutionary approach to develop an up-to-date e-learning system

An evolutionary approach, which is recommended when developing new types of systems, was used. The approach is recommended because it provides opportunities to include new information at different points during the development of the system. The evolutionary

approach was made possible through the incorporation of formative evaluation for each of the three sub-systems in the new Integrated Design Process. The feedback obtained was used to refine the features of the system prototype before the final version was deployed.

As part of formative evaluation, the adaptive user interface was evaluated to find out the kind of experience the users were given, the usability of the adaptive features to them and how they perceived the value of these features. Evaluation was also done to find out whether the adaptive features were suitable to the learner's knowledge level and preference to use additional materials. In addition, the investigation was to establish the extent to which learner control was allowed by the system and also whether the features interfered with learning or not.

In the evaluation of the instructional sub-system, feedback was obtained from various types of experts such as content-specific educators, subject matter experts, instructional design experts and adaptive learning experts. Feedback was also obtained from learners in a field trial of the learning system.

For the problem solving support system, feedback was collected to establish the presence of constructivist approaches in the system and whether the scaffolds for problem solving processes and metacognitive awareness were effective.

4. Model for guiding design of adaptive interfaces for learning

A model was developed to guide the design of adaptive user interfaces for learning. The model incorporated a domain model, learner model and adaptation model. The domain model constituted course goals, concepts and knowledge modules. The learner model had three layers. It had a top layer constituting both domain dependent and domain independent attributes. The model also had a goal level layer which constituted learner details such as level knowledge for the goal. Finally, there was the concept level layer that was an overlay over the domain model, with each concept having the learner's knowledge level and indication of use of additional materials for the concept. The overlay model was updated with values inferred from

previous learner activities, by using 2 Machine Learning algorithms. These algorithms were Heterogeneous Value Difference Metric (HVDM) and Naive Bayes Classifier (NBC). For the actual adaptation, adaptation model heuristics were used together with appropriate learner model attribute values, an appropriate abstraction of the domain model and hypermedia adaptation technologies developed by Brusilovisky (1999).

5. Investigation and recommendation of suitable Machine Learning algorithms

Three flavors of the K-Nearest Neighbour (k-NN) algorithm were tweaked and tested to ensure they could be used in adaptive user interfaces for learning. They were tested in a field trial during formative evaluation of the study, with students similar to those who were to participate in the actual study. The aim was to identify one version of the algorithm suitable for making inferences of concept test scores at the beginning of learning, to guide learner-centric adaptation of interfaces used for learning. The three flavors of k-NN were: Heterogeneous Overlap Euclidean Metric (HOEM) and HVDM with the parameter $Q = 1$ or 2 . The difference between HOEM and HVDM is that, whereas HOEM uses the overlap measure to determine the distance between 2 nominal attributes values, HVDM uses the Value Difference Metric (VDM). Unlike the overlap measure which only checks sameness of 2 nominal attribute values, VDM checks sameness of the attribute values and also considers the proportion of training examples with the attribute values and their classification. HVDM with $Q=2$ was found to give the highest prediction accuracy of the learner's knowledge level for each of the concepts.

Besides the availability of this information, other details were provided, such as how to represent instances and training examples, use inferred values for adaptation and evaluate the algorithm. This information would give anybody interested in using HVDM for adaptive interfaces for learning a head start and will reduce time taken in testing the algorithm before it is used.

6. Development of a number of re-usable instructional material components and research instruments

A number of instructional material components and instruments were specifically developed for the research. These include:

- a) **Learning objects** for the OOP course which were developed and sequenced, taking into consideration instructional theories such as Component Display Theory (Merrill, 1994), Elaboration Theory (Reigeluth, 1999) and Cognitive Load Theory (Kalyuga, 2009; Van Marrienoer & Sweller, 2005). The objects were used to achieve adaptation.
- b) **Question prompts** used were a refinement of the general question prompts used by Ge (2001). They were customized by being expressed in terms of OOP and the ill-structured problem solving processes.
- c) **Coding scheme** for establishing the level of metacognitive awareness was also developed. It was guided by Brown's (1987) version of the components of metacognitive knowledge and the Metacognitive Awareness Index developed by Schraw and Dennison (1994). OOP-based examples on how the codes would manifest themselves in the protocols were developed. The reliability of the codes was established before they were used.
- d) **Coding scheme** for the problem solving processes was also developed. It was guided by the models of Voss and Post (1988) and Sternberg (2009). It was defined in operational terms using examples from OOP.
- e) **A questionnaire for determining perception** of the value of adaptive features of an online learning system. The questionnaire was validated before being used for the study.

Finally, in terms of the specific objectives of the research, the following conclusions were made:

- 1) The system's adaptive support features can improve the student's learning achievement,

expressed in terms of learning scores, when the presentation of the knowledge elements is gradual. For example, from simple (few elements presented at first) to complex (all the elements presented). This is a way of managing cognitive load. In this research, the group which learned with the plain system before using the system's adaptive features performed slightly better than the group presented with the full complexity of the system at once i.e. the web-based learning system with OOP concepts and adaptive user interface features. The difference between the groups was not significant, though, and this could be due to other factors beyond the system, such as pressure to pass exams.

It was also established that the students' perceived value of the adaptive features was quite positive, meaning that the features contributed to improving the students' learning experience.

- 2) The research established that both question prompts and task schedule scaffolds improved the students' level of awareness of their metacognitive knowledge. The results show that the scaffolds are effective to the same extent. It was established that if the scaffolds can be used intentionally, they can improve the students' level of metacognitive awareness.
- 3) The research also established that both the task schedule and the question prompts scaffolds have a positive effect on the student's level of awareness of their problem solving processes. Therefore, scaffolds can be used intentionally to improve the students' awareness of their ill-structured problem solving processes. It was established in this research that both scaffolds are equally effective in increasing the students' level of awareness of complex problem solving processes and the quality of the final product of problem solving.

7.3 Limitations of the Study

Provided below are some of the challenges which were experienced during the research and remedies for some of them.

Firstly, due to the way the research was designed, especially Study 2 which was an experimental study with three treatment groups, the number of students needed to participate was big - at least 90 plus a margin to cater for attrition. As a result, it took quite a while to get the right cohorts for the experiment. At one point, there was one university willing to allow students to participate in the experiment but their access to the Internet was so poor and restricting for students that the research could not go on after some time. Eventually, it was possible to get the right number of students of the OOP course. Because the course was online, it was possible. The names of the students were randomly divided into the two experimental groups for Study 1 and three experimental groups for Study 2. The groups were confirmed to be homogenous during Study 1 because of equal variances of scores in CAT 1 and CAT 2, which were taken by both groups of students. As the experiment went on, some students dropped out, especially from the parts of the research which were not contributing to the students' end-of-semester grades. The sample size reduced and as a result, the data was analyzed using non-parametric methods.

Secondly, because of being in two separate campuses, the student's did not have equal access to the Internet. In one of the campuses, the number of computers available was so limited for the number of participating students that something had to be done to ensure the research went on. An arrangement was made to provide the students with a dedicated computer laboratory during the period of the research study.

Thirdly, self-report questionnaires were used to establish the students' perception of the value of the adaptive user interface features. They were also used to establish change in the level of metacognitive awareness and problem solving processes as a result of using scaffolds. The self-report method has been criticized for requiring students to engage in abstract recall (Tobias & Everson, 2002). This puts students under pressure as they try to recall past experiences stored in the Long Term Memory, which as a result has to be retrieved to Working Memory first before it can be externalized. This can lead to memory errors (Someren et al., 1994). The method also predisposes students to making errors because they might not understand some of the questions. The students might also want to give responses to make the researcher happy (Van

Marrienboer & Sweller, 2005). The students are also not conscious of the automated knowledge and therefore this knowledge might not be reflected in their responses to the self-report questionnaire items.

An attempt was made to improve the measurement of the effects of the scaffolds by selecting some students to participate in think aloud protocol sessions as a source of more information. The students were able to say what they knew or did not know in their response to the questionnaire items and they were also given an opportunity to apply the knowledge to a task i.e. ill-structured problem solving during the think aloud sessions. This was an attempt to implement the method proposed by Tobias and Iverson (2002).

