QUALITY OF SERVICE MANAGEMENT FOR MULTIMEDIA COMMUNICATIONS:

INVESTIGATIONS ON USABILITY OF INTERFACES FOR DESKTOP AND PORTABLE SYSTEMS

Mladen Georgievski

SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

SCHOOL OF COMPUTER SCIENCE & MATHEMATICS,

FACULTY OF HEALTH, ENGINEERING AND SCIENCE

VICTORIA UNIVERSITY

PO BOX 14428
MELBOURNE CITY MC
VICTORIA, AUSTRALIA, 8001

2007

Contents – Brief

Contents – Detailed

List of Figures

List of Tables

List of Acronyms – General

3GPP	Third Generation Partnership Program
ABR	Application Bit Rate
ATM	Asynchronous Transfer Mode
BER	Bit Error Rate
bpp	Bits per pixel
bps	Bits per second
CBR	Constant Bit Rate
CTTE	Concur Task Tree Environment
DiffServ	Differentiated Services
DLE	Distributed Learning Environments
DRoPS	Dynamically Reconfigurable Protocol Stacks
EEG	Electroencephalograms
EoD	Education on Demand
fps	Frames per second
FVoD	Full Video on Demand
GUI	Graphical User Interface
HCI	Human Computer Interaction
HQML	Hierarchical QoS Modelling Language
IETF	Internet Engineering Task Force Group
Intserv	Integrated Services
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
ISO	International Standards Organisation
ISP	Internet Service Provider
MBR	Mean Bit Rate
MPLS	Multi-Protocol Label Switching
NPPs	Network Performance Parameters
NVoD	Near Video on Demand
OSI	Open Systems Interconnection Reference Model
PBR	Peak Bit Rate
PDA	Personal Digital Assistant
PER	Packet Error Rate
QM	Quality of Service Manager
QoP	Quality of Presentation
QoS	Quality of Service
QoS-A	Quality of Service Architecture
RSVP	Resource Reservation Protocol
RTP	Real-Time Transport Protocol
SAS	Synchronisation Accuracy Specification
SBM	Subnet Bandwidth Management
ST-II	Stream Protocol

TCP/IP	Transport Control Protocol / Internet Protocol
TINA	Telecommunication Information Networking Architecture
UMTS	Universal Mobile Telecommunications System
VBR	Variable Bit Rate
VoD	Video on Demand
XRM	Extended Integrated Reference Model

List of Acronyms – Specific to this Thesis

ACM	Application Classification Model
APL	Application Perspective Layer
APM	Application Profile Model
CCS	Cartesian Coordinate System
PUI	Physical User Interface
QCTT	Quality Cost Temporal Triangle
QPA	Quality of Service Parameter Architecture
TFS	Triangular Fractal System
TPL	Transmission Perspective Layer
TRAQS	Three Layer QoS Model
UPL	User Perspective Layer

Declaration of Originality

I, Mladen Georgievski, declare that the PhD thesis entitled “Quality of Service Management for Multimedia Communications: Investigations on Usability of Interfaces for Desktop and Portable Systems” is no more than 100,000 words in length, exclusive of tables, figures, appendices, references and footnotes. This thesis contains no material that has been submitted previously, in whole or in part, for the award of any other academic degree or diploma. Except where otherwise indicated, this thesis is my own work.

Signature: ______________________ Date: _____________

Mladen Georgievski

Acknowledgements

First and foremost, I would like to thank my supervisor, Associate Professor Nalin Sharda, for his continual guidance, timely advice, encouragement and support throughout the course of this thesis. He has been a great mentor, and I will always remember our lengthy philosophical conversations.

Furthermore, I owe my thanks to Mr. Jayden Kimber for applying his elite programming skills in developing the software applications used in this research and for his assistance in performing the usability studies. I would also like to thank Mrs. Roopa Jakkilinki for facilitating the usability tests.

I wish to thank Dr. Florica Cirstea, Mr. Gabriele Sorrentino, and Mr. Leon Gor for their continual technical and theoretical advice, and the feedback they have given me throughout this research. A special thanks goes to Ms. Lilly Petrovska for applying her exceptional English skills in copy editing this thesis.

I would also like to express my gratitude towards the technical support staff within the School of Computer Science and Mathematics: Mr. Mark Mojic, Mr. Cameron Giles, and Mr. Rajendran Ponnusamy, for their promptly and excellent technical support throughout this research. Furthermore, I would like to thank the Head of School, and the administrative staff for their support, and all my colleagues and friends for the encouragement and motivation they have given me.

I would also like to thank all test participants for their patience and valuable feedback they have given within the usability tests.

Last but not least, I would like to specially thank my Mum and Dad for their constant, unconditional support, and encouragement they have provided me throughout my entire life. Their inspiration and enthusiasm have always made me achieve my goals and ambitions I set in life.

Abstract

Multimedia communication is now possible via the use of high bandwidth channels, and communication protocols that support Quality of Service (QoS) requests. However, these technologies often fall short of user expectations by not providing effective user interfaces that can facilitate negotiations between the user and the system.

In this thesis, a set of conceptual models are developed to create a user-centred QoS management system for networked multimedia applications. Various interfaces are designed and developed for 'specifying' and 'negotiating' QoS prior to initiating a multimedia session, and 're-negotiating' QoS in real-time during the multimedia session.

The interfaces developed in this research provide the means for managing QoS for fixed as well as mobile multimedia communications. Usability investigations were conducted to evaluate the efficacy of these interfaces for specifying, negotiating and re-negotiating QoS on Desktop Systems and on Portable Devices – such as, Personal Digital Assistants (PDAs) and Mobile Phones. Conducting the usability tests for these interfaces required re-engineering of traditional usability testing processes – to improve the efficiency of performing each test and collecting data.

Ten separate usability studies were performed using Beginner, Intermediate, and Advanced users to investigate novel Human Computer Interaction (HCI) methods for: computer control and system feedback; QoS specification and negotiation on Desktop Computers; and QoS re-negotiation on Desktop Systems and Portable Devices.

Users agreed that there is a need for such interfaces that enable them to request, negotiate and re-negotiate QoS. Usability studies for these interfaces show good usability and learnability for technical as well as non-technical users. The innovative interaction methods developed in this research project, by far, enhanced the user's experience in managing QoS, however, suggestions for further improvements were also given by the participants.

Index Terms: Quality of Service (QoS), Networked Multimedia Systems, QoS Management, QoS Negotiation, Multimedia Communications, Usability Testing.

Publications Reporting Outcomes of this Research

Peer Reviewed International Journal Publication

- M. Georgievski and N. Sharda, "Re-Engineering the Usability Testing Process for Live Multimedia Systems," *Journal of Enterprise Information Management (EIMJ)*, vol. 19, pp. 223-233, 2006.

International Conference Publications, Full Paper Peer Reviewed

- N. Sharda, M. Georgievski, and K. Lim, "Mapping the TRAQS Quality of Service Model to the m-Apps Middleware," presented at M2USIC 2005: MMU International Symposium on Information and Communications Technologies, PJ Hilton Hotel, Petaling Jaya, Malaysia, 2005.

- M. Georgievski and N. Sharda, "Implementation and Usability of User Interfaces for Quality of Service Management," presented at Tencon'05, Annual Technical Conference of IEEE Region 10, Melbourne, Australia, 2005.

- M. Georgievski and N. Sharda, "Enhancing User Experience for Networked Multimedia Systems," presented at The 4th International Conference on Information Systems Technology and its Applications, ISTA2005, Massey University, Palmerston North, New Zealand, 2005.

- M. Georgievski and N. Sharda, "Re-Engineering the Usability Testing Process for Live Multimedia Systems," presented at The 2004 International Research Conference on Innovations in Information Technology, IIT2004, Dubai, UAE, 2004.

- M. Georgievski and N. Sharda, "Task Modelling for a Holistic Quality of Service Model – TRAQS," presented at IEEE India Council Annual Convention and Exhibition 2003, ACE2003, Hotel Le Meridien, Pune, India, 2003.

- M. Georgievski and N. Sharda, "Usability Testing for Real-Time QoS Management," presented at The 24th IEEE International Real-Time Systems Symposium, RTSS2003, Cancun, Mexico, 2003.

- M. Georgievski and N. Sharda, "A Taxonomy of QoS Parameters and Applications for Multimedia Communications," presented at International Conference on Internet and Multimedia Systems and Applications, IMSA2003 Kaua'i, Hawaii, USA, 2003.

- N. Sharda and M. Georgievski, "A Holistic Quality of Service Model for Multimedia Communications," presented at International Conference on Internet and Multimedia Systems and Applications, IMSA2002 Kaua'i, Hawaii, USA, 2002.

Chapter 1
Introduction

1.1 Background

In recent years, advancements in information and communication technology have led to the development of sophisticated multimedia applications which are now being used in almost every aspect of our lives. Improvements in the performance of computing devices, and bandwidth for wired line and wireless communication networks have transpired an evolution of new networked multimedia systems.

These improvements enrich the experience for carrying out communication, entertainment and edutainment. People now have the luxury of experiencing Video on Demand, Video Conferencing, Education on Demand, Virtual Classrooms on Desktop Computers and Portable Devices. However, as society adopts these technologies, they will become dependant upon efficient and effective delivery of multimedia information. Consequently, people would want guarantees in the provision of services. The Quality of Service (QoS) concept arises due to multimedia communications requiring end-to-end service guarantees in order for the transmitted multimedia content to be comprehended by the user. To allow the users to request such guarantees from a system, this requires the development of User-Centred Quality of Service (QoS) Management systems and interfaces that enable users to request and negotiate QoS.

1.2 Motivation for this Research

Transmission of multimedia information requires consideration of QoS from three perspectives, namely: User Perspective, Application Perspective, and Transmission Perspective. At the User Perspective, acceptable QoS levels are influenced by the user’s perceptual, auditory and visual faculties. At the Application Perspective, users require that an acceptable Quality of Presentation (QoP) is maintained so that multimedia information can be easily perceived and comprehended. Communication networking protocols are required to efficiently deliver multimedia information at a level that does not compromise the quality of the presentation. Current, communication networking protocols have been enhanced with QoS mechanisms to guarantee services at the Transmission Perspective.

To provide the user with the QoS that they desire, and guarantee the preservation of these services throughout the communication session; it is important that facilities are made available which assist the user to request, negotiate, and re-negotiate QoS efficiently. Such a solution requires primary consideration of the User Perspective; however, its relation to the Application and Transmission Perspectives must be taken into account.

These solutions need to employ better Human Computer Interaction (HCI) techniques that model the user's requirements, which can then be mapped into the system's functionality. Developing such a system would require a framework that integrates the functionality of the User, Application, and Transmission Perspectives for managing QoS.

If a QoS management system does not give users the means to specify their needs easily and effectively, it is likely that the users will not receive the service they require. Therefore, it is important that users can specify their requirements using intuitive interaction methods and interfaces. Developing such interfaces is the primary motivation for this research.

1.3 Originality of the Thesis

Currently, most QoS mechanisms operate at the Transmission and/or Application Perspectives. Communication protocols (such as TCP/IP [1]) have been enhanced with QoS management facilities that reserve and allocate resources for transmitting multimedia information. New communication protocols have also been developed that provide adequate QoS support for the delivery of multimedia information. For example, in wireless networks, the Universal Mobile Telecommunications System (UMTS) [2] includes QoS management facilities, but only at the Transmission Perspective.

Therefore, a holistic approach towards managing QoS is required that considers human factors related to the User Perspective, over and above the Application and Transmission Perspectives.

This thesis introduces a Three Layer QoS (TRAQS) model that gives required prominence to the User Perspective, thus providing a framework for developing a 'user-centred' QoS management system. Next, user interfaces are developed based on this model. These interfaces are then subjected to usability investigations.

1.4 Research Aims & Objectives

Aim:

This research aims to develop a holistic QoS model for QoS management; and investigate the usability of related interaction methods and interfaces for QoS specification, negotiation, and re-negotiation.

Objectives

Specific objectives for this research are as follows:

- Survey and identify the shortcomings of current technologies that support QoS guarantees for the delivery of multimedia information.

- Provide a framework for developing user-centred QoS management systems for networked multimedia applications.

- Investigate novel interaction methods and interfaces for technical and non-technical users to request and negotiate QoS prior to initiating a multimedia session, and re-negotiate QoS in real-time while the multimedia session is in progress.

- Study the usability of these novel user interfaces and their features for QoS management on Desktop Systems and Portable Devices.

1.5 Research Methodologies and Techniques

To conduct this research, the following methodologies and techniques were employed.

- *Literature Study* – A literature survey was carried out to identify the different networked multimedia applications that are widely used and to understand the basic function of each application. Formal definitions of QoS, QoS technology developments and applications, and QoS negotiation techniques were studied. Current communication protocols, architectures and frameworks were analysed and their limitations for the provision of QoS were identified. The concept of holistic management of QoS was investigated, different holistic QoS models were analysed, and their limitations were identified. Methods for performing usability studies for multimedia systems were investigated, and the different types of usability studies, and usability testing processes were identified.

- *Develop a Holistic QoS Model* – A holistic Three Layer QoS (TRAQS) model was developed using the top-down approach where the user's requirements were considered for the development of taxonomies and models to manage and negotiate QoS parameters. Task modelling technique was used to model the system behaviour and the user interaction involved for specifying and negotiating QoS. Task models facilitated the development of the interfaces.

- *Analyse and Re-engineer Traditional Usability Testing Processes* – Current usability testing processes and facilities were analysed for their efficiency in performing the usability studies. This required conducting specific usability studies to evaluate the efficiency of the testing facility configuration and the procedures used to perform the experiment. This assisted to streamline the usability testing procedure and re-configure the testing facility to improve its efficiency.

- *Investigate Novel Interaction Methods* – Preliminary investigations were conducted into evaluating the usability of different Physical User Interface (PUI) devices, to perform basic computer control using different Graphical User Interface (GUI) elements. This guided the development of innovative interaction methods that can be used in developing interfaces for QoS management.

- *Study Interfaces for QoS Specification, Negotiation and Re-negotiation* – Interfaces for QoS specification, negotiation and re-negotiation were developed based on the TRAQS model. Usability engineering techniques were used throughout the development cycle to ensure that the interfaces correctly modelled the user requirements. Controlled usability studies with real users were performed for each interface. The re-engineered usability testing process was used to conduct the various tests efficiently, and collate their results systematically. Pre-experiment questionnaires were used to gather background information about each participant. Participants were ranked and categorised according to their technical sophistication. Participant feedback was collected using post experiment questionnaires; Task Completion Times and Task Error Counts were also recorded. The usability of each interface was determined by analysing the collated data.

1.6 Contribution to Knowledge

This research contributes to the enhancement of knowledge in the following areas:

- Develop a better understanding of the user requirements for QoS management.
- Investigate a novel holistic QoS model, namely TRAQS, for developing user-centred QoS management systems, which integrates the three perspectives for multimedia information transmission.
- Develop innovative interaction methods for requesting, negotiating and re-negotiating QoS, thus improving the processes for QoS management.
- Provide a thorough usability analysis of the innovative interaction methods and interfaces developed for QoS management, and give recommendations for further research.

The above contributions to knowledge facilitate the ongoing developments of new QoS management systems for networked multimedia systems. It is anticipated that this research will foster the development of user-centred QoS management interfaces on real products that can be used even by a layperson.

1.7 Organisation of the Thesis

This thesis includes seven chapters. Succeeding to this Introduction chapter, Chapter 2 comprises a literature study of Networked Multimedia Systems, current QoS technologies, and usability studies for multimedia systems. Current QoS Communication Protocols, Architectures, and Frameworks are reviewed, and their limitations are identified.

Chapter 3 presents details of the TRAQS model, a taxonomy for QoS Parameters and Applications, the three performance aspects for multimedia communications, and task modelling of the TRAQS model.

Chapter 4 introduces the reader to usability testing; an overview of Usability Engineering, Usability Assessment Methods, and Usability Testing Facilities is presented. This chapter also presents the Re-engineered Usability Testing Process, Assessment Methods, and Usability Evaluation Criteria employed in this research.

Chapter 5 presents the motivation for usability testing of user interfaces for QoS management. User interfaces for QoS management are introduced, and an overview of the aims and objectives for the usability investigation of these interfaces is given. A description is presented for the usability testing procedures, and a participant ranking model – used to classify the usability data.

Chapter 6 presents the various usability investigations on interaction methods and interfaces for QoS management on Desktop Systems and Portable Devices. Ten separate usability studies are presented which investigate the usability of novel system control and feedback interaction methods; interfaces for QoS specification, negotiation, and re-negotiation on Desktop Systems; and interfaces for QoS re-negotiation on Portable Devices. Each usability study presents details of the user interface design, experiment design, a usability analysis, and specific conclusions.

Chapter 7 concludes this thesis; a summary of the major findings and suggestions for further research is presented.

Detailed appendices are included on a CD, which is attached to the back of this thesis.

Chapter 2
Literature Study

Summary

This chapter provides a literature review of current QoS technologies that fall short to provide consideration of the three perspectives for multimedia information transmission; namely: User Perspective, Application Perspective and Transmission Perspective. Current communication protocols, such as: TCP/IP, RSVP, ATM, MPLS, DiffServ, SBM, and architectures, such as: XRM, QoS-A, Heidelberg QoS Model, OSI QoS Framework, IETF QM, and other QoS supporting architectures, operate predominantly at the transmission and/or application perspectives. These technologies do not provide adequate consideration to the user perspective for the provision of QoS.

A Holistic view of QoS is a concept that many research bodies have adopted in the attempt to integrate all three perspectives for the seamless provision of QoS. Siemens have developed the AQUILA model that provides consideration of eight perspectives for managing QoS. Bauer and Patrick have extended the OSI model to incorporate human factors. Bridging the gap between multimedia, network protocols and users enables people to interact with the system and request QoS. To accomplish this there is a need to develop efficacious interaction methods and user interfaces. Such developments would require extensive usability studies for their application in QoS management of Networked Multimedia Systems.

2.1 Introduction

Development of communication technology has evolved and has enabled support for transmitting multimedia information over communication networks. This evolution has lead to development of various multimedia applications, which are now being used in almost every aspect of our daily lives. Section 2.2 provides an overview of current networked multimedia systems and their application in society.

As society becomes more dependant on this technology, it is important that some guarantees for Quality of Service (QoS) are made available. Transmission of multimedia information can be categorised into three perspectives, namely: User, Application and Transmission perspectives. To provide such guarantees, it is essential to understand the concept of QoS and its applicability in networked multimedia systems. Section 2.3 provides an overview of QoS, a review of formal definitions made by researchers, development of QoS technologies and applications, and QoS negotiation techniques.

Thus far, various technologies have emerged that guarantee QoS at the Application and Transmission perspectives. Current network communication protocols, such as: TCP/IP, RSVP and ATM, guarantee QoS at the transmission perspective but do not provide adequate consideration to the Application and User perspectives. Section 2.4 describes these communication protocols in further detail.

Current QoS architectures and frameworks, such as: XRM, QoS-A, Heidelberg QoS, and UMTS – for wireless networks, provide end-to-end QoS guarantees at the Application and/or Transmission perspectives. Section 2.5 presents a review of various QoS architectures and frameworks.

As users require different QoS guarantees for different multimedia applications, it is necessary that the users are able to request QoS based on their needs, and, the cost they are willing to pay. In this research a Three Layer QoS (TRAQS) model has been developed that gives required prominence to the User, Application and Transmission perspectives. Similar research in this area includes the following projects: ‘A Holistic View of QoS’, ‘Siemens AQUILA model’, described further in Section 2.6.

Developing a solution that bridges the gap between the user requirements and the system functionality would require employing efficacious Human Computer Interaction (HCI) techniques. Usability studies in QoS management for networked multimedia systems are also required to improve the HCI. Usability studies can be performed at three levels: Interface, System and Service level. Section 2.7 describes usability studies for networked multimedia systems, the different levels of usability studies, and usability testing processes.

2.2 Networked Multimedia Systems

2.2.1 Overview of Networked Multimedia Systems

Networked Multimedia Systems comprise of various real-time multimedia applications such as: Video on Demand (VoD), Video Conferencing, Collaborative Conferencing, and Education-on-Demand (EoD). These applications are used for: Communication, Entertainment, Education, Edutainment, Health and Medicine, Marketing and Research. Networked Multimedia systems require for effective and comprehensible transmission of multimedia content.

Without QoS guarantees, failure in the delivery of multimedia information can be critical in certain situations. For example, for a virtual organisation that heavily relies on video conferencing to maintain communications between offsite offices, a failure or inefficient transmission of information could paralyse communications, thus effecting the company’s operations [3]. In order to maintain efficient communications and unambiguous multimedia information transmission, it is imperative to provide the ability to specify the required Quality of Service (QoS).

Networked Multimedia Systems are being implemented in distributed environments, where the underlying architecture provides support for such services and applications. The following section describes the various networked multimedia systems and applications.

2.2.2 Networked Multimedia Systems and Applications

Video on Demand (VoD) is an amalgamation of the video hire and pay-TV concepts [4]. With VoD, the end user is able to select and play a desired movie from a menu system displayed on their TV using a remote control. In a VoD service, a video server transmits the movie over a network to which the user's TV is connected. Nowadays, VoD services are mainly available in hotels, aeroplanes, and on the Internet. With the introduction of Pay-Per-View Cable TV, VoD would be available in many households also. Some Pay-Per-View Cable TV services do not incorporate a true VoD system. There are two types of VoD services available, they are: *Full Video on Demand (FVoD)* and *Near Video on Demand (NVoD)* [4]. *FVoD* provides the user with full VCR-like controls including; play, stop, rewind, fast forward and pause. On the other hand, *NVoD* does not provide the user with VCR-like controls. The user is only able to select a movie and watch it right through. For this type of service, many video servers transmit each movie clip at regular intervals, thus, for a user to begin watching the movie they must wait for the next transmission interval.

Multicast VoD Services enables efficient distribution of video streams to multiple clients, which improves performance [5]. Mobile VoD services are currently being researched and developed to provide efficient transmission of video content to portable devices. This service is useful for on-the-go information access, such as News-on-Demand [6], Weather Updates [7], and Location Based Systems [8][9].

Video Conferencing allows two or more users from different sites to have virtual face-to-face communication. Users are able to communicate via text, voice, and video [4]. An enhancement to Video Conferencing is Collaborative Conferencing. *Collaborative Conferencing* enables sharing of applications where two or more users can work on the same project/document/application in real-time.

Multicast Video Conferencing enables two or more users to collaborate in a virtual-meeting. *Video/Collaborative Conferencing* is being used in the corporate, education, health, and marketing industries to form virtual organisations. Further enhancements to video conferencing systems have been developed, these include: *GAZE-2*, which uses eye-controlled camera direction to ensure parallax-free transmission of eye focus [10];

and *Shared Virtual Team User Environments* that comprise 3D video conferencing systems simulating a team environment [10].

Education-on-Demand (EoD) enables instant availability of education material over a communication network. Applications of EoD include virtual classrooms and distance education [11][12]. These applications simulate a rich, near-classroom experience to non-classroom students [13]. Virtual classrooms enable shared workspace for teachers and students to interact and work on a project simultaneously. These applications are similar to collaborative conferencing, however, enhancements are made that enable users to immerse themselves in a virtual classroom environment. Leonard et al present innovations using IP video and data collaboration techniques [13].

2.2.3 Conclusion

Networked Multimedia Systems are currently revolutionising how people in society perform their daily tasks. Communication systems are making a transition from analog to digital in many areas, including, television broadcasting, entertainment, education etc. With this transition society is becoming more dependent on networked multimedia systems. As this dependency grows, the issue of QoS will become more significant as users of such systems will require efficient communications. Much research is underway in developing QoS supporting technologies. However, sufficient research has not been conducted on how users can interact with this technology to request and negotiate desired QoS. Therefore, further research is required in developing QoS supporting technologies that enable negotiations between the user and the system. This thesis presents the outcomes of developing innovative and intuitive interfaces that enable efficient Human Computer Interaction (HCI) for requesting, negotiating and managing QoS.

2.3 Quality of Service

2.3.1 Overview of the Quality of Service Concept

Multimedia information includes text, images, audio, video and/or a combination of these. In order for the multimedia information to be fully comprehended, transmission of multimedia content in real-time requires stringent constraints on the synchronisation of the transmitted content – in particular for audio and video content. The level of synchronisation for transmitting multimedia information is specified in terms of Synchronisation Accuracy Specification (SAS), which includes delay, jitter, skew, and error rate [4]. For the multimedia content transmitted in real-time to be correctly comprehended by the user, a Quality of Service (QoS) specification and its processing mechanisms must be included in the system. QoS specification depends upon qualitative and quantitative requirements that are related to the user's auditory and visual perception.

Microsoft states that "the goal of QoS is to provide preferential delivery service for the applications that need it by ensuring sufficient bandwidth, controlling latency and jitter, and reducing data loss" [14]. Cisco states that QoS "refers to the capability of a network to provide better service to selected network traffic over various technologies, including Frame Relay, Asynchronous Transfer Mode (ATM), Ethernet and 802.1 networks, SONET, and IP-routed networks that may use any or all of these underlying technologies" [15]. Dictionary.com states that QoS is "the performance properties of a network service, possibly including throughput, transit delay, and priority. Some protocols allow packets or streams to include QoS requirements" [16]. The above definitions show that QoS has been viewed mainly from a technical perspective.

2.3.2 Defining Quality of Service

Many research bodies and organisations have developed their own definition of QoS. Currently the International Standards Organisation (ISO) defines QoS as "a set of qualities related to the collective behaviour of one or more objects" [17]. Wang et al state that QoS "refers to optimizing the performance of a network relative to a specific application and takes place through resource management. It is accomplished through management tools that reserve bandwidth, prioritize usage, monitor change, and scale resources to provide performance assurances" [18].

Internet Engineering Task Force (IETF) states "as the demand for networked real time services grows, so does the need for shared networks to provide deterministic delivery services. Such deterministic delivery services demand that both the source application and the network infrastructure have capabilities to request, setup, and enforce the delivery of the data. Collectively, these services are referred to as bandwidth reservation and Quality of Service (QoS)" [19].

Fluckiger states that QoS is "a concept based on the statement that not all applications need the same performance from the network over which they run. Thus, applications may indicate their specific requirements to the network, before they actually start transmitting data" [20]. Vogel et al states that QoS "represents the set of those quantitative and qualitative characteristics of a distributed multimedia system that are necessary to achieve the required functionality of an application" [21].

The above definitions identify QoS from a network and application point of view.

2.3.3 Quality of Service Technology Developments and Applications

Various research bodies have developed Networking Protocols with some QoS provision, such as: RSVP [22], ATM [23], MPLS [24], DiffServ [25], and SBM [26]. Other research in QoS Frameworks and Architectures include: XRM [27], QoS-A [28], Heidelberg QoS Model [27], OSI QoS Framework [27], IETF QM [27], and TINA [27]. The emergence of wireless / mobile communications has lead to the development of UMTS [2]. Ecklund et al conducted research on "A Dynamically-Configured, Strategic QoS Management Hierarchy for Distributed Multimedia Systems" [29]. These technologies effectively provide QoS by operating generally at the Application and/or the Transmission Perspectives.

Bauer and Patrick published research on extending the OSI model to incorporate human factors [30]. Allison et al have conducted research that introduces a holistic view of QoS for the delivery of multimedia information [31]. Similar research has been presented by Dabrowski et al, which introduce the 'Siemens AQUILA model' [32]. Ghinea et al have conducted research in bridging the gap between multimedia network protocols and users, and mapping quality of perception to QoS [33][34]. Chalmers and Sloman present a survey on QoS management in mobile computing environments, and introduce the notion *Static* and *Dynamic* QoS Management [35]. These and other major QoS technologies are discussed further in sections 2.4, 2.5, and 2.6.

2.3.4 Quality of Service Negotiation Techniques

In multimedia communication systems, QoS negotiation involves the process of exchanging QoS parameters between the communicating entities (e.g. the service provider and the user(s)), where the main objective is to achieve a common agreement upon the level of QoS, satisfying all communicating entities [36]. The following three QoS negotiation techniques can be distinguished.

Triangular Negotiation involves the three entities: *calling service user*, *service provider*, and *called service user* for QoS negotiations. The *calling service user* passes QoS specification to the *service provider*. The *service provider* can then downgrade parameter values. The resulting QoS specification is then passed on to the *called service user*. The *called service user* may in turn also downgrade the QoS parameters. This final specification is then returned to the *calling service user,* whereby, both the *called service user* and the *service provider* may reject the request. This method is further subdivided into the negotiation schemes: Information Exchange, Bounded Target and Contractual Value. Further details of these schemes have been published by Plagemann et al [36].

Bilateral Negotiation involves QoS negotiation taking place between *service users*. In this scheme the *calling service user* can propose QoS parameter values that can be accepted, or accepted with modifications, or rejected by the *called service* user. The *service provider* is not allowed to change the QoS specification [36].

Unilateral Negotiation represents a ‘take it or leave it approach’. The *calling service user* proposes QoS parameter values that can be accepted or rejected without modifications by the *service provider* [36].

These negotiation techniques demonstrate how QoS negotiation can be carried out between various communicating entities at the Transmission Perspective. To enable different users to negotiate QoS with network services and applications, user interfaces are required that employ efficacious interaction methods for negotiating QoS at the User Perspective. These interfaces need to employ usability engineering in the development process to ensure that even a layperson can easily negotiate QoS.

2.3.5 Conclusion

QoS plays an important role in the delivery of multimedia information. QoS takes crucial consideration of the SAS factors for transmission of multimedia information. Different organisations develop their own QoS definition and employ it in their research and development of QoS technologies.

QoS is perceived as a specification for network and application technologies to provide services that satisfy auditory and visual requirements of the user. Definitions of QoS have been developed based on a technical perspective. QoS technologies and applications have been developed based on the definitions presented in the above subsection. These technologies provide QoS at the application and transmission perspectives.

QoS negotiation techniques such as: Triangular, Bilateral and Unilateral enable users to negotiation services between each other and the service provider. QoS negotiation between the user and the system is essential. This is possible via the development of QoS definitions, models and user interfaces that allow the user to interact with the system to obtain a desired QoS. Such a facility requires viewing QoS from a user's perspective rather than a technical perspective, as addressed in this research project.

2.4 Quality of Service Communication Protocols

2.4.1 TCP/IP Protocol Suite

TCP/IP (Transport Control Protocol / Internet Protocol) has become one of the most widely used communication protocol suite [1]. Originally, TCP/IP was not designed to support transmission of multimedia content, and did not have QoS mechanisms. Over the years, modifications were made to TCP/IP to support multimedia communications. This led to the development of QoS protocols such as RSVP (explained in the following subsection), ST-II (Stream Protocol), and RTP (Real-Time Transport Protocol).

TCP/IP is the underlying technology for the Internet as it has become a standard for internetworking [1]. The architecture of TCP/IP is a 5-layer protocol stack, comprising of both network-oriented protocols and application support protocols, as shown in

Figure 2-1. TCP/IP operates at the Transmission Perspective and delivers connection-oriented and connectionless services.

TCP/IP Layers		
Layer No.	**Name**	**Protocols**
1	Application	MIME SMTP FTP TELNET HTTP
2	Host-to- host transport	TCP UDP
3	Internet	IP ST-II
4	Network access	RSVP
5	Physical	

Figure 2-1: TCP/IP – Five Layer Protocol Stack

2.4.2 Resource Reservation Protocol (RSVP)

The ReSource ReserVation Protocol (RSVP) is a signalling protocol that provides QoS guarantees for multimedia communication over the Internet by reserving resources, such as bandwidth and buffers [22]. The RSVP protocol provides the ability to set-up and control reservation enabling the Integrated Services (IntServ) model [37].

RSVP provides two fundamentally different types of Integrated Services: Guaranteed Service and Control Load Service. *Guaranteed Services* provides guaranteed bandwidth, thus emulating a dedicated virtual circuit [38]. *Controlled Load Services* is equivalent to “best effort service under unloaded conditions” [38].

RSVP sends a PATH message from the sender, which contains traffic (TSpec) information to the destination address. The destination address may be unicast or multicast. A receiver requests resources by sending a RESV message, which comprises a request specification (Rspec) that indicates the type of Integrated Services needed [22][38]. RSVP can run over the Internet Protocol (IP) version 4 [23] as well as version 6 [39]. Thus, the RSVP protocol handles QoS issues only at the Transmission Perspective.

2.4.3 Asynchronous Transfer Mode (ATM)

The Asynchronous Transfer Mode (ATM) is based on the principle of cell switching, using Switched Virtual Connections (SVCs), in which data is asynchronously transmitted in 53-byte packets called cells [23]. The ATM technology was conceived as a transmission service that can support a variety of data types, including multimedia traffic.

ATM can deliver synchronous, asynchronous as well as isochronous traffic. ATM includes QoS by providing bandwidth-on-demand to an end system through SVCs, which are monitored and controlled by the Signal Control Point (SCP). Permanent Virtual Connections (PVCs) are set-up, controlled and managed by a Network Management System [23].

The ATM protocol architecture (Figure 2-2) includes a User Plane, Control Plane and a Management Plane. The *User Plane* manages user data transfer including flow and error control. The *Control Plane* manages call connection control functions. The *Management Plane* comprises the Layer Management and the Plane Management functions. ATM operates primarily at the Transmission Perspective and does not include the ability to perform negotiations between the user, application and transmission system.

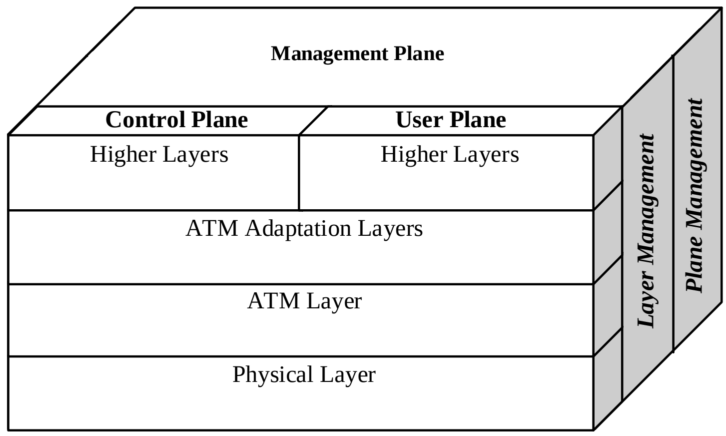


Figure 2-2: ATM Reference Model

2.4.4 Multi-Protocol Label Switching (MPLS)

Multi-Protocol Label Switching (MPLS) is a QoS protocol that marks traffic at ingress boundaries and un-marks at egress points [24]. MPLS resides only within routers and uses the marking priority scheme to determine the 'next-hop' for the traffic within the network. MPLS is more of a 'traffic engineering' protocol rather than a QoS protocol [38][40]. MPLS routing is used to establish 'fixed bandwidth pipes', similar to PVC in ATM and Virtual Circuits in RSVP, to improve QoS within the transmission perspective. MPLS is a multi-protocol system, which can be used with protocols other than TCP/IP, ATM, and RSVP, operating in the Transmission Perspective.

2.4.5 Differentiated Services (Diffserv) Model – Prioritisation

Differentiated Services (DiffServ) provide a clear-cut and unrefined method to categorise services for different applications [25]. Unlike IntServ, which functions on a signalled-QoS model, DiffServ functions on the provisioned-QoS model where network elements are set up to service multiple classes of traffic [25]. DiffServ operates on 'per hop behaviours' (PHBs), and classifies traffic into two service levels: Expedited Forwarding (EF) and Assured Forwarding (AF) [25]. Furthermore, DiffServ performs Packet Marking at ingress and egress points within the architecture in order to deliver end-to-end QoS in the Transmission Perspective [41].

2.4.6 Subnet Bandwidth Management (SBM)

Subnet Bandwidth Management (SBM) is a signalling protocol that allows communication between nodes and switches in the SBM Framework and enables mapping to higher-layer QoS protocols [26][38][42]. SBM consists of a Bandwidth Allocator (BA), which manages the bandwidth resources, and a Requestor Module (RM), which maps the high-layer QoS protocol parameters with the layer 2 (e.g. Ethernet) priority levels. SBM provides QoS to the end-system, operating in the Transmission Perspective.

2.4.7 Conclusion

As the demand for QoS supporting communication protocols is growing, many research bodies and organisations have made effort to modify existing communication protocols to cater for the provisioning of QoS at the Transmission Perspective. New technology, such as the ATM, have been developed and specifically targeted for supporting delivery of live multimedia information.

These protocols provide QoS by manipulating and reserving resources at the Transmission Perspective. Some of these communication protocols provide enough resources and QoS mechanisms that make possible for multimedia applications, such as: video conferencing, and video streaming, telephony to be used over communication networks. Although it is possible to use these applications over the communication networks, the quality and synchronisation of the transmitted multimedia information is of an unacceptable level.

As there is no such facility for users to specify a desired QoS, this leaves them with the burden to put up with poor communication quality. Developing QoS mechanisms that predominantly function at the Transmission Perspective only does not effectively provide QoS as there is insufficient consideration for the application and user requirements.

2.5 QoS Architectures & Frameworks

2.5.1 Extended Integrated Reference Model (XRM)

Extended Integrated Reference Model (XRM) developed by the COMET Group at Columbia University, is a real-time framework for control and management of multimedia communication networks [27]. XRM is divided into five distinct functional planes shown in Figure 2-3. Starting at the top of these are: Network and Systems Management, Resource Control, Data Abstraction and Management, Connection Management and Binding, User Information Transport and Computing [27]. The XRM is based on theoretical work done for guaranteeing QoS requirements in ATM networks and end-systems integrated with multimedia devices. The XRM operates in the

Transmission Perspective and provides QoS to the end-system by using management, traffic control and information transport functions [27].

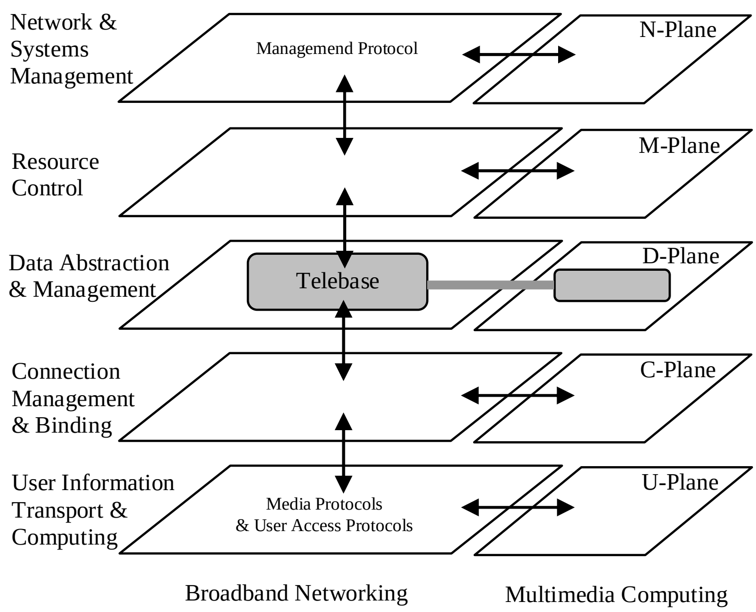


Figure 2-3: Five planes of the XRM Architecture

2.5.2 Quality of Service Architecture (QoS-A)

The Quality of Service Architecture (QoS-A) proposed by the COMET group at Columbia University, New York, USA is shown in Figure 2-4. The QoS-A is a multi-layer/plane architecture of services and mechanisms for QoS management and control of continuous media flows in multi-service networks [28]. The QoS-A model encompasses three key notions: Flows, Service Contracts and Flow Management [28].

Flows characterises the production, transmission and eventual consumption of single media streams related to QoS [28]. *Service Contracts* are induced agreements of QoS levels between users and providers. *Flow Management* monitors and maintains the contracted QoS levels. The QoS-A provides an end-to-end QoS framework at the Transmission Perspective.

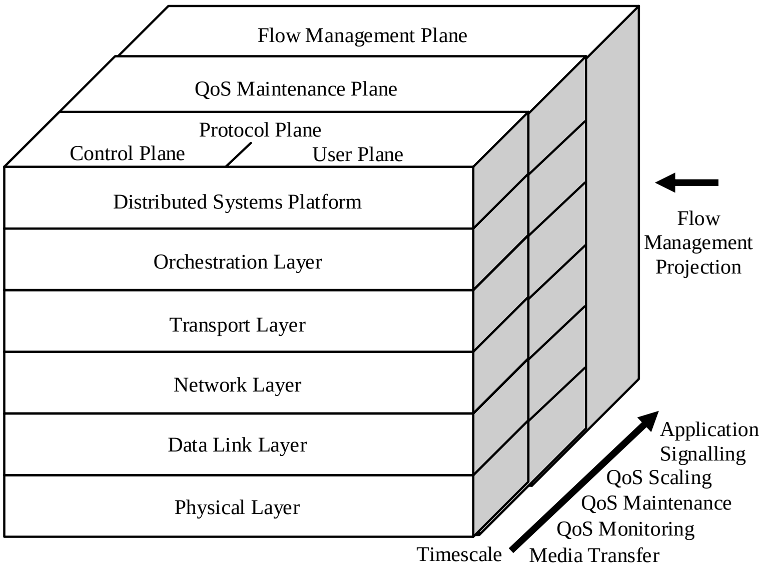


Figure 2-4: QoS-A Model

2.5.3 *Heidelberg QoS Model*

The Heidelberg QoS Model, shown in Figure 2-5, has been developed by the HeiProject at IBM´s European Networking Centre in Heidelberg. This model provides QoS guarantees in the end-systems and the network [27]. Its communications architecture includes a continuous media transport system (known as HeiTS/TP) that provides QoS mapping and media scaling [43].

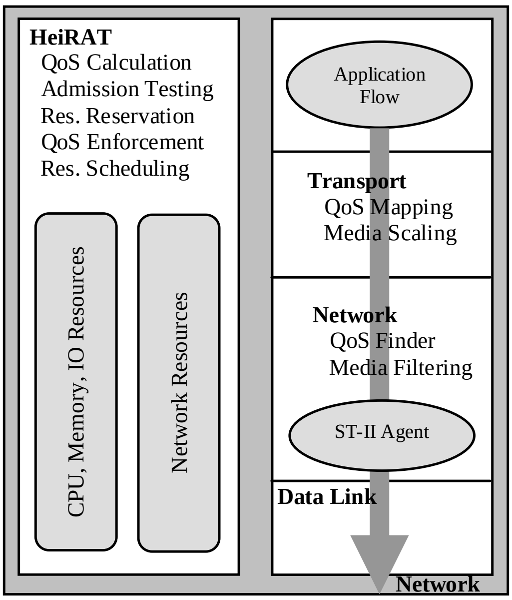


Figure 2-5: Heidelberg QoS Model

The Heidelberg QoS model consists of HeiRAT (Heidelberg Resource Administration Technique) [27], which provides end-to-end QoS negotiation, QoS calculation, admission control and QoS enforcement and resource scheduling. The Heidelberg Model facilitates QoS at the Application and Transmission Perspectives.

2.5.4 Open Systems Interconnection (OSI) QoS Framework

Open Systems Interconnection (OSI) QoS Framework developed by the International Standards Organisation (ISO) provides QoS support for OSI communications [27][44]. The OSI QoS Framework defines a set of terminology and notions for QoS and provides a model covering objects associated with QoS in the OSI standards.

The OSI QoS Framework defines the following three concepts: QoS requirements, QoS characteristics and QoS categories. End-to-end QoS monitoring, maintaining and controlling is performed in two types of entities: Layer-specific Entities and System-wide Entities [27]. The OSI QoS Framework operates and provides QoS to end systems at the Transmission Perspective.

2.5.5 IETF QoS Manager (QM)

The Internet Engineering Task Force (IETF) has developed a QoS Manager (QM), shown in Figure 2-6, for the TCP/IP suite of protocols. QM presents an abstract management layer, which isolates applications from the lower layers [27]. QM provides QoS negotiations between the applications and the communication protocols to provide the best possible QoS. The QM then determines the type of resources available and allocates them to applications.

QM supports heterogeneity, where application requirements are coordinated with the underlying QoS capability [27]. In addition, QM is transparent, extensible and provides for QoS negotiations between the Application and Transmission Perspectives.

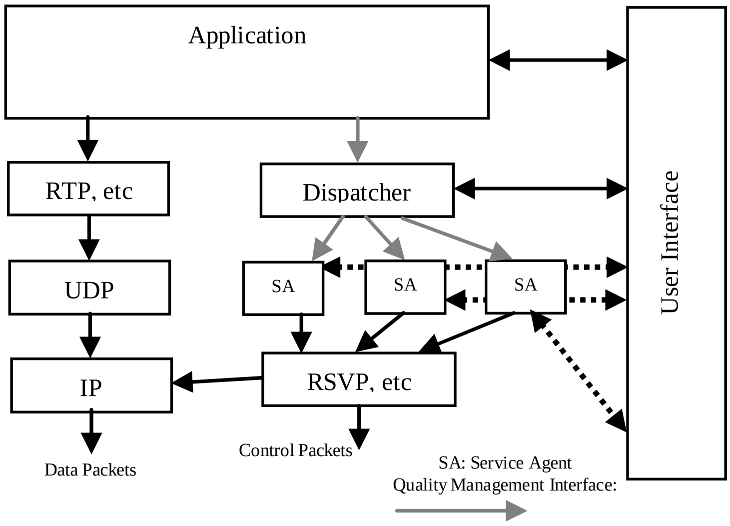


Figure 2-6: Internet Engineering Task Force (IETF) QoS Manager (QM)

2.5.6 Telecommunications Information Networking Architecture QoS Framework (TINA)

The Telecommunications Information Networking Architecture QoS Framework (TINA) developed by the Telecommunications Information Network Architecture Consortium (TINA-C), is a QoS framework designed to cater for distributed telecommunications applications where computational and engineering viewpoints are addressed [27]. The TINA framework is shown in Figure 2-7.

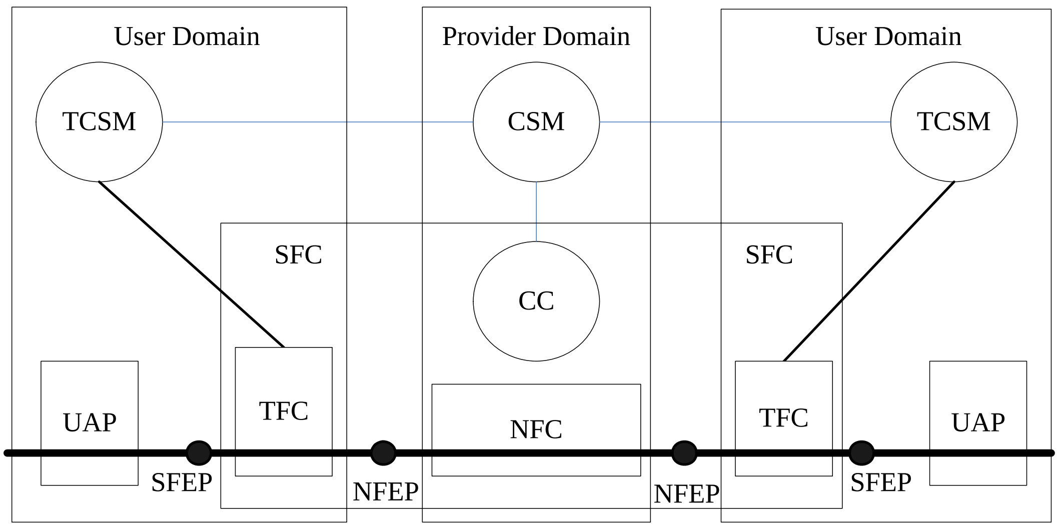


Figure 2-7: TINA Framework

QoS parameter requests are defined as objects known as *Service Attributes* [27]. *Distributed Processing Environments* manages QoS requests that are made by the underlying multimedia services. Transparency of QoS provisions is maintained by using resource managers that allocate required resources. This QoS management transparency relieves the burden of the applications in managing the QoS. The TINA architecture predominantly operates in the Application and Transmission Perspectives.