Lastly, evaluation of domain concepts, the quality of the product of problem solving and the extent of application of OOP concepts during problem solving contributed to the students' semester grade. This possibly added pressure to the students to do well because they needed to pass their semester exams. Thus, the students were motivated, due to extra pressure, to commit more resources to learning, including cognitive resources. This could be the reason why CAT scores and problem solving product quality scores were not significantly different among the treatment groups. The experiment needs to be carried out under different circumstances, perhaps with less emphasis on the need for students to pass their examinations.

7.4 Further Work

To further extend and enhance some of the findings from this study, further work needs to be done.

Firstly, the course needs to be studied under other sets of conditions so as to further observe the effects of various scaffolds on student scores and the quality of problem solving product. For example, the experiment needs to be carried out using a course that does not contribute to the students' semester grade. An example could be a life-long learning course or a course in a blended learning environment where the course only plays a supplementary role to face to face learning. The aim is to find out if there will be differences in the cognitive knowledge and

problem solving product quality scores as a result of using the scaffolds in a slightly different context.

Secondly, there is need to develop scaffolds which focus on part of the elements of knowledge of metacognition or some of the problem solving processes instead of having one scaffold to support all aspects of the cognitive processes. For example, introduce one scaffold to support knowledge of cognition and another scaffold to support Knowledge of Regulation of Cognition. But care needs to be taken because interacting with technology is a cognitive process and if many such scaffolds are introduced, there are chances of increase in intrinsic and extraneous cognitive load effects. This is due to need for the student to attend to multiple scaffolds and at the same time pay attention to the domain content being learned or problem being solved. The scaffolds might themselves need scaffolds in order to be used well!

Thirdly, Tobias and Everson (2002) proposed a two step process of measuring the effect of scaffolds on the knowledge of metacognition and other cognitive processes, such as problem solving processes. The evaluation is based on (a) what the student indicates he knows and can apply to a task, (b) what the student indicates he does not know but can apply to a task successfully, (c) what the student indicates he knows but is unable to successfully apply to a task, and (d) what the student indicates he does not know and is indeed unable to apply to a task (Tobias & Everson, 2002). The statistics collected on these 4 possible situations are used to establish the student's ability to carry out a cognitive process. This method needs to be applied in the evaluation of the cognitive processes in similar studies. The results should be compared with the results obtained from using the self-report method alone so as to obtain a better understanding of the effects of the scaffolds.

Finally, scaffolding approaches work by reducing undesirable cognitive effects and increasing the desirable Germane Cognitive Load. Germane cognitive load motivates the students to make use of available cognitive resources released due to reduction in undesirable cognitive load. There is need to incorporate into studies on scaffolding ways of measuring both the student's mental effort and performance. The learner's performance is the most used way of expressing

the learner's achievement in education, and it can be in the form of number of correct answers provided or the number of errors made in a task.

Other important aspects related to cognitive load should also be considered, such as mental effort, which refers to the cognitive resources expended in order to complete a task. This can give an indication of the student's level of expertise in the domain. If a lower amount of resources is spent to do a task, the student has a higher level of expertise. Information on the student's level of expertise can be used as a basis for adaptation during learning. It can also be an indication of the cognitive load experienced by the learner.

Bibliography

- Anderson, T., & Elloumi, F. (Eds.). (2004). *Theory and Practice of Online Learning*. Athabasca, Canada: Athabasca University.
- Azevedo, R., & Cromley, J. G. (2004). Does training on self-regulated learning facilitate students' learning with hypermedia? . *Journal of educational technology*, 96(3), 523-535.
- Azevedo, R., Cromley, J. G., & Seibert, D. (2004). Does adaptive scaffolding facilitate students' ability to regulate their learning with hypermedia. *Contemporary Educational Psychology*(29), 344-370.
- Azevedo, R., Cromley, J. G., Winters, J. G., Moos, D. C., & Green, J. A. (2005). Adaptive human scaffolding facilitates adolescents' self-regulated learning with hypermedia. *Instructional Science*, 33(5-6), 381-412.
- Azevedo, R., & Hadwin, A. F. (2005). Scaffolding self-regulated learning and metacognition - Implications for the design of computer-based scaffolds. *instructional Science*(33), 367-375.
- Baker, L. (1989). Metacognition, comprehension monitoring, and the adult reader. *Educational Psychology Review*, 1, 3-38.
- Becvar, L. A. (2005). Video-Based Reflective Tools for Developing Expertise in Hands-on Learning Environments. Department of Cognitive Science, UC San Diego.
- Benyon, D. R. (1993a). *Accommodating Individual Differences through an Adaptive User Interface*. Paper presented at the In Adaptive User Interfaces - Results and Prospects.
- Benyon, D. R. (1993b). Adaptive Systems; a solution to usability problems. *User Modelling and User Adapted Interaction*, 1-21.
- Benyon, D. R., & Murray, D. M. (1993c). Applying user modelling to human-computer interaction design *AI Review* (6), 43-69.
- Bergin, S., & Reilly, R. (2005). *Programming: factors that influence success*. Paper presented at the SIGCSE '05 Proceedings of the 36th SIGCSE technical symposium on Computer Science Education
- Bloom, B. S., & Krathwohl, D. R. (1956). *Taxonomy of educational objectives; the classification of educational goals, by a committee of college and university examiners* (1 ed.). New York: Longmans, Green.
- Boyle, T. (2003). Design Principles for Authoring dynamic, Reusable Learning Objects. *Australian Journal of Educational Technology*, 19(1), 46-58.
- Branch, J. L. (2000). Investigating the information-seeking processes of adolescents: The value of using think-alouds and think afters. *Library and Information Science Research*, 22(4), 371-392.
- Bransford, J. D., Brown, A. L., & Cocking, R. R. (2000). *How people learn: Brain, mind, experience, and school*. Washington, DC: National Academy Press.
- Brickell, G., & Herrington, J. (2006). Scaffolding learners in authentic, problem based e-learning environments: The Geography Challenge. *Australasian Journal of Educational Technology*, 22(4), 531-547.
- Brusilovskiy, P. (1996). Methods and Techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3), 87-129.

- Brusilovsky, P., Karagiannidis, C., & Sampson, D. (2004). Layered evaluation of adaptive learning systems. *International Journal of Continuing Engineering Education and Lifelong Learning*, 14(4/5), 402-421.
- Brusilovsky, P. (1999). Adaptive and Intelligent Technologies for Web-based Education. *Special Issue on Intelligent Systems and Teleteaching, Künstliche Intelligenz*(4), 19-25.
- Brusilovsky, P. (2001). Adaptive hypermedia. *User Modeling and User Adapted Interaction, Ten Year Anniversary Issue*, 11 (1/2), 87-110.
- Brusilovsky, P. (2003). *Developing adaptive educational hypermedia systems: From design models to authoring tools*. Dordrecht: Kluwer Academic Publishers, 377-409. Paper presented at the Authoring Tools for Advanced Technology Learning Environment.
- Brusilovsky, P. (2004). *Adaptive Educational Hypermedia: From generation to generation*. Paper presented at the Proceedings of 4th Hellenic Conference on Information and Communication Technologies in Education.
- Burak, G., & Mustafa, M. (2007). Machine Learning Based Learner Modeling for Adaptive Web-Based Learning *Machine Learning* 1133-1145.
- Burkhardt, J., Tienne, F., & Wiedenbeck, S. (2002). Object-Oriented Program Comprehension: Effect of Expertise, Task and Phase. *Empirical Software Engineering*(7), 115-156.
- Callear, S. (2000). Teaching Programming: Some Lessons From Prolog. *LTSN Centre for Information and Computer Science*.
- Chng, V. L. L. (2001). *Learning plans for student scaffolding*. Paper presented at the AARE 2001 International Education Research Conference - Crossing Borders: New Frontiers for Educational Research.
- Coombs, S. J. (2001). The teacher's role in using IT to manage student-centred learning. In A. Chang & I. Smith (Eds.), *Critical and Creative Thinking*. Singapore: National Institute of Education, Singapore.
- Dabbagh, N., & Kitsantas, A. (2004). Supporting self-regulation in student-centered web-based learning environments. *International Journal on E-learning*, 3(1), 40-47.
- Dabbagh, N., & Kitsantas, A. (2005). Using web-based pedagogical tools as scaffolds for self-regulated learning. *Instructional Science*, 33(5-6), 513-540.
- De Bra, P. (2000). Pros and cons of adaptive hypermedia in web-based education. *Journal of CyberPsychology and Behavior*, 3(1), 71-77.
- De Troyer, O. (2001). Audience-driven Web Design. In M. R. K. Siau (Ed.), *Information modelling in the new millennium*: IDEA Group Publishing.
- De Villiers, M. (2002). *The Dynamics in the Theory and Practice of Instructional Design*. University of Pretoria, Pretoria, South Africa.
- Domingos, P. (1995). *Rule Induction and Instance-Based Learning: A Unified Approach*. Paper presented at the The 1995 International Joint Conference on Artificial Intelligence(IJCAI-95).
- Ericsson, K. A., & Simon, H. A. (1993). *Protocol analysis: Verbal reports as data* Cambridge, Massachusetts: MIT Press.
- Fowler, M., & Scott, K. (2003). *UML Distilled: Applying the Standard Object Modeling Language* (Third ed.): Addison-Wesley
- Gagné, R. M. (1985). *Conditions of learning* New York: Holt, Rinehart and Winston.

- Gama, C. A. (2004). *Integrating Metacognition Instruction in Interactive Learning Environments*. University of Sussex, Brighton.
- Ge, X. (2001). *Scaffolding student's problem solving processes on an ill-structured task using question prompts and peer interactions*. Pennsylvania State University.
- Gouli, E., Papanikolaou, K., & Grigoriadou, M. (2002). *Personalizing assessment in adaptive educational hypermedia systems*. Paper presented at the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, and Lecture Notes in Computer Science.
- Greene, B., & Land, S. (2000). A qualitative analysis of scaffolding use in a resource-based learning environment involving world wide web. *Journal of Educational Computing Research*, 23(2), 151-179.
- Hadjerrout, S. (2006). *Creating web-based learning systems: An evolutionary development methodology*. Paper presented at the information science and IT Education Joint Conference.
- Hadjerrout, S. (2007). Applying a System Development Approach to Translate Educational Requirements into E-Learning. *Interdisciplinary Journal of E-Learning and Learning Objects*, 3.
- Hoffman, S. (1997). Elaboration Theory and Hypermedia: Is there a Link?. *Educational Technology & Society*, 37(1), 57-64.
- Johnstone, C. J., Bottsford-Miller, N. A., & Thompson, S. J. (2006). *Using the Think Aloud Method (Cognitive Labs) to Evaluate Test Design for Students with Disabilities and English Language Learners (Technical Report 44)*. Minneapolis, Minnesota: University of Minnesota, National Center on Educational Outcomes. Document Number).
- Jonassen, D. (1997). Instructional design models for well-structured and ill-structured learning outcomes. *Educational Technology Research and Development*, 45(1), 65-94.
- Jonassen, D. H. (1999). Designing constructivist learning environments. In C. M. Reigeluth (Ed.), *Instructional design theories and models: A new paradigm of instructional theory*, (Vol. Volume II, pp. 215-239). Mahwah, NJ: Lawrence Erlbaum Associates.
- Jonassen, D. H. (2000). Towards a design theory of problem solving. *Educational Technology Research & Development*, 48(1), 63-85.
- Jonassen, D. H. (2006). On the role of concepts in learning and instructional design. *Educational Technology Research and Development*, 54(2), 177-196.
- Jonassen, D. H., McAleese, T. M. R., & Duffy, T. M. (1993). *A Manifesto for a constructivist approach to technology in higher education*. Paper presented at the Design of constructivistic learning environments: Implications for instructional design and the use of technology.
- Kalyuga, S. (2009). Managing Cognitive Load in Adaptive ICT-Based Learning. *Journal of Systemics, Cybernetics and Informatics*, 7(5), 16-21.
- Katia, P., & Granger, M. J. (2000). A Developmental Model for Distance Learning using the Internet. *Computers and Education*, 3(4), 1-15.
- Khan, B. H. (Ed.). (1997). *Web-Based Instruction*. Englewood Cliffs, New Jersey: Educational Technology Publications.

- Kirschner, P., Sweller, J., & Clark, R. E. (2006). Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational psychologist*, 41(2), 75–86.
- Koohang, A., Riley, L., Smith, T., & Schreurs, J. (2009). E-Learning and Constructivism: From Theory to Application. *Interdisciplinary Journal of E-Learning and Learning Objects*, 5-2009.
- Lake, C., & Tessmer, M. (1997). *Constructivism's implications for formative evaluation*. Paper presented at the Association for Educational Communications and Technology.
- Langley, P. (1997). *Machine Learning for adaptive user interfaces*. Paper presented at the Proceedings of the 21st German Annual Conference on Artificial Intelligence.
- Langley, P. (1999). *User modeling in adaptive interfaces*. Paper presented at the Proceedings of the seventh international conference on User modeling.
- Lin, X., Schwartz, D. L., & Hatano, G. (2005). Toward Teachers' Adaptive Metacognition. *Educational Psychologist*, 40(4), 245–255.
- McLoughlin, C., & Hollingworth, R. (2001). *The weakest link: Is web-based learning capable of supporting problem-solving and metacognition?* Paper presented at the Meeting at the Crossroads: Proceedings of ASCILITE Melbourne: University of Melbourne.
- Merrill, M. D. (1994). *Instructional Design Theory* New Jersey: Educational Technology Publications, Englewood, NJ
- Mitchell, T. (1997). *Machine Learning*. New York: McGraw-Hill.
- Moallem, M. (2001). Applying Constructivist and Objectivist Learning Theories in the Design of a Web-Based Course: Implications for Practice. *Educational Technology & Society* 4(3), 113-125.
- Montiliva, J., Sandia, B., & Barrios, J. (2002). Developing Instructional Web Sites - A Software Engineering Approach. *Education and Information Technologies*, 7(3), 201-224.
- Nam, C. S., & Smith-Jackson, T. L. (2007). Web-based learning environment : A theory-based design process for development and evaluation. *Journal of Information Technology Education*, 6, 24-43.
- Nielsen, J. (2001). Jakob Nielsen on e-learning/Elearning post.
- Oboko, R., Wagacha, P. W., Omwenga, E. I., & Libotton, A. (2009a). Comparison of different machine learning algorithms for the initialization of student knowledge level in a learner model-based adaptive e-learning system. *International Journal of computing and ICT Research*, 3(1), 50-56.
- Oboko, R., Wagacha, P. W., Omwenga, E. I., & Libotton, A. (2009c). *Using Adaptive Link Hiding to provide learners with Additional Learning Materials in a Web-based system for teaching Object Oriented Programming*. Paper presented at the VLIR Conference.
- Oboko, R., Wagacha, P. W., Omwenga, E. I., Libotton, A., & Odotte, Z. (2009b). Non-obtrusive determination of learning styles in adaptive web-based learning. *East African Journal of Information Sciences*, 1(2), 1-18.
- Oliver, K. (2003). Effective Uses of Online Course Tools. Retrieved November, 2009, from <http://www.edtech.vt.edu/edtech/id/ocs/>