2.5.7 The OMEGA Architecture

The OMEGA Architecture, developed at the University of Pennsylvania, is also an end-to-end architecture that provides QoS guarantees in distributed systems [27]. The main focus of the OMEGA architecture is to provide global and local resource management in distributed systems [27]. The OMEGA architecture is presented in Figure 2-8.

Three relationships are identified in the OMEGA architecture for the provisioning of QoS and they are: Application QoS Requirements, Local Resource Management, and Global Resource Management [27]. OMEGA comprises a QoS Broker, which is a middleware component that caters for QoS negotiation between the underlying network system, the operating system and the application. All QoS requests are made via the QoS Broker where resource negotiation and allocation is established with the operating system and the network.

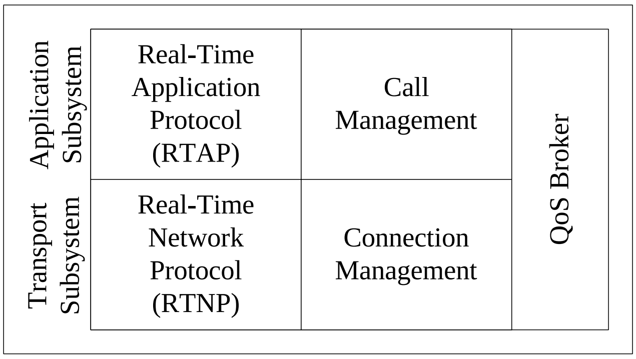


Figure 2-8: The OMEGA Architecture

At the application level a Real-Time Application Protocol (RTAP) administers call management, device management, synchronisation, and media delivery [45]. At the transmission level, a Real-Time Network Protocol (RTNP) administers connection management, error correction, rate control, and network access [45]. These protocols operate over communication channels that provide guaranteed services to the application through the QoS Broker. The OMEGA architecture manages QoS at the Application and Transmission Perspectives.

2.5.8 The Tenet Architecture

The Tenet architecture (shown in Figure 2-9) is developed by the Tenet Group at University of California at Berkley. Tenet comprises a suite of protocols that operate over an experimental wide area Asynchronous Transfer Mode (ATM) network [27].

This architecture includes a Real Time Channel Administration Protocol (RCAP), Real Time Internetwork Protocol (RTIP) [27], Continuous Media Transport Protocol (CMTP), Real-Time Message Transport Protocol (RMTP) and Real Time Control Message Protocol (RTCMP) [27]. RCAP provides signalling and control services that enable requests to setup and teardown real-time communication channels.

RTIP is a connection oriented protocol that enforces real-time requirements for packet delivery. RMTP provides a message-based abstraction on top of RTIP [27]. CMTP supports the delivery of continuous periodic traffic [27]. RTCMP was designed to detect and recover failures in data transfers. The Tenet architecture caters for QoS management at the Transmission Perspective.

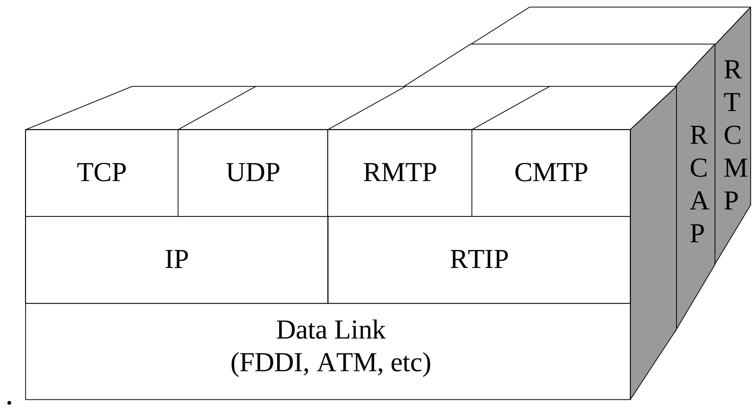


Figure 2-9: The Tenet Architecture

2.5.9 MASI End-to-End Architecture

The MASI End-to-End Architecture, known as the CESAME was developed at Laboratoire MASI, Universite Pierre et Marie Curie, and is an end-to-end framework that supports QoS for multimedia communications [27]. The MASI architecture comprises a generic QoS framework for specification of QoS requirements of distributed multimedia applications that operate over ATM networks [27]. Resource management is supported end-to-end incorporating the host operating system, host communication subsystem, and the ATM network.

The aim of the MASI architecture is to provide the ability to: “a) Efficiently map Open Distributed Processing (ODP) QoS requirements to specific resource modules, b) Maintain multimedia synchronisation of multiple ODP streams, c) Provide suitable communication protocol support for such multimedia services” [27]. The MASI architecture operates at the Application and Transmission Perspectives.

2.5.10 Washington University End System QoS Framework

The End System Framework, which is developed at Washington University by Gopal and Purulkar [27], is a QoS framework that operates within the end-system to provide QoS guarantees for networked multimedia applications. This framework comprises four components, namely: QoS Specification, QoS Mapping, QoS Enforcement and Protocol Implementations [27].

QoS Specification allows applications operating at a high level to specify QoS using a small number of parameters. *QoS Mapping* maps the QoS specification for each end-to-end application session into resources requirements, such as the CPU and network connection requirements. *QoS Enforcement* provides real-time processing guarantees using a *Real-Time Upcall (RTU)* facility [27].

Protocol Implementation structures protocol code as RTUs with attributes that are derived from high level specification by QoS mapping operations [27]. The Washington University QoS End System framework operates at the Application and Transmission Perspectives.

2.5.11 Universal Mobile Telecommunication System (UMTS)

The 3rd Generation Partnership Project (3GPP) has developed a Universal Mobile Telecommunications System (UMTS) [2] that supports QoS within the network services of a wireless/mobile network.

Within the UMTS architecture, bearer services provide the facility to setup QoS resources from the source to the destination. Various services are depicted in the UMTS architecture for QoS management, they are: End-to-End Service, UMTS Bearer Service, External Bearer Service, TE/MT Local Bearer Service, Radio Access Bearer Service, CN bearer Service, Radio Bearer Service, Iu Bearer Service, Backbone Bearer Service, UTRA FDD/TDD Service, and Physical Bearer Service [46].

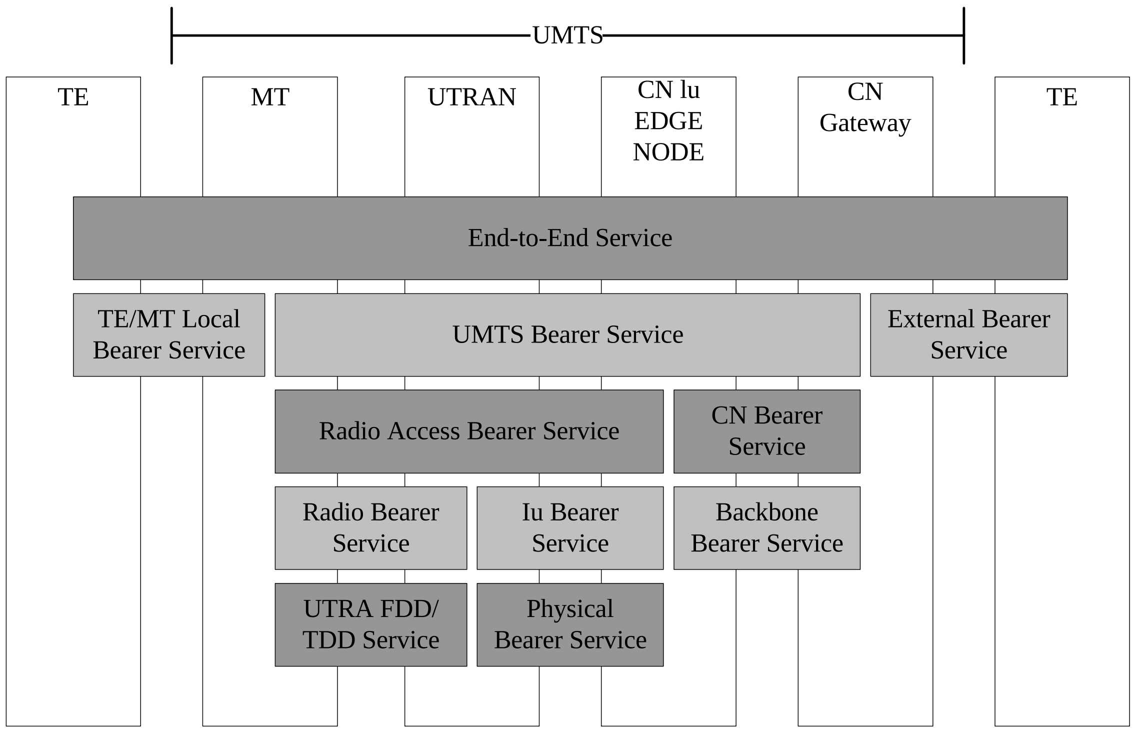


Figure 2-10: Universal Mobile Telecommunications System (UMTS) QoS Architecture

QoS traffic is categorised into four different classes, namely: Conversational Class, Streaming Class, Interactive Class, and Background Class [46]. *Conversational Class* caters for real-time traffic mainly used for voice applications. *Streaming Classes* caters for Real-Time and is traffic mainly used for streaming video. *Interactive Class* caters for best effort traffic, which is mainly used for web browsing. *Background Class* caters for best effort traffic and this is mainly used for telemetry and email applications. UMTS provides end-to-end QoS at the transmission perspective.

2.5.12 Other QoS Supporting Architectures

As QoS in multimedia communications is becoming a norm rather than an exception, many other research and industry bodies have contributed towards developing their own end-to-end QoS architectures. Ecklund et. al. present, a ‘Dynamically-Configured, Strategic QoS Management Hierarchy for Distributed Multimedia Systems’ [29].

The Distributed Systems Group (DSG) at Trinity College in Dublin, has conducted research for a Broadband Access Services Solution (BASS) that provides end-to-end IP-QoS through admission control and bandwidth reservation (Radius / BRAS) [47].

The CADENUS project (Creation and Deployment of End-User Services in Premium IP Networks), developed by Martel GmbH in Switzerland, is an end-to-end public network architecture separating ISP-level Services from Network Level Services [48].

The Global Communication Architecture and Protocols (GCAP) for new QoS services over IPv6 networks, developed by LAAS-CNRS (Laboratoire d'Analyse et d'Architecture des Systemes du Centre National de la Recherche Scientifique) in France, focuses on new end-to-end multicast and multimedia transport protocols, which are embedded in a new global architecture to provide a guaranteed QoS to advanced Multimedia Multipeer and Multi-network applications [49].

The QoSIPs project (Quality of Service and Pricing Differentiation for IP Services), developed at the University of Manchester Institute of Science and Technology DTG – Computation Department in the United Kingdom, aims to provide QoS real-time measurement and monitoring to support the Quality Based service innovation process of IP NSPs – and price each innovative service [50].

The TEQUILA project (Traffic Engineering for Quality of Service in the Internet, at Large Scale), funded by the European Commission, aimed to “study, specify, implement and validate a set of service definition and traffic engineering tools to obtain quantitative end-to-end Quality of Service guarantees” [51]. This was to be performed through careful planning, dimensioning and dynamic control of scaleable and simple qualitative traffic management techniques within the Internet (i.e., diffserv) [51].

A Common Object Request Broker Architecture (CORBA), developed by the Object Management Group (OMG), associates QoS requirements with media streams using an Object Request Broker (ORB) [52].

2.5.13 Conclusion

Various models and frameworks have been developed by many research bodies and organisations that provide end-to-end QoS. Many of these models provide QoS management mechanisms at the Application and/or Transmission Perspectives, without adequate consideration of the user interface from a non-technical user’s perspective. If the user is left out of the QoS management loop, then there is no guarantee that the service provided is fulfilling the user’s requirements. It is necessary that integration between the User, Application and Transmission Perspectives is made for QoS management.

A holistic approach towards QoS management has emerged, which gives consideration to all three perspectives in transmission of multimedia information. Research in this area is outlined in Section 2.6.

By considering the User Perspective for QoS management, this concept provides the prospect for developing systems that enable the user to request and negotiate services. Such advancements would require development of intuitive interfaces and interactions methods, which provide newfound Human Computer Interaction (HCI) for QoS management.

2.6 Holistic Quality of Service Models

Research in multimedia communications is moving towards developing holistic QoS management systems. ‘Holistic QoS’ implies that all perspectives in the multimedia communication process are integrated for managing QoS. As a convergence, holistic QoS models have been developed for communication systems operating over wired and wireless networks.

2.6.1 A Holistic View of Quality of Service

At the University of St Andrews in Scotland, research has been conducted on a holistic view of QoS for Distributed Learning Environments (DLE) [31]. In this research a conceptual model was developed that is based on three aspects: Usability, Middleware, and Infrastructure [31]. The *Usability* aspect manages abstractions for tutors and students, groups, interaction, real world input, and interface issues. The *Middle* aspect manages security, availability, responsiveness, concurrency control, and replication. The *Infrastructure* aspect manages networks, operating systems, server and client platforms. A TAGS [31] framework is defined, which facilitates the construction, deployment and management of DLEs. Research has been carried out for the QoS provision of all three aspects for the application of DLEs. Important aspects of the user's perception are considered for QoS management in this model.

2.6.2 Siemens AQUILA Model

The Siemens AQUILA model is a holistic model as it provides QoS negotiation and management at the User, Application and Transmission Perspectives [32]. The AQUILA model further subdivides these three perspectives into eight perspectives, they are: User, Application, Application Profiles, Access Network, Network Services, Admission Control, Resource Control, and Internet.

The AQUILA model introduces a software layer for distributed and adaptive resource control. AQUILA is oriented towards differentiated services (DiffServ) where intra-domain resource management is carried out using a Dynamic Resource Pool (DRP) and Inter-Domain uses Border Gateway Reservation Protocol (BGRP) [53].

The AQUILA network services comprise five classes for different types of network traffic, they are: Premium Constant Bit Rate, Premium Variable Bit Rate, Premium Multimedia, Premium Mission Critical, and Best Effort [53]. The AQUILA model operates at the User, Application and Transmission perspectives.

2.6.3 Bridging the Gap between Network Protocols and Users

Ghinea et al has presented research that focuses on bridging the gap between multimedia network protocols and users, and mapping Quality of Perception (QoP) to QoS [33][34]. In this research, human requirements are considered in the delivery of distributed multimedia. A user's perspective of QoS is defined as Quality of Perception (QoP). QoP is described as "a term which encompasses not only a user's satisfaction with the quality of multimedia presentations, but also his/her ability to analyse, synthesise and assimilate the informational content of multimedia displays" [33].

Dynamically Reconfigurable Protocol Stacks (DRoPS) architecture is defined which supports low cost reconfiguration of individual protocol mechanisms. DRoPS provides the ability to map user's perception of quality into technical QoS parameters. DRoPS enables to maintain QoP in the presented multimedia information in situations where degradation and fluctuation of QoS may occur. The DRoPS communication system is comprised of micro-protocols that perform subjective protocol functions. DRoPS carries out adaptable functionality for QoS using four protocol mechanisms, namely: (a) Sequence Control, (b) Flow Control, (c) Acknowledgement Scheme, and (d) Checksums [54]. DRoPS integrates the User, Application and Transmission Perspectives for the provision of QoS.

2.6.4 Extending the OSI Model to Incorporate Human Factors

Bauer and Patrick present research in extending the OSI (Open Systems Interconnection Reference Model) model to incorporate human factors [30]. In this research, the original OSI model that comprises the seven layers: 1-Physical, 2-Data Link, 3-Network, 4-Transport, 5-Session, 6-Presentation and 7-Application, has now been enhanced with the integration of the layers: 8-Display [30], 9-Human Performance [30], and 10-Human Needs [30]. It is conceived that with the extension of these three layers, three distinct aspects of Human Computer Interaction (HCI) will be incorporated within the functionality of QoS management.

The three aspects are [30]:

1. "What a user wants to do in the abstract sense (i.e. the need)."
2. "How that need is acted upon by the human."
3. "The artefacts that the user employs (hardware, software, etc).

HCI	10.	Human Needs (communication, education, acquisition, security, entertainment…)
	9.	Human Performance (perception, cognition, memory, motor control, social…)
	8.	Display (keyboard, GUI/CLI, vocal, bpp, ppi, ppm…)
OSI	7.	Application (http, ftp, nfs, pop…)
	6.	Presentation (ps, lz, iso-pp…)
	5.	Session (dns, rpc, pap…)
	4.	Transport (tcp, udp, rtp…)
	3.	Network Layer (ip, dhcp, icmp, aep…)
	2.	Data Link Layer (arp, ppp…)
	1.	Physical Layer (10bt, xDSL, V.42…)

Figure 2-11: The Ten-Layer OSI Model that Incorporates the HCI Extension

The Ten-Layer OSI model enables to link the needs of the User, Application and Transmission Perspectives. The *Display* layer (layer 8) encompasses the hardware, software and interfaces aspects that the user experiences [30]. This layer interacts with the *Application* layer (layer 7) and *Human Performance* layer (layer 9). In the former, human outputs are translated into a form that the application layer can process, and in the latter, signals from the application layer are translated into a form that the user can understand, which is displayed on a presentation device.

The Human Performance layer captures the information processing features and limitations of users, which are related to cognitive psychology, psychophysics, learning, memory, and human sensory organs [30]. These limitations are translated into parameters for specification of QoS. The *Human Needs* layer (layer 10) captures the essence of "why a user would interact with technology to get something done to satisfy a need" [30]. This layer considers the motivation behind the user's actions in carrying out a particular task; which may include communication, acquisition of knowledge and goods, entertainment and edutainment [30]. This research integrates the User, Application and Transmission Perspectives for the provision of QoS.

2.6.5 A Survey of QoS in Mobile Computing Environments

A survey has been carried out for specification and management of QoS in mobile computing environments. In this study, QoS is categorised into Static QoS Management, and Dynamic QoS Management [35].

Static QoS Management functions relate to properties or requirements which remain constant throughout some activity [35]. Static QoS Management is carried out prior to the initialisation of a session where a pre-contract of QoS requirements is established. *Dynamic QoS Management* is carried out in real-time (on an ongoing basis) where contract specifications of QoS are established based on the change within the environment [35].

This research suggests that techniques developed for QoS provision in specific computing environments/technologies such as: portable laptop computers, palmtops, and personal digital assistants with integrated communication, should be integrated into a single generic and flexible QoS management system [35]. Specific research in order to achieve this system would require prominence given to the User, Application and Transmission Perspectives for QoS provision.

2.6.6 Conclusion

The development of sophisticated multimedia communication systems has revealed that QoS management is now being applied to all perspectives of multimedia information transmission. Research bodies are now developing models/concepts that integrate the three perspectives for QoS provision (User Perspective, Application Perspective, Transmission perspective). Prominence is given to each perspective in the consideration of QoS for a variety of computing environments and multimedia applications.

A holistic view of QoS is now emerging where consideration is given to all three perspectives in transmission of multimedia information. Major focus is being applied to the User Perspective for QoS provision. Efforts are now being made in providing the ability for non-technical users to interact with systems to request QoS. This bridges the gap between multimedia network protocols and users.

Integrating the three perspectives for QoS management will prove to be beneficial for future QoS management systems operating in diverse computing environments. As the direction of future QoS architectures for Multimedia Communications is steering towards a holistic view, it is inevitable that further research is required into refining the process of managing QoS at each perspective. For the User Perspective, it is clearly visible that usability studies are required in developing user interfaces for efficient and effective HCI for QoS management.

2.7 Usability Testing for Networked Multimedia Systems

Usability testing involves measuring the ‘quality of user experience’ for particular applications, such as: software applications, websites, electronic devices, mobile phones etc. According to the International Standards Organisation (ISO), usability is the “effectiveness, efficiency and satisfaction with which a set of users can achieve a specified set of tasks in a particular environment." [55].

Usability testing emerged during the Second World War, when intensive research was carried out into the use of new technologies. It was discovered that through good interface design for new technology, the efficiency and performance for automated tasks increased.

As technology advancements led to the computer and telecommunication age, the need for research into developing usable equipment emerged, and thus, a usability revolution transpired [56]. As a result of this extensive research, usability engineering, testing, studies, methodologies, evaluation techniques, and laboratories have surfaced and evolved to the degree where there are international standards for producing usable equipment.

In computer science, the usability testing methodologies have been developed generally towards testing software applications and websites that operate in non-real-time environments. Current procedures in usability testing prove to be cumbersome and inefficient for usability studies in real-time systems.

2.7.1 The Need for Usability Studies

Currently, researchers and developers are beginning to introduce and integrate usability engineering within their product development cycle to create more 'usable' products. With various research bodies beginning to incorporate usability studies into their research for improving HCI, it is anticipated that in the near future there will be a growing demand for usability testing methods and procedures adapted for QoS Management interfaces for Networked Multimedia Systems.

There is a need for integrating usability engineering with research and development of user interfaces for managing QoS in Networked Multimedia Systems. By doing this, it will vastly improve the HCI aspects of these systems, which will eventually lead to innovative interaction methods and interface designs. To conduct usability studies on user interfaces QoS management interfaces, many standard usability techniques, methodologies, procedures, evaluation strategies and laboratory configurations need to be customised. This results in the development of new usability testing processes to cater for all aspects and scenarios encountered in QoS management systems.

2.7.2 Levels of Usability Studies

Usability studies can be conducted at three levels:

- *At the interface level,* various factors must be considered such as learnability, memorability, efficiency, reliability, and satisfaction [57].
- *At the system level*, the factors to be considered are reliability, latency (delays), and security (confidence) [57].
- *At the service level*, the factors to be considered are desirability, cost of change over, and retraining cost [57].

Many researchers and developers tend to develop the underlying technology of a system first and then attempt adapt/map the functionality of the system according to the users' needs. This is the bottom-up approach. This causes a predicament in the HCI of the system as the functionality of the system is mismatched with the demands of the user, which renders the system inefficacious. With a usability engineering focus, systems are developed using the top-down approach. User requirements, expectations, demands, experience and knowledge are taken into consideration first, and then used to inform the system design process [57].

2.7.3 Usability Testing Processes

Networked multimedia systems comprise of applications that require stringent constraints on delay and timing factors as they operate in 'real-time'. These include multimedia applications such as; video on demand, video conferencing, telephony conferencing etc [4]. Current, traditional usability procedures [58] and laboratories are mainly designed for non-real-time software applications, comprising stored applications that require less stringent constraints on delay and other timing factors. These applications include: websites, mobile devices, database software systems, etc [59][60][61][62]. To conduct usability studies on networked multimedia systems, this requires a different approach compared to non-real-time systems [63].

Usability Inspection & Assessment Methods

Usability experimentation can be conducted using empirical testing, where a participant is performing the experiment in a controlled environment such as a usability laboratory [64]. Usability testing can also be carried out 'off-site' where participants can use the software in their natural working environment and then provide feedback via questionnaires.

The most common usability testing method is the empirical test. In this testing method, a participant performs a scenario of tasks in a laboratory environment that simulates their working environment [64]. A facilitator assists and may observe the participant whilst an experiment is taking place [64]. An observer which is situated in another room, known as the observation room, performs the function of observing participants' responses [64].

Empirical testing usually requires a video monitoring system that records the participants' responses. Various usability inspection methods can be applied in evaluating the usability of software applications. These methods are:

- *Heuristic Evaluation* is a usability inspection method where usability specialists, developers, and/or system testers perform an 'in-house' usability test on a user interface [65].
- *Cognitive Walkthrough* is a technique where expert evaluators develop a scenario of tasks from a specification of an application and then 'role-play' the tasks that a user is to perform. Hence they 'walkthrough' the interface [65].
- *Pluralistic Walkthrough* comprises users and experts meeting and 'walking through' the scenario of tasks and evaluating each element of interaction [65].
- *Feature Inspection* focuses on analysing the features of a given interface where it is best applied during the middle stages of development [65].
- *Consistency Inspection* refers to ensuring that the consistency of multiple interfaces is maintained [65].
- *Standards Inspection* ensures compliance with industry standards [65].
- *Guideline Checklists* ensure that the original principles in the design of the interface have been satisfied within the implemented interface [65].

Other usability evaluation methods have been developed for the evaluation of software applications. These assessment methods are focused towards obtaining efficacious feedback from the participants. The following assessments are defined:

- *Feedback Questionnaires* is a method for the elicitation, recording and collecting of information [65].
- *Performance Measurement* enables to determine the participants' efficiency in performing each task [65].
- *Thinking Aloud Protocol* is a technique where during testing, participants are encouraged to vocalise their thoughts, feelings, and opinions for each task of a scenario [64].
- *Interviews* enable to query participants about their experiences and fondness of certain aspects of user interfaces used within the experiment [64].

Usability Laboratories

The classical usability laboratory configuration includes a participant room adjacent to an observer room. The participant room comprises a series of computer workstations, and the observer room comprises a monitoring system [64][66]. A typical usability laboratory configuration is given in Figure 2-12.

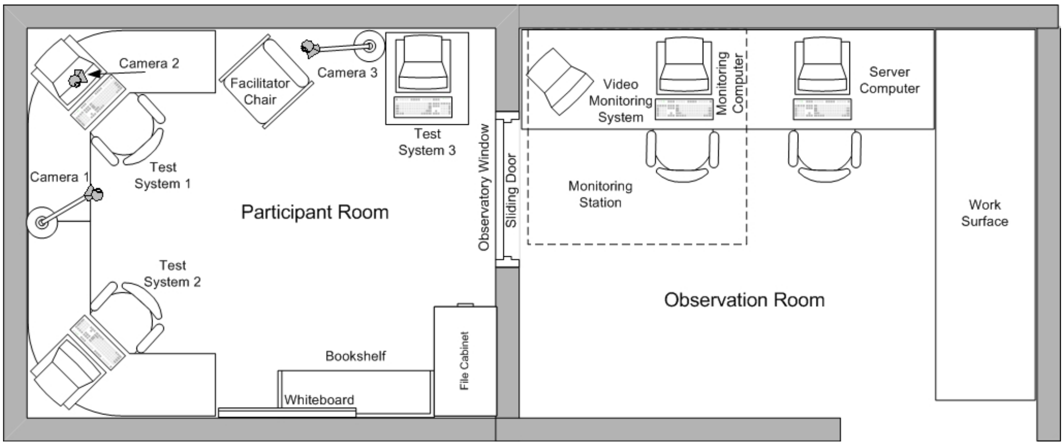


Figure 2-12: Traditional Usability Testing Laboratory

2.7.4 Conclusion

In this section a review is presented on usability of Networked Multimedia Systems. It can be deduced that usability studies of networked multimedia systems is at a beginning stage where further research of methodologies and techniques are required. As usability itself is a new area of study, major focus has been put into the usability of applications that operate in a 'menu-driven' metaphor. As networked multimedia systems operate in real-time, this imposes inherent restrictions on timing factors. Traditional usability testing methods do not apply for applications that function in the time varying domain.

As there is a need for QoS management user interfaces, which function in real-time, there is a need for applying these new methods in evaluating the usability of these interfaces. Some of these methods apply to usability testing of real-time systems and some do not apply. Therefore it is necessary to develop new formal methods that can be adopted for usability testing of real-time networked multimedia systems.

2.8 Conclusion of Literature Study

As networked multimedia systems have evolved over the recent years, sophisticated multimedia applications have emerged. Through this, a revolution in the transmission of multimedia information over wired and wireless communication technologies has transpired. Society is now becoming more dependent on such technologies, which are used in almost every aspect of peoples' daily lives, including: communications, entertainment, education, marketing, research, health and medicine.

To provide the user with effective experience in using these networked multimedia applications, it is imperative that optimum Quality of Service (QoS) is delivered. A wide range of QoS communication protocols, architectures, frameworks and negotiation techniques have been developed for QoS management. The problem with most of these technologies is that QoS is handled primarily within the Application and/or Transmission Perspectives, without reference to the User Perspective. There is a lack of QoS integration and coordination between the User, Application and Transmission Perspectives.

It is evident that a holistic view of QoS is required, which integrates the user, application and transmission perspectives for managing QoS. Adopting the holistic QoS concept enables to develop a QoS management system that takes into consideration the user requirements, which can and then be mapped into the system's functionality for the delivery of QoS. Such a 'user-centred' QoS management system requires development of novel and intuitive interfaces, and better Human Computer Interaction (HCI) techniques to bridge the current gap between the user requirements and the system functionality. To ensure that these novel interfaces satisfy the user requirements, it is imperative that usability studies are performed on them.

Chapter 3

A Model & Taxonomy for QoS Management

Summary

This chapter introduces a Three Layer QoS (TRAQS) model that is based on three Quality of Service (QoS) perspectives (user, application and transmission) for the transmission of multimedia information. The three layers in the TRAQS model are called: User Perspective Layer (UPL), Application Perspective Layer (APL), and Transmission Perspective Layer (TPL).

A taxonomy for classifying applications and QoS parameters is defined and modelled as a QoS Parameter Architecture (QPA). The QPA categorises parameters into four taxas, namely: Independent Parameters, Dependent Parameters, QoS Profiles, and Application Profiles. These four groups are further subdivided into three performance aspects, where the relationship between these three performance aspects is defined in a Quality, Cost and Temporal Triangle (QCTT) Model. The QCTT depicts that it is only possible to achieve 'more desirable' parameters for two performance aspects, while the third is forced to the 'less desirable' state.

An Application Classification Model (ACM) is defined, which demarcates the spatial and temporal variations in multimedia content into four classes. The ACM enables the specification of hard and soft requirements of a multimedia application that assist in the process of adjusting parameters to compensate for the degradation in the provisioning of QoS. Task modelling TRAQS enabled to model the behaviour and the Human Computer Interaction (HCI) involved for static QoS management. This assisted the development of efficient methods and user interfaces for QoS management. Research presented in this chapter aims to bridge the gap between the user and the system for the seamless provision of QoS and QoS management.

3.1 Introduction

Interest in design and development for the provision of QoS for transmission of multimedia information has increased considerably over the recent years. Research bodies have developed QoS technologies that operate at the Application and/or Transmission Perspectives. Research has revealed a focus towards adopting a holistic approach towards QoS management of multimedia information transmission. Holistic QoS implies taking consideration of the User, Application and Transmission Perspectives. This approach requires QoS management of the various parameters that fall into each perspective and the intersection of two or more perspectives. Section 3.2 provides an overview of QoS perspectives.

A Three Layer QoS (TRAQS) model provides integration between the User, Application and Transmission Perspective Layers. This model provides the ability for non-technical users to negotiate QoS with each perspective layer via intuitive interfaces that operate within the User Perspective Layer [67]. Details of the TRAQS model are given in section 3.3.

To negotiate and manage QoS, this requires specification and manipulation of independent and dependent parameter values. For a non-technical user to request QoS, it is necessary to eliminate any technical aspects from the user. Currently there is no formal method that makes this possible. Section 3.4 and 3.5 solves this problem by providing a QoS Parameter and an Application Taxonomy depicted in a QoS Parameter Architecture (QPA).

Application Taxonomies provide the foundation specifying parameter values based on different categories of multimedia applications. An Application Classification Model assists to determine the hard and soft requirements for multimedia applications. This facilitates to adjusting particular parameter values according to the importance of the spatial and temporal variation of the multimedia application. An Application Profile Model facilitates non-technical users to specify QoS via a profile system. Section 3.6 provides details of using application and QoS profiles to request QoS.

QoS performance can be controlled by parameter values that are categorised into three aspects: quality, cost and temporal. Relationship between these performance aspects are bound by a triangular dependency. Section 3.7 depicts the stringent constraints between each performance aspect in a Quality Cost Temporal Triangle (QCTT) model.

User interfaces provide the ability for users to interact with a system. In order to develop efficacious interfaces for QoS management, this requires task modelling the user and system requirements. Section 3.8 gives details of modelling the functionality of the TRAQS model, which assists the development of user interfaces for QoS management. Section 3.9 comprises a conclusion for this chapter.

3.2 Quality of Service Perspectives

Multimedia transmission and its QoS can be categorised into three perspectives, i.e. User Perspective, Application Perspective and Transmission Perspective. As shown in Figure 3-1, some QoS parameters are related to more than one perspective where they fall on the intersection of two or more perspectives.

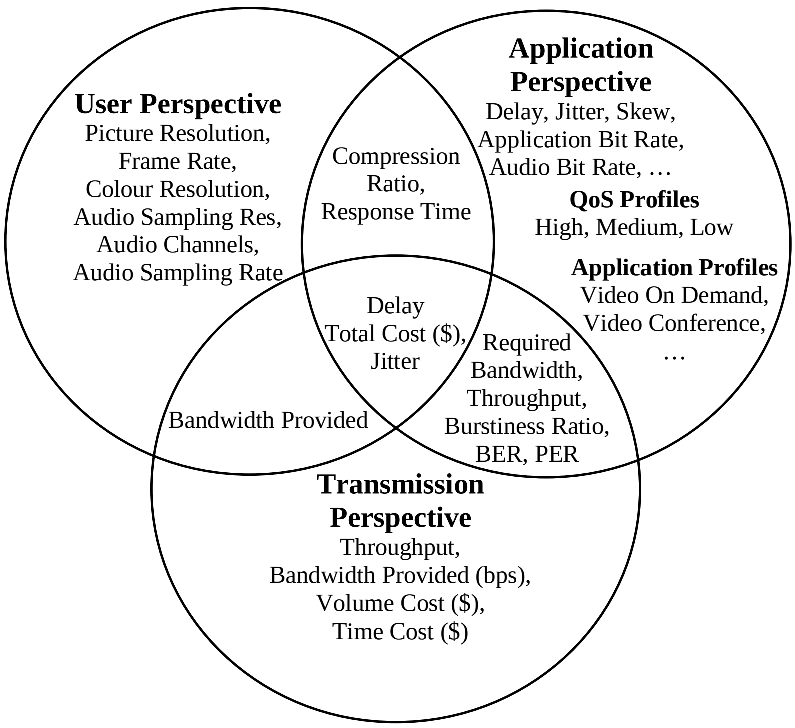


Figure 3-1: The Three QoS Perspectives Presented in a Venn diagram

Depending on the type of multimedia information transmitted, the functions of the three perspectives overlap for some QoS parameters. *Compression Ratio*, *Response Time*, *Delay*, *Total Cost ($)*, *Jitter*, *Required Bandwidth*, *Throughput*, *Burstiness Ratio*, *Bit Error Rate (BER)*, *Packet Error Rate (PER)*, and *Bandwidth Provided* fall on the intersection of two or more perspectives. These parameters are processed and managed by the intersecting perspectives.

3.2.1 User Perspective

The User Perspective relates to the quality output of multimedia presentations that are distinguished by human senses. In this perspective, quality of a multimedia presentation can be specified into parameter values such as: *Picture Resolution*, *Frame Rate*, *Colour Resolution*, *Audio Sampling Resolution*, *Audio Channels* and *Sampling Rate* etc.

A typical user would prefer to have the best-quality presentation that can be delivered by the system. Quality of a presentation reflects upon the cost to be paid for the type of services provided to the user. Therefore, users may want to also specify cost parameters for a particular service. If a user is unable to pay for a particular service, then they may want to decrease the quality to a level that they can afford.

Parameters such as: *Compression Ratio*, *Response Time* fall on the intersection of the Application Perspective and the User Perspective. Management of these parameter values may be required to be carried out by both perspectives. The *Bandwidth Provided* parameter falls on the intersection of the User and the Transmission Perspectives. *Delay*, *Total Cost ($)* and *Jitter* fall on the intersection of all three perspectives. Therefore, these parameters are managed by all three perspectives.

3.2.2 Application Perspective

The Application Perspective includes “parameters directly related to the performance of the application” [4]. The Application Perspective comprises the parameters: *Delay, Jitter, Skew* and *Error Rate*. These parameters affect the synchronisation of the transmitted multimedia content. The Synchronisation Accuracy Specification (SAS) defines acceptable level of synchronisation for transmission of multimedia information.

Other parameters for interactive multimedia applications such as *Gaming,* and *Virtual Reality* systems require crucial consideration of the *Response Time* parameter. Quality and performance of the presentation is dependant on other parameters included in the application perspective which are *Compression Ratio,* and *Response Time*. These parameters fall on the intersection of the User and Application Perspectives that require management by each perspective. *Required Bandwidth*, *Throughput*, *Burstiness Ratio*, *Transport Services*, *Bit Error Rate (BER)*, and *Packet Error Rate (PER)* fall on the intersection of the Transmission Perspective. *Delay*, *Total Cost ($)* and *Jitter* are covered by all perspectives.

QoS Profiles and *Application Profiles* reside in the Application Perspective as they are used to configure a set of parameters with a set of QoS values for a particular multimedia application.

3.2.3 Transmission Perspective

The Transmission Perspective includes parameters that are related to the transmission of multimedia information at the network level. These include Network Performance Parameters (NPPs): *Throughput*, *Delay*, *Delay Variance*, and *Error Rate*. Other parameters considered are, *Required Bandwidth*, *Throughput*, *Burstiness Ratio*, *Bit Error Rate (BER)* and *Packet Error Rate (PER)*.

The second set of parameters fall on the intersection of the Transmission and Application perspectives, thus, management can be carried out by either perspective. *Bandwidth Provided* intersects with the User and Transmission Perspectives, requiring coordinated management of this parameter by both perspectives. As mentioned earlier, *Delay*, *Total Cost ($)* and *Jitter* fall on the intersection of all perspectives. *Throughput*, *Bandwidth Provided (bps)*, *Volume Cost (in dollars)* and *Time Cost (in dollars)* reside within the Transmission Perspective, as they are directly related to the services provided by the network, through the network service provider.

3.2.4 Dependency between the Three QoS Perspectives

As explained before, some parameters fall on the intersection of two or more perspectives. This clearly indicates that these parameters could be managed by one or more perspectives in the QoS management system. The User Perspective is responsible for parameters that affect the quality of the visual presentation and thus takes direction from the user. However, the User Perspective also reflects on parameter management with the Application and Transmission perspectives.

The Application Perspective is responsible for parameters that affect the application quality and resources. Some of these parameters can be found in the User and Transmission Perspectives. The Application Perspective may also require managing these parameters that are reflected in the other perspectives. The Transmission Perspective is responsible for parameter management with the network services and takes consideration of parameter values that are also related to other perspectives.

As shown in Figure 3-1, *Delay*, *Total Cost*, and *Jitter* are the core parameters managed by the three perspectives (User, Application, and Transmission). *Compression Ratio* and *Response Time* are managed by the User and Application Perspectives. *Bandwidth Provided* is managed by the User and Transmission Perspectives. *Required Bandwidth*, *Throughput*, *Burstiness Ratio*, *Bit Error Rate (BER)* and *Packet Error Rate (PER)* are managed by the Application and Transmission Perspectives [4].

The three QoS perspectives and their overlapping function for parameter management can be depicted as a three layered architecture for QoS management. In section 3.3, the Three Layer Quality of Service (TRAQS) model is explained in further detail.

3.3 A Three Layer Quality of Service (TRAQS) Model

The TRAQS model is based on the three perspectives involved in multimedia communications. Each layer can be further partitioned into three main functions or modules: Assess, Map and Negotiate [4]. The A*ssess* module performs the function of assessing the QoS request within the specific perspective layer. The *Map* module performs the function of mapping the QoS request onto QoS parameters, and the *Negotiate* module performs parameter negotiation with other perspective layers. These functions can be implemented as software modules. The TRAQS model provides the foundation for building a QoS management system for multimedia communications. In this research, user interfaces that incorporate QoS negotiation and management facilities are developed based on the TRAQS model and concept.

3.3.1 Overview of the TRAQS Model

The TRAQS model (shown in Figure 3-2) comprises three layers for QoS management in multimedia communications. These three layers are: User Perspective Layer (UPL), Application Perspective Layer (APL) and Transmission Perspective Layer (TPL). As mentioned earlier, each layer performs QoS processing via three modules: Assess, Map and Negotiate.

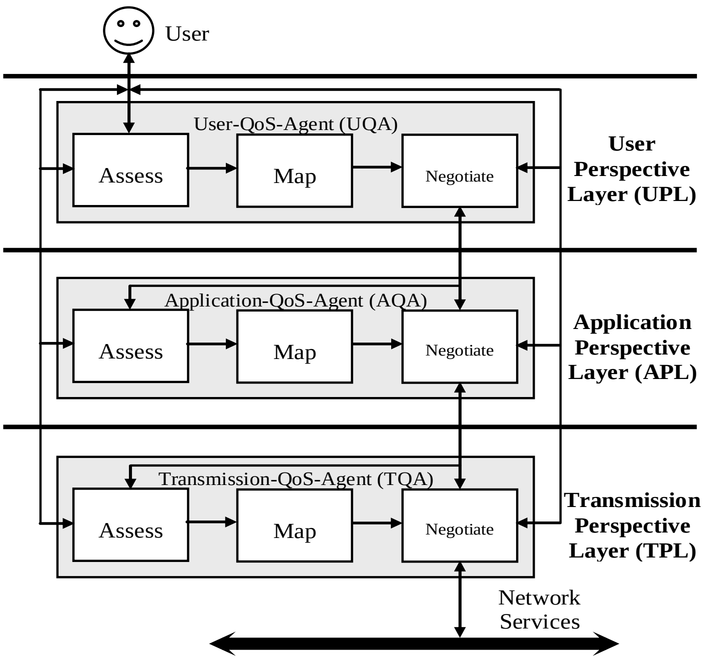


Figure 3-2: Three Layer QoS (TRAQS) Model [4]

Each layer performs QoS processing for a set of QoS parameters that are related to the particular perspective layer. However, depending on the multimedia content used for communication, some parameters can relate to more than one perspective layers. This leads to the notion that the functions of the three-perspective layers overlap. The main functions of the three perspective layers are:

- *User Perspective Layer (UPL)* interacts with the user and performs QoS negotiations with the user and the APL.

- *Application Perspective Layer (APL)* caters for the needs of the multimedia application.
- *Transmission Perspective Layer (TPL)* negotiates with the network to obtain appropriate communication services.

3.3.2 User Perspective Layer (UPL)

The User Perspective Layer (UPL) performs the main function of providing QoS to the user through QoS negotiations. The UPL receives the user requirements and performs the QoS negotiations with the user and the APL. The UPL enables technical and non-technical users to specify quality.

A technical user has privileges to specify individual parameter values, parameter rangers, or select parameters based on examples. For example, using the UPL, the user can specify a 640x480 resolution and 25 frames per second (fps) for a video stream by manual input, or the user can specify the parameters as the range between 640x480 & 20fps and 800x600 & 25fps, or the user can select the QoS based on three or more graphical examples.

A non-technical user can specify quality based on application and QoS profiles that configure parameters and their value based on the type of application used. For example, to specify high quality for a Video on Demand application, a non-technical user would be required to select the 'Video on Demand' application profile and a 'high quality' QoS profile. The APL would then map parameter values based on these profiles.

3.3.3 Application Perspective Layer (APL)

The Application Perspective Layer (APL) is related to the performance of the application. The APL considers the Synchronisation Accuracy Specification (SAS) factors, and other parameters that are related to the user and the network when performing negotiations with the UPL and the TPL. The parameter values can either be specified by the user, or obtained by other means through the UPL. These parameter values are then processed by the Assess and Map modules. The Negotiate module then performs the QoS negotiation with the TPL. For example, the APL differentiates between the parameters required for a Video Conference and those for a Video on Demand (VoD) session and can negotiate with the TPL to obtain the most suitable service.

3.3.4 Transmission Perspective Layer (TPL)

The function of the Transmission Perspective Layer (TPL) is to negotiate with the network to obtain appropriate communication services. The TPL receives QoS request parameters from the APL, or from the UPL via the APL, and processes these parameters through the Assess and Map modules. Finally, QoS negotiations between these layers and the network are performed to obtain the desired QoS. The TPL's QoS request parameters are split into two categories, hard requirements and soft requirements. Hard requirements must be satisfied, i.e. network must provide a guaranteed service. Soft requirements are to be met by using 'best-effort' services.

3.4 A Taxonomy for QoS Parameters

This section gives details of a taxonomy for categorising QoS parameters. This QoS taxonomy provides the foundation for building newfound methods to specifying quality for multimedia applications. Moreover, this taxonomy facilitates the development of user interfaces for QoS negotiation and management.

3.4.1 Categorising QoS Parameters into a Taxonomy

Currently, there is no standard for categorising QoS parameters in multimedia communications. To fill this gap, a formal approach is adopted in this thesis to categorise QoS parameters into four groups: *two parameter categories* and *two profiles*. The two categories are: *Independent Parameters* and *Dependant Parameters*. The two profiles are: *Quality of Service Profiles* and *Application Profiles*. These groups share three performance aspects, namely: *Temporal Facet*, *Quality Resolution* and *Cost Factor* [68]. This concept is depicted in a QoS Parameter Architectural (QPA) model, shown in Figure 3-3. Each group is denoted as a layer and each performance aspect as a segment running through the layers. These four groups and the three performance aspects are further described in the following sections.

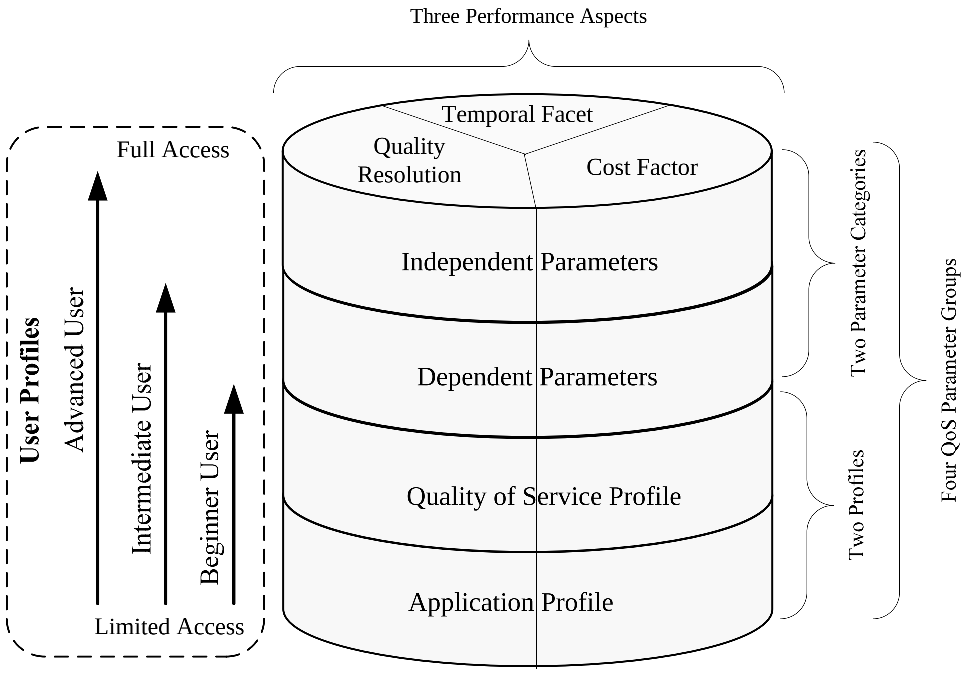


Figure 3-3: The QoS Parameter Architecture (QPA)

Independent and Dependent Parameters

The Independent Parameters category comprises of parameters that can be defined without any dependency on other parameters. These parameters can be specified by the user or can be defined through a Quality of Service Profile and/or an Application Profile.

The Dependent Parameters category comprises of parameters that are derived from the Independent Parameters. Some dependent parameters may be dependent upon some external aspects that are beyond the scope of this classification framework. Typical specifications of the Independent and Dependent Parameters are given in Table 3-1. For Example, in Table 3-1, the Jitter value type is specified in milliseconds, and the value can be greater than zero but less than the maximum acceptable jitter J_M. This parameter value is negotiated between the UPL, APL and TPL.

Table 3-1: Independent and Dependent Parameters Specification

Independent Parameters			
Parameter	Examples of Value Type	Constraints	Negotiating Entity
Jitter	Jitter = 0.01ms	$0<Jitter<J_M$	UPL, APL, TPL
Skew	Skew = 0.01ms	$0<Skew<S_M$	UPL, APL, TPL
PER	PER = 0.1%	0<PER<100	APL, TPL
BER	BER = 0.1%	0<BER<100	APL, TPL
Session Time (ST)	ST = 12:01:03	$0 < ST < ST_M$	TPL
Picture Resolution	Pic Res: X*Y X*Y= 640*480	176*144 <Pic Res < 1920*1080	User, UPL, APL
Frame Rate	fps = 25	1 < fps < 30	User, UPL, APL
Colour Res	bpp = 16 bits	8 < bpp < 32	User, UPL, APL
Audio Chan	Chan = 2	1 < Chan < 8	User, UPL, APL
Audio Res	ARes = 16 bits	7 < ARes < 16	User, UPL, APL
Audio Rate	ART=44.1KHz	8 < ART < 48	User, UPL, APL
Comp Ratio	CR = 50%	0 < CR < 100	User, UPL, APL
BW Provided	BWP=10Mbps	$0<BWP<BWP_M$	APL, TPL, ISP
Time Cost	TC = $1	$0 < TC < TC_M$	APL, TPL, ISP
Volume Cost	VC = $1	$0<VC<VC_M$	APL, TPL, ISP
Dependent Parameters			
Delay	Delay = 10ms	$0<Delay<D_M$	UPL, APL, TPL
ABR	ABR=15kbps	0 < ABR < X	APL
Burstiness	Burst = 2	$0<Burst<Burst_M$	APL, TPL
Throughput	TP = 15Mbps	$0 < TP < TP_M$	TPL
Required BW	RBW=9Mbps	$0<RBW<RBW_M$	APL, TPL
File Size	FS = 10MB	$0 < FS < FS_M$	APL, TPL
Total Cost	TC = $200	$0 < TC < TC_M$	User, UPL, APL, TPL, ISP

QoS Profiles

QoS Profiles are defined to provide a pre-configured set of QoS values according to the specific Application Profile. The Quality of Service Profiles category incorporates three profile parameters that define three different levels of QoS: High, Medium, and Low. The QoS Profiles configure the Independent and Dependant Parameters to a value that is related to the corresponding level of QoS for a particular Application Profile.

Application Profiles

A set of Application Profiles have been defined to provide a simple mechanism for specifying the values for each QoS parameters. By doing this it eliminates the need for the user to specify values for individual QoS parameters. Application Profiles configure and specify the acceptable value for each QoS parameter from a predefined set. In the first instance six Application Profiles have been defined that configure the Independent and Dependant parameters and their values for all subcategories to the most appropriate setting.

- *Chat* is a basic text messaging service that is used over communication networks.
- *Audio Conference* implies transmission of live audio between two or more communicating nodes.
- *Video Conference* implies transmission of live audio and video between two or more communicating workstations.
- *Near Video on Demand* is a service that provides video streaming but without VCR-like controls.
- *Full Video on Demand* is a service that provides video streaming with VCR-like controls.
- *Online Education/Presentation* comprises transmission of various multimedia objects; this includes text, audio, static graphics, graphic annotation and motion video.

Other Application Profiles – e.g. Virtual Reality – may be added as required in the future.

User Profiles

User Profiles are introduced to define the various methods a user can specify QoS which is linked to the technical sophistication of the user. User profiles filter the level of access at which different users with different technical competency can specify QoS. A technical user is given privileges to specify QoS for individual parameters. A non-technical user is limited to using the Application and QoS Profiles.

Three types of user profiles have been defined, they are: Beginner, Intermediate, and Advanced.

- *Beginner* is limited to specifying only by using the Application and QoS Profiles, they are not allowed to use technical aspects such as specifying individual parameter values for QoS.

- *Intermediate* can specify QoS by using the Application or QoS profiles. Users are also given the option to specify individual quality for audio and video via a slider adjust user interface element. Intermediate users are refrained from specifying individual parameter values for QoS.

- *Advanced* can specify QoS by using the (a) Application or QoS profiles, (b) set individual audio and video quality levels via the slider adjust user interface element, or (c) configure individual Independent and Dependent Parameters. This profile gives the user full access to the system to configure QoS.

User Profiles reside at the interface level and interact with the user and the UPL for QoS management. The Application and QoS profiles reside at the APL where they configure QoS for groups of parameters that are related to the particular application. User Profiles have been implemented in a user interface for static specification of QoS. Details for this user interface are given in section 5.3.3.

3.4.2 Quality of Service Performance Aspects

Earlier, three performance aspects were defined, namely; Temporal Facet, Quality Resolution, and Cost Factor. Each performance aspect includes parameters that affect presentation of multimedia information based on its characteristic. Quality Resolution comprises parameters that affect the quality of the presentation. Cost Factor includes parameters that relate to the costs and bandwidth for transmitting multimedia information. Temporal Facet includes parameters relate to timing and delays. The following sections provide further details of the three performance aspects and their relation to each parameter.

Quality Resolution

Quality Resolution comprises of parameters that are directly related to the specification of the output quality for the multimedia content. A user can define the parameters in this aspect directly, or by selecting a desired Application Profile and QoS Profile. A particular QoS Profile for a given Application Profile can be used to define the value for these parameters.

Independent Parameters

The independent parameters include parameters that specify the quality of the multimedia presentation. For a video content, it is possible to specify a:

- *Picture Resolution:* The horizontal and vertical pixels per frame.
- *Frame Rate:* Defines frames per second (fps).
- *Colour Resolution:* Bits per pixel (bpp).

Quality specification of audio content includes configuring the parameters:

- *Audio Channels:* The number of channels to be used for the audio stream.
- *Audio Sampling Resolution:* The number of bits used for storing the amplitude of the audio samples.
- *Audio Sampling Rate:* The number of samples taken per second for the audio stream.

For compressed media, it is possible to specify the compression ratio that affects the quality of the media. Applying a high compression can reduce quality using compression techniques. Decreasing the compression ratio will increase the quality and most likely use lossless compression techniques. The following parameter value can be specified to configure the quality of the multimedia presentation.

- *Compression Ratio:* The compression ratio for corresponding multimedia objects. Different values may be used for audio, video and still images.

Dependent Parameters

The dependent parameters for Quality Resolution are calculated based on the configured values for the Independent Parameters. Configuring these parameter values may accordingly change some Independent Parameter values. For example if the sum of the independent parameter values exceeds the limit applied to Application Bit Rate (ABR); then some downgrading of the Independent Parameter values is required in order not to exceed the ABR limit. The following Dependent Parameters are defined for Quality Resolution.

- *Application Bit Rate (ABR):* Specifies the rate at which the application generates information bits. ABR can either be constant, or variable. For *Constant Bit Rate (CBR)*, the Independent Parameters within the QoS Resolution aspect can be used to calculate the ABR. For *Variable Bit Rate (VBR)* (which is generated by compressed media), the calculated value for the ABR is dependant on the Multimedia Standard and the compression ratios.
- *Burstiness Ratio:* The traffic Burstiness generated by a compressed media stream. This value is calculated by dividing *Mean Bit Rate (MBR)* by the *Peak Bit Rate (PBR)*.
- *Throughput:* The effective rate of transmission of information bits. The throughput is calculated by dividing the number of information bits transferred by time taken to transfer bits.
- *Required Bandwidth:* The bandwidth required for a particular multimedia transmission/session.
- *File Size:* The size, in Megabytes (MB), for the stored multimedia objects.

Cost Factor

Cost Factor comprises of parameters that are related to the multimedia transmission costs over a communication network. There are two types of costs; 1. Bandwidth, 2. Monetary. The Service Provider, Application Profile and QoS Profile define the values for each parameter in this performance aspect. Descriptions of the Independent and Dependent parameters that are included in this performance aspect are as follows.

Independent Parameters

The Independent Parameters include parameters that are related to the bandwidth and service costs. These parameter values are defined by the service provider, but can be negotiated by the user.

- *Bandwidth Provided (bits per second (bps)):* The Bandwidth that the service provider delivers.

- *Time Charge ($):* The cost per unit of session time based on a particular bandwidth provided.
- *Volume Charge ($):* The cost per transmitted/received megabyte.

Dependent Parameters

The cost factors defined in the Independent Parameters calculate to a total cost for the service provided. This can be negotiated between the user and the service provider. Changes to the total cost can affect the Independent Parameters.

Total Cost ($): The Total Cost depends upon the particular cost model that the service provider employs.

Temporal Facet

Temporal Facet comprises of parameters that affect the timing relationships between related objects within multimedia presentations, and parameters that define the run length time for particular sessions. For a good quality presentation, the aim is to maintain accurate inter-stream and intra-stream synchronisation. A particular QoS Profile within an Application Profile determines the values for these parameters. Some of the important parameters are:

Independent Parameters

The Independent Parameters for Temporal Facet include the timing factors that affect the transmission of multimedia information and the duration of the multimedia session. Acceptable parameter values are defined for the parameters that affect the synchronisation of the multimedia streams. Session times are defined for the duration of the multimedia session.

- *Jitter:* The instantaneous difference between the desired presentation times and the actual presentation times.
- *Skew:* The average difference between the desired presentation rate and the actual presentation rate.

- *Packet Error Rate (PER):* The acceptable number of corrupted packets per unit time.
- *Bit Error Rate (BER):* The acceptable number of corrupted bits per unit time.
- *Session Time (hh:mm:ss):* The run-length time for a particular session in progress.

Dependent Parameters

The Dependent Parameters comprise of the accumulated delays that are calculated from the independent parameters. An acceptable total delay can be configured for the total delay dependent parameter. The limit applied to this parameter value can affect the Independent Parameter values.

- *Delay:* There are two types of delays: Response Time and End-to-End Delay. *Response Time* is the roundtrip delay encountered in an interactive activity. *End-to-End Delay* is the sum of all delays encountered in the processing performed on the end systems and the network.

3.5 An Application Taxonomy

In this section, an application taxonomy is defined which divides the various applications into four classes based on spatial and temporal frequencies; this forms the Application Classification Model (ACM). QoS management for different multimedia application requires specification of different sets of parameters. The selection of parameters depends on the type of multimedia application used and the type multimedia information transmitted. An Application Profile Model (APM) indicates the collection of parameter that are configured for each multimedia application. The value of these parameters depends on the type of QoS profile selected. These models are further explained in the following subsections.

3.5.1 Multimedia Applications

Multimedia applications use different multimedia content that varies in the spatial and temporal domains. The application and multimedia content determine the spatial and temporal frequencies.

For example, for a Video on Demand (VoD) application, the type of content viewed determines whether the spatial and temporal frequencies are high or low. Action movies are comprised of high spatial and temporal frequencies. The viewer would consider it important that the high quality audio and video not be interrupted. Therefore, priority must be given to maintaining the quality level for the spatial and temporal frequencies for this application.

A video conference application usually produces low spatial and high temporal variations. In most cases, users need high audio quality. Often video is less important if only a talking-head is being transmitted.

The type of multimedia application and multimedia content determine the spatial and temporal frequencies, and the importance of maintaining quality for each frequency domain is determined by the user. Nonetheless, there is a need for a formal method of categorising common multimedia applications and content into various categories in terms of spatial and temporal frequencies. This classification will assist the user in specifying the desired QoS.

3.5.2 Classification of Multimedia Information

The variations in multimedia content comprise two orthogonal aspects: spatial and temporal variations. From this, an Application Classification Model (ACM) is defined (shown in Figure 3-4), which delineates the spatial and temporal variation into high and low frequencies. The ACM classifies various multimedia applications into four classes:

- High Temporal and High Spatial Frequency (HTHS)
- Low Temporal and High Spatial Frequency (LTHS)
- High Temporal and Low Spatial Frequency (HTLS)
- Low Temporal and Low Spatial Frequency (LTLS)

Now, each multimedia application can be placed into one of these four broad classes.

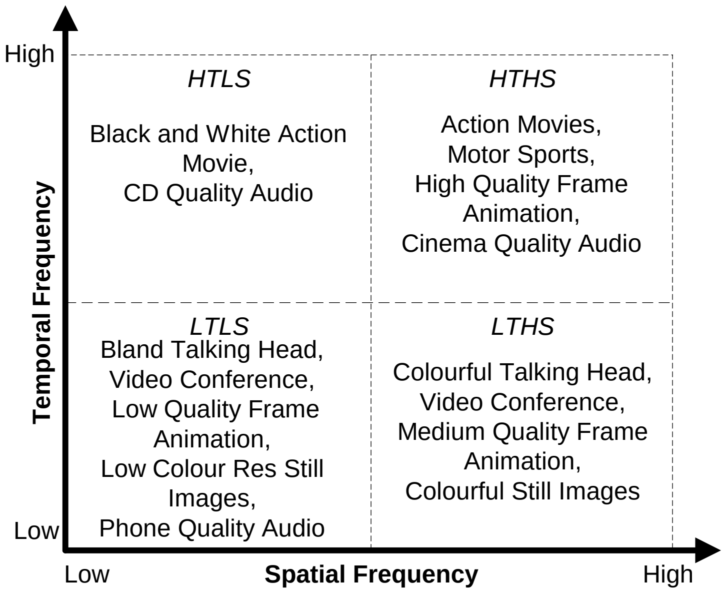


Figure 3-4: Application Classification Model (ACM)

The ACM can be used as a template for determining the important elements of the multimedia content/presentation and to allocate resources to cater for the demands of the application. Further description of these four classes is given in the following.

- *High Temporal and High Spatial Frequency (HTHS)* class denotes multimedia applications and presentations that have highly colourful content with fast action. For example: action movies, motor sports, high quality frame animation, and cinema quality audio.

- *Low Temporal and High Spatial Frequency (LTHS)* class denotes multimedia applications and presentations that have colourful content that does not change fast. For example: colourful talking head, video conference, medium quality frame animation, and colourful still images.

- *High Temporal and Low Spatial Frequency (HTLS)* class denotes multimedia applications and presentations that have low colour content with fast action. For example: black and white action movie and CD quality audio.

- *Low Temporal and Low Spatial Frequency (LTLS)* class denotes multimedia applications and presentations that have low colour content that do not change fast. For example: a bland talking head in a video conference, low quality frame animation, low colour resolution still images and phone quality audio.

The ACM concept is used to define the hard and soft QoS requirements of an application. If there is some degradation in the QoS, loss of quality of the presentation can be compensated by making adjustments, in the first instance, to the parameters that are defined in the soft requirements. However, if the degradation in the QoS is high to the point where it effects the hard requirements, then re-negotiating QoS is inevitable.

By using the concept of hard and soft requirements, the system can determine the quality facet that can be compromised. Hard requirements can not be compromised, as this will interrupt communication, while soft requirements can be comprised, if required. Implementation of the ACM will provide a QoS management system with the ability to give advisory feedback to the user when compromise in QoS is needed.

3.5.3 An Application Profile Model

Each application has a set of QoS parameters that are to be set to a specific value to provide the desired QoS. The Application Profile Model (APM) (Figure 3-5) provides a method for specifying the set of parameters required for successful operation of a given application. The APM includes the collection of parameters that are configured for each Application Profile and QoS Profile combination. For Example, the Chat Application Profile consists of the set of parameters that are indicated by a tick in Figure 3-5. In this case the Chat Application Profile can only control the Delay parameter value.

For the Delay parameter, the following values can be set according to the QoS Profile:

- High = 100ms, Medium = 1000ms and Low = 2000ms.

The APM assists in the parameter management and allocation of parameter values, which are dependent on the particular multimedia application used, and the type of QoS selected by the user. This model facilitates in developing a menu-driven interface used for specifying QoS via a Profile System for different multimedia applications.

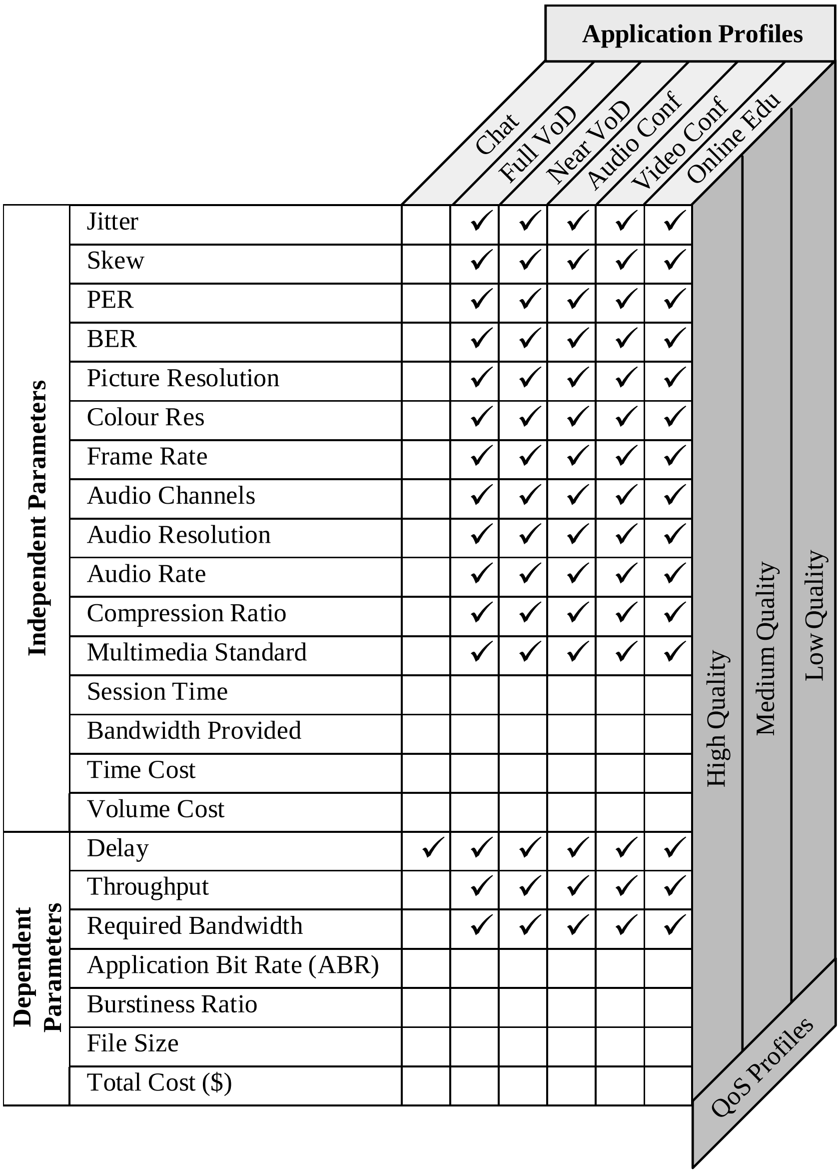


Figure 3-5: Application Profile Model (APM)

3.6 Specifying QoS with User, Application, and QoS Profiles

A User Profile defines the method at which QoS specification can be made based on the user's technical expertise. Intermediate and Advanced users are given privileges to specify QoS at a much more technical level. Application and QoS profiles provide non-technical users an easy method to specify a particular QoS required for a given multimedia application.

To specify QoS, the user initially selects their user profile. If the user selects a 'Beginner' user profile, then the system gives them *limited access* to specify QoS by allowing them to use only the profile system. The user is then required to select a specific Application Profile first, and then a particular QoS Profile. The system then configures the QoS parameter values according to the QoS Profile for the specified Application Profile.

'Intermediate' users are given *partial access* to the system, where they can either specify QoS using the profile system, or by selecting individual quality levels for audio and video. 'Advanced' users are given *full access* to the system, giving them the option to specify QoS either using the profile system, selecting individual quality levels for audio and video, or by configuring values for individual parameters.

User interfaces have been developed for static QoS management using the User, Application and QoS profiles. These user interfaces streamline the process for technical and non-technical users to request QoS. Usability studies have been performed for these interfaces in order to evaluate their efficacy from a user's perspective. Section 5.3.2 presents prototype interfaces for managing QoS, while chapter 6 presents the usability studies performed on these interfaces.

3.7 Relationship among the Three Quality of Service Performance Aspects

In almost every aspect of life, there are three related factors that affect the final outcome of achieving a desired goal. These factors include Quality, Cost, and Time. They form a triangular relationship where changing the value of one factor affects the state of the other two. It is possible to achieve a desired goal for two factors, whilst the third tends suffer. For example, take into consideration a motor vehicle company that is developing a new car. Reducing the costs and the development time will compromise the build quality of the car. To develop a car with a high quality for less time, will inevitably increase the costs. To develop a high quality car, but at the same time reduce the costs, will delay the development time due to lack of resources.

This same notion can be applied to negotiation of QoS for transmission of multimedia information. Each factor is viewed as a performance aspect that affects the quality of the presentation. The three performance aspects: Quality Resolution, Cost Factor, and Temporal Facet – are bound by the same triangular dependency that is modelled as a triangular relationship (as shown in Figure 3-6).

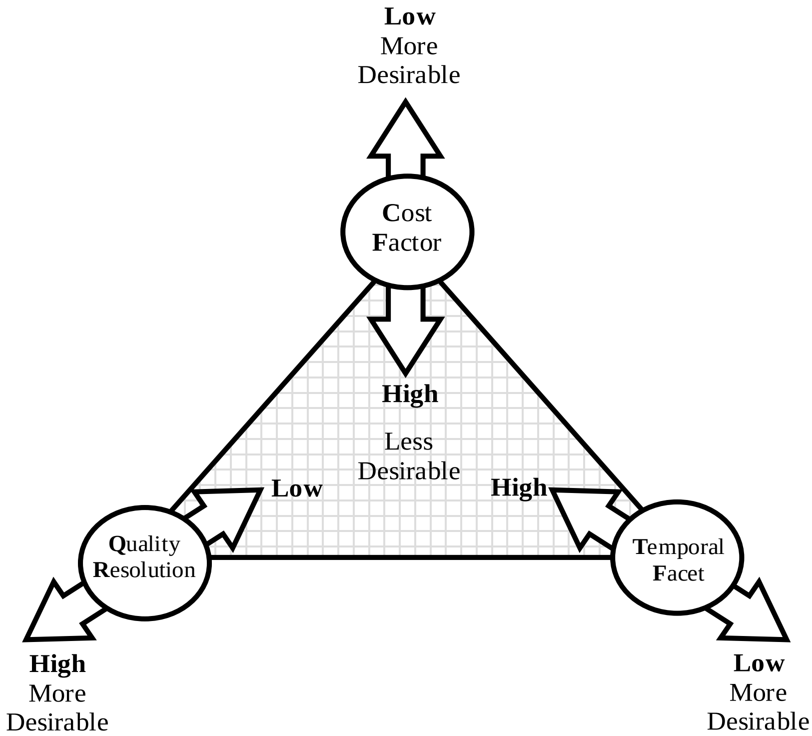


Figure 3-6: Quality, Cost, Temporal Triangle (QCTT) Model

This Quality, Cost & Temporal Triangle (QCTT) represents the dependency between these three performance aspects. The QCTT concept depicts inherent restrictions on the delivery of QoS. For the provision of QoS, it is only possible to achieve ‘more desirable’ parameter values for two performance aspects, whilst the third aspect is forced to the ‘less desirable’ parameter values.

In the following formal description of the QCTT model: High = H, Low = L, Cost Factor = CF, Temporal Facet = TF, Quality Resolution = QR, then the following assertions can be made, keeping in mind that while low CF and TF are more desirable, low QR is less desirable.

$$HQR \,\&\, LTF \rightarrow HCF \tag{3-1}$$

$$HQR \,\&\, LCF \rightarrow HTF \tag{3-2}$$

$$LCF \,\&\, LTF \rightarrow LQR \tag{3-3}$$

For example, the first predicate (3-1) states that if a user chooses to have high Quality Resolution and low Temporal Facet, the Cost Factor will have to increase. Similar concepts have been applied by the authors England and Finney in Multimedia Project Management [69].

By incorporating the Quality, Cost, Temporal Triangle (QCTT) model in a user interface, it provides the ability to dynamically manage QoS whilst a multimedia session is in progress. Further details of implementing the QCTT model in an interface for real-time QoS management are given in section 6.6.2. This user interface provides a novel approach for re-negotiating QoS in real-time.

3.8 Task Modelling the TRAQS Functionality

Task Modelling provides the ability to model the operation of a system based on the user's requirements [70]. This facilitates in developing user-centred systems as the requirements of the user are mapped into the system's functionality. Task Modelling assists in improving Human Computer Interaction (HCI) by modelling the system behaviour from the user's perspective. This guides the development of intuitive interfaces for different applications. Thus, it can be said that Task Modelling forms a bridge between modelling the user requirements and the application functionality.

In this thesis, Task Modelling is used to model the TRAQS functionality for managing QoS. Task models were developed for Static QoS Management and Dynamic QoS Management. *Static QoS Management* implies specifying and negotiating QoS in non-real-time prior to initiating a multimedia session. QoS negotiation can be user-initiated or system-initiated. *Dynamic QoS Management* implies re-negotiating QoS in real-time whilst a multimedia session is in progress. Re-negotiation can be initiated by the user or by the system.

Task Modelling the TRAQS functionality assisted in the development of interfaces for Static and Dynamic QoS Management. Chapter 6 presents the design of these interfaces and usability studies performed on them.

3.8.1 TRAQS Task Model

In order to develop a task model, it is necessary to determine the functional requirements for the system. In this case, the function requirements for the TRAQS model are based on Static and Dynamic QoS Management. Prior to developing the task model, certain questions require to be answered:

- What is required for this model to do?
- What set of functions does the model have to carry out?
- How will the model meet the functional requirements?
- How can it be verified and validated that the model behaves as intended, and with utmost efficiency?

The answer to these questions can be provided by Task Modelling technique, which allows translating functional requirements into sets of tasks and subtasks. Various options are investigated for defining the TRAQS model functional requirements as a task model. Using this task model, the internal communication between the system and the user can be modelled. The communication model for TRAQS determines how the user and the system interact for managing QoS [71]. The following sections provide an overview for Task Modelling technique, Concur Task Tree Environment (CTTE), and the TRAQS task model.

3.8.2 Task Modelling Technique

Task Modelling is a technique used to model the behaviour of a system from a user's perspective, thus enabling the requirements and actions of the system to be defined as sets of tasks.

The overall behaviour of the system is modelled as a scenario of tasks. Task modelling allows system designers to develop efficacious interactive systems by “focusing on the users and their tasks” [72]. Doing this makes it possible for the designers to consider and improve the Human Computer Interaction (HCI) aspects when designing the operation of a system [73].

Task models delineate the collection of tasks supported by an interactive system, their relationships and thus can be represented in diagrammatic forms, such as a hierarchy of tasks illustrated as a tree diagram. Generally, there are two types of task models, a User Task Model and a System Task Model. A *User Task Model* states the tasks to be performed by a system and thus consists of overlapping user scenarios [72]. Actors involved in a User Task Model are generally human; however it may include external systems and ‘the environment’. A *System Task Model* forms the basis for specifying a solution in the form of system requirements [72]. Actors involved in a System Task Model are generally subsystems, interfaces and/or humans. The TRAQS task model is created using a tool called Concur Task Trees Environment (CTTE).

3.8.3 Concur Task Tree Environment

The Concur Task Tree Environment (CTTE) is a Java Applet based tool, developed by the Human Computer Interaction Group – ISTI in Pisa [74]. CTTE provides the ability to build task models from a visual perspective. The user can define and structure the tasks in a logical fashion using a graphical editor provided in the tool. CTTE enables the user to focus on the activities of their model, thus allowing the user to identify the requirements of the model and organise them into a logical hierarchy of tasks and subtasks. A task model is presented using task objects and temporal operators that define the temporal transitions and paths for executing tasks.

CTTE provides the ability to build two types of models, Single User Task Models and Cooperative Task Models. A *Single User Task Model* is used to represent systems where a single user controls the operation or the behaviour of the system. A *Cooperative Task Model* is similar to Single User Task Model; however, it also includes tasks that are executed by two or more users. A set of tasks to be performed by a specific user is known as a role. Cooperative Task Models are also used to represent multi-user applications. The *Interactive Task Model Simulator* enables to walkthrough the task model operation and ensures it functions as originally intended.

3.8.4 TRAQS Functionality Requirements

Prior to determining 'how' the system should function, it is necessary to define 'what' the system should do. By doing this the functional requirements for the system are defined. A functional requirement describes a system's services or functions. For the TRAQS model some of the main functional requirements are as follows:

- Provide Seamless QoS for any Multimedia Application
- Provide pre-configured User, Application and QoS Profiles to the user, using the QPA conceptual architecture
- Provide a user interface for static QoS specification and negotiation
- Provide a facility for re-negotiating QoS based on the QCTT model.

3.8.5 TRAQS Communication Model

A Communication Model defines the interaction taking place between the user, the system, and internal components of the system. A Communication Model enables to define the various stages of operation for the system, and the interaction taking place for each stage. A robust Communication Model can streamline the process for Static and Dynamic QoS Management. In this research, a Communication Model has been developed for TRAQS. The aim of this Communication Model is to minimise the overhead for interaction between the user, the system and the internal system components. A five stage communication model has been developed, which represents the lifecycle of the TRAQS model for QoS management.

- *Stage 1 - Session Initialisation:* TRAQS components start-up the services, and wait for human input or instructions.
- *Stage 2 - QoS Negotiation:* This stage performs initial negotiation between the user and the TRAQS components to attain an agreement on the acceptable QoS.
- *Stage 3 - QoS Provision:* TRAQS components begin execution to deliver QoS.

- *Stage 4 – Dynamic QoS Management:* TRAQS performs QoS monitoring in real-time, thus continuously re-evaluating the services. Minor adjustments (re-negotiation) for the QoS can be made in real-time to compensate for any loss of resources. However, if there are insufficient resources to continue the multimedia session, the model then sends a signal to all components to return to Stage 2, where the user and system have to negotiate QoS again.
- *Stage 5 - Session Termination:* Initiates 'tear-down' process to free resources.

In order to develop the TRAQS Communication Model, consideration needs to be made on the tasks performed by each component in the system, and then determine the sub-division of its functionality. Other factors to be considered are the message interaction between the user and system.

3.8.6 TRAQS Task Definition

Task Modelling for TRAQS requires analysing the functional requirements, and defining sets of tasks to be performed. CTTE allows defining different types of tasks, such as: Abstract, User, Interaction, and Application tasks.

- *Abstract Tasks* define the set of subtasks to be performed at a conceptual level.
- *User Tasks* denotes the operation/tasks executed by the user.
- *Interaction Tasks* represent tasks executed for communication between the user and the system.
- *Application Tasks* comprises of tasks executed by the system or application entities in the process model.

A *Single User Task Model* was developed for the TRAQS model, which represents the overall processes for requesting, negotiating and re-negotiating QoS from the user perspective. Using the five-stage Communication Model presented earlier, a series of Abstract Tasks are formulated. Each Abstract Task contains its own collection of subtasks that perform the internal operation or specific function for the particular communication stage.

The operation of the TRAQS model is represented by five Abstract Tasks that relate to the five-stage Communication Model.

- *Stage 1: Session Initialisation* is responsible for the provision and initialisation of the system and its services, and represents the collection of tasks that perform QoS specification carried out by the user.
- *Stage 2: QoS Negotiation* represents the collection of tasks that perform the QoS negotiation carried out between the user, system and the service provider.
- *Stage 3: QoS Provision* represents the collection of tasks that supply the required QoS.
- *Stage 4: Dynamic QoS Management* represents the collection of subtasks that perform continuous QoS monitoring and control.
- *Stage 5: Session Termination* releases resources and performs other house-keeping tasks when a session ends.

At the next level, the model operation is described in further detail by using additional Abstract, User, Application, and Interaction Tasks. For brevity the most important tasks that represent the critical functionality of the TRAQS model are described.

For *Stage 1: Session Initialisation* Abstract Task, the following subtasks have been defined.

- *StartUp_Services* – This Application Task represents the function of loading the TRAQS user interface, application, and network services.
- *Display_User_Prof* – This Application Task performs the function of compiling the user profiles and have them displayed to the user.
- *Select_User_Prof* – This Interaction Task represents the user selecting their user profile.
- *Display_App_Prof* – This Application Task performs the function of compiling the application profiles and have them displayed to the user.
- *Select_App_Prof* – This Interaction Task represents the user selecting their desired application profile.

- *Display_QoS_Prof* - This Application Task performs the function of compiling the QoS profiles and have them displayed to the user.
- *Select_QoS_Prof* - This Interaction Task represents the user selecting their desired QoS profile.
- *Submit_Request* – This Interaction Task represents the user's selection being submitted into the system and then being processed.
- *Cancel_Request* – This Interaction Task represents the user cancelling the request and thus the system returns to idle state.

For *Stage 2: QoS Negotiation* Abstract Task, the following subtasks have been defined.

- *System_Admit* – Once the user submits a request, the system processes it, and if it can satisfy the requirements, it will 'admit' the request. This Abstract Task comprises of the set of tasks to be performed once the system admits the request.
    - *SystemAdmit:Display_Cost(s)* – Once the system has processed the user request, this Application Task returns a cost value.
    - *User_Accept* – This Abstract Task represents the user accepting the service provided and the system branching to the Stage 3 tasks.
    - *User_Reject* – This Abstract Task represents the user rejecting the request, the system then returns to Stage 1.
    - *Req_StepUp* – This Abstract Task represents the user increasing the QoS profile level if the system can provide higher QoS.
    - *Req_StepDown* – This Abstract Task represents the user decreasing the QoS profile level if the system is unable to provide the requested QoS.
    - *User_Cancel* – This Abstract Task represents the user cancelling their request which forces the system to return to *Stage 1: QoS Initialisation.*
- *System_Deny* - Once the user submits a request, the system processes it and if it can't satisfy the requirements, it will 'Deny' the request. This Abstract Task comprises the set of tasks to be performed, once the system denies the request. These subtasks are given in Figure 3-9 and follow the same method for the *System_Admit* subtasks.

- *System_StepUp* – If some resources are freed up and the system can provide a better QoS, then the system may ask the user to increase their QoS profile to a higher level. This Abstract Task comprises of the set of tasks to be performed once the system increases the QoS level. These subtasks are given in Figure 3-9 and follow the same method for the *System_Admit* subtasks.
- *System_StepDown* – If the services have degraded and the system cannot meet the request, it may decrease the QoS that the user has requested. This Abstract Task comprises the set of tasks to be performed as the system 'steps down' the QoS level. These subtasks are given in Figure 3-9, and follow the same method for the *System_Admit* subtasks.

For the *Stage 3: QoS Provision* Abstract Task, the following subtasks have been defined.

- *Execute_Resource_Reservation* – This Abstract Task comprises a set of subtasks that perform the network and system resource reservations. The subtasks included are as follows:
    - *Launch_Multimedia_Application* – Is an Interaction Task that loads the selected application profile from the application profile list.
    - *Apply_Application_Parameters* – Is an Interaction Task where the parameters are mapped and applied to the running application.
    - *Reserve_System_Resources* – This Interaction task performs the resource reservation on the local system, e.g. memory, CPU processing time etc.
    - *Reserve_Network_Resources* - This Interaction task performs the resource reservation on the network services through the service provider.
- *Execute_Multimedia_Communication_Session* – This Abstract Task represents the set of tasks that execute the multimedia communication session once all configurations have been completed by the previous communication stages.
    - *Establish_Session_Connection* – This Interaction Task performs the connection establishment between the user and the service provider.

- *Run_Multimedia_Application* – This Interaction Task executes the application and enables communication to take place.

For the *Stage 4: QoS Management* Abstract Task, the following subtasks have been defined.

- *Monitor_System_Services* – This Abstract task monitors the system services in real-time and detects any degradation in the service provided. Three degradation levels have been defined: High, Medium and Low Degradation. The selected parameters to be adjusted by the system are determined by the hard and soft requirements defined by the Application Classification Model (ACM). When in monitoring stage, the user is able to make parameter adjustments (re-negotiate QoS) using the QCTT model. The three degradation levels are defined as components of tasks, which are described in the following.
  - *SysServ:Degredation_low* – This Application Task is executed if the degradation in system services is low. Only minor adjustments are made to the parameter values to compensate for this, thus not requiring a re-negotiation of QoS.
  - *SysServ:Degredation_Med* – This Application Task is executed if the degradation in system services is medium. In this case, the user is prompted to make minor adjustments made to the parameter values and continue with the multimedia session, or the user can re-negotiate the services, which will interrupt the multimedia communication session.
  - *SysServ:Degredation_High* – This Application Task is executed if the degradation in system services is high and affect the hard requirements. The user is prompted to re-negotiate the services, which will interrupt the multimedia communication session.
- *Monitor_Quality_of_Presentation* – This Abstract Task monitors the quality of the presentation and detects any degradation in the quality. This task uses the QoS degradation detection and control method described in *Monitor System Services* Abstract Task.

- *Monitor_Network_Services* – This Abstract Task monitors the network services and detects any degradation in the QoS. This task uses the same QoS degradation detection and control method used in the *Monitor System Services* Abstract Task.

For the *Stage 5: Session Termination* Abstract Task, the following subtasks have been defined.

- *Session_Completed* – This Application Task represents a completed multimedia communication session.
- *Session Terminated* – This Application Task represents the function of the session being terminated.
- *Release_Resources* – This Application Task performs the process of releasing all reserved resources.
- *Exit_System* – This Application Task exits the system.

3.8.7 TRAQS Task Model in CTTE

The TRAQS model behaviour has been modelled in CTTE. The transition notations included in CTTE provide the ability to define the temporal relationships for each task and the execution sequence for the task model.

The temporal relationship notations in CTTE are described in Table 3-1.

Table 3-2: Temporal Relationship Notations in CTTE

Syntax	Notation	Description
T1 [] T2	Choice	A choice between two or more tasks.
T1 ||| T2	Interleaving	Independent concurrent execution of two or more tasks.
T1 >> T2	Enabling	Enables the proceeding task to be executed, once the preceding task has completed.
T1 []>> T2	Enabling with Information Exchange	Enables the proceeding task to be executed with information exchange, once the preceding task has completed.
T1 |> T2	Suspend/Resume	Enables the preceding task (T1) to be interrupted by the proceeding (T2) and then reactivate the preceding task.

The overall function of the TRAQS model, implemented as a Single User Task Model, is represented as a Task Tree Diagram in CTTE. The execution sequence for each task in the Task Tree Diagram is defined using the temporal relationship notations, presented in Table 3-2. The Task Tree Diagrams developed for the TRAQS model are presented in the following:

- *TRAQS Abstract Model* (Figure 3-7), presents the five communication signalling stages as Abstract Tasks where the completion of the former task enables the latter task with information exchange.

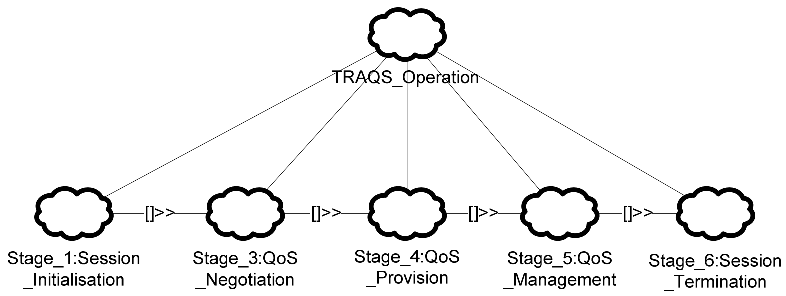


Figure 3-7: TRAQS Abstract Task Model

- *Stage 1: Session Initialisation* (Figure 3-8*)*, presents the sequence of tasks executed where the completion of some tasks enables the latter task with and/or without the exchange of information. This task model demonstrates the task sequence for a user specifying and requesting QoS via User, Application and QoS profiles.

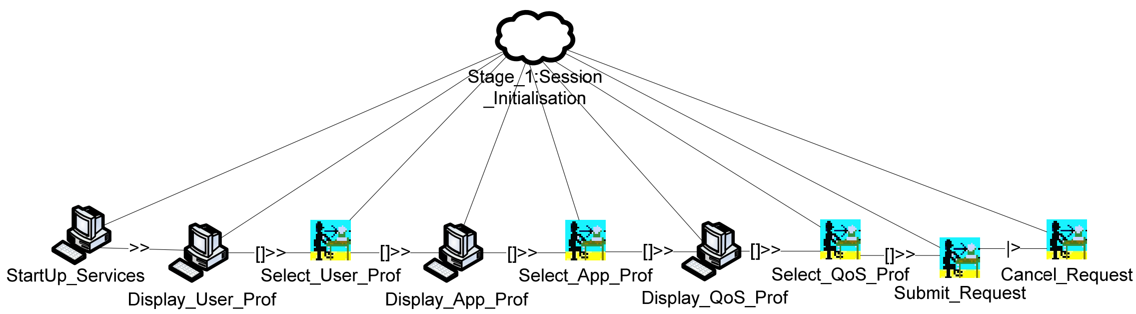


Figure 3-8: Stage 1 – Session Initialisation

- *Stage 2: QoS Negotiation* (Figure 3-9), presents the sequence of tasks to be completed when the system admits the user request for QoS. The choice notations demarcate the set of tasks to be completed based on the choices made by the user and the system.

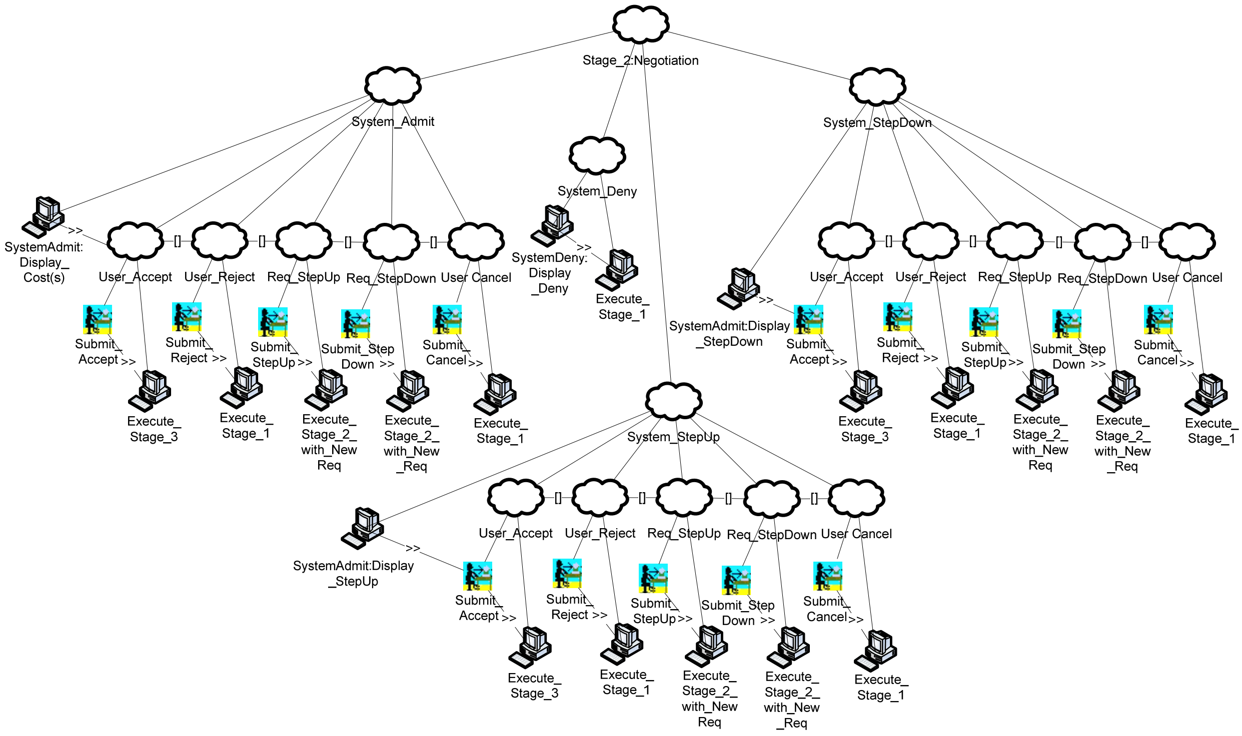


Figure 3-9: Stage 2 – QoS Negotiation

- *Stage 3: QoS Provision* (Figure 3-10), presents the sequence of tasks to be completed for QoS provision. The *Execute Resource Reservation* and *Execute Multimedia Communication Session* Abstract Tasks carry out the provision of QoS, where some of the subtasks are completed concurrently.

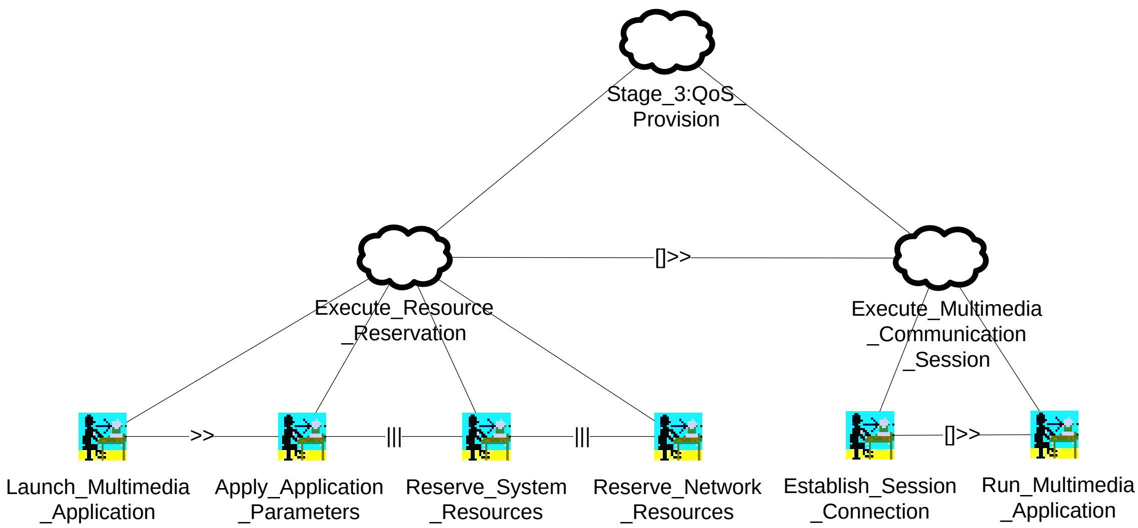


Figure 3-10: Stage 3 – QoS Provision

- *Figure 3-11 – Stage 4: Dynamic QoS Management*, presents the series of tasks to be completed for management of QoS in real-time. Dynamic QoS management is performed concurrently via the subtasks included in the *Monitor System Services*, *Monitor Quality of Presentation*, and *Monitor Network Services* Abstract Tasks.