- Oliver, R., & Herrington, J. (2001). *Teaching and Learning online: a beginner's guide to e-learning and e-teaching in higher education*. Mt Lawley, Western Australia: Centre for Research in Information Technology and Communications, Edith Cowan University.
- Paas, F., Renkl, A., & Sweller, J. (2003). Cognitive Load Theory and Instructional Design: Recent Developments. *Educational Psychologists*, 38(1), 1-4.
- Papanikolaou, K. A., Grigoriadou, M., Kornilakis, H., & Magoulas, G. D. (2003). Personalizing the Interaction in a Web-based Educational Hypermedia System: The case of INSPIRE. . *User Modeling and User-Adapted Interaction*, 13(3), 213-267.
- Piaget, J. (1985). *The equilibration of cognitive structures*. Chicago: University of Chicago Press.
- Pressley, M., & Ghatala, E. S. (1990). Self-regulated learning: monitoring learning from text. *Educational Psychologist*, 25, 19-33.
- Quintana, C., Reiser, B. J., Davis, E. A., Krajcik, J., Fretz, E., Duncan, R. G., et al. (2004). A scaffolding design framework for software to support science inquiry. *Journal of the Learning Sciences*, 13(3), 337-386.
- Quintana, C., Zhang, M., & Krajcik, J. (2005). A Framework for Supporting Metacognitive Aspects of Online Inquiry through Software-Based Scaffolding. *Educational Psychologist*, 40(4), 235-244.
- Reigeluth, C. M. (1999). *Instructional Design Theories and Models: A New Paradigm of Instructional Theory* (Vol. II). Mahwah NJ: Lawrence Erlbaum Associates, Inc.
- Savery, R., & Duffy, T. (1995). Problem Based Learning: An Instructional model and its' constructivist framework. *Educational Technology Research and Development*, 35, 31-38.
- Schmidt, C. P. (2005). *Cognitively Flexible Hypertext in an Object-Oriented Programming Course: Effects of Case-Based Instructional Support on Student Learning*. Kansas State University, Manhattan, Kansas.
- Schmidt, H. G. (1989). The rationale behind problem-based learning. In H. G. Schmidt, M. W. de Vries, M. Lipkin Jr & J. M. Greep (Eds.), *New directions for medical education: Problem-based learning and community-oriented medical education* (pp. 105-111). New York: Springer-Verlag.
- Schraw, G. (1998). Promoting general metacognitive awareness. *Instructional Science*(26), 113-125.
- Schraw, G., & Dennison, R. S. (1994). Assessing metacognitive awareness. *Contemporary Educational Psychology*(19), 460-475.
- Sekaran, U. (2004). *Research Methods For Business, a Skill Building Approach*. Carbondate, Illinois: John Wiley & Sons.
- Sia, K. C., Zhu, S., Chi, Y., Hino, K., & Tseng, B. L. (2007). Capturing User Interests by Both Exploitation and Exploration. *LECTURE NOTES IN COMPUTER SCIENCE*, 4511/2007, 334-339.
- Sison, R., & Shimura, M. (1998). Student Modeling and Machine Learning. *International Journal of Artificial Intelligence in Education*, 9, 128 - 158.
- Someren, M. v., Barnard, Y. F., & Sandberg, J. A. C. (1994). *The think aloud method : a practical guide to modelling cognitive processes* London Academic Press.

- Sorden, S. D. (2005). A cognitive Approach to Instructional Design for Multimedia Learning. *Informing Science Journal*, 8, 263-279.
- Stern, M. K., Beck, J. E., & Woolf, B. K. (1999). *Naive Bayes Classifiers for User Modeling*. Paper presented at the Proceedings of the Conference on User Modeling.
- Sternberg, R. J. (2009). *Cognitive Psychology* (5 ed.). Belmont, California: Wadsworth, Belmont, California.
- Stoyanov, S. (2001). *Mapping in Educational and Training Design.*, University of Twente, Enschede.
- Tobias, S., & Everson, H. T. (2002). *Knowing what you know and what you don't: further research on metacognitive knowledge monitoring*. New York: College Entrance Examination Board. Document Number)
- Valcke, M. (2002). Cognitive load: updating the theory? *Learning and Instruction*, 12(1), 147-154.
- Van Marriënboer, J. J. G., Clark, R. E., & de Crook, M. B. M. (2002). Blueprints for Complex Learning: The 4C/ID-Model. *Educational Technology Research and Development*, 50(2), 39-64.
- Van Marriënboer, J. J. G., Kirschner, P. A., & Kester, L. (2003). Taking the load off a learner's mind: Instructional design for complex learning. *Educational psychologist*, 38(1), 5-13.
- Van Marriënboer, J. J. G., & Sweller, J. (2005). Cognitive Load Theory and Complex Learning: Recent developments and future directions. *Educational Psychology Review*, 17(2), 147-178.
- Ventura, R. P. (2005). Identifying Predictors of Success for an Objects-First CS1. *Computer Science Education* 15(3), 223 – 243.
- Virvou, M., & Tsiriga, V. (2004). A Framework for the Initialization of Student Models in Web-based Intelligent Tutoring Systems. *User Modeling and User-Adapted Interaction* (14), 289 - 316.
- Vrasidas, C. (2000). Constructivism versus objectivism: implications for interaction, course design, and evaluation in distance education. *International Journal of Educational Telecommunications*, 6(4), 339-362.
- Wiedenbeck, S. (2005). *Factors affecting the success of non-majors in learning to program*. Paper presented at the Proceedings of the 2005 international workshop on Computing education research.
- Wilson, D. R., & Martinez, T. R. (1997). Improved Heterogenous Distance Functions. *Journal of Artificial Intelligence Research*(6), 1-34.
- Wixom, B. H., & Todd, P. A. (2005). A theoretical integration of user satisfaction and technology acceptance. *Information Systems Research*, 16(1), 85-102.
- Wolf, C. (2004). iWeaver: Towards 'Learning Style'-based e-Learning in Computer Science Education. Faculty of Education, Language and Community Studies RMIT University, Australia.
- Woods, R. (1994). *Problem-based Learning: How to gain the most from PBL*. Hamilton, Ontario, Canada: Waterdown.

- Yau, J. Y. (2004). *Learning Objects and the Teaching of Java Programming*. University of Warwick, Warwick.
- Zhang, K. (2004). *Effects of peer-controlled of externally structured and moderated online collaboration on group problem solving processes and related individual attitudes in well-structured and ill-structured small group problem solving in a hybrid course*. Pennsylvania State University.
- Zimmerman, B. (2000). *Attaining self-regulation: A social-cognitive perspective*. San Diego, California: Academic Press.

Appendix

Appendix A Summary of comments from one evaluator during the design walkthrough

- The personal information description should be made friendlier
- The friends functions under the goal of friends should be removed because it is only found in C++ and therefore not part of core OOP
- There is a problem with pre-test on “introduction to class features” because some of them are not correct
- Some of the answers to concept evaluation tests are not correct
- There was a misspelling on a goal name - inheritance
- Goal names should be changed to topics
- Try to make goal names more intuitive
- The title of home page had a misspelling for e-learning
- There should be feedback on login attempt e.g. successful login or check the login status
- The number of written programs can be many. Change the presentation of this so that the page can look better
- Examples and exercises do not have a good presentation under user level
- When a concept has no pre-requisites, there should be a message, say, “This concept has no pre-requisites”
- When wrong answers are given in the pre-test, the prompt message includes the words “or you can cancel” but there is nowhere for this.
- The concept evaluation questions should be numbered 1, 2, 3, 4, 5, 6 to give the student a feeling of progress when answering them
- The link called “current goal” in the list of goals is inactive. Suggestion - Can it read the name of the current concept or find another way of showing the current concept
- There is need to provide some feedback after submitting concept evaluation- a Javascript alert
- The time offered for pre-test is long. May be it should be 2 minutes
- The concept evaluation test should be given less time – may be 10 minutes instead of 15 minutes
- Examples and exercise headings are too small

Appendix B Goal concepts and how they are related through prerequisite links

Goal 1: Describe the basic concepts of object oriented programming

NO	Layer	Outcome Concepts	Prerequisite concepts
G1_1	1	Introduction to Object oriented thinking and to key object oriented concepts (Objects, Classes, Message passing, Inheritance, Polymorphism)	none

Goal 2: Demonstrate the use of classes and objects in object oriented programming

NO	Layer	Outcome Concepts	Prerequisite concepts
G2_1	1	Introduction to classes and class members (Introduction to classes and class members, class construction, Scope for Member functions, Arrays of objects)	G1_1
G2_2	2	Data hiding and Encapsulation	G2_1
G2_3	3	Static Data members	G2_1
G2_4	3	The <i>this</i> pointer	G2_1

Appendix C Summary of Knowledge modules per concept

Concept code: G2_2 Concept Name: Data Hiding and Encapsulation

Concept	No	Content Type	Performance Level
G2_2		Overview	
G2_2		Learning objectives	
G2_2	1	Presentation	Remember
G2_2	2	Example	Use
G2_2	3	Exercise	Use
G2_2	4	Exercise	Use
G2_2	5	Find level task	Find
G2_2		Summary	
G2_2		Additional Resources	

Concept g2_2: Data Hiding and Encapsulation

Overview

In this concept, the students learn how to use private data members and public functions.

Objectives

- By the end of the concept the learner should be able to use private data members and public functions in a working program

g2_2a_m1 - Theory Presentation

In terms of C++ classes, data hiding means that you can make data members of a class inaccessible to functions that are not part of the class; that is, you can make them private.

A private member of the Employee class cannot be changed by a function that is not a member of the Employee class.

Private is the default access specifier i.e. unless specified otherwise, all data and functions in a class will be private (i.e. inaccessible).

However, there will be need to access the data or functions in a class. Hence, some functions will need to be public – accessible by both member and non-member functions. You communicate with the private members of a class by sending messages to the public member functions.

Besides private and public, class members can be protected. The specifier *protected* is most often used when member functions of new classes are inherited from existing classes.

g2_2a_m2 - Instance – Example

Alter the Employee class by dividing it into the private section for data members and public section for the member functions.

```
//declaration section
```

```
class Employee
```

```
{
```

```
private:
```

```
int idNumber;
```

```
double salary;
```



```

public:

void assignValues(int id, double sal);

//prototype for assigning values

void displayValues(void);

//prototype for displaying values

};

//Implementation section - - the functions will go here

```

g2_2a_m3 - Exercise

Add private and public sections to the MYCLASS function class.

- Add keyword private: just before the declaration for numerator and denominator
- Add keyword public: after integer declaration and before function declarations.

g2_2a_m4 - Exercise

Define a class that holds years, months, and days someone has been on the job. Its functions assign values to years, months, and days, and then display the values. Remember to add the private and public keywords to the declarations. Add a member function to the class to add 1 month to an object. Once the months exceed 12, years should be increased.

Declare two instances of the EmployeeAge class as ageOfPeter and ageOfMary.

Use C++ syntax.

Find Level Task

In real life, there are things details about you which you do not disclose to the school/department where you are studying and there are others which you do not. Yet, some of these are revealed to your sponsor/employer. Identify some of the things about the student which are not disclosed to the school. Out of these, identify those which are known to your sponsor and not to the school. Also identify those which are not known to either of the two (but of course known to you). Summarize this in a short report.

Summary

- When you create a class, usually you want to make data items private, and to make functions public

Additional Resources

1. Encapsulation – An Introduction:

<http://www.exforsys.com/tutorials/c-plus-plus/encapsulation--an-introduction.html>

2. Abstraction, Encapsulation and Information hiding: <http://www.toa.com/pub/abstraction.txt>

<http://www.toa.com/pub/abstraction.txt>

3. Data Encapsulation:

<http://www.cs.mun.ca/~donald/bsc/node13.html>

Q.No	Question	Options	Answer	Difficulty
1	Which of the following is not a data type?	<ul style="list-style-type: none"> (A) char (B) int (C) float (D) void 	B	Easy
2	Which of the following is not a data type?	<ul style="list-style-type: none"> (A) void (B) int (C) float (D) char 	B	Easy
3	Which of the following is not a data type?	<ul style="list-style-type: none"> (A) void (B) int (C) float (D) char 	B	Easy
4	Which of the following is not a data type?	<ul style="list-style-type: none"> (A) void (B) int (C) float (D) char 	B	Easy
5	Which of the following is not a data type?	<ul style="list-style-type: none"> (A) void (B) int (C) float (D) char 	B	Easy
6	Which of the following is not a data type?	<ul style="list-style-type: none"> (A) void (B) int (C) float (D) char 	B	Easy
7	Which of the following is not a data type?	<ul style="list-style-type: none"> (A) void (B) int (C) float (D) char 	B	Easy
8	Which of the following is not a data type?	<ul style="list-style-type: none"> (A) void (B) int (C) float (D) char 	B	Easy
9	Which of the following is not a data type?	<ul style="list-style-type: none"> (A) void (B) int (C) float (D) char 	B	Easy
10	Which of the following is not a data type?	<ul style="list-style-type: none"> (A) void (B) int (C) float (D) char 	B	Easy

Appendix E Sample pretest for the concept “Data hiding and Encapsulation”

G2_2			Data hiding and Encapsulation		
	1	Novice	Two access specifiers in C++ are _____ a) public and private b) int and double c) formal and informal d) void and free	A	4
	2	Novice	Which of the following is an access specifier? a) particular b) shielded c) protected d) safe	C	4
	3	Intermediate	The default access specifier is _____ a) Shielded b) Private c) Protected d) public	B	5
	4	Intermediate	Which is true? a) All class functions are private. b) Most class functions are private. c) Most class functions are public. d) Most classes contain no functions.	C	5
	5	Advanced	The access specifier <u>protected</u> for data members in a class specification means a) The data members are only directly accessible to the member functions of the class b) The data members are accessible from the implementation of the class directly c) The data members can be inherited and accessed in new classes that are inherited from the current class. d) None of the above	C	6
	6	Advanced	//declaration section	B	6

		<pre>class Employee { private: int idNumber; public: double salary; void assignValues(int id, double sal); //prototype for assigning values void displayValues(void); //prototype for displaying values };</pre> <p>What is the likely problem of defining the attribute salary like this:B</p> <ul style="list-style-type: none">a) It is accessible even to only to the member functions of the classb) It defeats the whole idea of encapsulation since it is accessible even to non-member functionsc) It makes the variable to be unavailable to the other classes inheriting from the current classd) None of the above		
--	--	---	--	--

Appendix F Those who participated in the validation of instruments and evaluation of the system for learning

Name	Affiliation	Role
Prof. Arno Libotton	VUB	Supervisor; pedagogy; validation of instruments
Dr. Peter Wagacha	UoN	Supervisor; adaptive interface; validation of instruments
Dr. Elijah Omwenga	UoN	Supervisor; pedagogy; subject matter validation; validation of instruments; also expert in usability design due to experience teaching Internet based technologies
Prof. Bernard Manderick	VUB	Adaptive interface (ML algorithm) and design of content (feedback)
Ms Muthoni Masinde	UoN	Course lecturer; subject matter validation; instruments; formative evaluation with her class; also expert in usability design due to experience teaching Internet based technologies
Mr. Christopher Chepken	UoN	Course lecturer; subject matter validation; instruments; actual evaluation of learners after problem solving
Ms Christine Ronge	UoN	Course lecturer; subject matter validation;
Mr. Evans Miriti	UoN	Course lecturer; subject matter validation;
Dr. Slavi Stoyanov	Open University of Netherlands	Adaptive interface; features the system should adapt to; pedagogy
Mr. Wainaina	JKUAT	Course lecturer; subject matter validation; validation of instruments; validation of evaluation scored for objective instruction part; taught pre-requisite, hence important source of what to expect from the class and also the pre-requisite course marks
Mr. Njuguna	JKUAT	Course lecturer; subject matter validation; evaluation using focus group, field trial and one-on-one. His class participated in formative evaluation

Appendix G Sample Ill-structured programming problem for object oriented programming

Problem 1

The new Minister for Nairobi Metropolitan would like the Bus Terminals in Nairobi computerized in order to make the town more orderly. Below is a description of how the terminals work:

Bus Terminals in Nairobi are situated on various streets/roads, for example, GPO Bus Terminal on Kenyatta Avenue. The Bus Terminal has a maximum number of spaces (that is it can only hold a number buses at time) where buses can queue up waiting for passengers to board them. Passengers also have spaces to queue at the stand. There are many queues depending on the Routes; for example a queue for Route 111. The Passengers' queues are technically unlimited in size since people generally will queue all the way down the street if required. For security reasons, Passengers are identified by names as well as ID numbers (if they are adults). Buses are identified by a registration numbers and the Company that owns them, e.g. Easy Coach. Each bus has a maximum number of seats and a value representing the number of Passengers already in the Bus. The system needs to be able to model the following:-