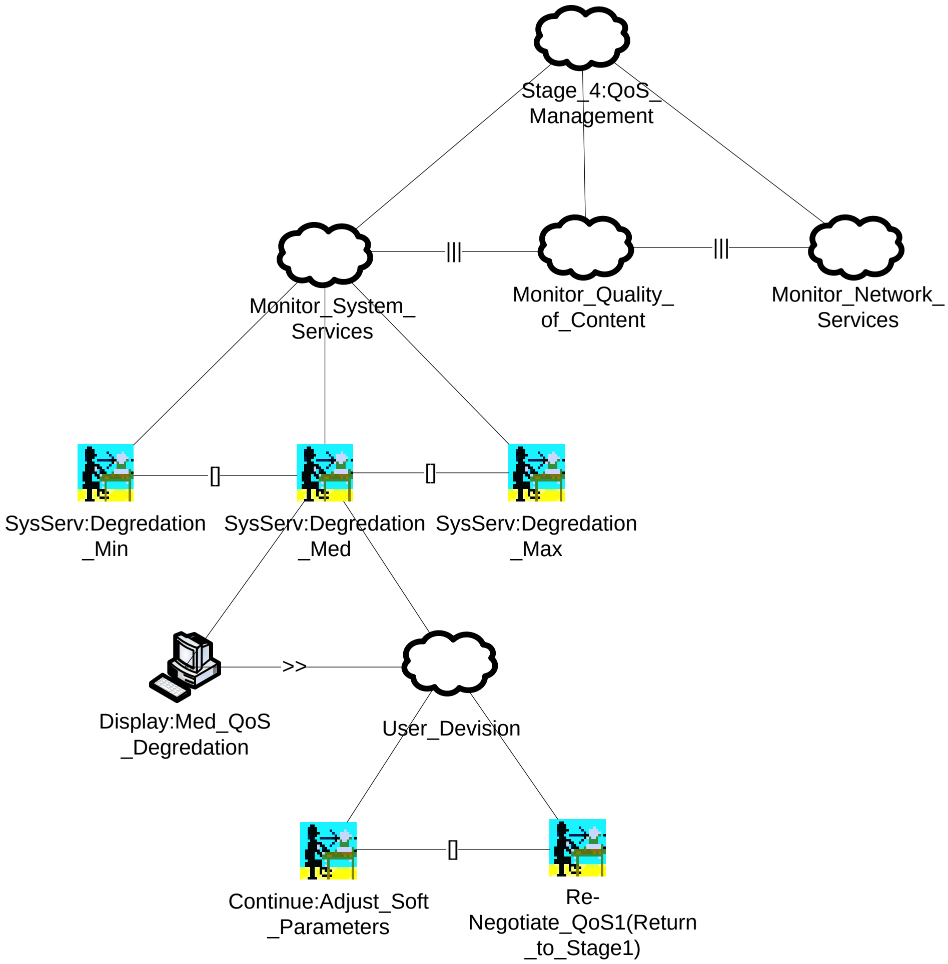


Figure 3-11: Stage 4 – Dynamic QoS Management (Re-negotiation)

- *Stage 5: Session Termination* (Figure 3-12), illustrates the task sequence for the session termination stage. Either a *completed* or *terminated* session enables the task to release the resources and exit the system.

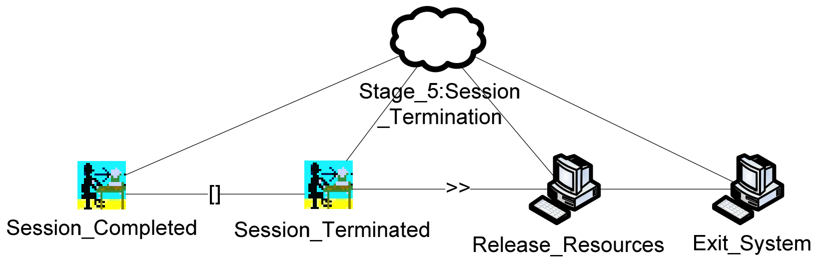


Figure 3-12: Stage 5 – Session Termination

3.8.8 TRAQS Task Model Simulation

The CTTE Interactive Simulator can simulate the behaviour of the model to validate and verify that the model performs its functions as originally intended. The simulator enables the user to execute a sequence of tasks in a discrete fashion, where the user interactively selects the tasks to be executed. The simulator allows one to develop and compare a series of task models, and determine the most efficient model. This can be used to select the communication model and the process used for QoS management. Consequently, it facilitates the design of the user interfaces for QoS management.

3.9 Conclusion

This chapter presented a QoS Management framework for multimedia communications. The Three Layer QoS (TRAQS) model provides the overall framework for QoS management, which integrates the functionality of the User, Application, and Transmission Perspectives. The QoS Taxonomy and the QoS Parameter Architecture (QPA) provide a formal model for categorising parameters, which assist the process of mapping user requests to parameter values. The Application Classification Model (ACM) facilitates the process for providing correct advisory feedback to the user for managing hard and soft requirements. The Quality, Cost, Temporal Triangle (QCTT) depicts the relationship between three performance aspects which are used for Dynamic QoS Management. The models presented in this chapter have not been implemented as a real-working QoS management system. Instead they are used as the framework and motivation for designing user interfaces for QoS management.

Task Modelling has led to the elaboration of the TRAQS model and determines the nature of its behaviour and the Human Computer Interaction (HCI) involved for the provisioning of QoS. The TRAQS Task Model informs the process for developing interaction methods for Static and Dynamic QoS Management. The models presented in this chapter are incorporated in the development of various intuitive user interfaces and interaction methods for QoS management. Development and usability investigations of these user interfaces are presented in Chapter 6.

Chapter 4
Overview of Usability Testing

Summary

This chapter introduces the reader to usability testing. Usability testing measures the quality of user experience in the development of different interfaces and applications. Usability Engineering considers the Human Computer Interaction (HCI) aspects for the development of a system and its user interface.

Usability testing of user interfaces requires: planning and preparing the experiment; recruiting participants; organising and configuring equipment; preparing and conducting the usability test; and collecting, analysing and reporting data. The usability engineering lifecycle is comprised of eleven stages that assist in developing ‘user-friendly’ applications. Empirical testing involves performing usability experiments with participants interacting with the interface.

To conduct tests with human subjects, it is necessary to consider human ethics. Ethics and code of conducts have been established, which require researchers to have respect, maintain trust and integrity for the participants.

Usability assessment methods facilitate in gathering participant feedback and recording experiment data. These methods include: Onsite Monitoring, Participant Feedback Questionnaire, Performance Measurement, Thinking Aloud Protocol, Interviews, Observation, and Heuristic Evaluation.

Usability laboratories provide a facility to conduct usability studies for different applications. A usability testing facility should be configured to cater for the type of application being tested. A fixed usability testing facility provides for testing desktop applications, and a mobile usability testing facility caters for testing portable applications.

Research in the usability testing facility configuration resulted in development of a streamlined usability testing process. The re-engineered usability testing process proved more efficient as it reduced the time taken to complete an experiment without affecting the experiment results.

4.1 Introduction

As computing technology proliferates in the modern society, people interact with computers and electronic devices regularly. These devices are being used in almost every aspect of our daily lives, including: communications, entertainment, education, marketing, research, health and medicine, etc. As society grows to be dependant on efficient and effective interaction with various computer/electronic equipment, it becomes essential that research and development caters for creating more usable/user-friendly interface systems for interacting with such equipment.

Usability testing encompasses different methods that measure the usability of interfaces. Usability engineering helps to improve usability of applications, and should be incorporated throughout the software development process. Usability engineering includes various activities, approaches, methods, and techniques for the development of user-centred applications. Section 4.2 provides details of usability engineering.

Usability assessment methods facilitate collecting feedback from participants and measuring their performance. The nature of the experiment and the type of feedback required determine the assessment method to be used. Section 4.3 gives an overview of the different assessment methods.

Usability studies are often carried out in a usability testing facility. The type of application being tested and its operating environment decide the test facility configuration, which may be fixed or mobile. Section 4.4 provides an introduction and configuration details of usability testing facilities.

To effectively study the usability of an application, it requires development of usability processes and re-configuration of the usability testing facility. The configuration of the usability testing facility assists in the development of streamlined testing processes. These processes should be time-efficient and less cumbersome, which in-turn, should not affect the experiment results. Section 4.5 presents an overview of a study for re-engineering traditional usability testing processes to facilitate experiments performed in this research, and section 4.6 includes conclusions for this chapter.

4.2 Usability Engineering

Usability engineering takes crucial consideration of Human Computer Interaction (HCI) and usability aspects for the development of a system intended for use by a user. Usability engineering incorporates various activities, approaches, methods, and techniques for the development of user-centred applications. These development aspects are all part of the development cycle for 'user-friendly' software systems. Usability Engineering involves specifying quantifiable measures of product performance, documenting them in a usability specification and assessing the product against them [66].

4.2.1 Usability Testing Processes

Usability testing processes comprise various stages that enable usability evaluation of user interfaces / applications to be carried out. A typical usability testing process comprises the following stages [66]:

- *Plan and Prepare* – Consists of preparing the usability investigation goals, aims, objectives, planning for the experiment, and describing the type of participants to be used.
- *Recruit Participants* – Involves recruiting participants based on a set participant background, knowledge, and experience criteria.
- *Organise and Configure Equipment* – Involves configuring the usability laboratory equipment.
- *Prepare the Usability Test* – Consists of preparing test scenarios, tasks, data collection methods, develop usability assessment and evaluation methods. Conduct heuristic evaluation to eliminate flaws in the application, testing process, and administration of information collection.
- *Conduct Usability Tests* – Involves performing the usability experiments with selected participants.
- *Collect and Analyse Data* – Comprises collecting, transcribing, collating, and performing analysis of data.

- *Report Analysis* – For 'industry' usability studies, this requires preparing a technical document that comprises an analysis, conclusions and recommendations for improving the usability of a particular application. For 'research' usability studies, collection and analysis of data should be included in a thesis or research report document.

Various aspects of the usability testing process that require critical analysis include: Usability Testing Life Cycle, Empirical Testing, Ethical Aspects, Choosing Participants, Test Scenarios, and Test Tasks. These aspects of the usability testing process are elaborated in the following subsections.

4.2.2 Usability Engineering Life Cycle

The usability engineering life cycle facilitates the process of developing more usable systems and should take place throughout the development of an application. This life cycle comprises eleven main stages, which are listed in Table 4-1. All eleven stages of the usability life cycle need not be applied to all usability investigations. The application type and the style of the usability study determine which stages should be used.

Table 4-1: Stages of the Usability Engineering Life Cycle [64]

Usability Engineering
1. User and Application Characteristics o Individual User Profiling o User Activity and Task Analysis o Application Analysis o User and Application Evolution 2. Competitive Analysis 3. Usability Goal Setting 4. Parallel Design 5. Participatory Design 6. Coordinated Design of the Total Interface 7. Apply Guidelines and Heuristics Analysis 8. Prototyping 9. Empirical Testing 10. Iterative Design 11. Collect Feedback from Field Use

1. *User and Application Characteristics* requires the developer of the system to understand the user, their requirements, knowledge of the system, experience, and other relevant background information. This is formally categorised into the following:
    - *Individual User Profiling* considers the user's individual qualities, such as work experience, education level, age, previous computer experience etc.
    - *User Activity and Task Analysis* involves modelling the goals, information needs and tasks that the user is required to perform.
    - *Application Analysis* involves considering the functions that the system is required to perform based on the task analysis. *Functional Analysis* enables to optimise the system to cater for the needs of the user. The necessity and importance of each task should be verified by the system developer.
    - *User and Application Evolution* considers the user and their job function. Users continuously find new ways to use an application. This leads to using an application beyond its original capabilities. Eventually, the application evolves and is used for completing more tasks than it is designed for. For example, spreadsheets were originally designed and used for calculations; users now have improvised and use spreadsheets as databases too. Therefore, when developing a system, some consideration of a flexible design needs to be taken into account to enable future changes to be made with ease.
2. *Competitive Analysis* involves using competitor's products to carry out a usability test. This enables one to determine the qualities and drawbacks from their product and use this information in designing a better system. Competitive analysis is mainly used in industry for product development.
3. *Usability Goal Setting* requires setting usability goals such as Target Task Completion Time and Target Error Count. This enables one to determine the acceptable and unacceptable values for both criteria. It also shows how usable the system is.

4. *Parallel Design* is mainly used in large organisations where various teams work on different designs of the same product. The best design is selected for implementation.

5. *Participatory Design* involves the user participating during the design stage of the system. This enables closer examination of the user requirements. This enables the developers to design a system that better models the user.

6. *Coordinated Design of the Total Interface* is an important aspect of user interface design, where the consistency of the entire interface must be maintained. This includes the formatting, themes, and structure of the interface. Consistency of the total interface can be maintained by code sharing, guideline preparation, standards and/or assigning a team that specifically focuses on the user interface design.

7. *Apply Guidelines and Heuristics Analysis* provides advice about the usability characteristics of the user interface. Different levels of guidelines can be developed through these characteristics. Some of these are: general guidelines (which apply to user interfaces), category-specific guidelines (focused towards the kind of system being developed), and product specific guidelines (focused towards the individual product). An example of a guideline is that the system must provide feedback to the user about any critical errors encountered. Heuristic analysis enables to enhance the user interface and eliminate any problems encountered.

8. *Prototyping* implies that developers prepare a trial product throughout various stages of the software development process. Prototypes can be tested with the user in order to assure that the system is satisfying the user's requirements. Two dimensions of prototyping are identified:

    - Horizontal Prototyping: maintains the features but eliminates depth of functionality.
    - Vertical Prototyping gives full functionality for a few features.

    Nielsen states that "scenarios are the ultimate minimalist prototype in that they describe a single interaction session without any flexibility for the user" [64].

9. *Empirical testing* involves pilot testing the system. A usability study is performed with the participant in a usability laboratory. User assessment is performed via participant feedback questionnaires. Empirical testing enables one to determine usability problems that were not encountered in the heuristic evaluation and their severity.

10. *Iterative Design* involves identifying the usability flaws within the user interface after conducting a usability test, then using this information to improve the user interface and develop a new version. Further usability tests and improvements of the new design can be repeated and carried out as many times.

11. *Collect feedback from Field Use* involves collecting feedback from users via questionnaires once the system has been deployed in the intended operating environment. Usability problems can be overcome by the developer tweaking the software and eliminating any bugs.

The usability engineering lifecycle presented in this section is a general specification of the steps to be taken when developing a system. Due to the nature of the system and its use, some of the stages in the lifecycle can be modified to suit different development processes. The stages presented in this section are designed for product development and evaluation for the commercial industry. Usability studies performed for research purposes require a different approach, in which some new stages may require to be introduced.

4.2.3 Empirical Testing

Empirical Testing involves conducting usability experiments with participants interacting with the user interface. Empirical testing is usually performed in a controlled environment such as a usability laboratory. Empirical testing involves monitoring the user’s actions and behaviour towards the system. Within such a controlled environment, various usability inspection methods can be used in order to evaluate the usability of the use interface. These methods include [65]:

- *Heuristic Evaluation* involves specialists judging and evaluating each element of the user interface for its usability.

- *Cognitive Walkthroughs* uses an explicitly detailed procedure, such as a scenario of tasks, to simulate the user's action in solving a problem.
- *Formal Usability Inspections* employ a six-step procedure with strictly defined roles to combine heuristic evaluation and a simplified form of cognitive walkthroughs.
- *Pluralistic Walkthroughs* involve *users*, *developers*, and *human factors people* collectively to step through a scenario and discuss each aspect and element of the user interface.
- *Feature Inspection* involves listing a sequence of features to be evaluated and requesting the users to complete a set of tasks that they would not naturally perform. This method requires the user to have extensive knowledge of the system, as they will be using features that are not generally used by a layperson.
- *Consistency Inspection* involves designers representing multiple projects to inspect the interface and ensure that it operates in the same fashion as their own designs. This ensures consistency of the user interface across multiple products from the same development effort [65].
- *Standards inspection* involves a specialist to inspect the user interface and evaluate whether the interface complies with guidelines and standards.

The empirical testing methods listed above are mainly used in industry for usability studies of commercial products. All of these methods do not apply for 'research' usability studies. A combination of Heuristic Evaluation, Cognitive Walkthrough, Formal Usability Inspection, and Feature Inspection testing methods were used to conduct the usability studies on the QoS management interfaces developed in this research.

4.2.4 Ethical Aspects of Tests with Human Subjects

Ethics are "beliefs regarding right and wrong behaviour where society forms a set of rules that establishes the boundaries of universally accepted behaviour. These established rules form the moral code which society lives by" [75]. Individual's ethical behaviour and values are developed and influenced by a combination of family, influences, life experiences, education, religious beliefs, personal values, and peer influences. It is important that an individual's principles and integrity are not violated [75].

Ethics and code of conducts have been established for almost every aspect in society, such as corporate, business, science, research, education and computing environments etc. Many ethical issues that can arise in computing context relate to creating and marketing applications, as well as in societal and research processes [76]. Research ethics apply where investigations involve human subjects. Research ethics require that researchers have respect for participants and that trust and integrity are maintained [75].

As usability investigations involve human users, it is crucial that all ethical aspects be considered. It is necessary to evaluate whether the usability experiment will pose any risks to the participant, such as: physical, psychological, social or legal. Furthermore, it is important to identify any other risks that may affect the user. If such risks are identified, it is important to assess the level of risk the participant is placed at, and evaluate the potential benefits to the participants or contributions to the body of knowledge, which must outweigh the risks.

To perform usability studies with participants in a university, it requires ethics clearance. Ethics clearance was requested and approved for conducting the usability studies presented in this thesis.

In usability experiments, it is essential that the participant is made to feel they are in a comfortable environment; otherwise factors such as discomfort may affect one's performance in the experiment. It is therefore important to create a comfortable working atmosphere by using less intrusive monitoring equipment. Ideally, the usability laboratory should be setup to appear as their normal working environment. Investigators should have everything prepared and should not keep the participant waiting, as this may discourage them.

Participants should be given access to light refreshments as it will help them feel relaxed throughout the experiment. The working environment should reflect how participants normally work. The investigator should ensure that the participant is comfortable and should assist them if necessary to create the appropriate environment.

The experiment should not be interrupted. When the experiment is complete, the investigator should end it on a positive note and inform the participant that they were very helpful.

4.2.5 Choosing Experimenters and Participants

It is necessary that in a usability experiment, appropriate experimenters and participants are selected. The experimenter's task is to supervise the usability test. It is preferred that the experimenter have some experience in the usability testing process and/or using the system as they will be required to resolve any problems faced with the system.

The experimenter should know what decisions to make in problematic situations. More than one experimenter can be involved in a small-scale usability experiment. One experimenter can serve as a facilitator to provide the participant with assistance, and the other can serve as an observer to monitor the experiment.

It is recommended that the system designer not be involved as an experimenter, as they may have the tendency to assist and explain too much about the system. This will affect the experimental results. The experimenter only needs knowledge of how the system works rather then its implementation details [64]. To avoid such problems, it is recommended that developers observe, while a usability specialist handles the relations with the participants [77].

Selection of participants can range between Beginner and Advanced. The following two methods can be used for conducting usability tests.

1. *Use a different test for Beginner and Advanced participants:* This method is employed when the user interface is developed for specific type of users. A Beginner user makes basic use of the system, while an Advanced user uses it to complete complicated tasks. The system must then be tested for both scenarios. Participant feedback is used to improve the usability of the system for completing basic and advanced tasks.

2. *Use the same test for both Advanced and Beginner participants:* This method is employed when the user interface is developed for general purpose use. Participant feedback can be used to see how their expertise relates to the usability of the system.

This research employs the second test method. Participants with different computing expertise are selected to evaluate user interfaces for QoS management. Nielsen states that testing five users would reveal an average of 85% of the usability problems [64]. Generally, ten participants were selected for each usability study during this research.

4.2.6 Test Scenarios and Tasks

Usability test scenarios and tasks enable the experimenter to evaluate a particular feature or aspect of the system. A scenario is used to provide the participant with background information of the test, which describes the situation that the user is faced with when completing the tasks. A scenario should be described in such a way that it presents the user interface's goal, the user's role, and the tasks required to achieve the goal. The scenario must be as close to the real environment as possible.

Test tasks should be designed in such a way that they enable thorough evaluation of the system's features. The test tasks should precisely specify the result the user is to produce and the goal that is required to be achieved. This can be achieved through task modelling the user's duties. Test tasks should not be complicated or trivial. It should be possible to complete each task in realistic time.

4.3 Usability Assessment Methods

Various assessment methods are used in usability studies to record different type data from the experiments. The assessment methods used in the usability studies performed for this research are: Onsite Monitoring, Participant Feedback, Task Performance Measures, Usability Evaluation Criteria, Thinking Aloud Protocol, Interviews, Observation and Heuristic Evaluation. These methods are further described in the following.

4.3.1 Onsite Monitoring

Onsite monitoring involves the facilitator monitoring the participant's actions and errors during the experiment from within the participant room. An observer sits in the adjoining observation room and monitors the participant's responses, while managing the monitoring system. Usability tests conducted in this research employed a facilitator to record participant error logs in real-time and the observer to note task completion times.

The usability laboratory configuration used in this research included facilities for video recording the participant from four different views:

- Face (front)
- Face (side profile)
- Hands
- Screen capture.

The video recordings of each experiment were used to authenticate participant feedback, and the observer's and facilitator's notes.

4.3.2 Participant Feedback Questionnaire

Feedback Questionnaires provide qualitative and quantitative data about the participant's attitude towards the system, knowledge of the system (mental model), level of satisfaction, and suggestions about particular aspects that can be improved [78]. Feedback questions are based upon standard usability criteria, such as: Ease of Use, Learnability, Memorability, Perception, Comprehension, Usefulness, Effectiveness, Efficiency, Satisfaction and Stress factor.

The post-questionnaires used in the usability studies for this research employed a five point rating system similar to the Likert Scale [64]. Some questions required simple ‘yes’ or ‘no’ answers. An example of the rating system used is given in the following.

- 1 = very poor,
- 2 = poor,
- 3 = neutral,
- 4 = good
- 5 = very good.

The usability for the different interfaces was evaluated based on the ratings participants gave for the selected criteria.

4.3.3 Usability Evaluation Criteria

Usability criteria were formulated based on the main goals of usability from a user’s perspective, these goals are: Ease of Use, Learnability, Memorability, Perception, Comprehension, Level of Distraction, Usefulness, Effectiveness, Efficiency, Satisfaction, and Stress Factor. The goal and nature of the experiment determine the set of criteria to be used. Further details for each criterion are given in the following.

Ease of Use – Eason (1988) defines Ease of Use as, “the degree to which users are able to use the system with the skills, knowledge, stereotypes and experience they can bring to bear.” [79].

Learnability – (also known as Ease of Learning) delineates how easy it is for users to accomplish basic tasks the first time they encounter the user interface [64].

Memorability – refers to the degree at which users can re-establish proficiency when returning to use the interface after not using it for a certain time. Improving the learnability for an interface makes it easier for users to remember how to use it.

Ease of Perception – defines how easy it is for users to perceive the objects, events, sounds and information displayed in the user interface. Roth (1986) refers perception to how information is acquired from the environment, via different sense organs (e.g., eyes, ears, fingers) and transformed intro experiences of objects, events, sounds, and tastes [80].

Ease of Comprehension – defines how easy it is for users to understand the information presented in the user interface.

Level of Distraction – measures the extent which users feel distracted by the interface when carrying out multimedia communications.

Usefulness – is a measure of whether the system can achieve a desired goal [81].

Effectiveness – refers to “how good a system is at doing what it is supposed to do” [66]. In other words, effectiveness is the accuracy and completeness with which users achieve certain goals [81].

Efficiency – refers to the way a system supports users in carrying out their tasks [66]. Efficiency can also be defined as the relation between the certain goals and the resources expended in achieving them [81].

Distinguishing Usefulness, Effectiveness and Efficiency

Usefulness can be very easily mistaken for effectiveness and efficiency, as there is some correlation between these aspects. To clarify the distinction between each aspect it can be said that, an item is useful for a task if it can accomplish the same; even if it is not effective or efficient in doing it.

For example, a pencil is useful in writing on paper, but not for writing on a whiteboard. However, a pen is more effective for writing on paper, as it produces more striking effect than the grey lead pencil. Something is more efficient if it takes less effort or time in accomplishing the task. You may be able to write on a whiteboard with a fine tip felt pen, however, the whiteboard marker is not only more effective but also more efficient. Because, with a fine tip pen you will have to overdraw many times to produce a thick line, where as, with a marker it will be done with one stroke.

Thus:

1. A pencil is useful and efficient for writing on paper, but not effective. It is useless for writing on a whiteboard.
2. A fine tip felt pen is useful, effective and efficient for writing on paper. It is useful for writing on a whiteboard, but not effective or efficient.
3. A marker is useful, effective and efficient for writing on a white board.

Satisfaction – refers to the participant's comfort and attitude towards the use of the user interface [64]. Satisfaction can be determined subjectively by asking the participant for their opinion with respect to this evaluation criterion. Satisfaction is an important usability attribute for user interfaces that are used in a non-work environment, such as education, entertainment and edutainment [4].

Stress Factor – refers to the participant's discomfort and negative attitude towards the use of the user interface. Stress Factor (also known as User Cost) measures the user's state of mental, emotional strain, or suspense developed by using the user interface [82]. Stress can be measured psycho-physiologically such as Electroencephalograms (EEG), pupil dilation, heart rate, skin conductivity, blood pressure and level of adrenaline in the blood [83]. However, such methods as these are intrusive and intimidating to the user, which may result in the user not feeling relaxed, and therefore affect the results.

4.3.4 Task Performance Measures

Task Performance Measures include Task Completion Time and Error Count.

Task Completion Time implies the time taken to complete a task, or a set of tasks. In this research, Task Completion Time is used as an indicator of the overall usability of the user interfaces. *Response Time* is another measure that refers to the time taken for the user to respond to system prompts. This measure is useful for assessing the effectiveness of the interaction methods used for system feedback.

Task Error Count implies the number of errors a user makes per completed task, or a set of tasks. An error is considered when a participant performs an action not specified in the scenario of tasks. The number of errors a participant makes per task indicates the difficulty they experience in using the interface to complete the task. A high Task Error Count implies poor usability for the user interface.

4.3.5 Thinking Aloud Protocol

Thinking Aloud Protocol is a technique that encourages participants to vocalise their thoughts, feelings, and opinions for each task during testing. This technique should be encouraged throughout the experiment as participants tend to forget to vocalise their thoughts. By doing this the observer and facilitator can understand how the participant approaches the user interface, their continuous opinion and the mental model they build in their mind of the user interface. In this research, Thinking Aloud Protocol is mainly used to improve the user interface by identifying any major flaws that the participant indicates.

4.3.6 Interviews

Participant Interviews enable to query participants about their experiences and fondness of certain aspects in the user interfaces and the system. Participant Interviews should be structured and require direct interaction with the participants. Participants can state their opinions, experience, and attitude towards the usability test. Participant Interviews are usually carried out at the end of the usability test. In this research, participants were interviewed at the end of each usability test; they were informal and facilitated in supporting feedback given in the participant feedback questionnaire.

4.3.7 Observation

Observation involves watching and listening to the participants while usability test is in progress. Participants' behaviour can be recorded using an onsite monitoring system within a usability testing facility. Facilitators and observers can observe the participant's actions in real-time while the usability test is in progress and by reviewing the video record. Observation can also be performed in a field environment (natural), where by the observer can be a participant observer looking on the outside spectrum of the usability test. In this research, observation was performed by the facilitator and observer within a usability testing facility.

4.3.8 Heuristic Evaluation

Heuristic Evaluation is a usability inspection method that involves usability specialists, developers, and/or system testers to perform an 'in-house' usability test on the system. Specialists can eliminate any major flaws or usability issues within the system prior to conducting a usability test with participants.

For an effective heuristic evaluation, the following ten usability heuristic principles must be satisfied [84]:

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation

For this research, heuristic evaluations were performed for each user interface. This enabled to eliminate any flaws and usability issues for each interface.

4.4 Usability Testing Facilities

Purpose-built usability testing facilities (laboratories) are often used for conducting usability tests. However, it is not mandatory that all usability experiments take place in a usability laboratory. A typical usability testing facility comprises a test room and an observation room. Other rooms such as an executive observation lounge and participant waiting room are provided for large scale usability tests. The usability test takes place in the test room, and the observation room is used to observe the experiment [85]. The observation and test room are (usually) separated by a one-way mirror so that the observation is not intrusive.

The test room typically comprises of a test system to execute the tested application, and various cameras that enable monitoring and recording of the participants' actions. The observation room comprises a monitoring system that shows each camera view, and an event-logger workstation. Configuration of software and hardware is dependent on the type of usability experiment conducted. A typical usability testing facility is presented in Figure 4-1.

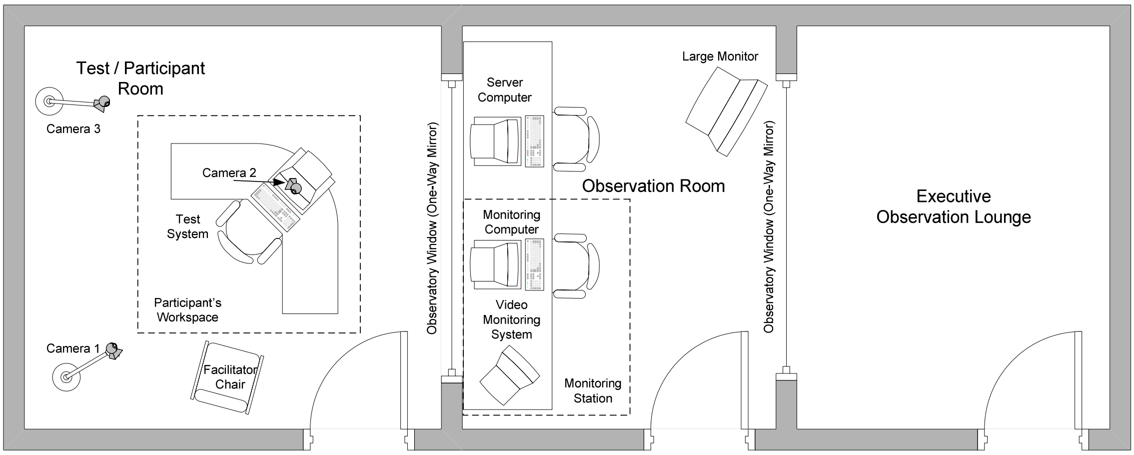


Figure 4-1: Typical Architecture of a Fixed Usability Test Facility

The usability testing facility should be configured to cater for the type of application being tested, which can be fixed or mobile. A Fixed Usability Testing Facility is a stationary onsite based laboratory. Mobile Usability Testing Facilities are portable and mainly used for field studies. Research has been carried out by Kimber et al in developing usability test systems and procedures for mobile tourism services [86]. The usability studies carried out for this research were performed in a Fixed Usability Testing Facility.

4.5 Re-engineering Usability Testing Processes

To efficiently carryout usability studies, the usability testing facility configuration must be customised for the type of tests to be performed. This also requires fine-tuning usability testing processes accordingly to improve the efficiency of performing the experiments [87].

Prior to carrying out the usability studies on the interfaces developed for this research, the traditional usability testing facility configuration and process was evaluated for its efficacy in conducting usability tests on QoS management interfaces.

For the purpose of this investigation, a usability test was formulated and conducted using the traditional testing facility configuration and testing processes. It was discovered that the traditional usability testing facility configuration resulted in employing inefficient usability testing processes, which made it time consuming and cumbersome to the conduct usability test.

Consequently, a re-engineered usability testing facility configuration was developed, which resulted in developing a streamlined usability testing process. After completing the same usability test using the re-engineered usability testing facility and processes, it was discovered that this approach reduced the time for completing the usability test and did not affect the experiment results.

Details of this study have been published by Georgievski and Sharda, which can be found in [88] and [89]. An overview of this study is presented in Appendix A.

4.6 Conclusion

This chapter provided an overview of the fundamentals of usability testing. Usability testing facilitates for enhancing the usability of user interfaces and applications. The various stages involved in usability engineering, assist in the development of 'user-friendly' applications. This can only be achieved if usability engineering is applied throughout the software development process. Usability assessment methods extract valuable feedback from participants. The type of application and the feedback required determine which assessment methods are applicable to the usability test.

Usability testing facilities should be configured according to the application type and the test environment. Fixed usability testing facilities are suitable for stationary applications. Mobile usability testing facilities are best suited for portable applications. The usability testing facility should simulate the environment in which the application is likely to operate in.

Re-configuring the usability test facility and re-engineering testing processes are required to efficiently perform usability studies. Re-engineering the traditional usability testing process streamlined and enhanced the productivity of usability testing by reducing the experiment time without affecting the experiment results.

The usability testing process and assessment methods presented in this chapter were used for testing different QoS management interfaces developed in this research project. The re-engineered usability testing process was used to conduct all the usability studies for this research.

Chapter 5
Usability Investigation Design

Summary

This chapter introduces the motivation for usability testing of user interfaces for QoS management. Developing QoS management user interfaces requires usability studies performed on them to ensure that the user's needs for requesting and negotiating services are fulfilled. Usability studies can take place at three levels for networked multimedia systems, namely: Interface, System and Service Level. This research focuses on usability studies at the interface level where different user interfaces for QoS management are developed and tested.

An overview is presented for testing various user interface elements with different Physical User Interface (PUI) devices that can be used for QoS management. Various PUI devices such as the Keyboard and Mouse combination, Joystick, Game-pad Controller, Steering Wheel with Foot Pedals, Pen & Touchpad, and Touch Screen – which can provide novel interaction methods – are introduced in this chapter.

A prototype user interface is presented, which incorporates different QoS management interfaces that were designed based on the QoS taxonomies and models presented in Chapter 3. The prototype interface provides an overview of how the various QoS management interfaces can be integrated with a networked multimedia application for performing QoS specification, negotiation, and re-negotiation.

5.1 Introduction

With the emergence of networked multimedia systems, QoS guarantees have become a necessity. As there are no interfaces that provide easy interaction for QoS management, this makes it difficult for non-technical users to request and negotiate QoS with networked multimedia applications and network services. This predicament provides motivation for developing user interfaces that enable different users to interact with QoS management systems to request and negotiate QoS.

Section 5.2 introduces the rationale for conducting usability testing of interfaces for QoS management. Section 5.4 provides an overview of the usability studies performed on the user interfaces developed for Static and Dynamic QoS Management on Desktop Systems, and Portable Devices. Sections 5.5, 5.6 and 5.7 respectively present the usability investigation process, usability testing procedures and assessment methods, and the method for selecting and classifying participants. Section 5.8 presents the conclusion for this chapter.

5.2 Rationale for Usability Testing of Interfaces for QoS Management

5.2.1 Motivation for Usability Investigation

As networked multimedia systems have evolved over the recent years, so has the emergence of networked multimedia systems. Through this, a revolution in the transmission of multimedia information over wired and wireless communication technologies has transpired. Society is now becoming more dependent on such technologies, which are used in almost every aspect of peoples' daily lives, including: communications, entertainment, education, marketing, research, health and medicine.

To provide the user with effective experience in using these networked multimedia applications, it is important that optimum Quality of Service (QoS) is delivered. This requires innovative solutions for QoS management. These solutions need to employ better Human Computer Interaction (HCI) techniques and bridge the current gap between the user requirements and the system functionality. If a QoS specification methodology does not consider the user’s needs, it cannot be guaranteed that the user will get the service they desire. Therefore, a mechanism for specifying QoS from a user’s perspective is vital. As efficient and effective specification of QoS would be desired by most users, a usability study for QoS management interfaces is paramount.

Developing interfaces for QoS management requires extensive usability studies performed on them as it is expected that users with diverse backgrounds and skills will be interacting with these interfaces. It is important that the user interface satisfies the requirements for a range of users, and then assist in achieving their goals, rather than make tasks more cumbersome. To ensure that these systems facilitate the user in completing their tasks efficaciously, a usability investigation of the system under development is a must.

5.2.2 Strategy for Usability Investigation

Usability studies can take place at three levels for networked multimedia systems, these are [57]:

- *Interface level* – considers various factors that are related to the user interface, such as: learnability, memorability, efficiency and satisfaction.
- *System level* – considers various factors that are related to the operational aspects of the system, such as: reliability, latency (delays), and security (confidence).
- *Service level* – considers factors that are related to the provision of the service such as: desirability, cost of change over, and retraining cost.

Generally, system developers have the tendency to use the bottom-up approach when developing a system – where the underlying technology of a system is developed first and then the functionality is adapted/mapped according to the users’ requirements. This development process often leaves a gap in the usability of the system because system functionality is not well matched with the user’s requirements, which renders the system inefficacious.

By employing usability engineering principles, systems are developed using the top-down approach. User requirements, expectations, demands, experience and knowledge are taken into consideration first, and then used to inform the system design process

[57]. This facilitates the development of a 'user-oriented system' that enhances the user's experience in completing various tasks. In this research, usability studies are carried out at the *Interface level* for user interfaces developed for QoS management. Conclusions drawn from this study can then inform further research and development of the underlying technology (i.e. at the system and service levels) to manage and deliver the desired QoS.

5.3 User Interfaces for QoS Management

User interfaces that allow non-technical users to easily specify and negotiate QoS have not been developed to date. Some user interfaces have emerged that enable the user to tweak parameter values for QoS control. However, this requires technical knowledge of multimedia communication systems, which is suitable for a technical user only.

Various programming languages and environments have been developed that support creation of QoS management applications. QoS-Talk is an Integrated QoS Programming Environment which is a general-purpose and application-independent framework [91]. QoSTalk is based on a Hierarchical QoS Modelling Language (HQML) – an extended version of Hypertext Mark-up Language (HTML) [92]. These QoS programming environments enable developers to develop QoS management applications for different networked multimedia applications. These environments help in developing the underlying technology for QoS management, but cannot support user interface development.

There is a need for developing user interfaces for QoS management of multimedia applications operating on Desktop Systems. With the rapid evolution of Portable and Mobile Devices, executing multimedia applications on these devices is now possible. Performance and bandwidth issues relating to mobile networks and communication devices impede guaranteeing QoS.

In this research, user interfaces operating on Desktop Systems, Portable and Mobile Devices have been developed for QoS management. These user interfaces have then been tested to determine their usability.

5.3.1 Physical User Interfaces

In recent years, novel Physical User Interface (PUI) devices have provided innovative methods for Human Computer Interaction (HCI). Each PUI device is designed for a specific class of applications enabling the user to perform tasks/functions with upmost efficiency and effectiveness. The various PUI devices that are commercially available include: the traditional Mouse & Keyboard combination, Joystick, Game-Pad Controller, Steering Wheel with Foot Pedals, Pen & Touchpad combination, and Touch-screen. Prior to developing QoS management interfaces, preliminary studies are required to determine the suitable devices for interacting with different user interface elements. This explores the question of 'how the user can control the interface to specify, request, and negotiate QoS'. Commonly used PUIs include the following:

- *Mouse & Keyboard* – enables users to perform numerous tasks on a Personal Computer (PC).

- *Joystick* – consists of a hand-held stick that can be pivoted to different directions, where the angle in two dimensions is transmitted to the computer. Joysticks also comprise buttons that activate specific functions. Joysticks are mainly used in gaming, robotics, flight simulation, and specific applications for the disabled.

- *Game-Pad Controller* – consists of a four-way directional controller buttons and activation buttons. Game-Pad controllers are usually connected to a PC or a gaming console and are used for gaming (arcade style), general purpose gaming and robotics.

- *Steering Wheel with Foot Pedals* – consists of a steering control similar to that in any automobile. The steering wheel control is used for directional (left, right) control, whilst the foot pedals are used for forward/reverse motion and stopping control. These are usually connected to a PC or a gaming console.

- *Pen & Touchpad* – enables one to draw directly into a computer memory in real-time. A pen (stylus) is used on a large flat surface for drawing on. This simulates writing with a pen on paper. Pen & Touchpad devices are used for graphics design, fashion design, and CAD applications.

- *Touch-screen* – is a touch sensitive display panel that includes display overlays that are either pressure-sensitive (resistive), electrically-sensitive (capacitance), acoustically-sensitive (SAW – surface acoustic wave) or photo-sensitive (infrared). Users can interact with these devices by simply touching any part of the screen surface that will result in performing an input action. These devices are used on desktop PCs, Notebook Computers (Tablet PC), and Portable Digital Assistants (PDAs); and for applications such as: Automatic Teller Machines, Information Kiosks, and Gaming Machines.

5.3.2 Graphical User Interfaces

A prototype user interface (shown in Figure 5-1) has been designed for a videoconference application using the TRAQS concept and other related models mentioned in chapter 3. This interface provides a general overview of how the various user interface elements can operate in a desktop multimedia application for QoS management [90]. This user interface is used to request and negotiate QoS prior to starting multimedia application, and manage (re-negotiate) QoS in real-time during the multimedia session. Figure 5-2 shows a flow chart of the tasks, actions, and interactions used to specify, negotiate and re-negotiate QoS.

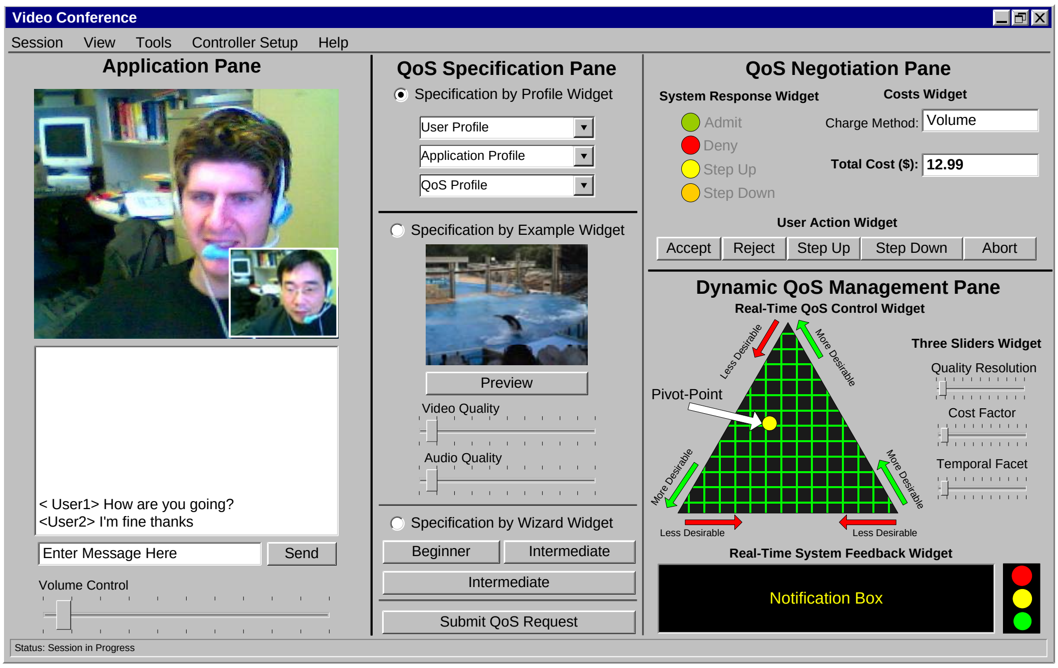
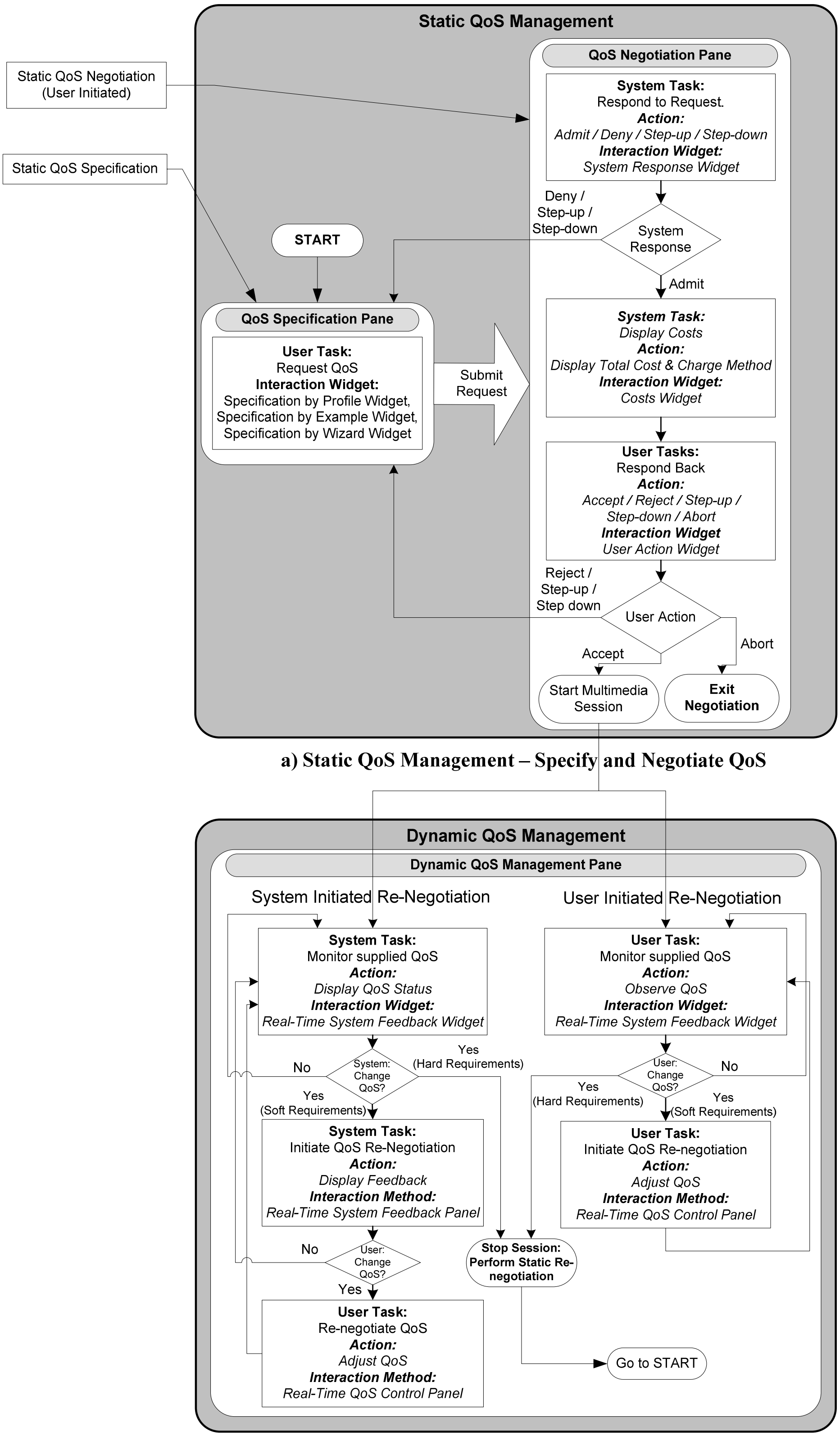


Figure 5-1: Prototype QoS Management Interface



a) Static QoS Management – Specify and Negotiate QoS

b) Dynamic QoS Management – User & System Initiated Re-negotiation

Figure 5-2: Process for Specifying, Negotiating and Managing QoS

The user interface presented in Figure 5-1 is made of the following panes; where each pane includes a number of widgets.

- *Application Pane* comprises the video screens for both the *caller* and the *called* parties. These users are able to communicate via audio and text. The application pane can be used for other applications also, such as: Video on Demand, Education on Demand, Online Presentations etc.