1. A Bus can join the Bus Terminal queue, but only if there are spaces in the queue.
2. A Passenger can join the passenger queue.
3. The system should be able to report the number of Passengers and buses in the queues at any time.
4. Any passenger in the queue can walk away from the queue without getting into a bus.
5. A bus at the front of the queue can drive away from the queue without picking up Passengers.
6. A Passenger or Passengers can get into a bus, before it drives away. Only the passenger at the front of the queue can enter the first bus in the queue. The bus can then drive away.
7. However, the next Passenger and so on may enter the bus until the bus becomes full or Passenger s stop getting in. The bus will then drive away

Appendix G Sample Ill-structured programming problem for object oriented programming

Problem 1

The new Minister for Nairobi Metropolitan would like the Bus Terminals in Nairobi computerized in order to make the town more orderly. Below is a description of how the terminals work:

Bus Terminals in Nairobi are situated on various streets/roads, for example, GPO Bus Terminal on Kenyatta Avenue. The Bus Terminal has a maximum number of spaces (that is it can only hold a number buses at time) where buses can queue up waiting for passengers to board them. Passengers also have spaces to queue at the stand. There are many queues depending on the Routes; for example a queue for Route 111. The Passengers' queues are technically unlimited in size since people generally will queue all the way down the street if required. For security reasons, Passengers are identified by names as well as ID numbers (if they are adults). Buses are identified by a registration numbers and the Company that owns them, e.g. Easy Coach. Each bus has a maximum number of seats and a value representing the number of Passengers already in the Bus. The system needs to be able to model the following:-

1. A Bus can join the Bus Terminal queue, but only if there are spaces in the queue.
2. A Passenger can join the passenger queue.
3. The system should be able to report the number of Passengers and buses in the queues at any time.
4. Any passenger in the queue can walk away from the queue without getting into a bus.
5. A bus at the front of the queue can drive away from the queue without picking up Passengers.
6. A Passenger or Passengers can get into a bus, before it drives away. Only the passenger at the front of the queue can enter the first bus in the queue. The bus can then drive away.
7. However, the next Passenger and so on may enter the bus until the bus becomes full or Passenger s stop getting in. The bus will then drive away

Appendix H Class Design Test
Class Design Test

Student's Number _____ Date _____

Instructions:

Answer all Questions.

Section (A) Circle the letter next to the most correct answer (14 marks)

1. The ___ variables are sometimes called class variables, class fields, or class-wise fields
 - A. Public
 - B. Private
 - C. Constant
 - D. Static
2. You must use both class name and the ___ operator when you implement a class function, because together, they tie the function to the class and allow every instantiated class object to use the function name
 - A. Address
 - B. Dot
 - C. Scope resolution
 - D. Pointer-to-member
3. A ___ class member exists, even when you have not instantiated any objects of the class
 - A. Public
 - B. Private
 - C. Static
 - D. Const
4. When you use an object, the ___ operator, and a non-static function of the object's class, you actually pass the specific object's address to the function as an unseen argument.
 - A. Scope Resolution
 - B. Conditional
 - C. Dot
 - D. Address
5. ___ is an example of a correct use of the this pointer.
 - A. *this.employeeldNum
 - B. (*this).employeeldNum
 - C. *(this.employeeldNum)
 - D. *(this).employeeldNum
6. ___ functions return information about an object's state, or display some or all of an object's attributes.
 - A. Inspector
 - B. Manager
 - C. Mutator
 - D. Auxiliary
7. The ___ directive means that you have reached the end of the block you are defining

- A. #define
- B. #endif
- C. #ifdef
- D. #ifndef

Section (B) Short Answer (24 marks)

1. Consider the following class:

```
class Student
{
    private:
    int idNum;
    string lastName;
    double gradePointAverage;
};
```

Add a public function void displayStudentData(). Next, write an implementation for the function.

- 2. What are the four special member functions that every C++ class object automatically has?
- 3. Write a class declaration that has three private fields and three set functions for the private data. Provide an implementation for one of the functions.
- 4. Answer a), b) and c)
 - a) How do constructor functions differ from other member functions?
 - b) Why does a constructor need to have the same name as its class?
 - c) Implement a default constructor for the following class:

```
class Employee
{
    private:
    int idNum;
```

```

double hourlyRate;

public:

Employee();

void setValues(const int, const double);

void displayValues();

};

```

5. Answer both questions a) and b)

a) Can you modify the "this" pointer? Explain.

b) In what situation would you want to use the "this" pointer?

6. Modify the following class definition so that it uses the #ifndef, #define and #endif directives:

```

class Salesperson
{
private:
    int idNum;
    string name;
public:
    Salesperson(int, string);
    void displayPerson();
};

Salesperson::Salesperson(int num, string name)
{
    idNum = num;
    this->name = name;
}

void Salesperson::displayPerson()
{

```

```
cout<<"Salesperson #" <<idNum<<" " <<name<<endl;
```

```
}
```

Section (C) Long Question (20 marks)

- a) In terms of state and behaviour, use an example to explain the statement “a superclass is a subset of any of its subclasses” 4 Marks
- b) A class is constructed in two sections – declaration and implementation. What is the role of each of these two?: 2 Marks
- c) Give any two rules for destructor names. 2 Marks
- d) Assume that you have been hired a consultant for Safaricom. The management would like you to demonstrate how a C++ program would capture and display the following Personnel information:
- Names of employees
 - Personnel numbers
 - Salary

The management would like an object-oriented program that manipulates the above information.

- e) Create the class 4 Marks
- f) Write any one relevant member function. The member function could include one to get personnel data and another to display personnel data 3 Marks;
- g) Suppose there was a minimum wage of 20000 which you would like to store as a static data member. Change the Employee class to reflect this. 2 Marks
- h) Safaricom has 100 Employees. Declare an array of Employees. Write a statement which you could place in the main function to get personnel data for a specific employee in the array. 3 Marks.

Scaffolding learners for Task Schedule Preparation, Maintenance and Use

About the task schedule:

1. The task schedule is a popular learning support technique among many students
2. A task schedule contains information which students can use to measure and evaluate their current progress against all the tasks planned to be done
3. A task schedule is made by:-
 - i. First understanding the main task to be done i.e. problem to be solved. If there is not enough information in the problem statement and what the student already knows is not adequate to enable him to understand the problem, then further information may be sought by reading or consulting. E.g. go to the school library and seek clarifications on some of the terms and processes.
 - ii. Dividing the problem into activities or tasks which are to be done so as to solve the main problem. For instance, in object oriented programming, this could be in the line of problem definition (what is the precise definition of the problem?), problem analysis to identify classes, relationships, ..., solution design (how to maneuver the different components available so as to solve the problem e.g. classes, relationships (inheritance), attribute design, method design, header and implementation file design, constructor and destructor design,...), integration design,....
 - iii. Determining when each task is to start, when each task is to end and the estimated duration say, in days
 - iv. Establishing the order to the tasks by identifying the predecessor task to each task
 - v. Recording in an organized way the tasks and their details so as to produce a learning schedule. This could be done by using screens provided for keying the details into a database which can be visualized later using a chart
 - vi. Updating the learning schedule whenever there is progress e.g. a higher percentage of task completion achieved or full task completion, so that the schedule shows the students actual current progress in the programming task
 - vii. From time to time while solving the programming task, comparing the student's actual current progress against all the tasks planned to be done, and the time already spent against the total time set aside for completing the main task
 - viii. If the progress is not according to plan, finding out why and what to do to improve

Appendix J Question Prompts

Something to Think About

As you work through the programming problem, please read and think about the following questions.