- *QoS Specification Pane* comprises three widgets for static specification of QoS: Specification by Profile widget, Specification by Example widget, and Specification by Wizard widget. This aspect of the interface is explored to determine the most efficacious method for specifying QoS.
    - *Specification by Profile widget* allows the user to select a User Profile, Application Profile and QoS profile for specifying QoS.
    - *Specification by Example widget* enables the user to specify the desired QoS by viewing and selecting an example clip played onscreen. The user can individually adjust the audio and video quality further if required.
    - *Specification by Wizard widget* provides a step-by-step walkthrough wizard for specifying QoS. Three types of wizards are defined: Beginner, Intermediate and Advanced, which are based on the User Profiles.

- *QoS Negotiation Pane* is used to negotiate QoS prior to executing a multimedia session. Upon making a request, the system provides feedback via one of the following widgets:
    - *System Response widget* provides system feedback for the QoS request using a colour scheme. Red represents request denied, and green indicates request admitted. The system can then recommend the user to step-up (increase) or step-down (decrease) their QoS request based on the availability of system and network resources.
    - *Costs widget* displays the Charge Method and Total Cost that the system can provide.

  - *User Action widget* provides the user the ability to negotiate QoS with the system. The user is given the option to Accept, Reject, Step-up, Step-down or Abort their QoS request.

- *Dynamic QoS Management Pane* includes the *Real-Time QoS Control widget* and *Real-Time System Feedback widget* for managing QoS during a multimedia session.

  - *Real-Time QoS Control widget* includes the QCTT model implemented as an interface, which enables one to adjust and re-negotiate QoS dynamically for the three performance aspects. The user can make these adjustments by directly interacting with a *Pivot-Point*, used as an indicator of the QoS achieved. Alternatively, the user can control the *Pivot-Point* using the *Three Sliders widget*.

  - *Real-Time System Feedback widget* uses a colour scheme to display the condition of the supplied services. The system monitors the QoS in real-time, and informs the user of any degradation and failures in the services. A Notification Box is used to give additional feedback to the user.

This research covers usability studies for Static and Dynamic QoS management using the different user interface features presented in the prototype interface, shown in Figure 5-1. This user interface and the flow chart given in Figure 5-2 provide a guide to how the different QoS specification, negotiation and re-negotiation functions proceed. The QoS Specification, Negotiation and Dynamic QoS Management panes were further enhanced and implemented as separated interfaces for managing QoS. These interfaces are described in the following sections.

5.3.3 Static QoS Management

QoS management includes Static QoS Management, and Dynamic QoS Management. *Dynamic QoS Management* is covered in section 5.3.4. *Static QoS Management* involves configuring QoS services in non-real-time, prior to initiating a multimedia session. Configuration of QoS takes place in two stages: Static QoS Specification, and Static QoS Negotiation, as shown in Figure 5-2a.

Static QoS Specification

Static QoS Specification is used to specify a desired QoS before executing a multimedia session. The user specifies the desired quality and the cost that they are willing to pay. Static QoS specification user interfaces (shown in Figures 5-3, 5-4, and 5-5) have been developed to enable different types of users to specify QoS. The underlying models used for developing these user interfaces are the QoS Parameter Architecture (QPA) and the Application Profile Model (APM). These models were presented in sections 3.4 and 3.5, respectively.

Using the QPA and APM models, a user is able to specify QoS by selecting User, Application and QoS Profiles:

1. *User Profile* – defines the method a user can specify QoS, which is related to the technical sophistication of the user. Three types of user profiles are defined: a) Beginner, b) Intermediate, c) Advanced.

2. *Application Profile* – configures and specifies the acceptable value for each QoS parameter from a predefined set of application profiles, e.g. Video on Demand, Video Conference, and Text Chat.

3. *QoS Profile* – can be High, Medium, or Low, and they provide a pre-configured set of QoS values according to the selected Application Profile.

Three different methods for Static QoS Specification are used:

a. *Profile Only* (Figure 5-3) – utilises a menu driven interface that enables the user to request QoS based on different profiles, namely: User Profile, Application Profile and QoS Profile.

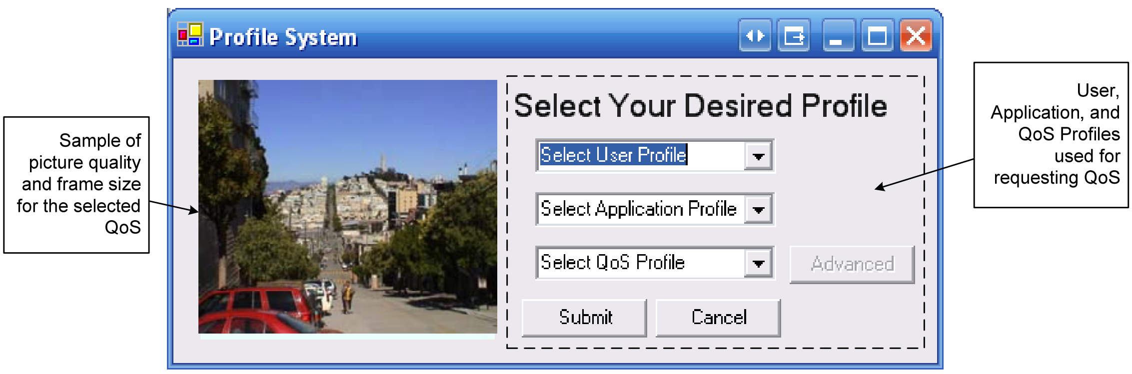


Figure 5-3: Menu Driven QoS Specification User Interface using Profiles

b. *Profile with Example* (Figure 5-4) – uses the same menu driven interface where a user can specify QoS based on a profile system, with the added ability to view an example similar to their request.

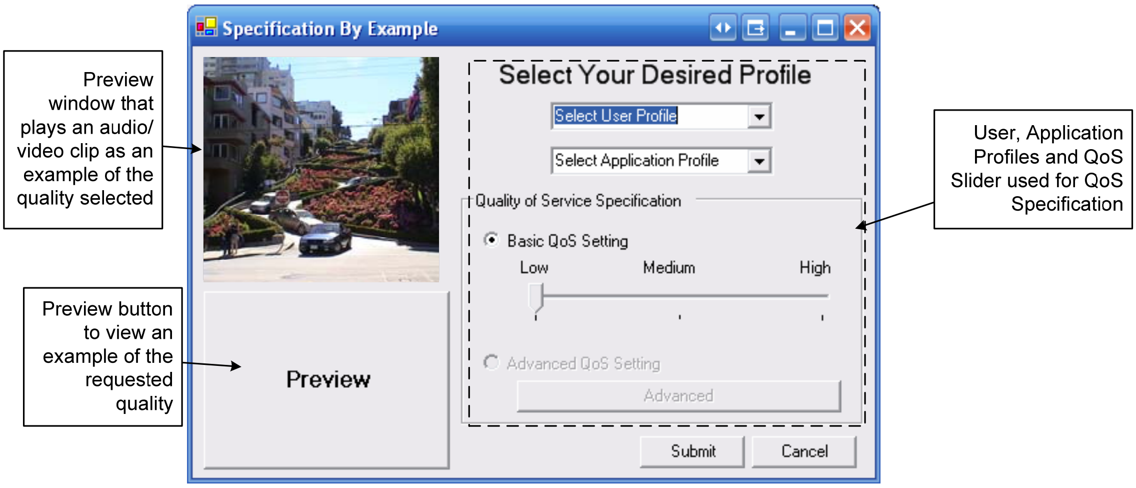
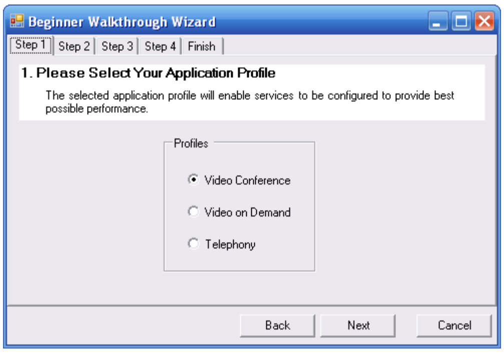
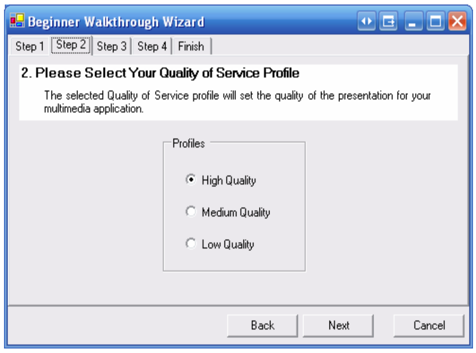


Figure 5-4: Menu Driven QoS Specification User Interface using Profiles with Examples

c. *Walkthrough Wizard* (Figure 5-5) – guides the user step-by-step in specifying QoS. The number of steps for specifying QoS is related to the user profile. A Beginner User is limited to specifying QoS using profile selection only, which can be done in four steps. An Intermediate user is able to specify QoS based on profiles and individual quality for audio and video, which requires six steps. An Advanced user is given access to individual parameter values. The user requires ten steps to walk through the various parameter groups.

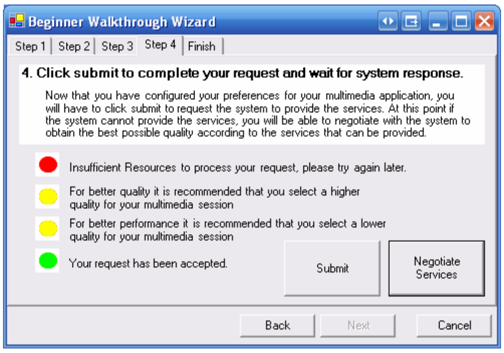


(a) Step 1: Select Application Profile



(b) Step 2: Select QoS Profile

(c) Step 3: Preview Example of Request



(d) Step 4: Submit Request

(e) Finish: Start Multimedia Session

Figure 5-5: QoS Specification using Walkthrough Wizard (Beginner User)

Static QoS Negotiation

Static QoS Negotiation involves the user negotiating the QoS with the system. This process is similar to a customer bargaining with a sales person. An agreement must be achieved between the two parties in order to conclude negotiations. In this case, the user is the 'customer' who wants to purchase a particular QoS, and the system is the 'sales person' who is selling different levels of QoS. The bargaining factors in this negotiation process are Quality, and Cost. In other words, Quality comes at a Price. The user and the system must come to agreement on the cost for a particular QoS.

QoS negotiation can be *System Initiated* or *User Initiated*.

- *System-Initiated QoS Negotiation* – in this case the system lists the available levels of QoS and its price. The user then selects the desired level of QoS from this list. This process requires users to 'know what they want' prior to making a selection from the list of services. Some technical knowledge in multimedia communication is required, which makes it more suitable for technical users.

- *User-Initiated QoS Negotiation* – requires the user to request QoS without knowing what the system can provide. The system can then accept or deny the request. If the system accepts the request, then the cost is negotiated next. If the initial request is denied, the user and the system continue to negotiate the quality first, and then the cost. The user and system must come to an agreement on the quality and its cost to conclude a negotiation session. This process does not require the user to have technical expertise. The user can select QoS based on a profile, and wait for the system to respond. The flow chart of this negotiation process is given in Figure 5-2a.

A *Static QoS Negotiation* user interface, presented in Figure 5-6, has been developed based on the *User-Initiated QoS Negotiation* model. This interface includes user interface elements that assist the user in negotiating QoS with the system. The different user interface elements, in the *User-Initiated QoS Negotiation* model, are evaluated for their usability.

System-Initiated QoS Negotiation method was not investigated as this method requires a simple process of selecting a desired QoS from a list; which does not introduce any novel user interface aspects that need to be investigated.

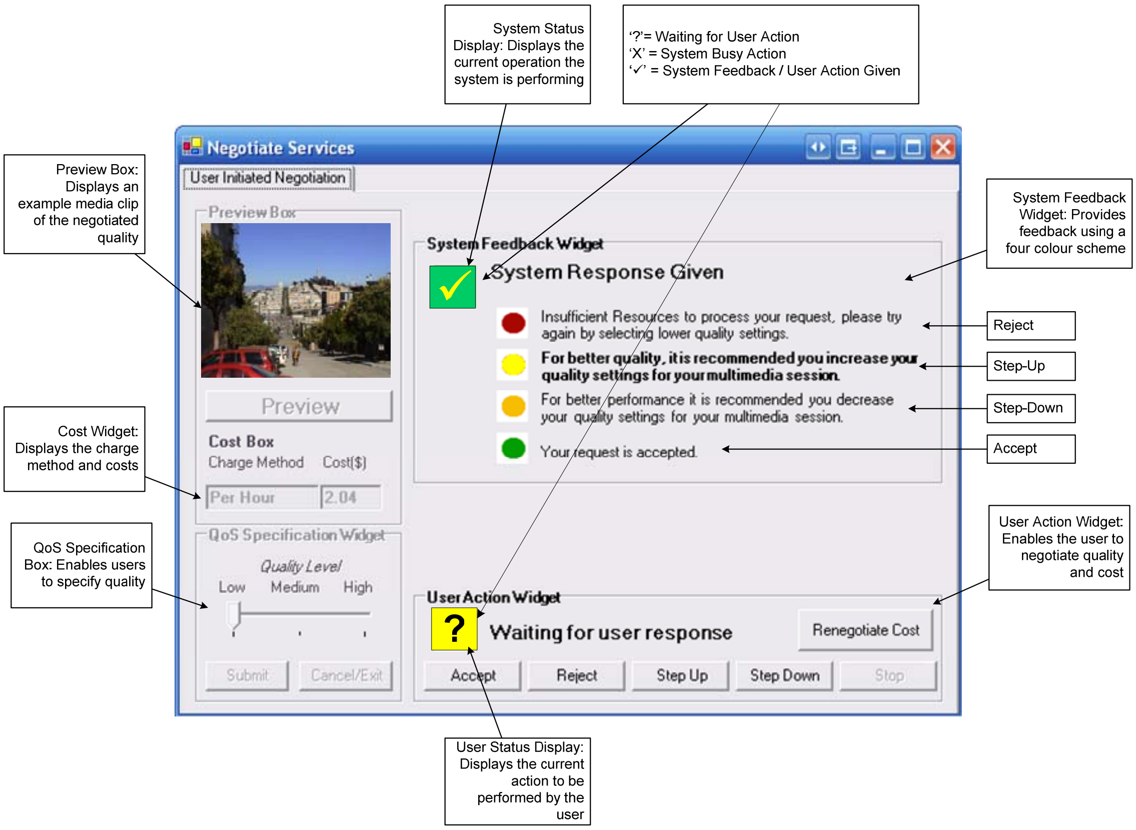


Figure 5-6: Static QoS Negotiation Interface

Usability investigations of the user interfaces presented in this section for Static QoS Specification and Negotiation are given in sections 6.4 and 6.5, respectively.

5.3.4 Dynamic QoS Management

Dynamic QoS Management involves re-negotiating QoS in real-time while a multimedia session is in progress. Re-negotiation of QoS can be *User-Initiated* or *System-Initiated*, as shown in Figure 5-2b.

User-Initiated Re-negotiation – in this case, the user instigates QoS re-negotiation by making adjustments to the quality in real-time during a multimedia session. The Quality Cost Temporal Triangle (QCTT) model (presented in section 3.7) has been implemented as a user interface to provide the user with the ability to dynamically manage QoS in real-time. An example of the user interface developed and tested is presented in Figure 5-7. This interface enables the user to adjust QoS using the QCTT triangle, where repositioning a Pivot-Point within the triangle controls the Quality, Cost and Temporal performances aspects.

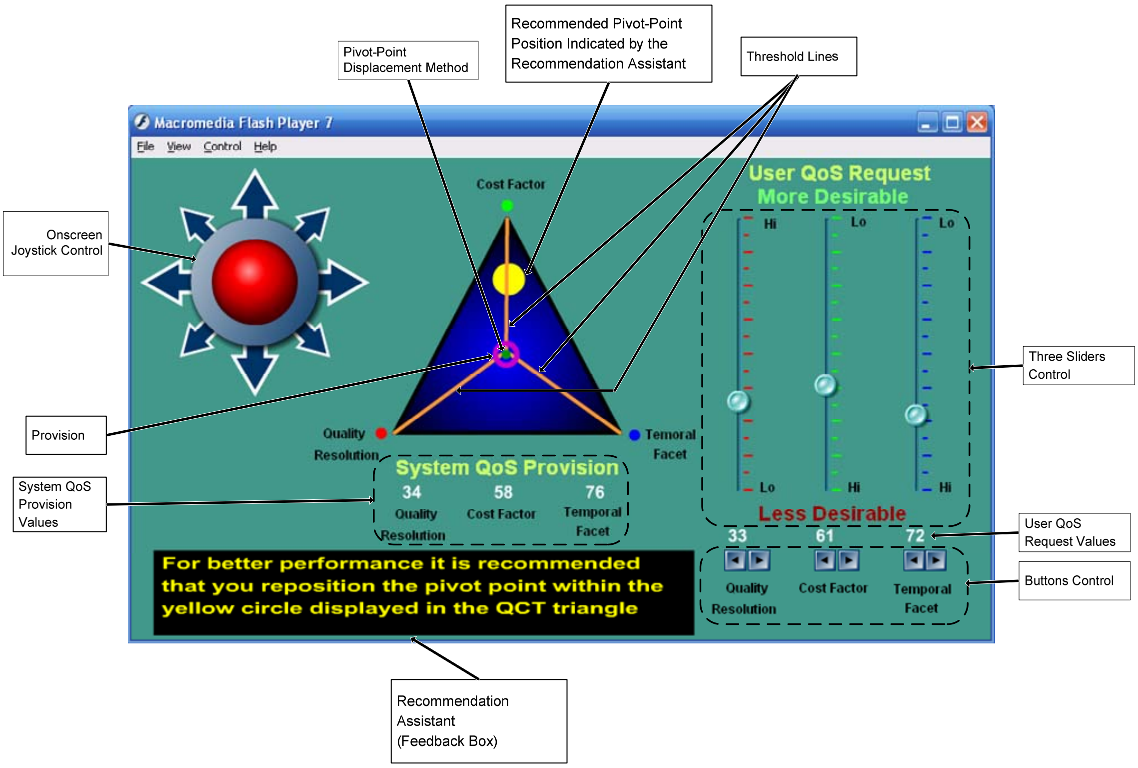


Figure 5-7: Dynamic QoS Management Interface

User Interface features, such as: *Onscreen Joystick*, *Three Sliders*, *Buttons*, and *Pivot-Point Displacement* are the different interaction methods used to re-position the Pivot-Point for re-negotiating QoS. The *User QoS Request Values* display the desired QoS value for the three performance aspects.

System feedback is given via:

- *Provision Ring* – displays the supplied QoS.
- *System QoS Provision Values* – displays the values for the supplied QoS for the three performance aspects.
- *Threshold Line* – uses a three scheme to provide feedback for displaying desirable and non-desirable values for each performance aspect in the QCTT interface.

It is intended that in a real system, when the Pivot-Point position is adjusted, changes to 'soft requirements' are made based on the Application Classification Model (ACM), while the 'hard requirements' remain unchanged. If degradation in the supplied QoS affects the hard requirements, then the multimedia session must stop and QoS re-negotiation must take place using the *Static QoS Negotiation* process.

System-Initiated Re-negotiation – is instigated by the system. The system prompts the user to adjust the requested QoS if changes in the available services are detected. The possible level of disruption to the multimedia session determines the type of feedback given to the user. A *critical* feedback response implies that the user must make necessary adjustments. An *advisory* feedback is a recommendation for adjusting the QoS level that was originally requested by the user.

A QoS *Recommendation Assistant* has been developed and implemented in a user interface to carry out *System-Initiated Re-negotiation*. The Recommendation Assistant, presented in Figure 5-7, informs the user via a text box message to adjust the Pivot-Point to a particular location in the QCTT interface, indicated by a yellow or red circle. A *yellow circle* indicates an advisory response, and a *red circle* indicates a critical response. Repositioning the Pivot-Point will make changes to parameters based on the ACM model.

This thesis includes a usability study of the user interfaces presented in this section for Dynamic QoS Management for networked multimedia applications operating on Desktop Systems and Portable/Mobile Devices.

5.4 Usability Investigation Overview

Usability studies for this research take place at the Interface level, where novel interaction methods are investigated prior to developing interfaces for Static and Dynamic QoS management. Usability studies are conducted in various stages, each stage guides the development of each user interface, and the type of usability test required to be performed on them. An overview of the three areas investigated in this research is given in the following. Figure 5-8 presents the structure of this investigation, showing the different usability studies conducted.

- *Investigation of Innovative Interaction Methods* – Combinations of Graphical User Interface (GUI) elements and Physical User Interface (PUI) devices are studied for basic computer control. Different user interface elements are also tested for providing real-time system-feedback during a multimedia presentation session.

- *Static QoS Management* – Different user interfaces and methods are studied for Static QoS Specification and Negotiation. User interfaces for Static QoS Management were tested on Desktop Computers only. User interfaces developed for static QoS management are menu driven, and do not introduce many new usability issues for portable devices, i.e. the research issues are the same as for Desktop Systems. Therefore, investigations of these user interfaces are limited to Desktop Systems.

- *Dynamic QoS Management* – investigates different user interfaces for Dynamic QoS Management. Dynamic QoS management includes performing QoS re-negotiation while a multimedia session is in progress. The QCTT model is implemented as an interface for performing User-Initiated and System-Initiated QoS Re-negotiation. This user interface is tested for Dynamic QoS Management on both Desktop and Portable Systems.

Classifying Desktop and Portable Systems

Current computing systems include Desktop Computers, Notebook/Laptop Computers, Mobile Phones, and Personal Digital Assistants (PDAs). These are generally categorised into the three groups: Desktop Systems, Portable Systems, and Handheld Devices. However, these groups overlap. For example, a Laptop/Notebook can be used as a Desktop System as well as a Portable System, and can be placed in either category depending upon the context.

In this research two categories are defined: Desktop Systems and Portable Systems. Desktop and Laptop/Notebook computers are considered as Desktop Systems, as they use similar interfaces for interaction, i.e. QWERTY keyboard, mouse, and a large screen. Handheld Devices, such as Mobile Phones and PDAs are categorised as Portable Systems, as their interfaces are similar, i.e. small screen size, (phone) keypad and touch screen – in some devices.

Each usability study aims to evaluate different features in the user interfaces for QoS management. *Controlled Usability Tests* have been performed for each interface. This research adopts a formal usability testing process for each experiment, which includes: developing documentation, recruiting participants, organising and configuring equipment, developing interfaces as software applications, preparing the usability test (including scenarios and tasks), preparing and selecting assessment methods, re-configuring the usability test facility, and analysing recorded results.

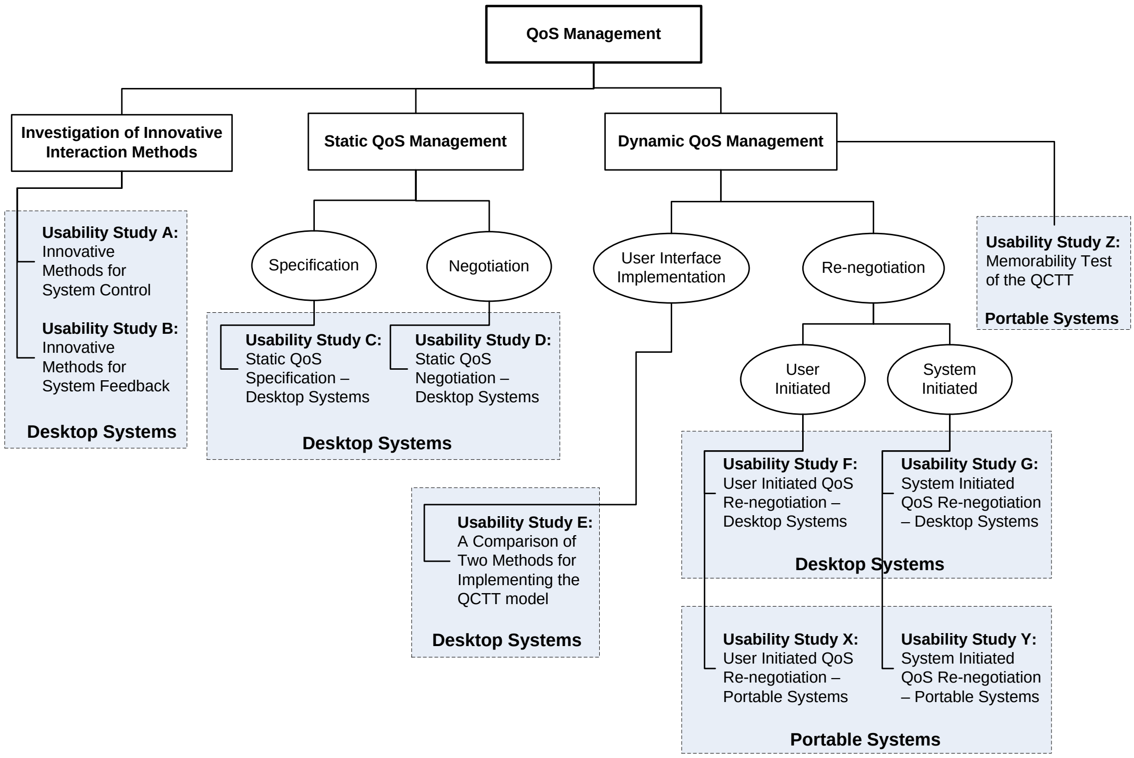


Figure 5-8: Overview of Usability Investigation

5.4.1 Overall Aims and Objectives

The overall aim of this investigation is to create efficacious user interfaces for QoS management and study their usability for requesting, negotiating, and re-negotiating QoS. This general aim can be divided into the following list of specific objectives.

- Investigate new Human-to-Computer Interaction (HCI) methods for QoS Management.

- Investigate the ability for technical and non-technical users to specify, negotiate, and re-negotiate QoS without requiring any technical knowledge of QoS and multimedia communication systems.
- Investigate novel methods and interfaces for managing QoS in real-time and do not disrupt the communication process taking place.
- Provide recommendations for developing user-centred QoS management system which enhance the user experience through use of efficacious interfaces that enable even a layperson to request, negotiate and re-negotiate QoS. Such a system would be required to inter-operate with existing and new communication technologies that support the provision of QoS.

5.4.2 Individual Usability Study Aims and Objectives

This investigation is divided into ten separate usability studies, which investigate the usability of novel interaction methods for computer control and system feedback; interfaces for QoS specification, negotiation, and re-negotiation on Desktop Computers; and interfaces for QoS re-negotiation on Portable Systems. Specific aims for each study are given in the following.

Usability Study A: Innovative Methods for System Control

Usability Study A aims to determine the most suitable Physical User Interface (PUI) device for basic control of various Graphical User Interface (GUI) elements. Various PUI devices, namely: Mouse & Keyboard combination, Joystick, Game Pad, Steering Wheel with Pedals, and Pen & Touchpad combination; are interfaced with the GUI elements: Menu Item, Slide-bar, Radio Buttons, Check Box, List Box, Combo Box, Push Button, Scrollbar, and Tab Control. For each task, participants are required to perform a series of actions that involved clicking on objects and navigating a user interface.

Usability Study B: Innovative Methods for System Feedback

Usability Study B aims to determine the most suitable user interface element for basic system feedback. Various user interface elements, namely: Audio Alert, Three Colour Alert (with & without Audio), Four Colour Alert (with & without Audio), and Pop-up Alert (with & without Audio), are tested to provide three random responses, namely; Critical, Informative and Advisory.

Usability Study C: Static QoS Specification – Desktop Systems

The aim of *Usability Study C* is to determine the most suitable method and related user interface for specifying QoS prior to the execution of a multimedia session. Three interfaces, namely: a) Profile Only, b) Profile with Example, and c) Walkthrough Wizard, and specification methods: (i) Limited Control, (ii) Partial Control, and (iii) Full Control are investigated for Static QoS Specification on a Desktop System.

Usability Study D: Static QoS Negotiation – Desktop Systems

The focus of *Usability Study D* is to investigate the usability of the User-Initiated QoS Negotiation method and interface for Static QoS Negotiation. Participants are required to evaluate the usability of the negotiation method and related user interface for negotiating quality and cost prior to executing a multimedia session. Users are required to request a desired quality, and then interactively negotiate with the system to obtain a feasible cost and QoS.

Usability Study E: Comparison of Two Methods for Implementing the QCTT model

In *Usability Study E,* two methods (namely Cartesian Coordinate System (CCS) and Triangular Fractal System (TFS)) for implementing the QCTT model as an interface are investigated. This study evaluates the usability of the TFS and CCS interfaces and the interaction methods for controlling the Pivot-Point in the QCTT triangle. Outcomes from this study guide the implementation of the QCTT interface, which is later used to perform Dynamic QoS Management.

Usability Study F: User-Initiated QoS Re-negotiation – Desktop Systems

Usability Study F focuses on the interface elements for performing User-Initiated QoS Re-negotiation. Users are able to re-negotiate QoS by controlling the Pivot-Point within the QCTT interface using the following three methods: Three Sliders Control, Buttons Control, and Pivot Point Displacement Method. System feedback is given via two intuitive interface elements, namely: Provision Ring, and Threshold Lines. This study investigates the usability of the Provision Ring and the Threshold Lines for performing User-Initiated QoS Re-negotiation on Desktop Systems.

Usability Study G: System-Initiated QoS Re-negotiation – Desktop Systems

Usability Study G investigates the usability for the Recommendation Assistant to perform System-Initiated QoS Re-negotiation on Desktop Systems. Users are required to respond a series of system feedback alerts given by the Recommendation Assistant.

Usability Study X: User-Initiated QoS Re-negotiation Portable Systems

Usability Study X investigates the usability of the QCTT model implemented in a Video Player that is used on: an emulated Personal Digital Assistant (PDA), a real PDA and an emulated Mobile Phone. As the screen size of a portable system is much smaller than that of a Desktop System, usability of the QCTT interface may be different because the small screen affects the accuracy for specifying QoS.

Three different versions of the QCTT model are implemented as interfaces and tested for usability. Each interface catered for different levels of accuracy. The QCTT is partitioned into fractal zones, such that each fractal zone defines preset values for Quality, Cost and Temporal performance aspects. The following versions of the QCTT model are implemented and identified as:

1. *Large QCTT Quad Fractals* – defines four fractal zones within the QCTT. Each fractal zone is assigned specific parameter values for the three performance aspects.

2. *Small QCTT Quad Fractals* – is similar to the Large QCTT Quad Fractals interface. However, the size of each fractal is reduced to better utilise the small screen of portable devices.

3. *Large QCTT 10,000 (10k) Fractals* – This breaks the QCTT interface into 10,000 fractals for much higher accuracy as compared to the previous two interfaces.

The above mentioned user interfaces are evaluated for performing real-time User-Initiated Re-negotiation of QoS during the progress of a multimedia session.

Usability Study Y: System-Initiated QoS Re-negotiation – Portable Systems

Usability Study Y investigates the usability of the Recommendation Assistant integrated into the *Large QCTT 10,000 (10k) Fractals* interface to perform System-Initiated QoS Re-negotiation on the three devices: Emulated PDA, Emulated Mobile Phone, and PDA.

Usability Study Z: Memorability Test of the QCTT

Usability Study Z measures the memorability for the different features of the QCTT interface to perform User-Initiated and System-Initiated QoS Re-negotiation. A multiple choice test, comprising twenty five questions, is given to participants twenty minutes after completing the usability tests X and Y.

5.5 Usability Investigation Process

Usability investigations performed in this research employed a systematic testing process. For each usability study, the following process was used.

Plan and Prepare Investigation – involved developing usability investigation goals, aims, objectives, planning the experiment, and selecting participants.

- *Identify Aims and Objectives:* General aims and objectives were identified for the usability investigation, which provided an overview for focusing on the specifics of this research. Specific aims and objectives were then outlined for each usability study, which indicated the goals that needed to be achieved.
- *Plan Usability Study:* This involved designing and developing each experiment based on the defined aims and objectives. For this research, experiment planning involved 'storyboarding' each experiment stage, which included the development of:

  - Application Interfaces.
  - Experiment test scenarios and tasks.
  - Processes for selecting and recruiting participants from diverse backgrounds and computer literacy skills.
  - Experiment documentation, including: experiment overview information for participants, participant consent form, participant contact details form, revocation of consent form, pre-experiment questionnaire, post-experiment questionnaire, task performance evaluation form (task error and task completion time logging), and usability testing process checklist form.
  - Application for ethical clearance to use human subjects for conducting the usability studies.

- *Recruit Participants:* Participants were recruited at random and were categorised as Advanced, Intermediate or Beginner Users based on a user ranking model presented in section 5.7. Participants were ranked based on: (a) Computing & Gaming Experience, (b) Knowledge of Networked Multimedia Systems and (c) User Experience with Networked Multimedia Applications. Generally, ten participants were recruited for each usability study.
- *Organise and Configure Testing Facility:* Re-engineering of the usability testing facility and usability testing processes was required to perform the usability studies. An overview for re-engineering the usability test facility and testing processes is presented in section 4.5, and specific details for this process are presented in Appendix A.
- *Conduct Usability Test:* Each user interface was subjected to heuristic evaluation to eliminate glitches and bugs in the application and the testing process. Empirical testing was then performed with the selected participants, which required them to complete a scenario of tasks using the specific user interface. An example of a test scenario used in this research is given in Appendix B. Video monitoring was used to record participant behaviour and to authenticate the experiments. Participants were encouraged to use the Thinking-Aloud-Protocol as described in section 4.3.5.

- *Collect and Analyse Data:* Pre-experiment questionnaires were used to collect demographical data from participants, and to classify them as either Beginner, Intermediate or Advanced users. An example of the pre-experiment questionnaire used in this research is given in Appendix C. Post-experiment questionnaires were used to collect feedback about the user interfaces. An example of a post-experiment questionnaire used in this research is given in Appendix D. Task Completion Times and Task Error Counts were recorded for Task Performance analysis.
- *Report Analysis:* Analysis of recorded data for each usability study was carried out consistently throughout the research. The analysis for each usability study is presented in Chapter 6.

5.6 Usability Test Procedure

Usability tests were performed in a controlled usability test environment by using the same usability test facility for each experiment. Participants were tested individually and guided through the experiment procedure. The testing processes that were used are discussed in the following:

- *Interviewing the Participant:* Participants were interviewed to gather information on their demographics, as well as computing skills. Information about the nature of the test and procedures involved were presented to the participants. Participants were provided with ample time to read this material and absorb the information. If the participant voluntarily agreed to participate in the usability test, only then were they required to fill out a Contact Form, and a Consent Form.
- *Briefing the Participant:* Participants were briefed on the testing process, scenario and the nature of the tasks for each experiment. A sample scenario and task was given to the participants as an example, to help them become familiar with the testing procedure. Participants were also given the opportunity to ask questions.
- *Synchronising for the Usability Test:* Participants were given the opportunity to become comfortable for conducting the tests. Facilitators and observers would then perform final equipment test, which included sound checks, camera positioning, and setting up the video tape for recording.

- *Conducting Usability Test:* While the usability test was in progress, the participant performed the test tasks given in the experiment scenario. Usability tests were used to record users' Task Performance and Feedback using a set of Usability Evaluation Criteria. After completing a group of test tasks in a scenario, the participant then filled out a post-experiment questionnaire. Participants were given the opportunity to take breaks between task groups or scenarios.
- *Debriefing the Participant:* Once the usability test concluded, participants were debriefed and were given the opportunity to provide further comments.

5.7 Selection of Participants

When recruiting participants for usability studies, it is important that the participant's knowledge of the system, computer literacy, and other external factors are considered. To conduct a thorough usability study, the right number of participants should be selected. As mentioned earlier, Nielsen states that testing five users would reveal on average, 85% of the usability problems [64]. For this research, to include users with diverse backgrounds it was decided to generally use ten participants for each usability study.

Table 5-1: Participant Experience/Knowledge Ranking Model

<table>
<tr><th>Three Dimensions of User Experience</th><th>Evaluation Criteria</th><th>Credit Points</th></tr>
<tr><td rowspan="4">Experience: Computing / Gaming</td><td>Gaming Frequency</td><td>10</td></tr>
<tr><td>Level of Experience in Computer / Gaming</td><td>30</td></tr>
<tr><td>Computer Usage Frequency</td><td>25</td></tr>
<tr><td>Weighting</td><td>65</td></tr>
<tr><td rowspan="2">Knowledge of Domain: Networked Multimedia Systems</td><td>Level of Knowledge for Networked Multimedia Systems</td><td>20</td></tr>
<tr><td>Weighting</td><td>20</td></tr>
<tr><td rowspan="4">User of System: User Experience with Network Multimedia Applications</td><td>Level of Internet Usage</td><td>5</td></tr>
<tr><td>Video Conferencing Usage</td><td>8</td></tr>
<tr><td>Prior participation in usability testing for Networked Multimedia Systems</td><td>2</td></tr>
<tr><td>Weighting</td><td>15</td></tr>
</table>

Participants were categorised as Advanced, Intermediate or Beginner Users based on the accumulation of points for the three dimensions of user experience [64]. These three dimensions are: Computer Experience, Knowledge of Domain, and User of System. For this research, these three dimensions translate to (Table 5-1): Computer / Gaming

Experience, Knowledge of Networked Multimedia Systems, and User Experience with Networked Multimedia Applications, respectively.

A model was created for ranking each participant based on their responses given in the Pre-Experiment Questionnaire. This model is presented in Table 5-1. Credit points are given to reflect the importance of the dimension. This helps to evaluate the participant's profile. *Experience (Computing & Gaming)* was given the highest weighting of 65, *Knowledge of Domain (Networked Multimedia Systems)* was given a weighting of 20, and *User of System (User Experience with Networked Multimedia Applications)* was given a weighting of 15. Each dimension is further divided into criteria shown in Table 5-1.

Based on the feedback given in the Pre-Experiment Questionnaire, each participant accumulated credit points. The participant profile (Advanced, Intermediate, and Beginner) was determined by the total credit points a participant scored as per Table 5-2. This user ranking model was found to be in line with heuristic ranking considered appropriate by users themselves. Formal validation of this user classification model can be taken as an area for further research.

Table 5-2: Participant Profile Rank Chart

User Category	**Credit Points**
Advanced User	75 -100
Intermediate User	50 – 74
Beginner User	0 – 49

5.8 Conclusion

This chapter introduced the motivation for developing user interfaces to manage QoS. As networked multimedia systems proliferate in modern society, people are becoming dependant on these technologies. It is vital to develop interfaces that provide users the ability to efficiently request, negotiate, and re-negotiate QoS.

Developing such interfaces enable users to specify a desired QoS, negotiate its cost, and receive service guarantees throughout the multimedia session. If such guarantees fail, then the user can re-negotiate QoS in real-time without disrupting the multimedia session.

The prototype user interface presented in this chapter, provides a general overview of how various user interfaces for QoS management can operate in a desktop multimedia application. This prototype interface is used as a guide for developing and enhancing novel interaction methods and user interfaces for Static and Dynamic QoS Management.

This research covers usability studies at the Interface level, where novel interaction methods and interfaces are developed and tested for their usability in managing QoS. These studies are presented in Chapter 6 and employ the usability investigation process, testing procedure, assessment methods, and participant classification methods that are presented in this chapter.

Chapter 6
Usability Studies on QoS Management Interfaces

Summary

This chapter presents usability studies of various user interfaces for QoS management on Desktop Computers and Portable Devices. Prior to developing interfaces for QoS management, interaction methods were investigated for basic system control and system feedback – for real-time multimedia applications. Usability tests confirmed that the Mouse & Keyboard combination is most suitable for interacting with Desktop Computers, and the Pen & Touchpad is most suitable for Portable Devices. It was also discovered that a combination of visual and audio feedback is necessary for system feedback.

Usability studies were carried out on interfaces for performing Static QoS Specification, User-Initiated QoS Negotiation, and QoS Re-negotiation on Desktop Systems. These studies revealed that users found the Application and QoS Profiles effective for specifying QoS; particularly when combined with the option to preview an example of the content with the requested quality. The User-Initiated QoS Negotiation method and interface was acknowledged by all users for its good usability to negotiate QoS on Desktop Computers. However, suggestions were made by some users that the interface required some further refinement.

A usability comparison of two methods for implementing the QCTT model as an interface for managing QoS revealed that the Triangular Fractal System (TFS) worked better than the Cartesian Coordinate System (CCS). Users found the QCTT interface to be rather useful for performing User-Initiated and System-Initiated QoS re-negotiation on Desktop Computers. The Provision Ring, Threshold Lines and Recommendation Assistant enhanced the user experience for re-negotiating QoS; while the Provision Ring substantially reduced task completion times.

Usability studies were conducted on the QCTT interface for performing real-time User-Initiated and System-Initiated QoS Re-negotiation on an Emulated PDA, Real PDA, and Emulated Mobile Phone. These studies involved investigating the usability of the QCTT interface for accurately specifying QoS on a small screen. Tests revealed that the QCTT interface is suitable for performing User-Initiated QoS Re-negotiation on Portable Systems, and using fractal zones improves the accuracy with which QoS can be specified on small screens. The Recommendation Assistant received good usability ratings for performing System-Initiated QoS Re-negotiation, and using visual feedback only without textual information caused less distraction to the user. A memorability study conducted for these interfaces revealed good usability as all users recorded a high memorability score.

6.1 Introduction

Interfaces for managing QoS on Desktop Systems and Portable Devices requires extensive usability studies performed on them in order to make certain that these interfaces cater for the needs of different users. Developing novel interfaces for QoS management requires investigation of various Human Computer Interaction (HCI) aspects, such as: methods for Physical User Interaction (Hardware); User Interaction Design for control and feedback; and User Interface Design, i.e. the layout and visual representation of user interface elements.

This chapter presents usability studies of interfaces used to manage QoS for Networked Multimedia Systems operating on Desktop Computers and Portable Devices. Each study answered some important research questions, while raising many more. Answering each of these new questions is beyond the scope of a single research thesis; therefore, these have been listed as recommendations for further research.

Section 6.2 investigates innovative Physical User Interface (PUI) methods for controlling different user interface elements. Section 6.3 investigates different feedback methods for real-time multimedia systems. Sections 6.4 and 6.5 investigate user interfaces for Static QoS Specification and Static QoS Negotiation on Desktop Computers, respectively. Section 6.6 provides a usability comparison of two implementation methods for the QCTT model. Sections 6.7 and 6.8 respectively present usability studies on interfaces for performing User-Initiated, and System-Initiated QoS Re-negotiation on Desktop Systems. Sections 6.9 and 6.10 investigate the usability of the User-Initiated, and System-Initiated QoS Re-negotiation interfaces operating on Portable and Mobile Devices, respectively. Section 6.11 provides a memorability analysis for the QCTT interfaces used on Portable and Mobile Devices, while section 6.12 presents the conclusion of this chapter.

6.2 Usability Study A: Innovative Methods for Control

6.2.1 Overview

Presently, the Mouse & Keyboard combination are the most widely used devices for computer control. Prior to developing QoS management interfaces, it must be made certain that the Mouse & Keyboard combination is the most suitable Physical User Interaction (PUI) device for interacting with different interface elements that may be used for real-time QoS management. Real-time QoS management requires interfaces and interaction methods that work quickly and do not disrupt the multimedia session. Therefore, it is crucial that prior to developing novel user interfaces for QoS management, suitable PUI devices are identified.

Usability Study-A investigates the usability of five PUI devices for interacting with different user interface elements. The five PUI devices, namely: (a) Mouse & Keyboard, (b) Joystick, (c) Game Pad, (d) Steering Wheel with Pedals, and (e) Pen & Touchpad; were interfaced with nine Graphical User Interface (GUI) elements, namely: (i) Menu Item, (ii) Slide-bar, (iii) Radio Buttons, (iv) Check Box, (v) List Box, (vi) Combo Box, (vii) Push Button, (viii) Scrollbar, and (ix) Tab Control; for performing basic computer interaction functions [93].

6.2.2 User Interface Design

The user interface developed for this study comprised of different GUI elements grouped into screens that were demarcated by tabs, as shown in Figure 6-1.

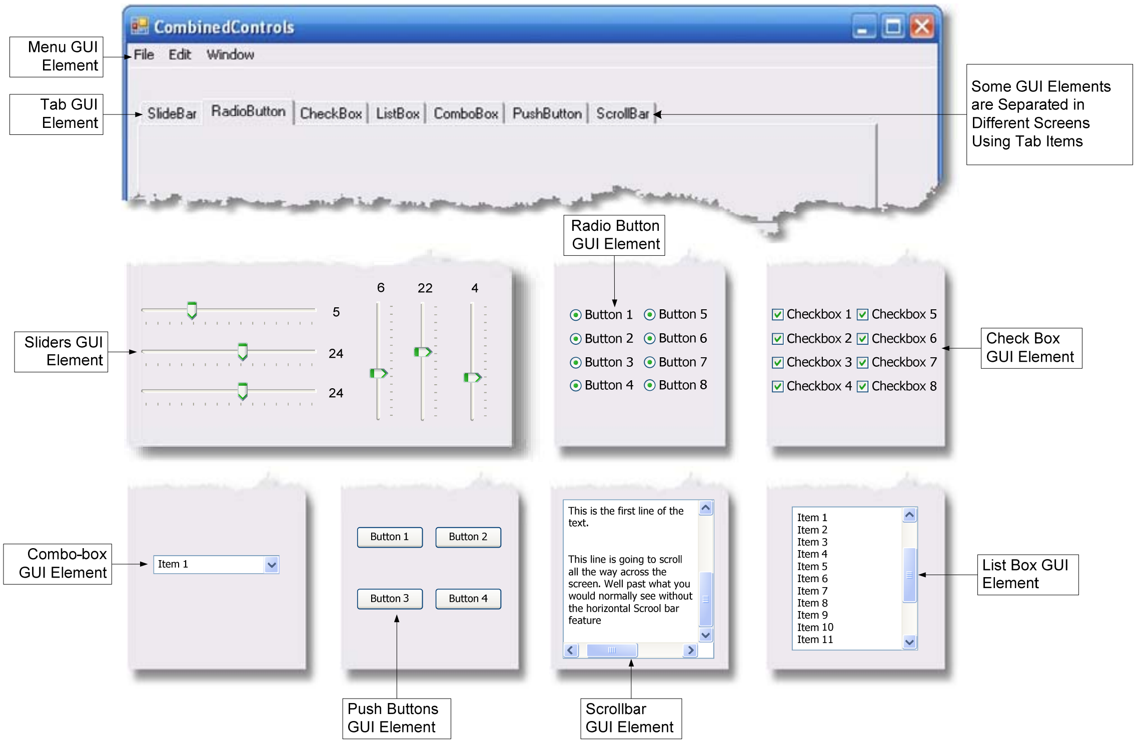


Figure 6-1: Usability Study A: User Interface,

Developed for the Convenience of Carrying Out Experiments Required for this Study

Each tab screen comprised of the following nine GUI elements:

- *Menu* displays a list of commands available to the user. ‘Menu items’ include: ‘File’, ‘Edit’, and ‘Window’.
- *Slide-Bar*, also called a track-bar control, enables a user to select a range of values by moving a slider.
- *Radio Buttons* are used to select one of several options, usually within a dialog box. A radio button contains a small circle with text next to it.
- *Check Box* comprises a list box combined with either a static control or edit control. The list-box portion of the control may be displayed at all times or may only drop down when the user selects the drop-down arrow next to the control.
- *List Box* is a scrollable list that contains a number of options or pieces of information. Multiple items can be visible at all times.

- *Combo Box* is a combination text box and list box. The list-box portion of the control may be displayed at all times or may only drop down when the user selects the drop-down arrow next to the control.
- *Push Button* is used to execute a particular command or function.
- *Scrollbar* enables a user to choose the direction and distance to scroll through information in a related window or list box.
- *Tab Control* defines multiple pages for the same area of a window or dialog box.

This user interface was specifically developed for the convenience of interfacing the five PUI devices with the different GUI elements as required for this study. Therefore, the design of the interface has no resemblance to the interfaces developed for QoS management, which are included in the forthcoming studies in this chapter.

The following five PUI devices were connected to the computer's USB port and used to interact with each of the nine GUI elements listed above.

- *Mouse & Keyboard* is the most prevalent method for HCI.
- *Joystick Controller* is a device designed for gaming purposes.
- *Game Pad* is a device that consists of four-way directional controller buttons that functions similar to a Joystick, however, without the hand-held stick.
- *Steering Wheel with Foot Pedals* is a device that consists of a steering wheel similar to that found in automobiles.
- *Pen & Touchpad* is a device that enables to draw directly on a computer screen.

6.2.3 Experiment Design

Aim:

This study aims to determine the most suitable Physical User Interface (PUI) device for controlling various Graphical User Interface (GUI) elements.

Objectives:

The deliverable objectives for this study are:

- Evaluate the usability of the (a) Mouse & Keyboard, (b) Joystick, (c) Game Pad, (d) Steering Wheel with Pedals, and (e) Pen & Touchpad; for interfacing with different GUI elements.
- Perform a Task Performance analysis for each device.
- Determine the preferred device based on user feedback.

Participants:

Ten participants were recruited and ranked based on the user ranking model presented in section 5.7. This study used three Advanced, five Intermediate, and two Beginner users.

Usability Evaluation Criteria:

Each PUI device was measured for: Ease of Use, Learnability, Usefulness, Effectiveness, Efficiency, Satisfaction, and Stress Factor. Participants provided feedback for each evaluation criterion based on the Likert scale, where for example 1=Very Difficult, 2=Difficult, 3=Neutral, 4=Easy, 5=Very Easy.

Task Performance was measured as a combination of average Task Completion Time and average Task Error Count. An error was recorded whenever the user performed an action not specified in the scenario of tasks.

Table 6-1 shows the full matrix of PUI devices, the Evaluation Criteria and Task Performance used in this study.

Table 6-1: Usability Study A: Evaluation Criteria and Task Performance Measures

PUI Device	Evaluation Criteria							Task Performance	
	Ease of Use	Learnability	Usefulness	Effectiveness	Efficiency	Satisfaction	Stress Factor	Task Completion Time	Task Error Count
1. Mouse and Keyboard	✓	✓	✓	✓	✓	✓	✓	✓	✓
2. Joystick	✓	✓	✓	✓	✓	✓	✓	✓	✓
3. Game-Pad Controller	✓	✓	✓	✓	✓	✓	✓	✓	✓
4. Steering Wheel with Foot Pedals	✓	✓	✓	✓	✓	✓	✓	✓	✓
5. Pen & Touchpad	✓	✓	✓	✓	✓	✓	✓	✓	✓

Test Scenario:

The testing scenario for study A comprised nine tasks for each PUI device, where each task involved using the PUI device to interact with one of the nine GUI elements. Each task required five actions performed on the GUI element, for example:

- Positioning the Slider Bars to five given values; and
- Click on five Radio Buttons.

6.2.4 Analysis

Figure 6-2 shows the average Task Completion Times for the five PUI devices and the three user types (Beginner, Intermediate, and Advanced). Figure 6-3 presents the average Task Error Count for the five devices and three user types, while Figure 6-4 presents the average usability ratings for the same.

Task Completion Time Analysis

Figure 6-2 shows a consistent trend where Beginner users recorded the highest Task Completion Time, followed by the Intermediate users. The Advanced users recorded the lowest Task Completion Time, as expected.

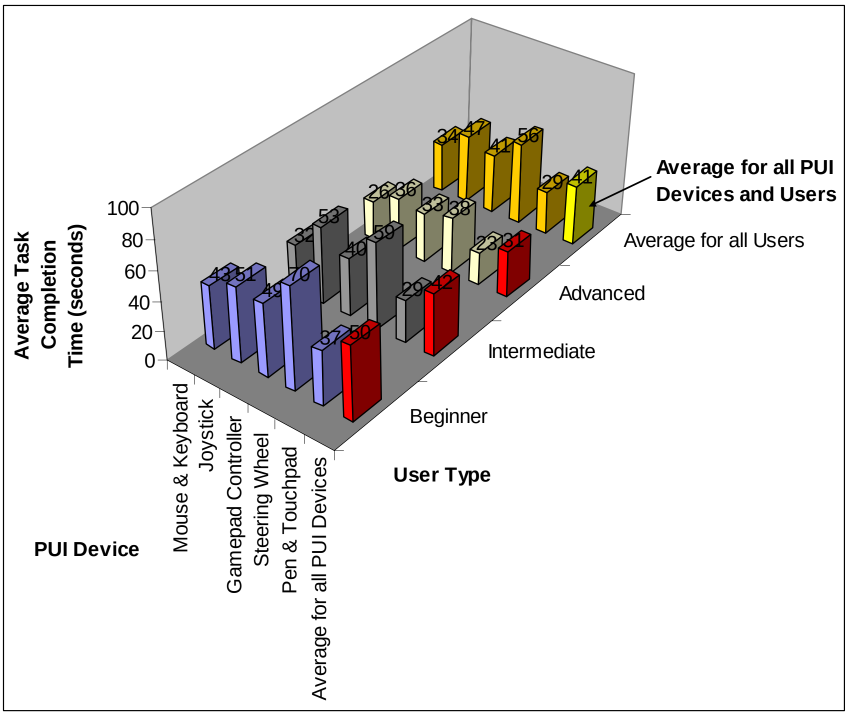


Figure 6-2: Usability Study A: Average Task Completion Times,
For the five PUI devices and the Three User Types

For all three user types, the Pen & Touchpad device recorded the lowest Task Completion Time, followed by the Mouse & Keyboard, Gamepad Controller, and Joystick, while the Steering Wheel recorded the highest Task Completion Time. This clearly indicates that the Pen & Touchpad is even more efficient than the commonly used Mouse & Keyboard combination.

Task Error Count Analysis

Figure 6-3 shows that Beginner users recorded lower number of errors for the Pen & Touchpad, followed by the Gamepad Controller and then the Steering Wheel. A high error count was recorded for the Joystick and, unexpectedly, for the Mouse & Keyboard. The facilitator observed that as the Mouse & Keyboard was used first in the experiment by all users, Beginner users were making more errors as they were familiarising themselves with the experiment.

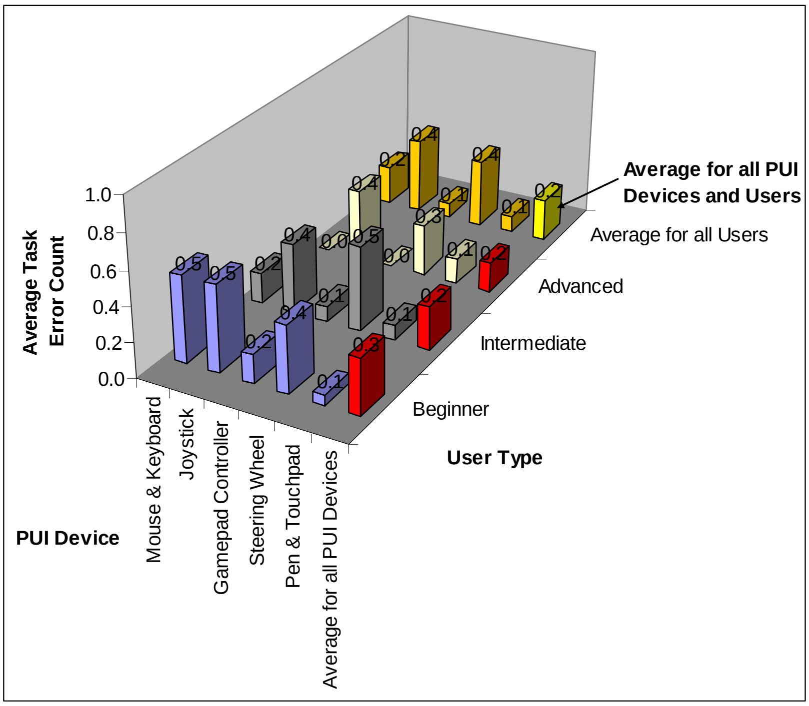


Figure 6-3: Usability Study A: Average Task Error Count, For the Five PUI devices and the Three User Types

Intermediate users made the lowest number of errors for the Pen & Touchpad and the Gamepad Controller, and marginally higher errors for the Mouse & Keyboard; while a high error count was recorded for the Joystick and Steering Wheel.

Advanced users made zero errors for the Mouse & Keyboard and the Gamepad Controller. Some errors were made for the Pen & Touchpad, while a high error count was recorded for the Joystick and Steering Wheel.

On average, Beginner users made the most number of errors followed by Intermediate and then Advanced users. Overall, the Pen & Touchpad and the Gamepad Controller recorded the lowest number of errors, followed by the Mouse & Keyboard; making them all suitable for Desktop Systems. The Joystick and Steering Wheel recorded a high error count as users found them cumbersome to use.

Usability Ratings Analysis

Figure 6-4 shows that on average Intermediate users gave good ratings for all devices, while Beginner and Advanced users gave marginally lower ratings.

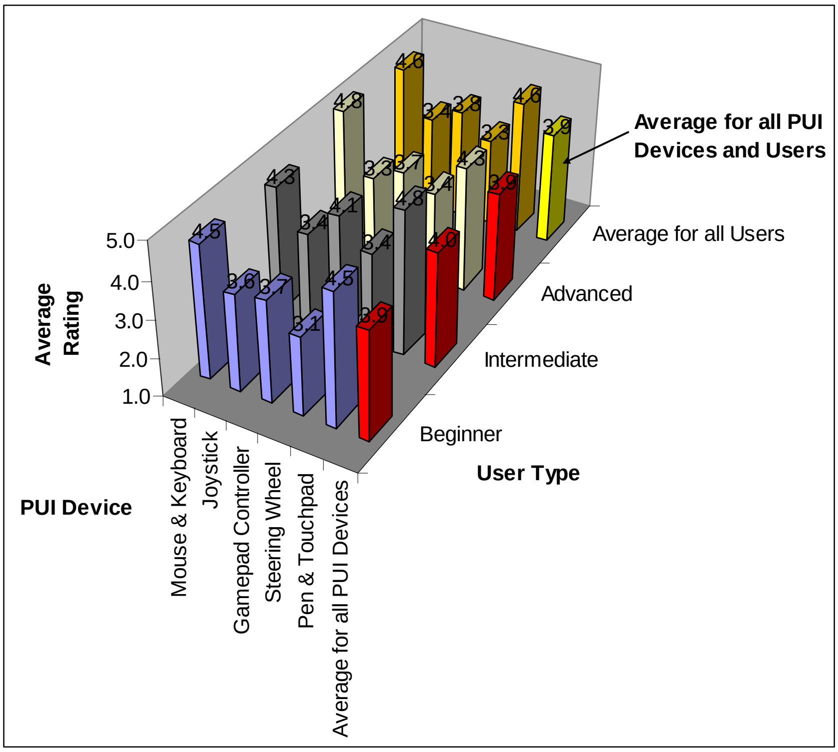


Figure 6-4: Usability Study A: Average Usability Ratings, For the five PUI devices and the Three User Types

Results clearly show that the Mouse & Keyboard and Pen & Touchpad devices were preferred by all three user types. The Gamepad controller was the second most preferred device, while the Joystick and Steering Wheel were second least and least preferred devices, respectively.

Participant Comments

Participants stated that it was cumbersome to position the cursor with the Steering Wheel and the Joystick. An Intermediate user failed to complete a task using the Steering Wheel as he/she found it too stressful, and gave up. On the other hand, some participants stated that the Mouse & Keyboard was easy to use, but they preferred the Pen & Touchpad as it was ‘different’ and ‘new’.

6.2.5 Conclusion

In conclusion, this study revealed the Pen & Touchpad to be most suitable for interacting with different GUI elements. The Mouse & Keyboard fared second best on all three measures.

It was decided that the Mouse & Keyboard device will be used as the primary Physical User Interaction (PUI) method for Desktop Computers, as this device is most widely available; and the Pen & Touchpad will be used as the primary PUI device for Portable Devices, being readily available on these.

6.3 Usability Study B: Methods for Feedback

6.3.1 Overview

This study evaluates the suitability of different methods to provide system feedback for real-time multimedia systems. System feedback that is given during a multimedia presentation should be relevant to the QoS issue that is to be reported, as users generally prefer minimal disruption to a communication session; informative and advisory alerts should get the user’s attention, while critical alerts should provide adequate feedback, such that the user can take effective action. Various user interface elements, namely: Audio Alert, Three Colour Alert (with & without Audio), Four Colour Alert (with & without Audio), and Pop-up Alert (with & without Audio) are investigated for providing system feedback in a real-time multimedia environment.

6.3.2 User Interface Design

The system feedback interfaces developed for this study used either audio alert only, or a combination of audio and visual feedback. A total of seven feedback methods were tested.

Audio Alert:

- *Audio Alert* – System feedback is given via audio only. This method includes a text-to-speech voice that vocalises system feedback responses.

Visual Alerts:

- *Three Colour Alert* – This is based on a three colour scheme. The significance of the feedback is indicated by a particular colour: red represents critical, yellow represents advisory, and green represents informative feedback. System Feedback is augmented with text. This feedback method is presented in Figure 6-5a.
- *Four Colour Alert* – This is similar to the previous feedback method. As shown in Figure 6-5b, the yellow light is replaced by a yellow and an orange light. Orange represents advice to increase requested quality, and yellow to decrease it.
- *Pop-up Alert* – This method gives the alert message in a dialogue box, which pops up in the current window. Response types are distinguished by different icons, and text is used for further description. Figure 6-5c shows the different alerts for this feedback method.

The three visual alert feedback methods were tested with and without the audio alert.

Participants watched a video presentation, while feedback alerts were given in real-time at random intervals. This simulated a real-time multimedia environment.

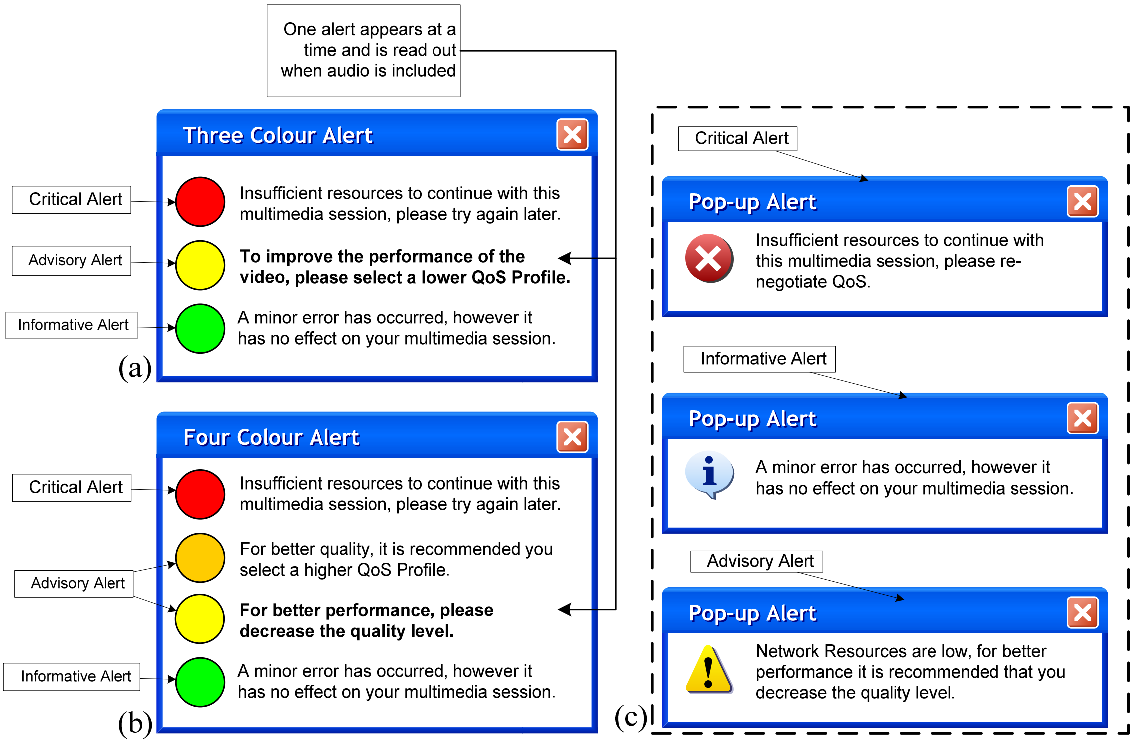


Figure 6-5: Usability Study B: Feedback Methods and Interfaces, (a) Three Colour Alert, (b) Four Colour Alert, (c) Pop-up Alert.

6.3.3 Experiment Design

Aim:

The aim of this study is to investigate the usability of different feedback methods and related user interface elements for real-time system feedback.

Objectives:

The deliverable objectives for this study are:

- Evaluate the usability of each feedback method and related user interface element.
- Perform Accuracy of Identification and Comprehension tests.
- Determine the preferred system feedback method.

Participants:

Ten participants were recruited and ranked based on the user ranking model presented in section 5.7. This study used five Advanced and five Intermediate users.

Usability Evaluation Criteria:

Each system feedback method was measured for: Accuracy of Identification, Ease of Perception and Comprehension, Usefulness, Satisfaction, and Stress Factor. Accuracy of Identification was measured as the number of alerts identified correctly. Participants provided feedback for the other criteria based on the Likert scale, where for example 1=Very Difficult, 2=Difficult, 3=Neutral, 4=Easy, 5=Very Easy. Table 6-2 shows the matrix of the Evaluation Criteria used to measure the usability for each system feedback method. In this study, each Evaluation Criterion was applied to each feedback method.

Table 6-2: Usability Study B: Evaluation Criteria

System Feedback Method / **Evaluation Criteria**	Ease of Perception	Ease of Comprehension	Usefulness	Satisfaction	Stress Factor	Accuracy of Identification
1. Audio Alert	✓	✓	✓	✓	✓	✓
2. Three Colour Alert *(without Audio)*	✓	✓	✓	✓	✓	✓
3. Three Colour Alert *(with Audio)*	✓	✓	✓	✓	✓	✓
4. Four Colour Alert *(without Audio)*	✓	✓	✓	✓	✓	✓
5. Four Colour Alert *(with Audio)*	✓	✓	✓	✓	✓	✓
6. Pop-up Alert *(without Audio)*	✓	✓	✓	✓	✓	✓
7. Pop-up Alert *(with Audio)*	✓	✓	✓	✓	✓	✓

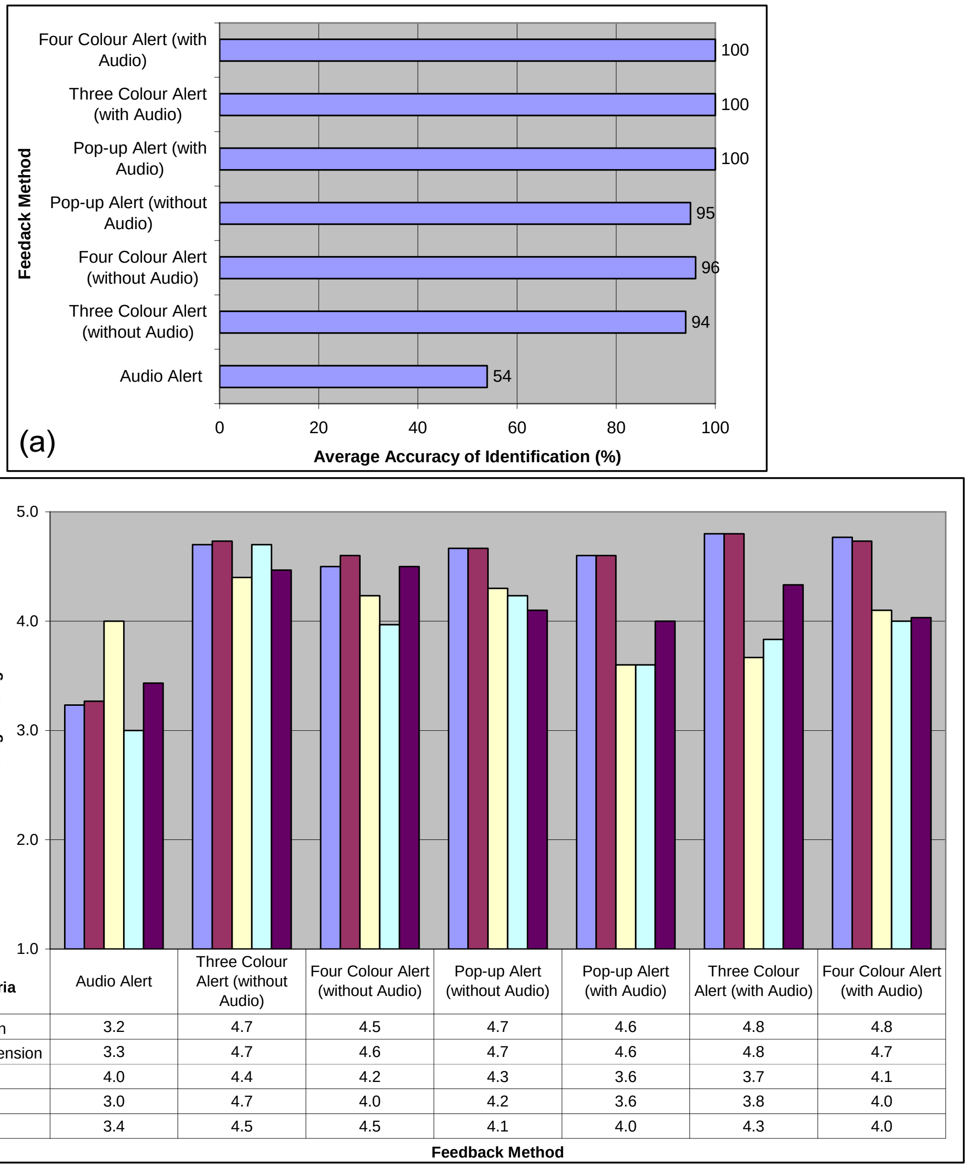
Test Scenario:

The testing scenario for this study comprised one task for each of the feedback methods. For each feedback method, ten alerts were given at random intervals in real-time while a video was playing. Critical, informative, and advisory alerts were given at random. The participant was required to identify the type of alert.

6.3.4 Analysis

Figure 6-6 shows the results for (a) Average Accuracy of Identification and (b) Average usability ratings for each system feedback method.

Accuracy of Identification **–** From Figure 6-6a, it can be seen that participants identified the alerts more accurately for: Three Colour Alert (with audio), Four Colour Alert (with audio), and Pop-up Alert (with audio). The combination of audio and visual alerts increased the participant's ability to identify the feedback correctly. Combined audio and visual alerts gave participants the option to either review the feedback visually or just listen to it. Audio Alert by itself scored the lowest. Participants reported that the sound in the video clip overpowered the voice in the audio feedback alert.



Evaluation Criteria	Audio Alert	Three Colour Alert (without Audio)	Four Colour Alert (without Audio)	Pop-up Alert (without Audio)	Pop-up Alert (with Audio)	Three Colour Alert (with Audio)	Four Colour Alert (with Audio)
Ease of Perception	3.2	4.7	4.5	4.7	4.6	4.8	4.8
Ease of Comprehension	3.3	4.7	4.6	4.7	4.6	4.8	4.7
Usefulness	4.0	4.4	4.2	4.3	3.6	3.7	4.1
Satisfaction	3.0	4.7	4.0	4.2	3.6	3.8	4.0
Stress Factor	3.4	4.5	4.5	4.1	4.0	4.3	4.0

Figure 6-6: Usability Study B: System Feedback Methods Analysis, (a) Average Accuracy of Identification (b) Average Usability Ratings

Figure 6-6b gives the average ratings given by participants for the following evaluation criteria.

Ease of Perception **–** In this criterion, the Three Colour Traffic Light Alert (with audio) is a marginal winner, making it the easiest to perceive, i.e. the user correctly recognises the purpose of the alert. Audio Alert feedback method was the most difficult to perceive, thus scoring the lowest rating.

Ease of Comprehension **–** All feedback methods, except Audio Alert, scored high ratings for ease of comprehension, i.e. the user could understand the meaning of the alert.

Usefulness – Results show reasonably high ratings for each feedback method, indicating that all methods were helpful in providing system feedback. The Three Colour Alert (with Audio) method is marginally higher; however, it cannot be unequivocally stated that this is the best option.

Satisfaction – Results show that participants were almost equally satisfied with all feedback methods, except for Audio Alert. This indicates that the participants found most of the methods pleasant to interact with [64].

Stress Factor – A high stress factor rating implies minimal stress on participants, while a low stress factor rating implies the opposite. Results indicate that all feedback methods, except Audio Alert, caused low stress to participants.

6.3.5 Conclusion

In conclusion, feedback methods that combine visual and audio alert are preferred for real-time multimedia systems.

6.4 Usability Study C: Static QoS Specification – Desktop Systems

6.4.1 Overview

This study evaluates different methods and related user interfaces for Static QoS Specification, i.e. request a quality level for a particular application prior to executing the multimedia session on a Desktop System. As multimedia systems are used by people with different technical knowledge, it is important to develop and test user interfaces that help them to request QoS easily.

Three different methods and related user interfaces, namely: a) Profile Only, b) Profile with Example, and c) Walkthrough Wizard have been investigated for Static QoS Specification. The development of these interfaces is based on the QoS taxonomy and Application Profile Model (APM) models introduced in sections 3.4 and 3.5.

6.4.2 User Interface Design

Menu-driven user interfaces were developed for users to specify QoS by selecting Application, QoS, and User profiles.

- *Application Profile* configures acceptable values for a collection of QoS parameters that are related to the type of application: Video on Demand (VoD), Video Conference and Telephony; as shown in Figure 6-7a.
- *QoS Profile* provides a pre-configured set of QoS values for an application based on three profiles: High, Medium and Low; as shown in Figure 6-7a.
- *User Profile* method relates to the user's technical abilities, i.e. Beginner, Intermediate and Advanced user.
    - *Beginner user* is limited to specifying QoS by profile only, as shown in Figure 6-7a. Such an interface is called *Limited Control* interface.
    - *Intermediate user* can specify discrete quality levels for audio and video, as shown in Figure 6-7b. Such an interface is called *Partial Control* interface.

- o *Advanced user* can specify individual values for different performance aspects (Quality Resolution, Temporal Facet, Cost Factor); giving them full access to all parameters, as shown in Figure 6-7c. Such an interface is called *Full Control* interface.

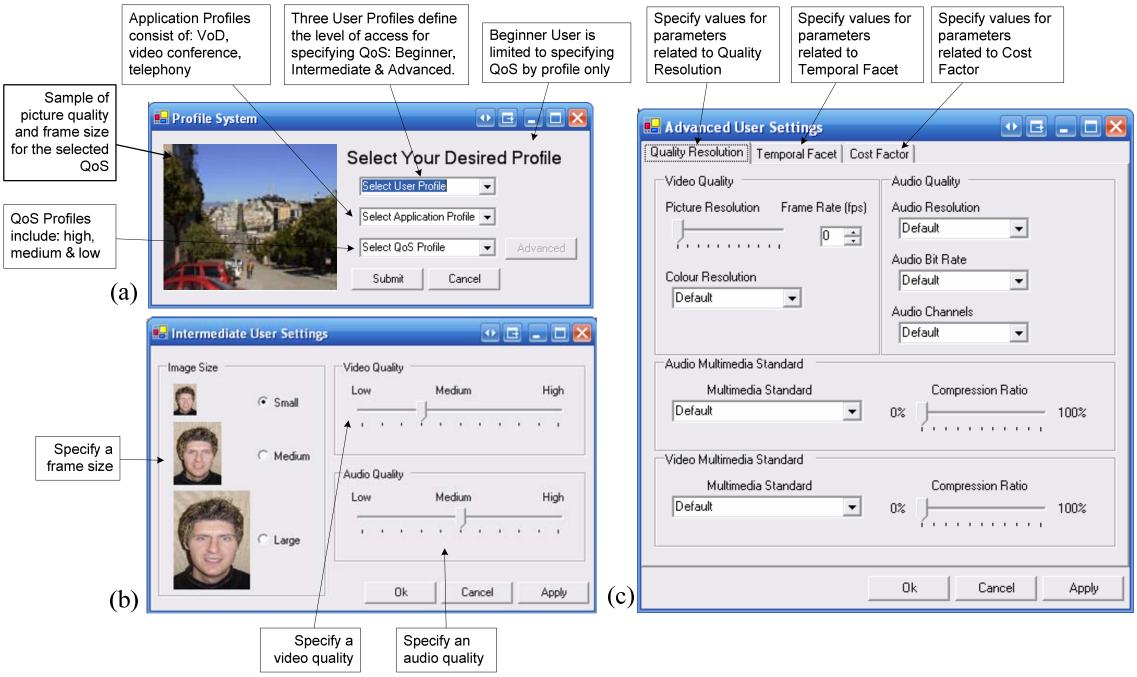


Figure 6-7: Usability Study C: Level of QoS Specification for User Profiles,
(a) Beginner (Limited Control), (b) Intermediate (Partial Control), (c) Advanced (Full Control).

Three methods for Static QoS Specification were tested. They are:

- *Profile Only* includes a menu-driven user interface that enables users to request QoS by selecting a User, Application and QoS profile, as shown in Figure 6-7a.

- *Profile with Example* is similar to the previous specification method; however, this method adds the ability to preview an example once the user has made their request, as shown in Figure 6-8. This should not be confused with *Specification by Example*, which requires the user to select QoS by viewing examples of different quality levels. Specification by example is recommended as further research as it introduces further usability issues.

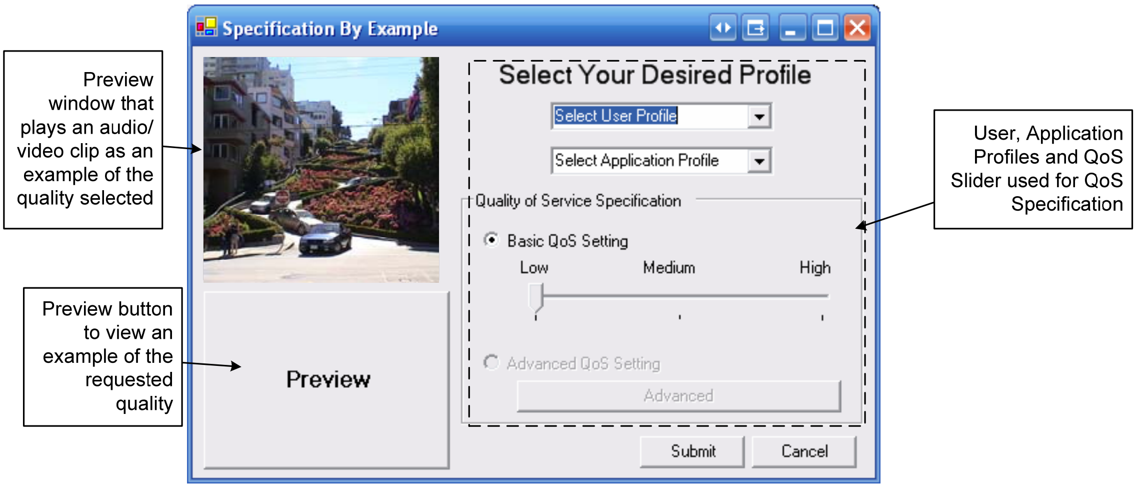
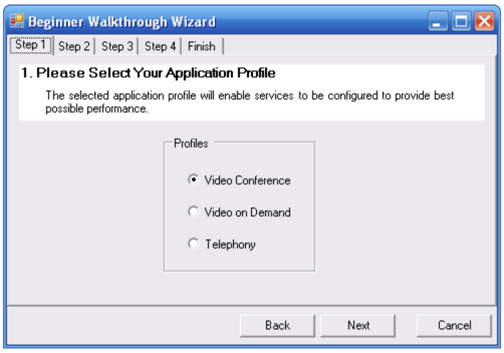
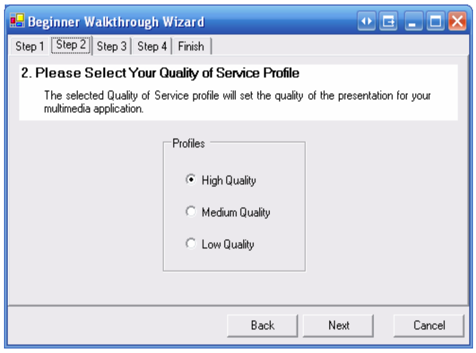


Figure 6-8: Usability Study C: Profiles with Examples Specification Method

- *Walkthrough Wizard* provides a step-by-step guide for specifying QoS, as shown in Figure 6-9. The number of steps used to specify QoS is related to the level of control. A Limited Control interface requires four steps, a Partial Control interface requires six steps, and Full Control Interface requires ten steps. In the final step, the user is able to preview an example of their selected quality.

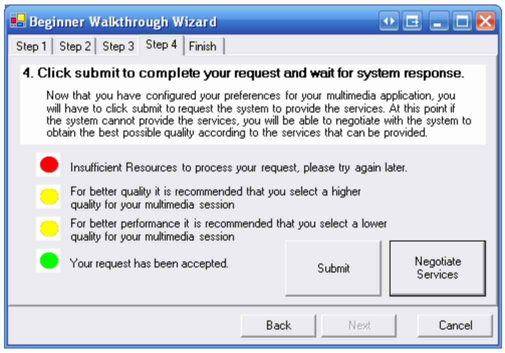


(a) Step 1: Select Application Profile



(b) Step 2: Select QoS Profile

(c) Step 3: Preview Example of Request



(d) Step 4: Submit Request

(e) Finish: Start Multimedia Session

Figure 6-9: Usability Study C: Walkthrough Wizard Specification Method, For Limited Control Interface

6.4.3 Experiment Design

Aim:

This study aims to determine the most suitable method and related user interface for Static QoS Specification.

Objectives:

The deliverable objectives for this study are:

- Evaluate the usability of Profile Only, Profile with Example, and Walkthrough Wizard methods.
- Determine the preferred specification method for Static QoS Specification.
- Evaluate the Task Performance for each specification method.

Participants:

Ten participants were recruited and ranked based on the user ranking model presented in section 5.7. This study used three Advanced, five Intermediate, and two Beginner users.

Usability Evaluation Criteria and Task Performance:

Each specification method was measured for: Ease of Use, Learnability, Usefulness, Effectiveness, Efficiency, Ease of Comprehension, Satisfaction, and Stress Factor. Participants provided feedback for each evaluation criterion based on the Likert scale, where for example 1=Very Difficult, 2=Difficult, 3=Neutral, 4=Easy, 5=Very Easy.

Task Performance was measured as a combination of average Task Completion Time and average Task Error Count. Table 6-3 shows that the full matrix of Specification Methods, the Evaluation Criteria and Task Performance was used in this study.

Table 6-3: Usability Study C: Evaluation Criteria and Task Performance Measures

	Evaluation Criteria								Task Performance	
Specification Method	Ease of Use	Learnability	Usefulness	Effectiveness	Efficiency	Ease of Comprehension	Satisfaction	Stress Factor	Task Completion Time	Task Error Count
1. Profile Only	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2. Profile with Example	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3. Walkthrough Wizard	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Test Scenario:

The test scenario comprised three tasks for each specification method. For each method, participants were required to specify QoS using the Limited, Partial and Full Control interfaces, irrespective of their own user type in the user ranking model. A total of nine tasks were completed for this study.

6.4.4 Analysis

Figure 6-10 shows the average Task Completion Times for the three user types (Beginner, Intermediate and Advanced) and the three QoS specification methods (Profiles Only, Profiles with Example, and Walkthrough Wizard) for: (a) Limited Control (specification with profiles only), (b) Partial Control (setting discrete values for audio and video quality), (c) Full Control (specifying values for individual parameters), and (d) overall average for a to c. Data from Figure 6-10 a to d are collated in Figure 6-11 and Figure 6-12 for the average Task Error Count, and the average Usability ratings, respectively. Figure 6-13, presents the overall results for this study.

Task Completion Time Analysis

Figure 6-10a shows that for Limited Control, the Profile Only method recorded the lowest Task Completion Time. This was followed by Profile with Example method, and the Walkthrough Wizard, which took the longest. This rise in Task Completion Time can be seen for Beginner, Intermediate as well as Advanced users. This indicates that the Profiles Only method is most suitable for the Limited Control interface.

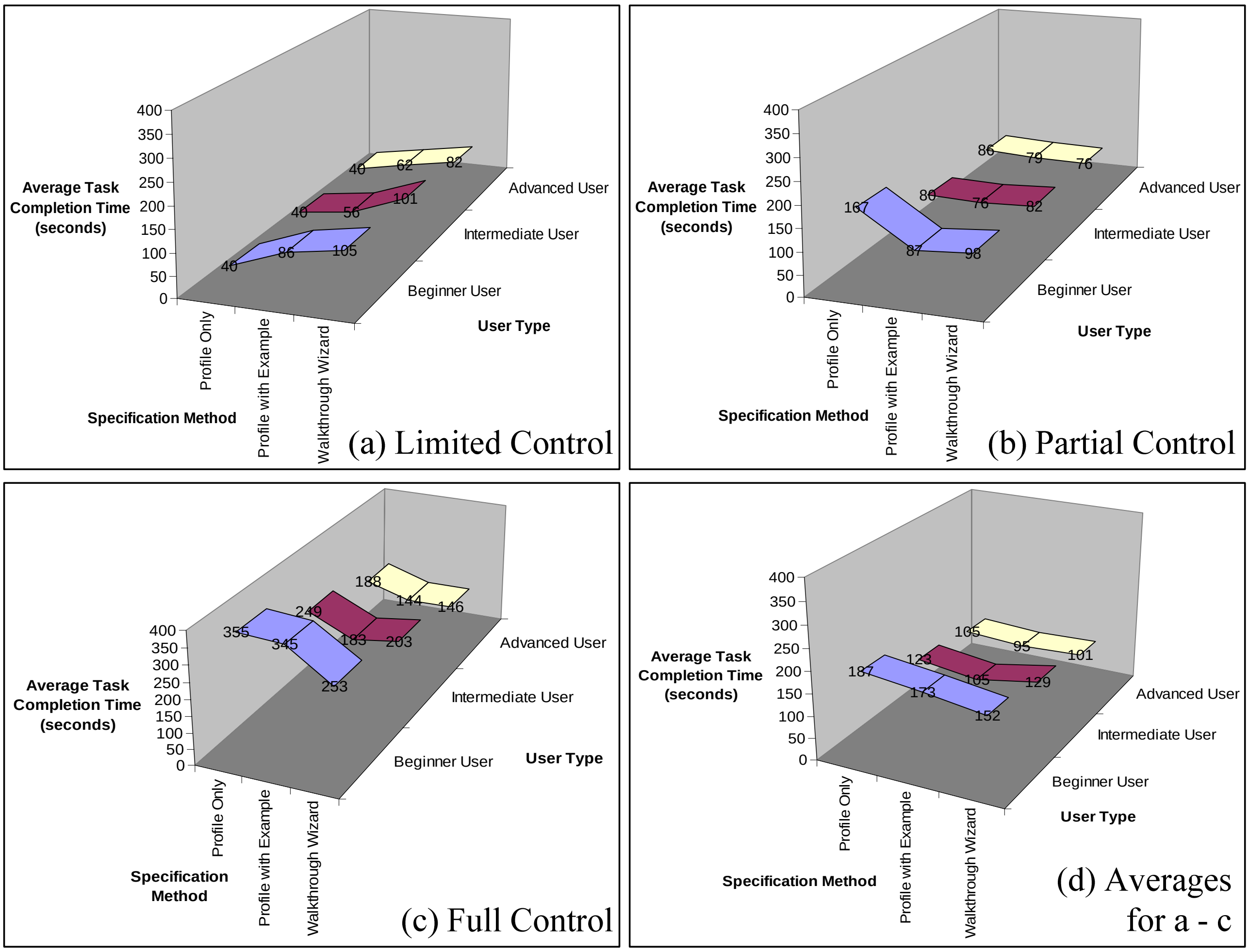


Figure 6-10: Usability Study C: Average Task Completion Times, For each Specification Method and User Type for: (a) Limited Control, (b) Partial Control, (c) Full Control, and (d) Overall Average.

For Partial Control, Figure 6-10b shows that Beginner users took the longest to specify QoS using the Profile Only method. Profile with Example and Walkthrough Wizard methods do not show much change in the Task Completion Times for all user types.

For Full Control, results given in Figure 6-10c show a decline in Task Completion Times from Profiles Only to Profiles with Example, and Walkthrough Wizard methods for all three user types (Beginner, Intermediate and Advanced).

Figure 6-10d shows that on average Beginner users had the lowest Task Completion Time for Walkthrough Wizard, and a high Task Completion Time for the Profile Only method. No significant change can be seen for all three methods for Intermediate and Advanced users. Overall, Advanced users completed QoS specification the quickest for all three specification methods, and levels of control. Intermediate users took longer, and Beginner users took the longest, as expected.

Task Error Count Analysis

Figure 6-11a shows that for Limited Control, some errors were made by Beginner and Intermediate users for the Profile with Example method. Advanced users made no errors.

For Partial Control, Figure 6-11b shows a high number of errors made by Beginner users for the Profile Only method; followed by a steep decrease for the Profile with Example method, and then a small increase for the Walkthrough Wizard method. This indicates that Beginner users found it difficult using the Profile Only method, but less cumbersome for the Profile with Example and Walkthrough Wizard methods. An improvement on the number of errors can be seen from Profiles Only to Profiles with Example, and Walkthrough Wizard methods for Intermediate and Advanced users. Overall, results show that the Profile with Example and Walkthrough Wizard methods were less problematic for the Partial Control interface.

Results given in Figure 6-11c show Advanced users to have little trouble with all three methods for specifying QoS using the Full Control interface. The gradual decline from Profiles Only to Profile with Example and Walkthrough Wizard indicates an improvement for Intermediate users, i.e. Walkthrough Wizard method to be less cumbersome than the other two methods. An unexpected phenomenon can be seen for Beginner users, in such that there is a steep increase in the number of errors from Profiles Only to Profile with Example and Walkthrough Wizard. It was expected that the Walkthrough Wizard method would reduce the number of errors; however, results show that this method turned out to be more cumbersome.

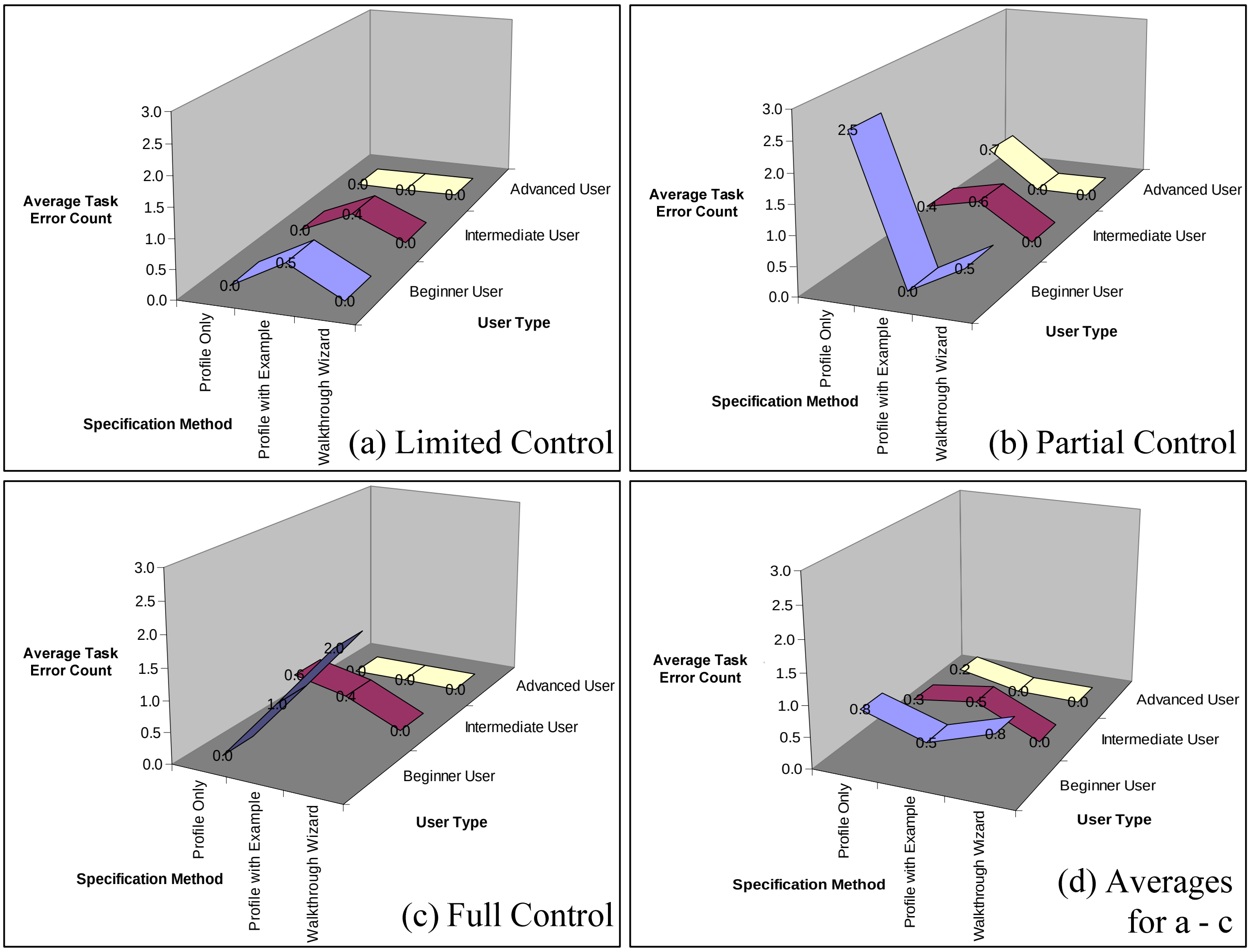


Figure 6-11: Usability Study C: Average Task Error Count,
For each Specification Method and User Type for: (a) Limited Control, (b) Partial Control, (c) Full Control, and (d) Overall Average.

Overall, Figure 6-11d shows that on average Advanced users experienced almost no difficulties, and Intermediate users faced some difficulties when using all three methods with the three levels of control. Beginner users found using each method to be cumbersome for each level of control. It can be deduced that all three methods are suitable for Intermediate and Advanced users; however the Profile with Example is more suitable for Beginner users.

Usability Ratings Analysis

Figure 6-12a shows the combined evaluation criteria ratings for each user type and specification method for Limited Control. The average ratings for these methods do not vary much between user types. Beginner users gave a slightly higher rating for the Profile Only method. Intermediate Users gave the highest rating for the Profile with Example method. Advanced users show a decline in the ratings from the Profile Only method to the Profile with Example, and the Walkthrough Wizard methods.