<p>How do I define the problem?</p> <ol style="list-style-type: none"> 1. What are the parts of the problem? 2. What are the technical components? 3. What information do you need for this system? How will the system be used, by whom, and for what? <ul style="list-style-type: none"> - Who would be the users? - What information do you expect the users need? - How would a user ideally interact with the proposed system? 	<p>Related object oriented programming questions</p> <ol style="list-style-type: none"> 1. What is the problem to be solved? 2. What are the sub-tasks that make the main task? 3. What are the classes in the described problem? 4. What are the attributes for each of the classes? 5. What are the methods that can be applied on the class objects?
<p>What solutions do I need to generate?</p> <ol style="list-style-type: none"> 4. What should the system do? 5. How should the different technical components of the proposed system interrelate? 6. What are the risks? <ul style="list-style-type: none"> - How do I generate the solutions? 	<ol style="list-style-type: none"> 6. Are there any relationships between the classes? Is there inheritance? Are there any "is a" or "is like" relationships? <ul style="list-style-type: none"> - Are there classes which have some common attributes for which you can introduce a parent class so as to avoid implementing these methods/responsibilities twice? 7. Will one need to use any static variables? 8. Will special initialization tasks be necessary for a class (a constructor will be needed) 9. Will any special clean up tasks be carried out when a class object goes out of scope (especially when dealing with pointers – then a destructor might be needed) 10. Will class data members be assigned values after their constructions? (then methods with assignment statements will be required) 11. Will class values need to be displayed? 12. Will class data members need operations (e.g. arithmetic, sorting, capitalization, etc) to be performed on them? 13. Will you need different forms of the same method, especially constructor functions with different parameters at different times?

	<p>14. Will I need to store variables whose memory sizes will keep varying? use pointers?</p> <p>15. How do I link the different components to form an integrated solution?</p>
<p>What are my reasons/argument for my proposed solution?</p> <p>7. How would I justify this specific system design?</p> <p>8. Do I have evidence to support my solution (that is, the specific system I have proposed)?</p> <ul style="list-style-type: none"> - What is my chain of reasoning to support my solution? - What are my reasons and arguments for the proposed solutions? 	<p>16. Why did I choose to represent the attributes, relationships, methods, etc as I did?</p> <p>17. How do I support my various decisions (over the alternatives) during system design? Static Vs non-static, inheritance Vs no inheritance, default constructors/destructors Vs user-defined, constructor initialization lists Vs assignment, user of pointers Vs fixed length attributes....</p>
<p>Am I on the right track?</p> <p>9. Have I discussed both the technical components and the issues with use, for example, usability and effectiveness?</p> <p>10. Are there alternative solutions?</p> <ul style="list-style-type: none"> - What are they? - How are they compared with my proposed system? - What argument can I make or what evidence do I have to convince that my solution is the most viable? - Am I on the right track? 	<p>18. Is the solution effective?</p> <ul style="list-style-type: none"> - Have I included the required components to solve the sub-tasks and the main task? - Are they integrating well? - Is the system usable? <p>19. Were there any alternatives to the inclusions and designs in the system?</p> <ul style="list-style-type: none"> - What are they? - How do they compare to this system - How do I prove that the solution is practicable/ workable? - Is my solution the best?

Appendix K Brief Description of the Problem Solving Process

The Problem Solving Process

A problem

A problem occurs when an individual is following some goal or target, but is uncertain about the action that he should undertake to reach the goal-achieve the target.

An ill-structured problem

An ill-structured problem is a problem which provides the problem solver with little or no information on the best way of developing a solution to the problem. Because there is no clear procedure on how to solve the problem, the problem solver should improvise.

Problem Solving

Problem solving is the process of defining this problem clearly, finding the best solution to the problem among the possible alternative approaches, implementing the solution and evaluating the solution to see if it meets the goal (is it effective) and is the best among the possible solutions.

Problem Solving Processes

These are the various stages the problem solver goes through during problem solving. The processes according to this study are four: problem representation, problem solution, justification, and monitoring and evaluation.

a) Problem Representation

This is an extremely important process for determining the solutions of ill-structured problems.

It involves data gathering and the definition of the problem.

- i. **Data gathering** – The problem solver starts to collect different **types of information** that is provided such as **who, what, where, when, why, which** and **how** of the situation. The problem solver examines the **concepts** and **the relationships among the concepts** in the problem situation. Any information **provided** in the problem statement is harnessed.
- ii. **Needed information not provided** – The problem solver also identifies information which **might be needed in the process** of solving the problem but which has **not been provided**. This can **be sought** or it can **be handled as assumptions**.

- iii. **Problem definition** – There may be **several ways of looking at the problem**. The problem solver generates **alternative problem definitions**. After this, a working definition of the problem is selected, and a justification (reason(s) for seeing the problem as such) is provided for the selection.
- iv. **Setting sub-goals** – The problem solver sets **specific targets or objectives to achieved**, which are such that when achieved and added up, they will **constitute the solution** to the problem being solved.

Relationship to object oriented programming – It parallels with the **requirements and analysis stages** of a programming task. What is the task? Why do you think so? What are the methods, classes, relationships, attributes ... Can you use inheritance, polymorphism, ... What data is to be stored, what information will be produced by the system,...

b) The problem Solution

For ill-structured problems, there is **no single solution**. The best solution has to be **selected among the alternatives** and it has to be **justified** with relevant and sufficient evidence and it has to be backed up with **supportive facts or suppositions/assumptions**.

The process involves solution generation and solution selection.

- i. **Solution generation** – The problem solver **selects or develops as many ideas/approaches as possible on how to solve the problem**. Look for possible ways to solve the problem, possibly coming in the form of different ways in which to select and combine the different components which might form the solution to the problem.
- ii. **Solution Selection** – Select the most promising approach/idea **out of those generated**. Some criteria may be used for the selection of the best approach/idea. Look at the possible consequences of the possible solutions.

The **selected solution** has to address the **sub-goals or specific sub-tasks** of the problem and **how they do this has to be clear**. The selection has to be **justified with a well-argued, convincing argument by the problem solver**.

Relationship to object oriented programming – It parallels with the **design stage**. A number of **design decisions are made and they have to be justified**. What is the design of the classes? What are the relationships? Which C++ constructs are to be used? What is the file organization? Do you have any reasons for the choices?

c) Making justifications for the proposed solutions

Because ill-structured problems are complex, ill-defined, with **multiple possible solutions**, **justification skills**, monitoring and evaluation are critical processes.

The problem solver must **select a good solution** from among the many possible solutions and therefore he must provide the **most viable, the most defensible and the most convincing argument to support the selected solution** (defend it against alternative solutions), and then go on to **produce evidence to support the positions taken in the arguments.**

Relationship to object oriented programming – The justification process parallels the project justification provided during requirements determination and problem definition and the justification for the various choices during design.

d) Monitoring and evaluation

- i. **Monitoring** involves the problem solver keeping an eye on the processes of **implementing the solutions and the progress from step to step, with any remedial decisions and actions taken or some steps which are repeated being identified.** Monitoring involves **identifying the task one is currently engaging in** (selection of the problem definition and the goals for problem solving), **evaluating the progress of that work, predicting what the outcome will be and taking necessary action to achieve the desired outcome** (e.g. a specific sub-goal).
- ii. **Evaluation** of the solution is done by examining the implemented solution to see **if it is effective i.e. if it solves the problem (and meets subgoals), providing the benefits of the solution, supported with relevant evidence.** The solution selected has to be justified in terms of whether **it can solve the problem and it is the best, with a convincing argument** and if there are **areas in which it can be improved and how they can be improved (normally comes as conclusion of project report).** The selected solutions should be **compared to alternative solutions** (which should be identified), giving its' **advantages and disadvantages.**

Relationship to object oriented programming – It parallels with problem definition and requirements/task objectives identification, and **implementation** in terms of **programming and testing the developed solution.** **Alternative solutions** arise when **different components** e.g. structures or C++ components or **different arrangements** of components had been used

Appendix L Sample Coding Scheme for the Element Declarative Knowledge under Knowledge of Cognition

A. Knowledge of Cognition -			
What the learners know about their cognition			
Construct	Operational Definition	Code	Pointers
Declarative knowledge- Knowledge about oneself i.e. one's skills, intellectual resources and abilities as a learner	1. Indication of awareness of one's capacities in terms of skills, abilities and intellectual resources	KC-DK-CAP <i>Awareness of own capacities</i>	Things about self awareness <ul style="list-style-type: none"> • Things one knows and which will be useful for the programming problem • Things he will easily do while solving the programming problem e.g. being good at organizing information, knowing which information is important to learn, knowing what the teacher expected one to learn, being good at remembering information, having control over how well one learns, being a good judge of how well one understands something, 'I learn more when interested in the topic'
	2. Indication of awareness of one's limitations in terms of skills, abilities and intellectual resources	KC-DK-LIM <i>Awareness of own limitations</i>	Things about self awareness <ul style="list-style-type: none"> • Things one does not know/understand but will be required for the assignment • Things one will find difficult to do while solving the programming problem
	3. Indication of awareness of difficulties as they arise during problem solving so that remedial action may be taken such as sudden change in direction of thinking	KC-DK-DIF <i>Awareness of difficulties as they arise</i>	<ul style="list-style-type: none"> • Sudden change in direction of thinking – all of a sudden talking about another thing/idea without completing discussion on a previous one • Stopping mid-sentence while talking – sudden halt with long silence • Words suggesting doubt on suitability of what is being done-questioning self like "I am not sure ..."

Appendix M Post Survey (After written exams in study 1)

Student's Number _____ Date _____

The statements below are designed to identify your online learning experiences while using the learning system for the Object Oriented Programming course. The questions have 7 possible responses. The responses range from 1 (Strongly Disagree) through 4 (Neither disagree nor agree) to 7 (Strongly Agree). If you have no opinion, choose response 4. Please read each statement and mark the one response that most clearly represents your degree of agreement or disagreement with your statement. Please respond to all the statements.

	Strongly disagree				Strongly agree		
	1	2	3	4	5	6	7
1. The course's organization into goals, concepts, pre-requisite concepts and the knowledge modules facilitated your study	1	2	3	4	5	6	7
2. The structure of the educational material in Different levels of performance i.e. Remember, Use and Find facilitated your study	1	2	3	4	5	6	7
3. The presentation of the content of each concept using different modules (i.e. theory, exercises, examples, activities, etc) supported you to understand the concept	1	2	3	4	5	6	7
4. The presentation of individual sub-sections of a concept's reading material (i.e. theory, sub-concepts, additional reading materials, etc) through links which are seen when you expand a concept by clicking on + next to the concept name facilitated your study	1	2	3	4	5	6	7
5. The provision of feedback (i.e. test scores and concepts which should be studied further after a test) facilitated your study	1	2	3	4	5	6	7
6. The proposed level of difficulty for a concept at which you start studying the concept (Novice, Intermediate, Advanced) supported you to understand the concept	1	2	3	4	5	6	7
7. The traffic light metaphor (Green, Orange, Red colors) on concept names showing the extent to which you are ready to study different concepts facilitated your learning	1	2	3	4	5	6	7
8. Do you think the way the links of additional materials at the bottom of your study page are presented (if you have been using additional materials) or are not presented (if you have not been using additional materials) was agreeable to you?	1	2	3	4	5	6	7
9. At the same time, you had the option to select the material that you prefer to study next. Is this option useful?	1	2	3	4	5	6	7
10. The different features proposed by the system (concept to study –through different colors, level of difficulty to follow, links to additional materials)	1	2	3	4	5	6	7

were disruptive to your study

11. The information I received from the system suited my needs very well

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

12. All things considered, I am very satisfied with the system

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

13. Overall, my interaction with the system is very satisfying

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

14. What did you like about the learning system?

15. What did you dislike about the learning system?

16. Did you try to access the course content through your mobile phone? If yes, what did you like/dislike about it?

17. Could you have preferred to receive on your mobile phone feedback from the system such as test scores and concepts which should be studied further after a test? Explain your answer

18. What did you like about the user interface of the system?

19. What did you dislike about the user interface of the system?

20. Did the learning system help you in understanding the subject matter? If so, tell us how it was helped. If not, tell us how it hindered your understanding.

21. How do you think the environment could be improved in the future?

Appendix N Metacognitive Awareness Inventory

The following questions ask about the way you learn and solve a problem, such as the programming task for this course. Please take a moment to respond to these questions. **Remember there is no right or wrong answers, just answer as accurately as possible.** Use the scale below to answer the questions. If you think the statement is very true of you, circle 7; if it is not at all true of you, circle 1. If the statement is more or less true of you, find and circle the number between 1 and 7 that best describes you.

	1	2	3	4	5	6	7
	not at all true of me			very true of me			
1. I ask myself periodically if I am meeting my goals.	1	2	3	4	5	6	7
2. I consider several alternatives to a problem before I answer	1	2	3	4	5	6	7
3. I try to use strategies that have worked in the past.	1	2	3	4	5	6	7
4. I pace myself while learning in order to have enough time.	1	2	3	4	5	6	7
5. I understand my intellectual strengths and weaknesses	1	2	3	4	5	6	7
6. I think about what I really need to learn before I begin a task	1	2	3	4	5	6	7
7. I know how well I did once I finish a test.	1	2	3	4	5	6	7
8. I set specific goals before I begin a task.	1	2	3	4	5	6	7
9. I slow down when I encounter important information.	1	2	3	4	5	6	7
10. I know what kind of information is most important when I learn.	1	2	3	4	5	6	7
11. I ask myself if I have considered all options when solving a problem.	1	2	3	4	5	6	7
12. I am good at organizing information	1	2	3	4	5	6	7
13. I consciously focus my attention on important information	1	2	3	4	5	6	7
14. I have a specific purpose for each strategy I use.	1	2	3	4	5	6	7
15. I learn best when I know something about the topic	1	2	3	4	5	6	7
16. I know what the teacher expects me to learn	1	2	3	4	5	6	7
17. I am good at remembering information.	1	2	3	4	5	6	7
18. I use different learning strategies depending on the situation	1	2	3	4	5	6	7
19. I ask myself if there was an easier way to do things after I finish a task.	1	2	3	4	5	6	7
20. I have control over how well I learn.	1	2	3	4	5	6	7
21. I periodically review to help me understand important relationships.	1	2	3	4	5	6	7
22. I ask myself questions about the material before I begin.	1	2	3	4	5	6	7
23. I think of several ways to solve a problem and choose the best one.	1	2	3	4	5	6	7
24. I summarize what I've learned after I finish.	1	2	3	4	5	6	7
25. I ask others for help when I don't understand something.	1	2	3	4	5	6	7
26. I can motivate myself to learn when I need to.	1	2	3	4	5	6	7
27. I am aware of what strategies I use when I study.	1	2	3	4	5	6	7
28. I find myself analyzing the usefulness of strategies while I study.	1	2	3	4	5	6	7
29. I use my intellectual strengths to compensate for my weaknesses.	1	2	3	4	5	6	7
30. I focus on the meaning and significance of new information.	1	2	3	4	5	6	7

31. I create my own examples to make information more meaningful.	1	2	3	4	5	6	7
32. I am a good judge of how well I understand something.	1	2	3	4	5	6	7
33. I find myself using helpful learning strategies automatically.	1	2	3	4	5	6	7
34. I find myself pausing regularly to check my comprehension.	1	2	3	4	5	6	7
35. I know when each strategy I use will be most effective.	1	2	3	4	5	6	7
36. I ask myself how well I accomplished my goals once I'm finished.	1	2	3	4	5	6	7
37. I draw pictures or diagrams to help me understand while learning.	1	2	3	4	5	6	7
38. I ask myself if I have considered all options after I solve a problem.	1	2	3	4	5	6	7
39. I try to translate new information into my own words	1	2	3	4	5	6	7
40. I change strategies when I fail to understand	1	2	3	4	5	6	7
41. I use the organizational structure of the text to help me learn.	1	2	3	4	5	6	7
42. I read instructions carefully before I begin a task.	1	2	3	4	5	6	7
43. I ask myself if what I'm learning is related to what I already know.	1	2	3	4	5	6	7
44. I re-evaluate my assumptions when I get confused.	1	2	3	4	5	6	7
45. I organize my time to best accomplish my goals.	1	2	3	4	5	6	7
46. I learn more when I am interested in the topic.	1	2	3	4	5	6	7
47. I try to break studying down into smaller steps.	1	2	3	4	5	6	7
48. I focus on overall meaning rather than specifics.	1	2	3	4	5	6	7
49. I ask myself questions about how well I am doing while I am learning something new.	1	2	3	4	5	6	7
50. I ask myself if I learned as much as I could have once I finish a task.	1	2	3	4	5	6	7
51. I stop and go back over new information that is not clear.	1	2	3	4	5	6	7
52. I stop and reread when I get confused.	1	2	3	4	5	6	7