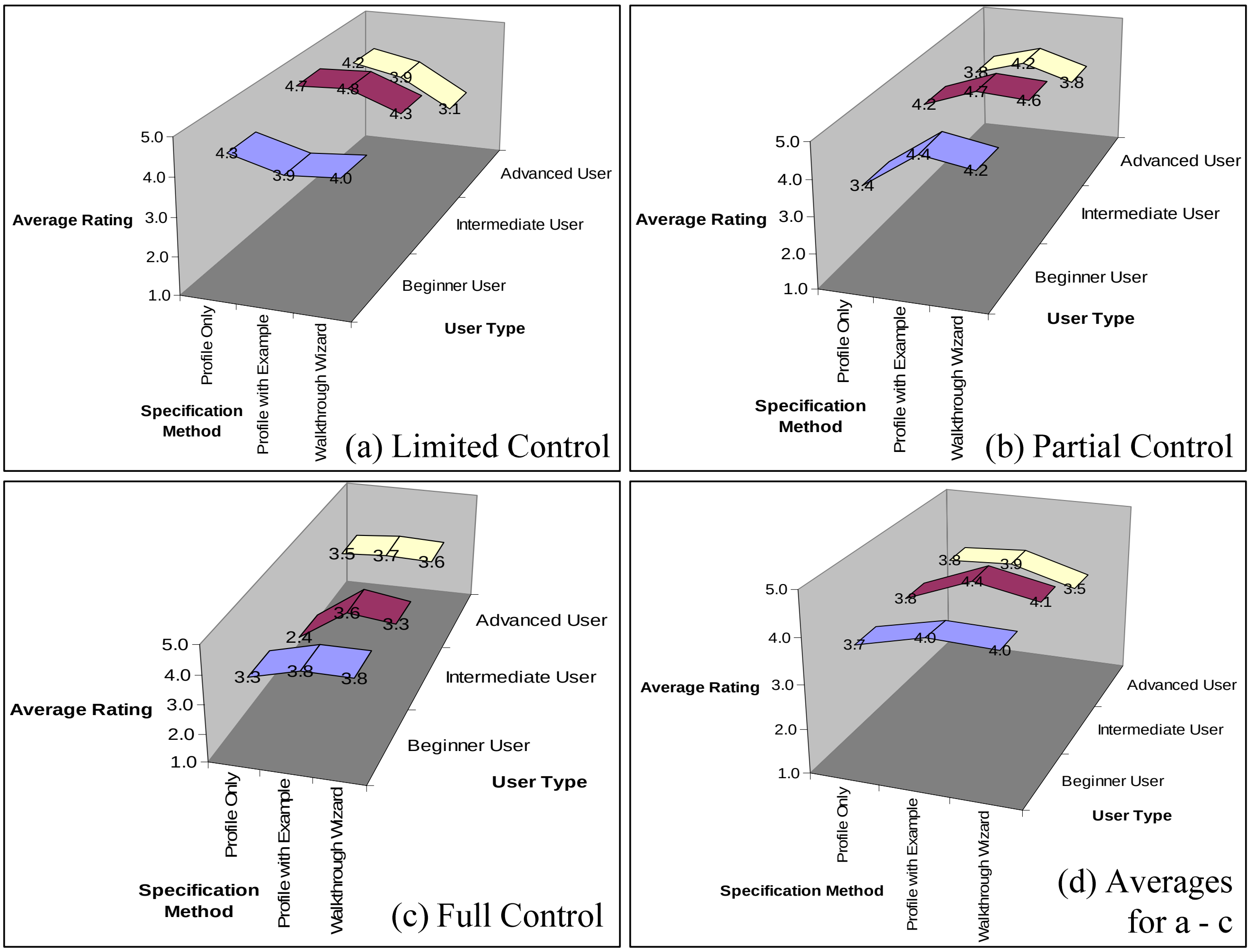


Figure 6-12: Usability Study C: Average Usability Ratings, For each Specification Method and User Type for: (a) Limited Control, (b) Partial Control, (c) Full Control, and (d) Overall Average.

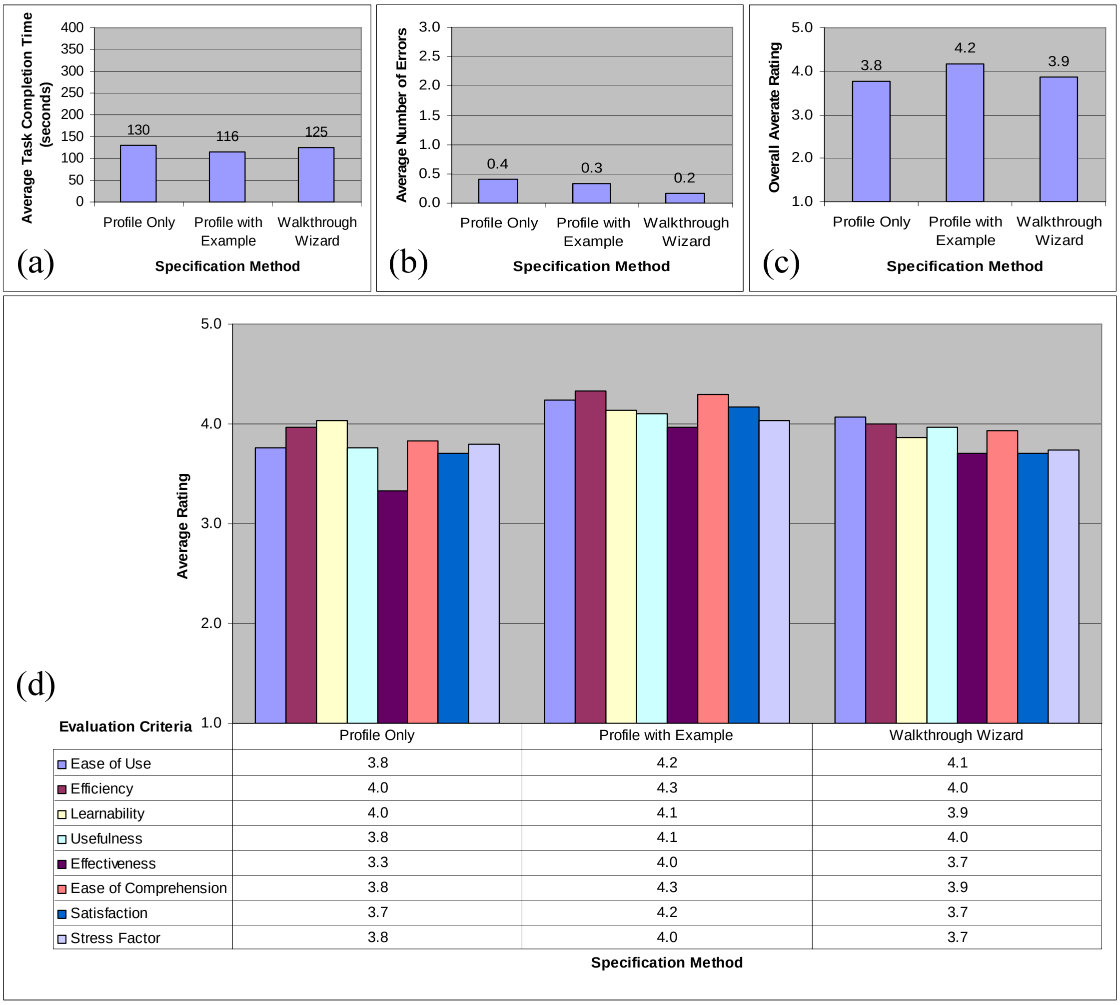
For Partial Control, results given in Figure 6-12b indicate that all three user types preferred the Profile with Example method. The Walkthrough Wizard method was the second most preferred.

Results presented in Figure 6-12c show that for Full Control, all methods received similar ratings for Beginner and Advanced users. Intermediate users provided similar ratings for Profile with Example and Walkthrough Wizard, while the Profile Only method was rated the lowest.

The overall averages, given in Figure 6-12d, show a similar trend for all user types. Profile with Example is the most preferred method, followed by the Walkthrough Wizard and the Profile Only methods.

Combined Averages for Usability Study C

Figure 6-13 shows the overall analysis for this study, where results are combined for the three user types as well as the level of control.



Evaluation Criteria	Profile Only	Profile with Example	Walkthrough Wizard
Ease of Use	3.8	4.2	4.1
Efficiency	4.0	4.3	4.0
Learnability	4.0	4.1	3.9
Usefulness	3.8	4.1	4.0
Effectiveness	3.3	4.0	3.7
Ease of Comprehension	3.8	4.3	3.9
Satisfaction	3.7	4.2	3.7
Stress Factor	3.8	4.0	3.7

Figure 6-13: Usability Study C: Overall Usability Analysis,
For each Specification Method showing (a) Average Task Completion Time, (b) Average Task Error Count, (c) Combined Usability Ratings, (d) Ratings for each Evaluation Criterion.

Task Completion Time: Results given in Figure 6-13a show that the Profile with Example method recorded the lowest Task Completion Time followed by Walkthrough Wizard and then the Profile Only method.

Task Error Count: Results given in Figure 6-13b show participants to have made the least number of errors with the Walkthrough Wizard, somewhat higher for the Profiles with Example, and the highest for the Profile Only method.

Combined Usability Ratings: Results given in Figure 6-13c show the Profile with Example method as the most preferred, and Profile Only least preferred.

Ratings for each Evaluation Criterion: Results given in Figure 6-13d show that on every evaluation criterion, the Profile with Example method rated the highest.

It can be deduced that the Profile with Example method proves to be the most suitable method to specify QoS for all user types and levels of control.

Participant Comments

Participants commented that for the Profile with Example method, they liked the idea of being able to preview their QoS request and found it easy to navigate through the options. Participants liked the Profile Only user interface, but preferred a preview option. Participants discovered that the Walkthrough Wizard was very structured and found it difficult to make changes to their QoS request if they changed their mind, as this required them to backtrack through each step, thus making it more cumbersome.

6.4.5 Conclusion

It can be deduced that Profile with Example method is most suitable for Static QoS Specification. This conclusion is made based on the following observations:

Task Completion Time: On average Intermediate and Advanced users performed static QoS specification using the Profile with Example method marginally quicker than for the Profile Only and Walkthrough Wizard methods. Beginner users were a little quicker with the Walkthrough Wizard. The difference in the task completion time is very small, which cannot justify as the Walkthrough Wizard to be a clear winner.

Task Error Count: Some errors were made by Beginner and Intermediate users when using the Profile with Example method. Advanced users did not make any errors for this method. Beginner users made some errors with the Walkthrough Wizard method, whereas Intermediate and Advanced users did not. The Walkthrough Wizard method should, technically, be the winner in this category. However, few errors were made for the Profile with Example method, indicating that this method was easy to use.

Participant Rating: All three user types rated the Profile with Example method as the most preferred.

Combined Averages: By combining the averages for all users, the Profile with Example method received the lowest Task Completion Time, a moderate error count, and the highest Usability Ratings.

Thus far, this study has revealed that the Profile with Example method is appropriate for Static QoS Specification on Desktop Systems.

6.5 Usability Study D: Static QoS Negotiation – Desktop Systems

6.5.1 Overview

This study evaluates the usability of a user interface for Static QoS Negotiation. This user interface implements a User-Initiated QoS Negotiation method for negotiating quality and cost prior to executing a multimedia session. Current multimedia communication systems do not provide mechanisms for users to negotiate QoS against cost and often fail to deliver the desired QoS even when the user is willing to pay for it.

The User-Initiated QoS Negotiation interface includes the following interaction elements (as shown in Figure 6-14): (a) Quality Specification Widget, (b) System Feedback Widget, (c) User Action Widget, (d) Cost Box, and e) Preview Box. Task modelling techniques were used for developing the user interface, as given in section 3.8.7.

6.5.2 User Interface Design

The development of the User-Initiated QoS Negotiation interface includes widgets for user interaction and system feedback. The design of these widgets and their interaction method were based on feedback given from Usability Studies A and B.

User-Initiated QoS Negotiation

To complete a successful QoS negotiation, the user and the system must come to an agreement for a particular quality and cost. In the case of User-Initiated QoS Negotiation, the user first makes a request for a particular quality. The system then accepts or denies the request. If the request is accepted, then the cost is negotiated. If the system denies the request, then the cost and quality are re-negotiated. The multimedia session is executed only when an agreement on quality and cost is reached. For the developed user interface, the system attempts to provide the best possible quality for the negotiated cost.

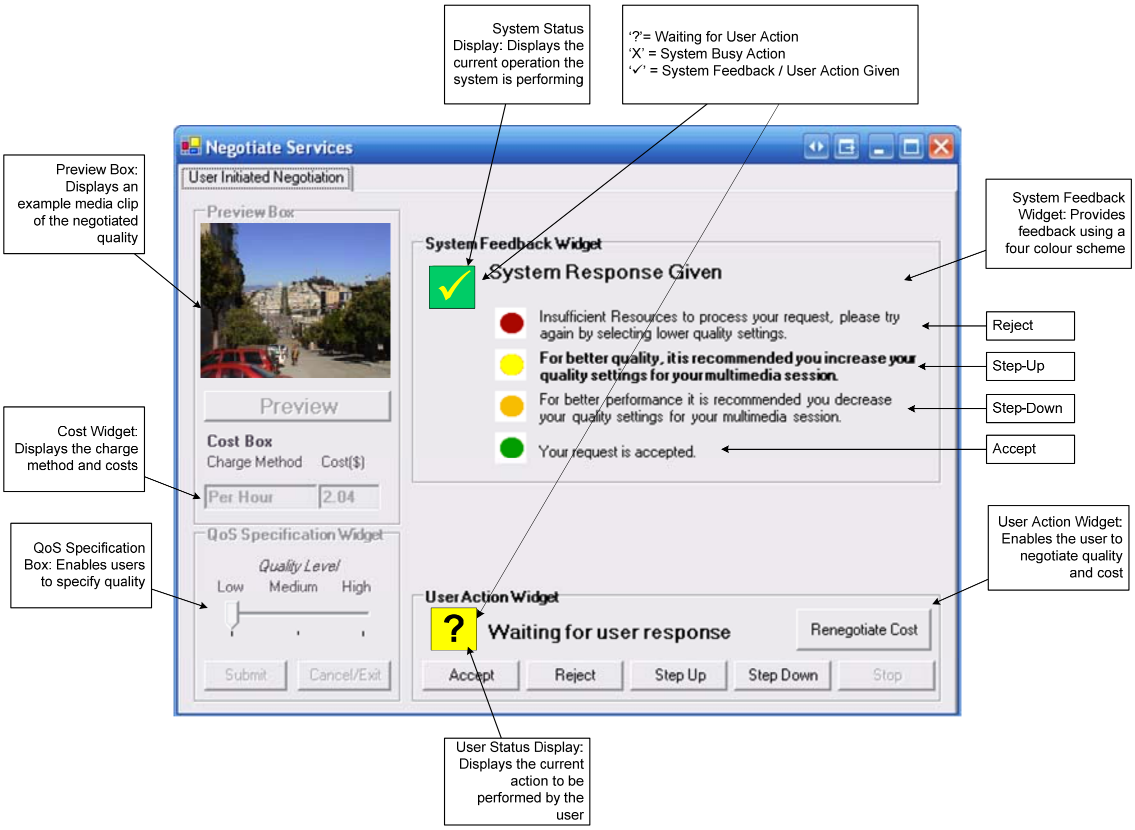


Figure 6-14: Usability Study D: Static QoS Negotiation Interface

User-Initiated QoS Negotiation Interface

The interaction process for the User-Initiated QoS Negotiation Interface is based on the flow chart given in Figure 5-2a. This user interface comprises the following user interaction and system feedback elements.

- *QoS Specification Widget* enables the user to specify a quality level to be negotiated.
- *System Feedback Widget* displays the system responses for the quality requested by the user. A Four Colour Scheme is used to display the following system feedback types:
    - *Reject* – A critical response (represented by the colour red) that indicates that the system cannot accommodate for the requested quality.
    - *Step-up* – An advisory response (represented by the colour yellow) that is used to advise the user to increase the quality level.
    - *Step-down* – An advisory response (represented by the colour orange) that is used to advise the user to decrease the quality level.
    - *Accept* – An informative response (represented by the colour green), which indicates that the system accepts the QoS request.

  A *System Status Display* is included in this widget, which uses icons accompanied with text to inform the user of the current function the system is performing. The icon denoted by the symbol: '?', indicates 'waiting for user action'. 'X' indicates a system busy response. '✓' indicates that the system has responded.
- *User Action Widget* enables the user to interact with the system during negotiation using the buttons: *Accept*, *Reject, Step-up* (increase quality), *Step-down* (decrease quality), or *Stop*. A *Re-negotiate Cost* button initiates re-negotiation of the cost for a given quality level based on the current condition of the network resources. A *User Status Display* is included in this widget, which prompts the user when they are required to take action. The function of this feedback element is identical to the *System Status Display*.
- *Cost Box* displays the charge method and the cost value in dollars for the negotiated quality.

- *Preview Box* enables the user to preview an example of the negotiated quality.

6.5.3 Experiment Design

Aim:

This study investigates the usability of the User-Initiated QoS Negotiation interface for Static QoS Negotiation.

Objectives:

The deliverable objectives for this study are:

- Evaluate the usability of the User-Initiated QoS Negotiation interface for Beginner, Intermediate and Advanced users.
- Evaluate the Task Performance of completing a successful negotiation for each user type.
- Evaluate the usability of the User Action Widget for user control.
- Evaluate the usability of the System Feedback Widget, the System Status Display, User Status Display and the Cost Box for system feedback.

Participants:

Ten Participants were recruited and ranked based on the user ranking model presented in section 5.7. This study used three Advanced, five Intermediate, and two Beginner users.

Evaluation Criteria and Task Performance:

In this study, the usability for the User-Initiated QoS Negotiation interface, the user control and system feedback elements were investigated for Static QoS Negotiation. Table 6-4 shows which usability measures (Evaluation Criteria and Task Performance) were recorded for each of the user interaction methods.

Usability for the User-Initiated QoS Negotiation interface was measured using the entire set of evaluation criteria (Table 6-4). This assessed the efficacy of the user interface and as well as the negotiation process. Task Performance was measured as a combination of average Task Completion Time and average Task Error Count for the entire negotiation process, and not for individual user interface elements.

The usability of the individual elements for user input (user control) and system output (system feedback) is measured with a limited set of criteria. Only criterions that have clear relevance to the usability of specific user interface elements are used, to avoid collecting redundant data.

Ease of Perception and Ease of Comprehension criteria apply to system outputs only, therefore, they were not used for measuring the usability of the User Action Widget. Satisfaction and Stress Factor criteria apply more to an entire task and therefore were not used for individual widgets, and display boxes. Ease of Use, Learnability, Effectiveness and Efficiency measures apply to user input elements, and therefore, were not used for evaluating system feedback user interface elements.

Table 6-4: Usability Study D: Evaluation Criteria and Task Performance Measures

<table>
<tr><th rowspan="2">Interaction Methods</th><th colspan="9">Evaluation Criteria</th><th colspan="2">Task Performance</th></tr>
<tr><th>Ease of Use</th><th>Learnability</th><th>Usefulness</th><th>Effectiveness</th><th>Efficiency</th><th>Ease of Perception</th><th>Ease of Comprehension</th><th>Satisfaction</th><th>Stress Factor</th><th>Task Completion Time</th><th>Task Error Count</th></tr>
<tr><td>1. User-Initiated Negotiation User Interface</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td></tr>
<tr><td>2. User Action Widget</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td colspan="2">Do not apply to inputs</td><td colspan="2" rowspan="5">Do not apply to individual widgets and display boxes.</td><td colspan="2" rowspan="5">Measured only for the complete negation process</td></tr>
<tr><td>3. System Feedback Widget</td><td colspan="2" rowspan="4">Do not apply to system feedback user interface elements</td><td>✓</td><td colspan="2" rowspan="4">Do not apply to system feedback user interface elements</td><td>✓</td><td>✓</td></tr>
<tr><td>4. System Status Display</td><td>✓</td><td>✓</td><td>✓</td></tr>
<tr><td>5. User Status Display</td><td>✓</td><td>✓</td><td>✓</td></tr>
<tr><td>6. Cost Box</td><td>✓</td><td>✓</td><td>✓</td></tr>
</table>

Test Scenario:

The test scenario comprised of six negotiation tasks. Each task involved performing QoS negotiations to obtain a particular quality and cost range. The task instructed participants to: a) request the quality b) negotiate until mutually acceptable quality is achieved, and c) negotiate until cost falls in the given range.

The system was programmed to guide the user in obtaining a best quality level, based on the available network resources and the cost the user is willing to pay. A task was deemed completed only when negotiations were concluded successfully.

6.5.4 Analysis

Figure 6-15 shows average Task Completion Times for each task against the three user types (Beginner, Intermediate and Advanced); and combined averages for each task, each user type, and for the overall experiment. Data from Figure 6-15 are collated in Figure 6-16 for average Task Error Count, and Figure 6-17 includes usability ratings for the: (a) User-Initiated QoS Negotiation Interface, (b) User Action Widget, (c) System Feedback Widget, (d) User Status Display, (e) System Status Display, and (f) Cost Box. Figure 6-18 shows the combined results for: (a) Usability Ratings for each User Interaction Method, and (b) Task Error Count and Task Completion Times

Task Completion Time Analysis

Figure 6-15 shows a gradual decline in Task Completion Times from *Task 1* to *Task 6* for each user type. It is 'not unexpected' that users take longer in the beginning as they familiarise with the interface and the negotiation process. It is noticeable that the trend in the average Task Completion Time is similar for all user types.

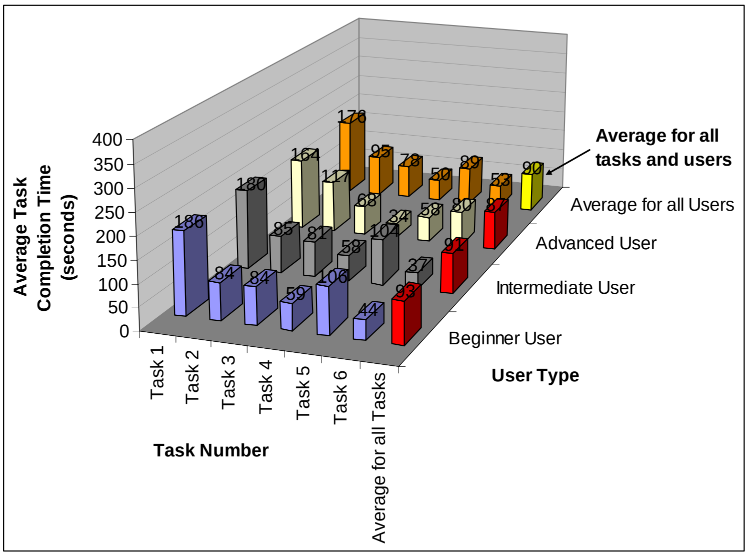


Figure 6-15: Usability Study D: Average Task Completion Times, For the User-Initiated QoS Negotiation Interface

There is not much difference in the average Task Completion Times between user types. A slight increase is shown for Beginner and Intermediate users in completing *Task 5*, and Advanced users for completing *Task 6*. The reason for this anomaly was recorded by the facilitator. The participants became complacent towards the end of the experiment, which affected their performance.

Task Error Count Analysis

Figure 6-16 shows that as expected all users made more errors towards the beginning of the experiment and fewer towards the end. Beginner users made more errors in the beginning than Intermediate and Advanced users. However, a substantial decrease in the number errors is seen for Beginner users from Task 4 onwards.

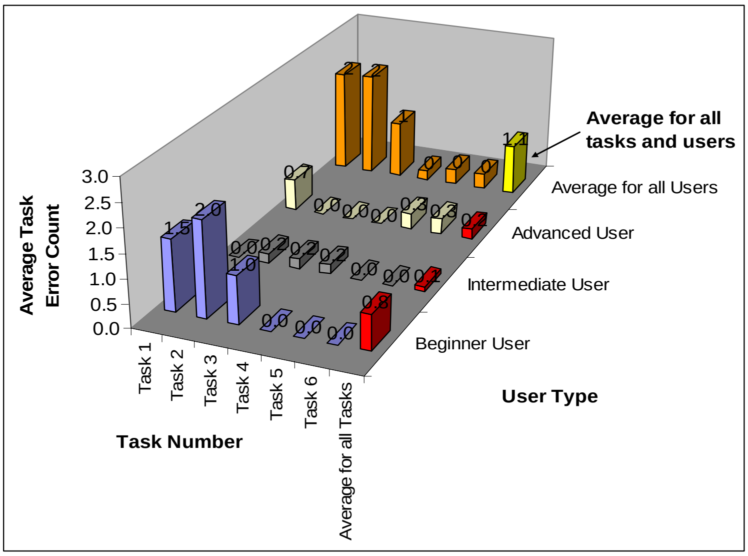


Figure 6-16: Usability Study D: Average Task Error Count, For the User-Initiated QoS Negotiation Interface

A declining trend is present for all users. Beginner and Intermediate users performed as well as Advanced users towards the end of the experiment. On average, Beginner users made the highest number of errors, Intermediate and Advanced users made very few errors.

Usability Ratings Analysis

Figure 6-17a shows the usability ratings given by Beginner, Intermediate and Advanced users for the User-Initiated QoS Negotiation interface. The average ratings given for each criterion do not vary much between the three user types. All ratings average to a value of 3.3, which implies marginally positive feedback according to the Likert scale.

A similar trend to that of Figure 6-17a is present for the results shown in Figure 6-17b – f, which includes evaluation ratings for the (b) User Action Widget, (c) System

Feedback Widget, (d) User Status Display, (e) System Status Display and (f) Cost Box. These results show equally positive feedback given by Beginner, Intermediate and Advanced users for each user interaction method.

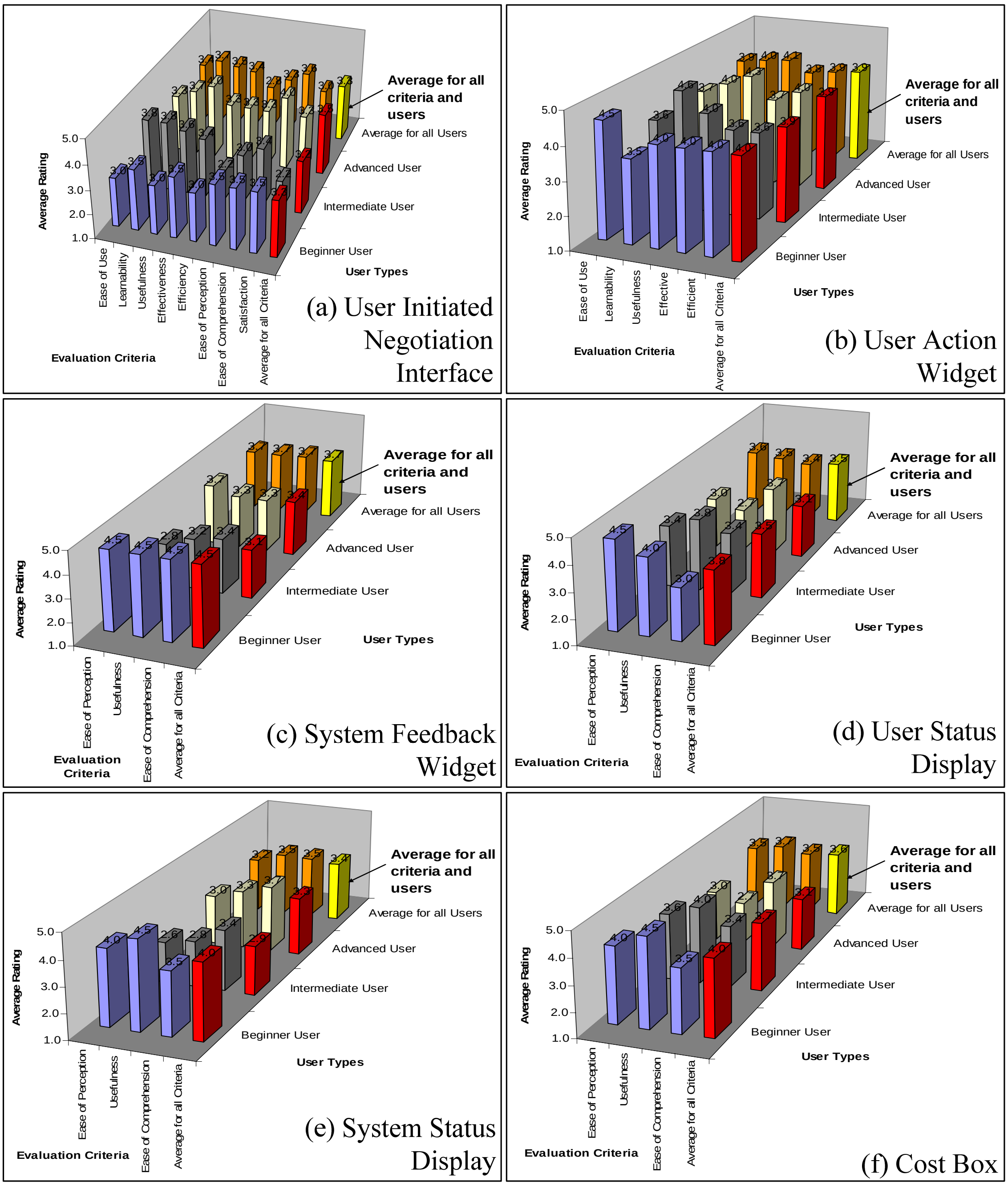


Figure 6-17: Usability Study D: Usability Ratings for each User Interaction Method, (a) User-Initiated QoS Negotiation Interface, (b) User Action Widget, (c) System Feedback Widget, (d) User Status Display, (e) System Status Display, (f) Cost Box.

Combined Averages for Usability Study D

Figure 6-18a shows the combined average ratings from the evaluation criteria set for the User-Initiated QoS Negotiation Interface as well as the specific interface elements: User Action Widget, System Feedback Widget, User Status Display, System Status Display and Cost Box. These results indicate acceptance of the User-Initiated QoS Negotiation process and interface.

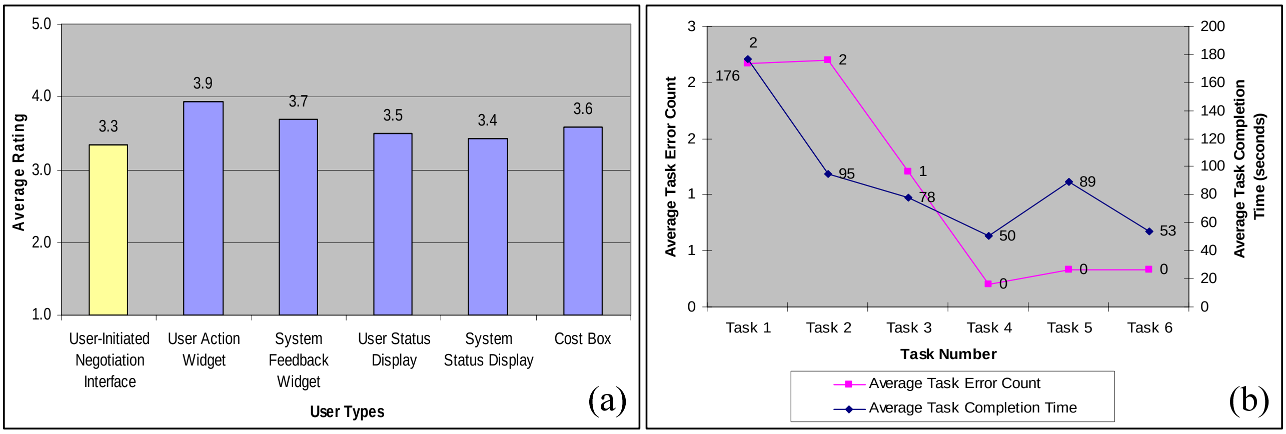


Figure 6-18: Usability Study D: Overall Usability Analysis, (a) Average User Interaction Method Ratings, (b) Average Task Error Count & Average Task Completion Times

Figure 6-18b shows the combined averages of Task Completion Time and Task Error Count for all users. The declining trend for the Task Completion Time and Task Error Count clearly show improvement for all users, indicating good learnability for the same. On average, it took less than four tasks for participants to become familiar with the user interface and perform negotiations with little difficulty. The total time for these four tasks adds up to less than seven minutes. A rubric for good interface design is that the users should be able to learn it in less than ten minutes [94], in which this interface has achieved.

Participant Comments

Participants commented that they were intrigued by the idea of being able to negotiate quality and cost with the system. Participants liked the 'look' and 'feel' of the user interface; however they felt that the system was a little restrictive when giving advisory feedback. Participants wanted the freedom to conclude a successful negotiation for a quality that they felt was suitable, which did not comply with the advisory system feedback. Participants suggested that the *Step-up* and *Step-down* buttons should use names that better defined their functionality.

6.5.5 Conclusion

It can be deduced that the User-Initiated QoS Negotiation process and Interface is suitable for Static QoS Negotiation. This conclusion is made based on the following observations:

Task Completion Time: Beginner, Intermediate and Advanced users improved on the time taken to achieve a successful negotiation.

Task Error Count: Each user type on average recorded a low Task Error Count. All users reduced the number of errors as they progressed through the tasks.

Participant Rating: Positive feedback was given for all user interaction methods, indicating that all users found the user interface suitable for negotiating QoS.

Combined Averages: Results given in Figure 6-18a confirm overall user acceptance. Results given in Figure 6-18b indicate that participants became familiar with the user interface and the negotiation process in less than ten minutes.

Thus far, this study has revealed that the User-Initiated QoS Negotiation process and interface is appropriate for Static QoS Negotiation on Desktop Systems. However, the System-Initiated QoS Negotiation method is not investigated within this thesis. After describing the System-Initiated QoS Negotiation method to participants, five participants stated they would prefer the System-Initiated QoS Negotiation method, and five stated they preferred the User-Initiated QoS Negotiation Method. Development of a System-Initiated QoS Negotiation interface and its usability testing would lead to an interesting comparison.

6.6 Usability Study E: Comparison of Two Methods for Implementing the QCTT model

6.6.1 Overview

In this study, the Quality Cost Temporal Triangle (QCTT) model is implemented as a user interface for Dynamic QoS Management. The QCTT model defines an inherent triangular dependency between the three performance aspects (Quality Resolution, Cost Factor, and Temporal Facet) for the delivery of multimedia information. In providing the desired QoS, it is only possible to achieve ‘more desirable’ parameter values for two aspects, while the third is forced to a ‘less desirable’ value.

Two methods, namely: Cartesian Coordinate System (CCS), and Triangle Fractal System (TFS) were developed for implementing the QCTT model to provide the facility for managing QoS in real-time. Due to practical implementation limitations, each method provided a different visual representation of the QCTT model and affected the controllability for the Pivot-Point.

A preliminary study conducted to determine the most suitable PUI device for interacting with the QCTT interface, confirmed that the Mouse & Keyboard combination was suitable for Desktop Systems and the Pen & Touchpad combination was suitable for Portable Devices.

Usability Study-E investigates the usability of the QCTT interface and the interaction methods for controlling the Pivot-Point. It compares the CCS and TFS interfaces for the three user types.

6.6.2 User Interface Design

Two user interfaces for the QCTT model were developed based on two implementation methods: Cartesian Coordinate System (CCS) and Triangular Fractal System (TFS), as shown in Figure 6-19 and referred to as the *CCS Interface* and the *TFS Interface*, respectively. The theory for implementing the QCTT using the CCS and TFS methods are given in Appendix E, and have been published in the paper: Implementation and Usability of User Interfaces for Quality of Service Management [95].

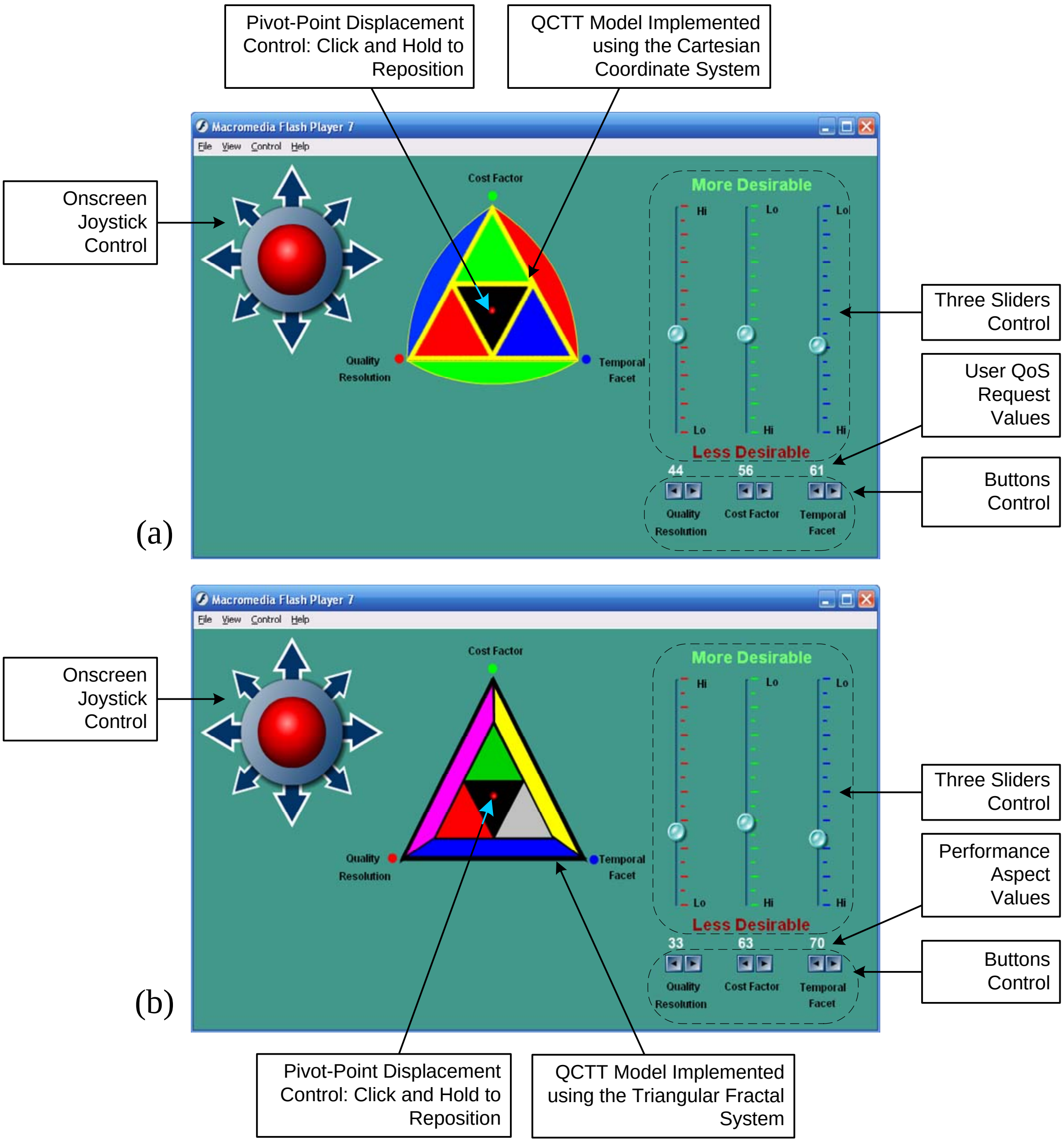


Figure 6-19: Usability Study E: QCTT Interfaces,

(a) Cartesian Coordinate System (CCS), and (b) Triangular Fractal System (TFS)

User Interaction

Both interfaces use the same user interaction and system feedback methods and similar visual presentation. The QCTT model includes a Pivot-Point used to indicate the level of QoS supplied with respect to the three performance aspects.

The following three user interaction methods control the position of the Pivot-Point.

- *Onscreen Joystick Control* method enables the user to click and hold on the joystick handle with a mouse, and then tilt it towards the direction they want the Pivot-Point to move.
- *Three Sliders Control* method comprises of three sliders, one for each performance aspect. Moving the sliders upwards sets ‘more desirable’ QoS values for each performance aspect, while moving them downwards sets ‘less desirable’ values. Based on the restrictions inherent in the QCTT model, it is not possible to move all sliders to the ‘more desirable’ position at the same time. Each performance aspect and Slider was colour coded using the colours: Quality Resolution - Red, Cost Factor – Green, and Temporal Facet – Blue. This made it easier to correlate the sliders with each performance aspect for controlling the Pivot-Point.
- *Buttons Control* method comprises buttons to increase and decrease the QoS value for each performance aspect.
- *Pivot-Point Displacement Control* method enables the user to click and hold the Pivot-Point with a mouse and move it to a desired position within the triangle.

The Graphical User Interfaces (GUIs) for these three methods are linked, such that changes made with one interface, show up in the other two interfaces.

User QoS Request Values show the QoS value for each performance aspect that is specified by the user. In Figure 6-19 a and b, the User QoS Request Values are shown just above the Buttons Control interface.

QCTT Visual Presentation

The theoretical boundaries set by the CCS method changed the visual representation of the QCTT model to a Reuleaux Triangle, as shown in Figure 6-19a. However, the TFS method retained the original triangle representation of the QCTT model, as shown in Figure 6-19b. Details on the limitations for the CCS and TFS methods are also given in Appendix E.

The visual representation of the QCTT model for the CCS and TFS interfaces included seven coloured regions that were used for completing the exercises given in the scenario of tasks.

6.6.3 Experiment Design

Aim:

This study investigates the usability of the QCTT interface for the two implementation methods: Cartesian Coordinate System (CCS) and Triangular Fractal System (TFS). Three user interaction methods to control the Pivot-Point in the QCTT interface are compared for Beginner, Intermediate and Advanced users.

Objectives:

The deliverable objectives for this study are:

- Evaluate the usability for the Onscreen Joystick, Three Sliders, Numeric Buttons, and Pivot-Point Displacement methods to control the Pivot-Point.
- Provide a usability comparison for the CCS and TFS interfaces.
- Provide a Task Performance Analysis for the CCS and TFS interfaces, and the three user interaction methods.

Participants:

Two groups of ten participants were recruited and ranked based on the user ranking model presented in section 5.7. Two separate groups of users participated in this experiment. Each group comprised of three Advanced, four Intermediate, and three Beginner users. Participants in Group A used the CCS Interface, and Group B used the TFS Interface.

Evaluation Criteria and Task Performance:

Each user interaction method was measured for: Ease of Use, Learnability, Usefulness, Effectiveness, Efficiency, Satisfaction and Stress Factor. Task Performance was measured for each user interaction method as a combination of average Task Completion Time and Task Error Count. Table 6-5 shows the matrix of user interaction methods, the evaluation criteria and task performance measured for the CCS and TFS interfaces.

Table 6-5: Usability Study E: Evaluation Criteria and Task Performance Measures, For Each User Interaction Method, for the CCS & TFS Interfaces

	Evaluation Criteria							Task Performance	
User Interaction Method for CCS & TFS Interfaces	Ease of Use	Learnability	Usefulness	Effectiveness	Efficiency	Satisfaction	Stress Factor	Task Completion Time	Task Error Count
1. Onscreen Joystick	✓	✓	✓	✓	✓	✓	✓	✓	✓
2. Three Sliders	✓	✓	✓	✓	✓	✓	✓	✓	✓
3. Buttons	✓	✓	✓	✓	✓	✓	✓	✓	✓
4. Pivot-Point Displacement	✓	✓	✓	✓	✓	✓	✓	✓	✓

Test Scenario:

The test scenario comprised of one task for each user interaction method. For each method, participants were required to position the Pivot-Point in the different coloured regions within the QCTT interface. A total of four tasks were completed for this study.

6.6.4 Analysis

Figure 6-20 a and b present the average Task Completion Times for using the user interaction methods: Onscreen Joystick, Three Sliders, Buttons, and Pivot-Point Displacement to control the Pivot-Point in the CCS and TFS interfaces. Figure 6-21 shows the average Task Error Count for these interfaces. Figure 6-22 and Figure 6-23, present the Usability Ratings for the CCS Interface and TFS Interface, respectively. An overall analysis is given in Figure 6-24, which includes combined averages for (a) Task Completion Time, (b) Task Error Count, and (c) Usability Ratings.

Task Completion Time Analysis

Figure 6-20 a and b show that all three user types completed tasks the quickest using the Pivot-Point Displacement method for the CCS as well as the TFS interface. The Buttons method took the longest for both interfaces. There is not much difference in the Task Completion Times between the three user types for all user interaction methods. On average, users completed the tasks marginally quicker for the TFS interface compared to the CCS interface.

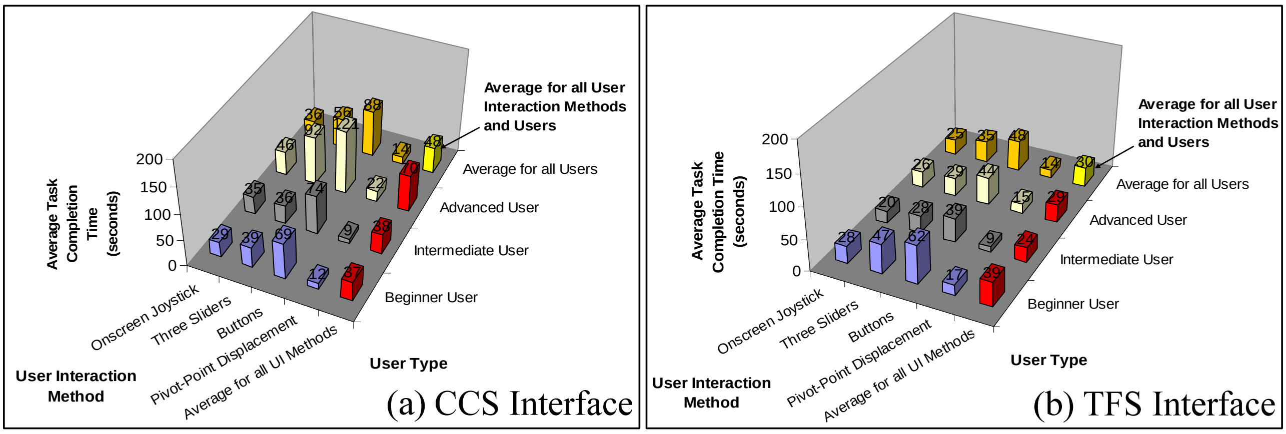


Figure 6-20: Usability Study E: Average Task Completion Times,
For each User Interaction Methods and Three User Types for: (a) CCS Interface, (b) TFS Interface

Task Error Count Analysis

For the CCS interface, Figure 6-21a depicts that the Buttons method recorded the highest number of errors on average, which were mostly made by Beginner users. Beginner users also made the highest number of errors for all user interaction methods. Intermediate and Advanced users made only a few errors.

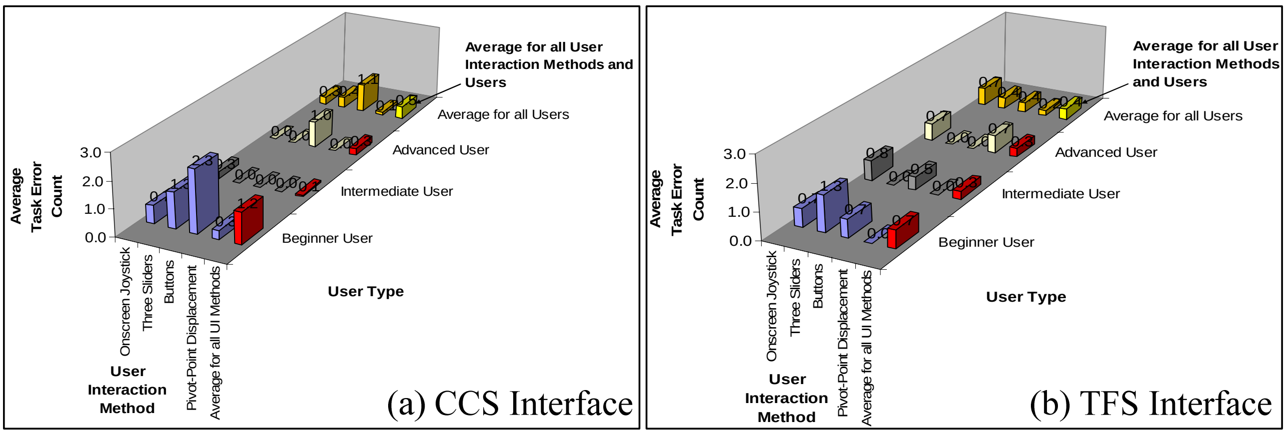


Figure 6-21: Usability Study E: Average Task Error Count,
For each User Interaction Methods and Three User Types for: (a) CCS Interface, (b) TFS Interface

For the TFS interface, Figure 6-21b shows a sporadic pattern of low number of errors for each interaction method. Overall, the number of errors for the CCS as well as the TFS interface is rather low, and there is not much difference between the two.

Usability Ratings Analysis

For the CCS interface, Figure 6-22a shows that for the Onscreen Joystick, all users gave equally high ratings for its Ease of Use and Learnability, and moderate ratings for Usefulness, Effectiveness, Efficiency, Satisfaction and Stress Factor. Figure 6-22b shows that Intermediate users gave the highest ratings for the Three Sliders method.

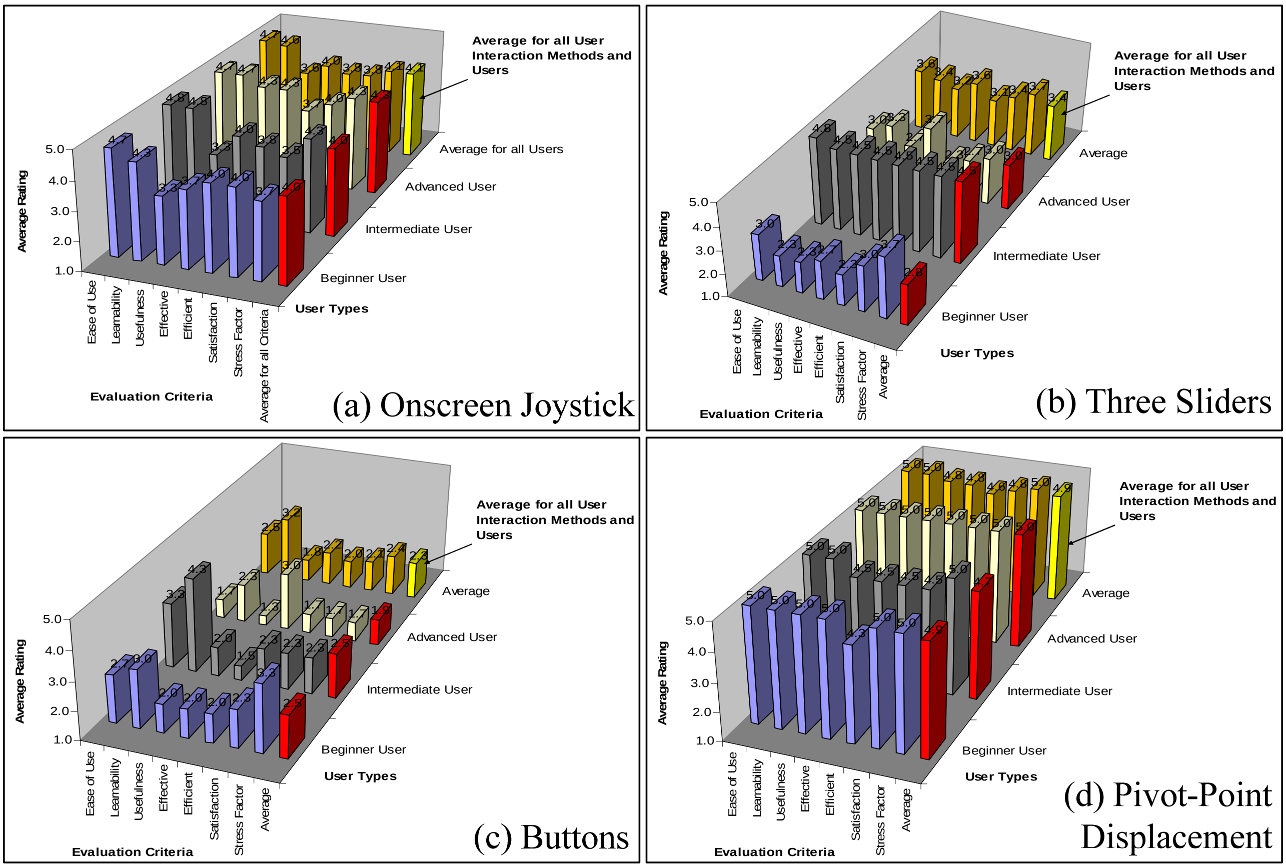


Figure 6-22: Usability Study E: Usability Ratings for the CCS Interface, Using the Interaction Methods: (a) Onscreen Joystick, (b) Three Sliders, (c) Buttons, (d) Pivot-Point Displacement

Figure 6-22c shows that all user types gave low ratings for the Buttons control method. However, Intermediate and Advanced users surprisingly gave the highest ratings for its

Ease of Use and Learnability. Figure 6-22d indicates high ratings given by all user types for the Pivot-Point Displacement method. Overall, the Pivot-Point Displacement method was most preferred, and the Buttons method was the least preferred.

For the TFS Interface, Figure 6-23 a to c shows that there are no clear trends in the ratings for the Onscreen Joystick, Three Sliders, and Buttons methods. On average, moderate ratings were given for these three interaction methods. The Pivot-Point Displacement method received the highest ratings by all user types, as shown in Figure 6-23d. Overall, the Pivot-Point Displacement method was most preferred; the Onscreen Joystick, Three Sliders, and Buttons methods were equally somewhat less preferred by all type of users.

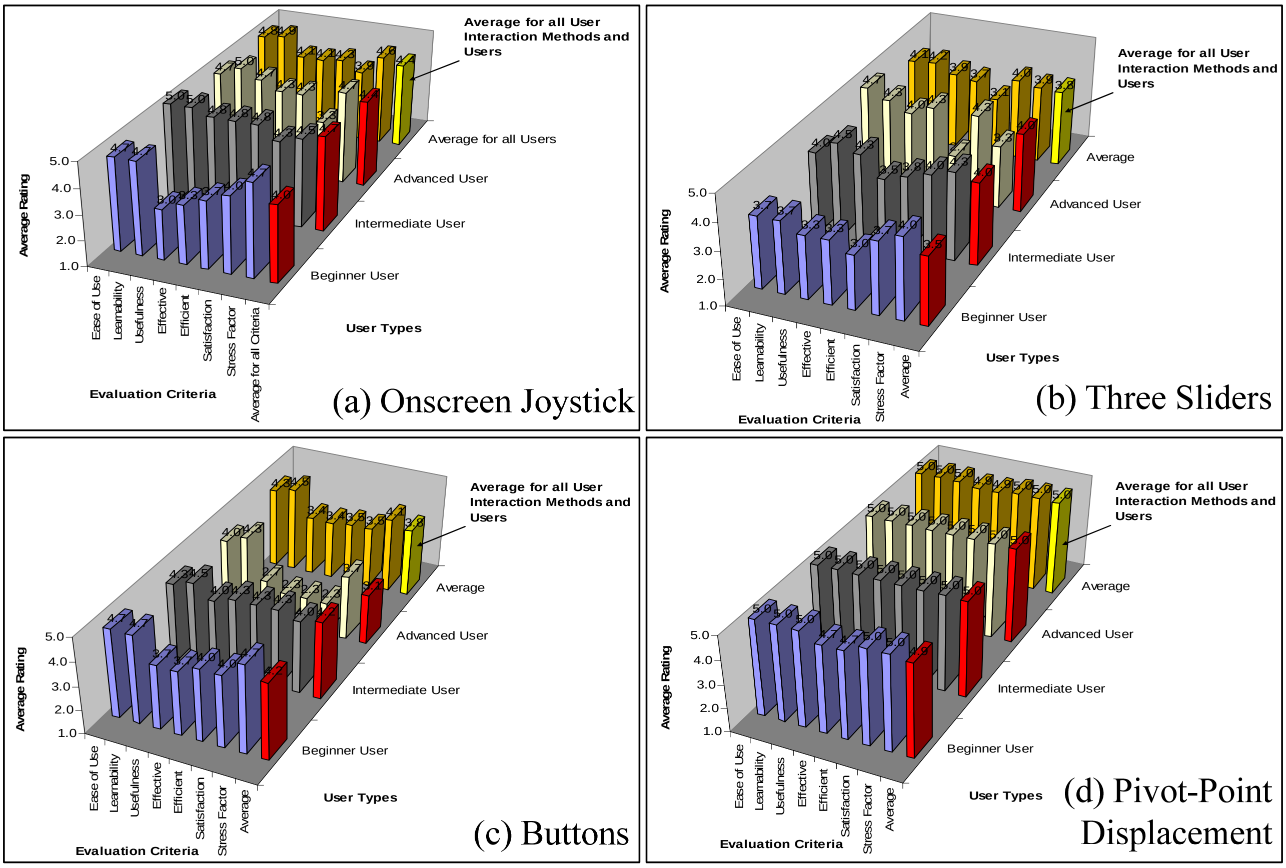


Figure 6-23: Usability Study E: Usability Ratings for the TFS Interface, Using the Interaction Methods: (a) Onscreen Joystick, (b) Three Sliders, (c) Buttons, (d) Pivot-Point Displacement

Combined Averages for Usability Study E

Figure 6-24a presents the combined average Task Completion Times for the TFS and CCS interfaces. The TFS Interface has the lowest Task Completion Times for the Onscreen Joystick, Three Sliders as well as the Buttons interaction methods; whereas, for the Pivot-Point Displacement method, both interfaces have equally low Task Completion Times. Figure 6-24b shows the Pivot-Point Displacement method to have the lowest number of errors for the CCS as well as the TFS Interface. The Buttons method recorded the highest number of errors for the CCS Interface, and moderate errors for the TFS Interface. The Onscreen Joystick and Three Sliders methods recorded a moderate number of errors for both interfaces.

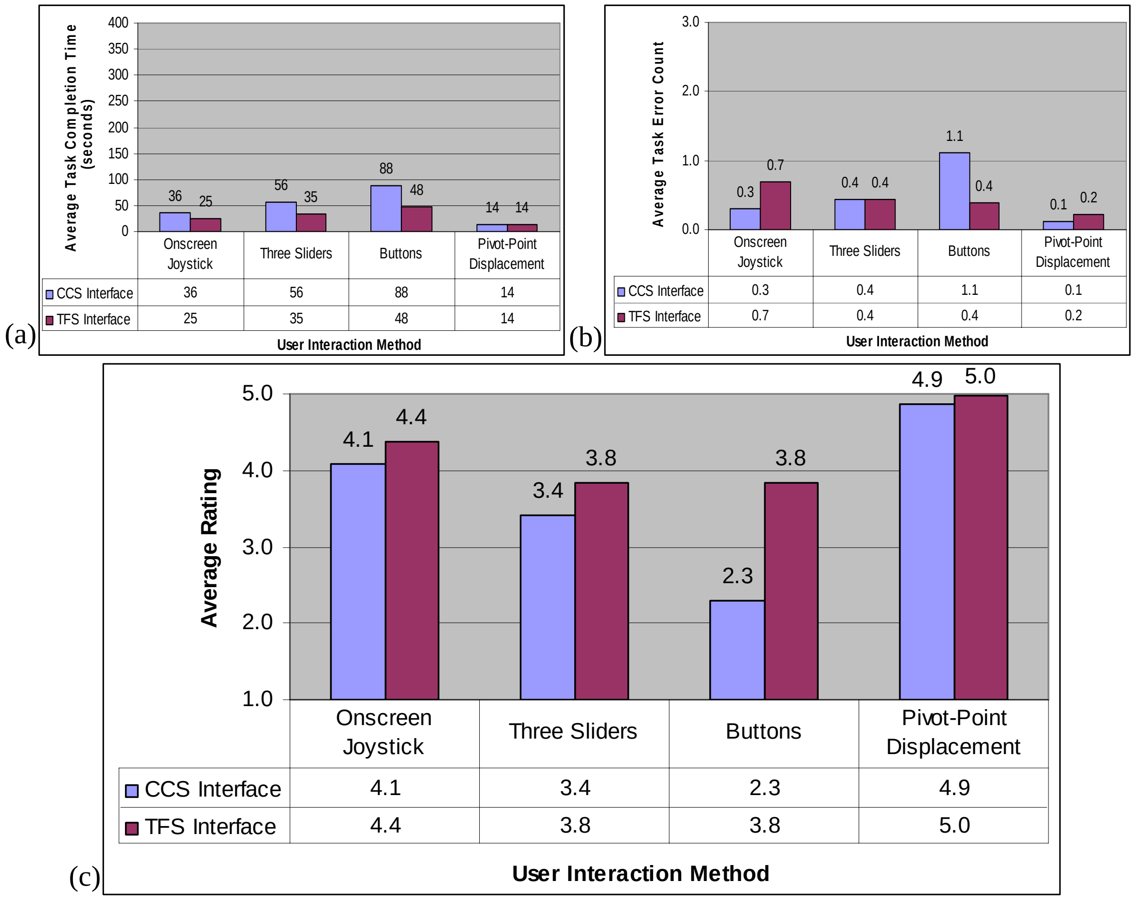


Figure 6-24: Usability Study E: Usability Comparison for the CCS and TFS Interfaces, Showing Combined Results for: (a) Average Task Completion Time, (b) Average Task Error Count, (c) Usability Ratings.

The combined average ratings given in Figure 6-24 clearly indicate that the TFS Interface is marginally better for usability than the CCS Interface. The Pivot-Point Displacement method is clearly the most preferred method for both interfaces, whereas the Buttons method is the least preferred, and more so for the CCS Interface. The Onscreen Joystick method is the second most preferred method for both interfaces, whereas the Three Sliders method is third most preferred.

Participant Comments

After briefing the Group A participants (who used the CCS Interface) about the QCTT model, they questioned the visual representation of the triangle. Participants stated that the Reuleaux Triangle visual representation did not correlate to the QCTT model closely as it did not look like a 'normal' triangle. Participants in Group B (who used the TFS Interface) did not make any such comments as the interface is a 'normal' triangle. Both, Group A and B participants found it cumbersome to use the Three Sliders and Buttons interaction methods to control the Pivot-Point. All participants were highly satisfied with the Pivot-Point Displacement method and found it to be the easiest to control.

6.6.5 Conclusion

It can be deduced that the TFS method is the better option for implementing the QCTT as an interface for Dynamic QoS Management. This conclusion is made based on the following observations:

- *Task Completion Time:* The TFS Interface had lower Task Completion Times for all interaction methods.
- *Task Error Count:* The TFS Interface has a low Task Error Count for all user types.
- *Usability Rating:* The TFS Interface ratings are marginally higher than the CCS Interface ratings, thus implying better usability.
- *Implementation:* The TFS method also proved to be less cumbersome to implement than the CCS method. Furthermore, the TFS graphical representation provides the ability to scale the visual representation of the QCTT interface and define more fractal regions for improving the resolution of QoS specification for different screen sizes.

This study has revealed that the Pivot-Point Displacement method is the most preferred for interacting with the QCTT interface, and should be the primary interaction method for Dynamic QoS Management. The Buttons method is the least preferred; however it should not be completely dismissed for interacting with the QCTT interface; as it can be used to fine adjusting the Pivot-Point position. The Onscreen Joystick and Three Sliders were nominally preferred and can be used as an optional interaction method for Dynamic QoS Management.

6.7 Usability Study F: User-Initiated QoS Re-Negotiation – Desktop Systems

6.7.1 Overview

This study evaluates the usability of new user interface elements introduced in the QCTT Interface (implemented using the TFS method) for performing User-Initiated QoS Re-negotiation on a Desktop System. As network traffic congestion and other external factors cause fluctuation in the provisioned QoS, current systems do not provide the facilities to re-negotiate QoS. This leaves the user stuck in the current session with inadequate QoS. There is a need to implement an interface that will allow the user to make adjustments to the QoS in real-time and without interrupting the multimedia session.

The interface presented in Usability Study-E was developed for re-negotiating QoS in real-time while a multimedia session is taking place. This interface did not give any system feedback for the QoS that the system can provide. It is crucial that the user is informed of the delivered QoS without interrupting the multimedia session during real-time re-negotiations. The user can use this information to make appropriate QoS requests in real-time.

The QCTT Interface has been enhanced with the two new features: (a) Provision Ring, and (b) Threshold Lines. This study evaluates the usability for these elements used in the QCTT Interface to perform User-Initiated QoS Re-negotiation. A comparison is made for performing QoS re-negotiations with and without the use of these elements.

6.7.2 User Interface Design

The QCTT Interface (shown in Figure 6-25) is enhanced with the addition of the following two system feedback elements:

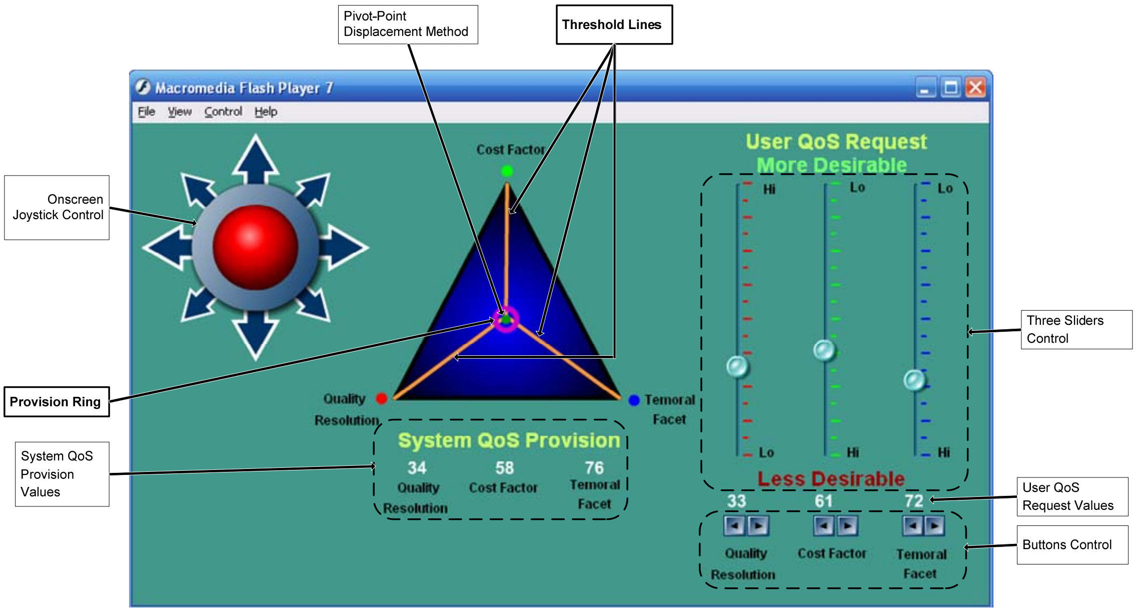


Figure 6-25: Usability Study F: QCTT Interface – Includes Two New Features, For User-Initiated QoS Re-Negotiation

- *Provision Ring* – This method gives feedback for the QoS that the system can deliver via the use of a QoS *Provision Ring* and *Numerical values*. The Provision Ring follows the Pivot-Point's movement within the triangle. The position of the ring indicates the level of QoS the system can provide. When the Pivot-Point moves beyond the boundaries of the ring, such that the ring is separated from the Pivot-Point, it indicates that the system cannot deliver the QoS that the user requests, as shown in Figure 6-26 d to f. Keeping the Pivot-Point within the ring implies that the system can deliver the QoS that the user requests as shown in Figure 6-26 a to c. The *System QoS Provision Values* interface displays the QoS values for the three performance aspects, and these values are linked to the position of the ring within the triangle. The aim of the Provision Ring is to provide the user with clear feedback about the QoS that the system can deliver.

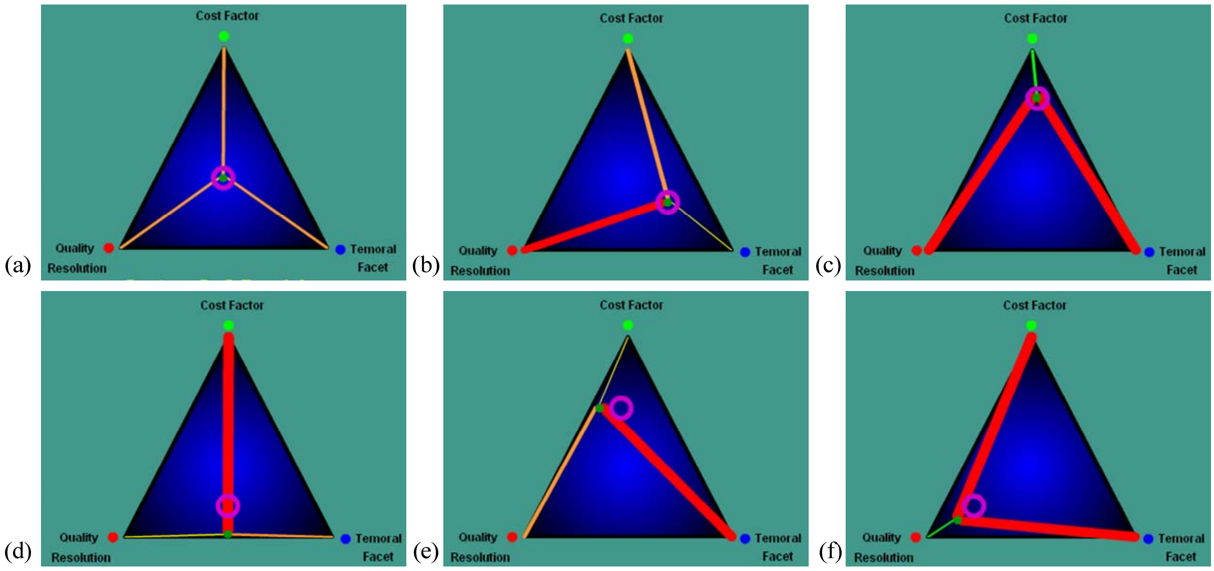


Figure 6-26: Usability Study F: Different System Feedback, Given by the Provision Ring, and Threshold Lines

- *Threshold Lines* – This feedback method uses three lines that are connected from each performance aspect vertex to the Pivot-Point. As the Pivot-Point moves farther or closer from each performance aspect vertex, the colour of the threshold lines change based on the following scheme:
    - Red implies non-desirable QoS values.
    - Yellow implies acceptable QoS values.
    - Green implies desirable QoS values.

  As the Pivot-Point moves farther away from a particular performance aspect, the line connected to this vertex becomes red and its width becomes thicker. Moving the Pivot-Point closer to a particular performance aspect makes the line thinner and the colour changes to green. Keeping the Pivot-Point at a moderate distance from any of the performance aspects causes the colour to change to yellow. Figure 6-26 a to f depict the different system feedbacks given by the Threshold Lines.

6.7.3 Experiment Design

Aims:

This study investigates the usability of the two new interface elements: (a) Provision Ring and (b) Threshold Lines for performing User-Initiated QoS Re-negotiation.

Objectives:

The deliverable objectives for this study are:

- Evaluate the usability for the QCTT Interface, without the assistance of any feedback elements to perform User-Initiated QoS Re-negotiation.
- Evaluate the usability of the (a) Provision Ring and (b) Threshold Lines.
- Provide a Task Performance analysis for using the: (a) QCTT Only (without feedback), (b) Provision Ring, and (c) Threshold Lines.

Participants:

Ten participants were recruited and ranked based on the user ranking model presented in section 5.7. This study used three Advanced, five Intermediate, and two Beginner users.

Evaluation Criteria and Task Performance

Table 6-6 shows that the full matrix of the evaluation criteria and task performance measures were used in this experiment.

Table 6-6: Usability Study F: Evaluation Criteria and Task Performance Measures.

	Evaluation Criteria									**Task Performance**	
Interaction Methods	Ease of Use	Learnability	Usefulness	Effectiveness	Efficiency	Ease of Perception	Ease of Comprehension	Satisfaction	Stress Factor	Task Completion Time	Task Error Count
1. QCTT Only	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2. Threshold Lines	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3. Provision Ring	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Test Scenario

The test scenario comprised of one task for using each interaction method. Each task included six exercises. Each exercise comprised a qualitative description instructing users to achieve a desired QoS goal with respect to the three performance aspects.

Task 1: Participants were required to achieve six desired QoS goals by repositioning the Pivot-Point within the QCTT interface without any system feedback.

Task 2: Participants were required to achieve six new QoS goals while taking into consideration the feedback given by the Threshold Lines.

Task 3: Participants were required to achieve a new set of six QoS goals while taking into consideration the feedback given by the Provision Ring.

6.7.4 Analysis

Figure 6-27 presents the average Task Completion Times for the three interaction methods: (a) QCTT Only, (b) Threshold Lines, and (c) Provision Ring. Figure 6-28 shows the average Task Error Count for these interaction methods. Figure 6-29 a to c present the usability ratings for each interaction method. An overall analysis is given in Figure 6-30, which includes combined averages for (a) Task Completion Times, (b) Task Error Counts, (c) overall Usability Ratings, and (d) Ratings for individual criterion.

Task Completion Time Analysis

Figure 6-27 shows that all three user types completed QoS re-negotiation the quickest using the Provision Ring feedback method. Beginner users took the longest using the Threshold Lines, and a moderate time was recorded for the QCTT Only method. Intermediate and Advanced users recorded a highest time for the QCTT Only method, and moderate times for the Threshold Lines method. On average, Beginner and Advanced users completed all three tasks with similar timing, while Intermediate users took marginally longer.

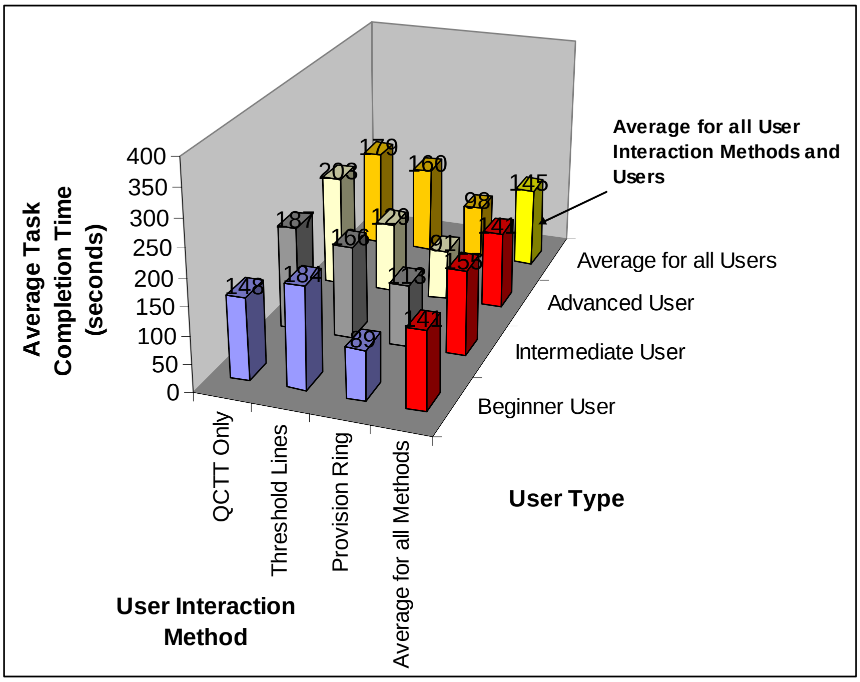


Figure 6-27: Usability Study F: Average Task Completion Times, For each Interaction Method and User Type.

Task Error Count Analysis

Figure 6-28 depicts that Intermediate users made the highest number of errors for the Provision Ring method, while Beginner and Advanced users made the highest number of errors for the Threshold Lines, and QCTT Only methods, respectively. On Average, a similar number of errors were made by all user types for each interaction method. For this experiment, Intermediate and Advanced users made a moderate number of errors with a similar average value, while Beginner users made the fewest errors on average; which seems counter intuitive. This is explained by the facilitator's observation that some Intermediate and Advanced users seemed over confident, leading to unwarranted errors.

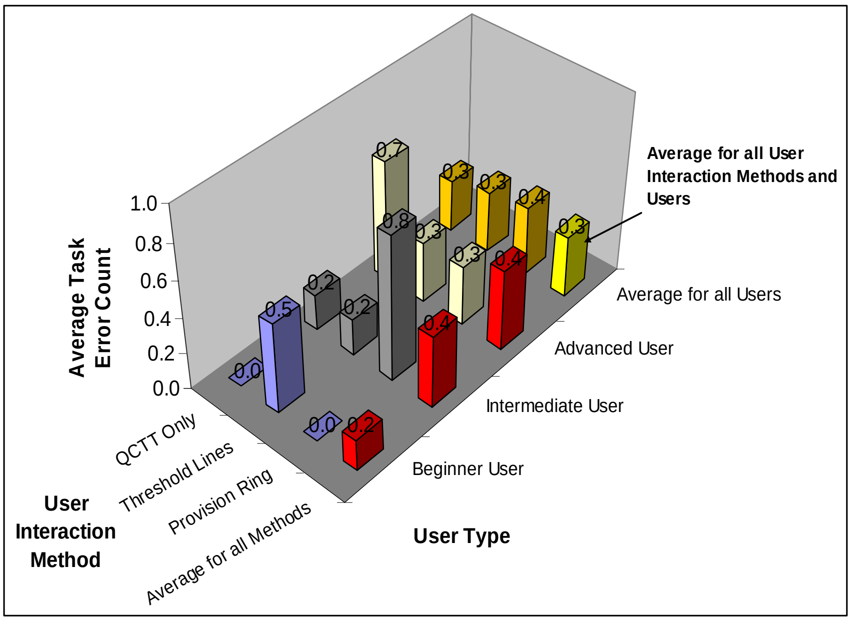


Figure 6-28: Usability Study F: Average Task Error Count, For each Interaction Method and User Type

Usability Ratings Analysis

Figure 6-29 a to c shows that there are no clear trends in the ratings for the three interaction methods. On average all user types gave similar positive ratings for each interaction method, with an average of around 4 on the Likert scale of 5. By combining the averages for all user types, it can be seen that the Threshold Lines received marginally higher overall ratings. This small difference in ratings demonstrates that all users found the three methods equally usable.

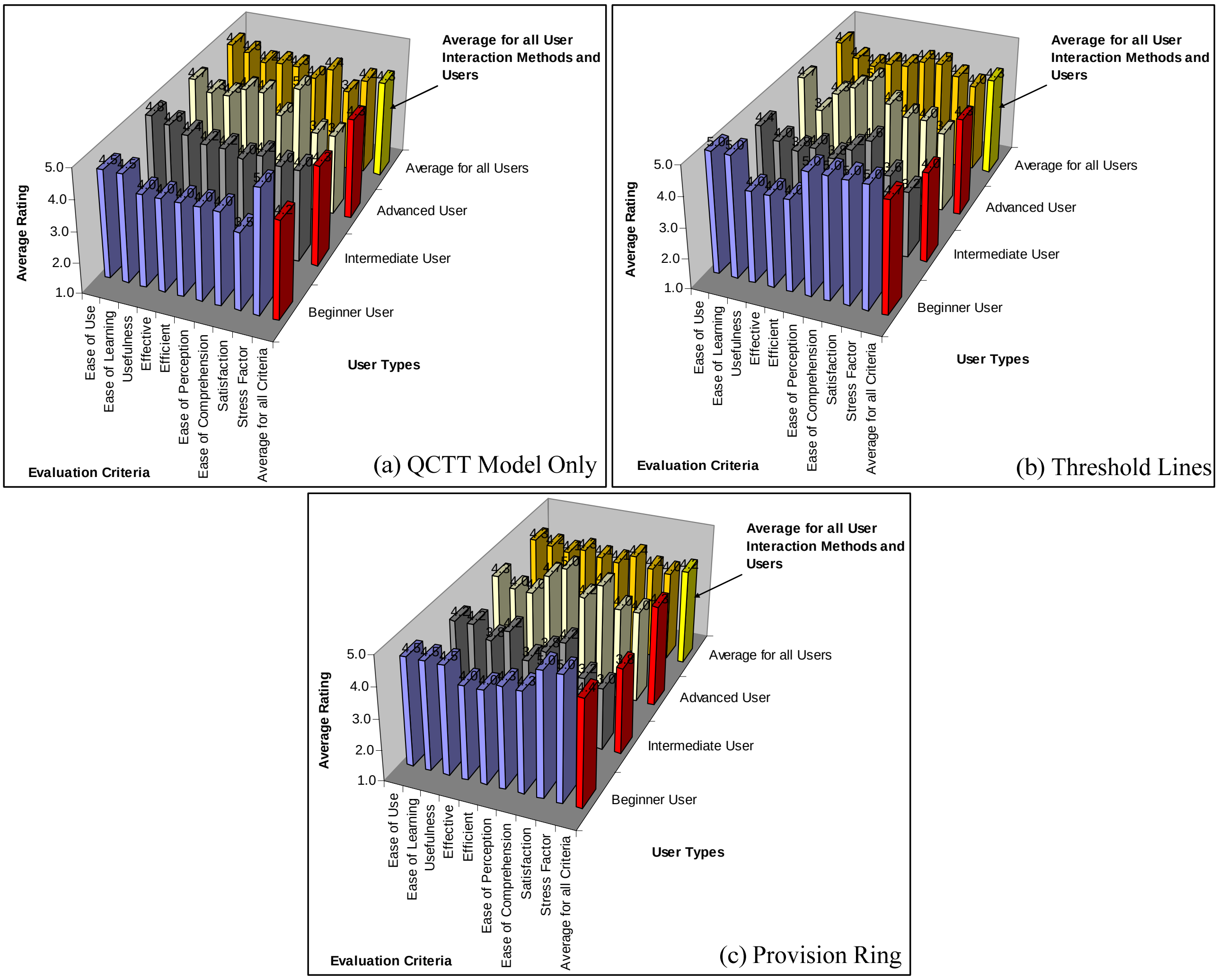
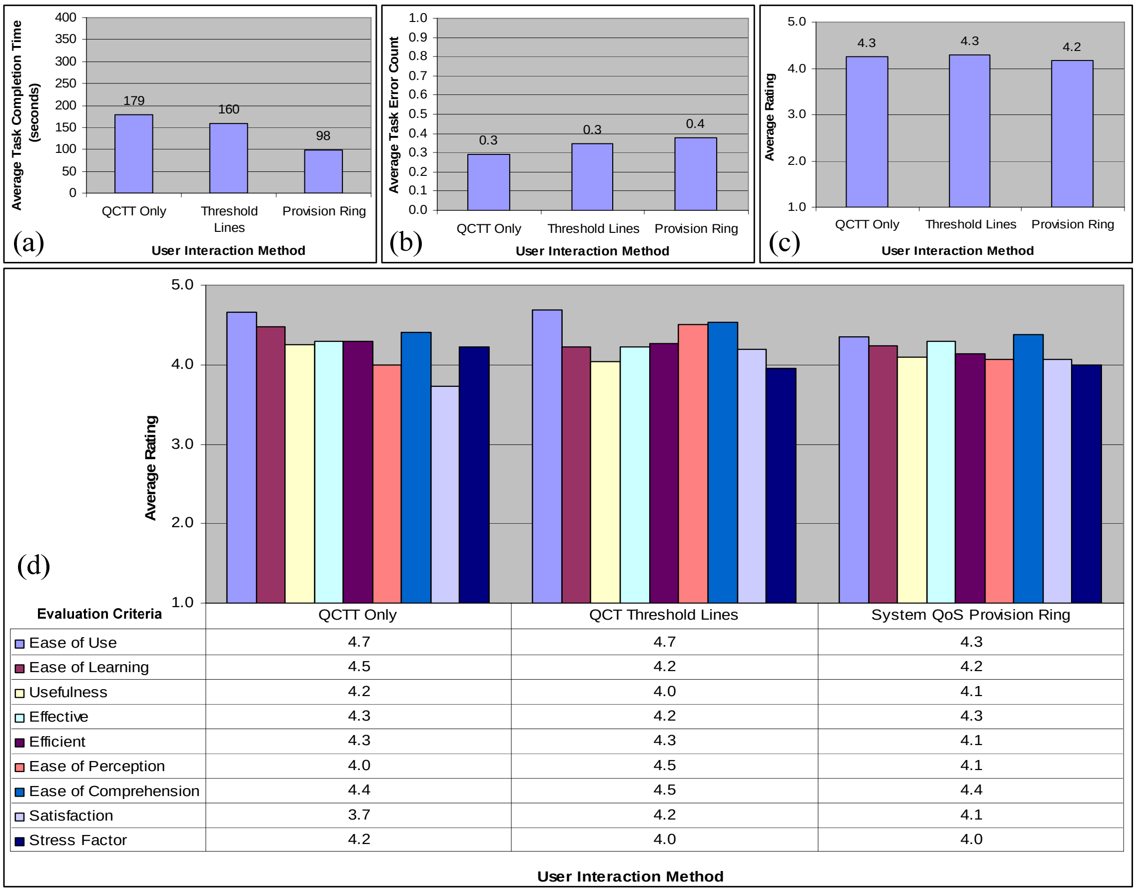


Figure 6-29: Usability Study F: Average Usability Ratings,
For the Interaction Methods (a) QCTT Only, (b) Threshold Lines, (c) Provision Ring.

Combined Averages for Usability Study F

Figure 6-30a presents the combined average Task Completion Times for each interaction method. The Provision Ring method has the lowest Task Completion Time, followed by the Threshold Lines method; the QCTT Only method took the longest. Figure 6-30b shows the QCTT Only and the Threshold Lines methods to have the least number of errors.



Evaluation Criteria	QCTT Only	QCT Threshold Lines	System QoS Provision Ring
Ease of Use	4.7	4.7	4.3
Ease of Learning	4.5	4.2	4.2
Usefulness	4.2	4.0	4.1
Effective	4.3	4.2	4.3
Efficient	4.3	4.3	4.1
Ease of Perception	4.0	4.5	4.1
Ease of Comprehension	4.4	4.5	4.4
Satisfaction	3.7	4.2	4.1
Stress Factor	4.2	4.0	4.0

Figure 6-30: Usability Study F: Overall Usability Analysis,
Showing Combined Results for: (a) Average Task Completion Time, (b) Average Task Error Count, (c) Combined Usability Ratings (d) Usability Ratings for Evaluation Criterion.

The combined average usability ratings presented in Figure 6-30c depict that the QCTT Only and Threshold Lines methods received marginally higher ratings than the Provision Ring method. The ratings for each individual criterion presented in Figure 6-30d show that the QCTT Only and Threshold lines received substantially high ratings for Ease of Use. The QCTT Only method also received marginally high ratings for

Learnability and Ease of Comprehension, while the Threshold lines received high ratings for Ease of Perception and Ease of Comprehension. Overall, the QCTT Only and Threshold Lines methods received marginally higher ratings than the Provision Ring method.

Participant Comments

Participants enjoyed using the user interface to complete QoS re-negotiations. Some participants stated that they were confused when using the Threshold Lines method. Because of this, for some tasks they ignored the feedback given by the Threshold Lines. Some participants questioned the meaning of the label 'Temporal Facet', and suggested that it be re-labelled with a more appropriate name that refers to 'Delay'. Participants did not indicate a clear preference for either method.

6.7.5 Conclusion

This study did not reveal a clear winner for the most suitable interaction method to perform User-Initiated QoS Re-negotiation. It was observed that the:

- *Provision Ring* method had the lowest Task Completion Times, but a marginal higher error rate, and a marginal lower usability rating.

- *Threshold Lines* method had a moderate Task Completion Time and Task Error Count, and a marginal higher usability rating.

- *QCTT Only* method had the highest task completion time, the lowest error count, and a marginal high usability rating.

While the two new feedback elements did not make much difference to the usability of the QCTT Only method, the Provision Ring reduced the Task Completion Time substantially. Further research has been conducted in using the interface presented in this study for performing User-Initiated QoS Re-negotiation on Portable Systems; this investigation is presented in section 6.9.

6.8 Usability Study G: System Initiated QoS Re-Negotiation – Desktop Systems

6.8.1 Overview

This study evaluates the usability of a new user interface element included in the QCTT Interface for performing System-Initiated QoS Re-negotiation. While a multimedia session is in progress, various factors such as delays, traffic congestion, connection dropouts can occur within the network service, which affect the supplied QoS. In this case, there is a need for the system to monitor the network services and prompt the user to make adjustments to the QoS. Such a process is called System-Initiated QoS Re-negotiation.

Currently, network services do not provide system feedback and do not warn the user of any degradation encountered in the supplied QoS. Therefore, users can experience unexpected interruptions in their multimedia sessions.

There is a need to implement an interface with the ability to provide feedback to the user to adjust the QoS for their multimedia session, and prevent any interruption to their communication session. Such an interface needs to include interface elements that provide non-intrusive feedback, i.e. without distracting the user from carrying out their multimedia session. This feature requires underlying technology that monitors the QoS degradation in the network. How this monitoring can be done is beyond the scope of this research and suggested as a future research problem.

The user interface developed in Usability Study-F enabled users to initiate QoS re-negotiation with the system, and obtain feedback based on the actions they make. This was useful for users that are not satisfied with the QoS they receive. For Usability Study-G, the QCTT interface is enhanced with a new feature named: Recommendation Assistant, which gives suggestions to the user for improving the QoS level during a multimedia session.

6.8.2 User Interface Design

The QCTT interface (shown in Figure 6-31) was enhanced with the addition of a QoS *Recommendation Assistant*.

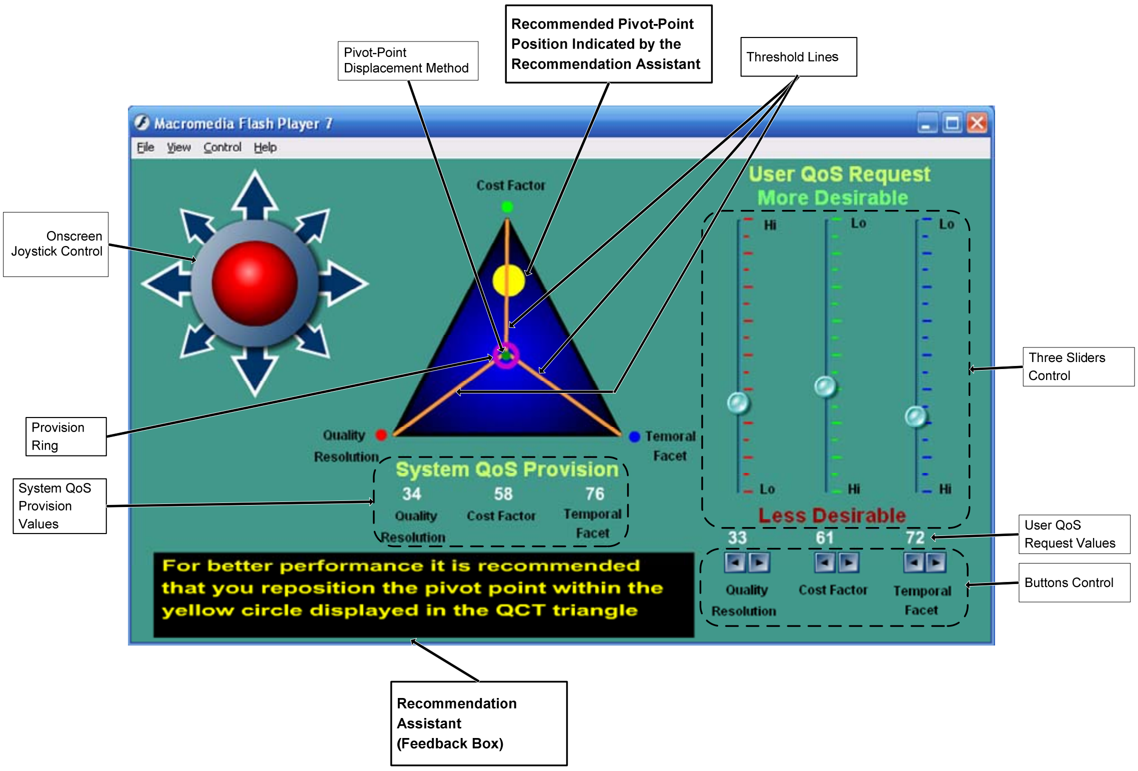


Figure 6-31: Usability Study G: QCTT Interface,

Includes Recommendation Assistant for System Initiated QoS Re-Negotiation

The *Recommendation Assistant* provides feedback to the user and suggests necessary changes to the requested QoS in order to prevent the multimedia session from being interrupted. As shown in Figure 6-31, textual responses are given in the feedback box that requests the user to make necessary adjustments to the QoS by moving the Pivot-Point to a particular position; indicated by a yellow or red circle.

- *Yellow circle* implies advisory feedback, where the user has the choice to make the recommended QoS adjustment, or leave it as it is. If the user does not make the recommended adjustment, then some degradation in the quality of the multimedia session may be experienced.

- *Red circle* implies critical feedback, where the user must make the recommended QoS adjustment to prevent the multimedia session from being interrupted.

Once the Pivot-Point is positioned on top of the coloured circle, the circle then disappears after a three second delay. This delay gives the user the opportunity to re-adjust the Pivot-Point within the boundaries of the coloured circle.

6.8.3 Experiment Design

Aims:

This study investigates the usability of the Recommendation Assistant to perform System-Initiated QoS Re-negotiation.

Objectives:

The deliverable objectives for this study are:

- Evaluate the usability of the Recommendation Assistant to perform System-Initiated QoS Re-negotiation.
- Evaluate the Level of Distraction in using the Recommendation Assistant.

Participants:

Ten participants were recruited and ranked based on the user ranking model presented in section 5.7. This study used three Advanced, five Intermediate, and two Beginner users.

Evaluation Criteria and Task Performance

Table 6-7 shows that the complete set of evaluation criteria is used in this experiment. *Level of Distraction* is measured to determine the degree to which the user felt distracted in using the Recommendation Assistant in real-time, while participating in a multimedia session. Participants provided Level of Distraction ratings based on the Likert scale, where 1=High Distraction, 2=Medium Distraction, 3=Neutral, 4=Low Distraction, 5=No Distraction.

Response Times were recorded to measure the time it took participants to become aware of the system feedback alert and adjust the QoS as recommended by the system. It was virtually impossible for participants to make errors in this study as the system feedback guided the user to adjust the QoS by moving the Pivot-Point to the coloured circle indicated in the QCTT interface. Consequently, Task Error Counts were not recorded as participants did not make any errors.

Table 6-7: Usability Study G: Evaluation Criteria and Task Performance Measures,

	Evaluation Criteria										**Task Performance**
Interaction Method	Ease of Use	Learnability	Usefulness	Effectiveness	Efficiency	Ease of Perception	Ease of Comprehension	Satisfaction	Stress Factor	Level of Distraction	Response Time
1. Recommendation Assistant	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Test Scenario

The testing scenario for this study comprised one task for using the Recommendation Assistant. Participants were presented with a sample video to simulate a Video on Demand session. Ten system feedback alerts, comprising a mixture of advisory and critical feedback were presented to the user at random intervals. For each feedback alert, participants were required to take necessary actions in adjusting the QoS as recommended by the system.

6.8.4 Analysis

Figure 6-32 presents the average response times for the ten system feedback alerts against the three user types. Figure 6-33 shows the Level of Distraction ratings for using the Recommendation Assistant to re-negotiate QoS. Figure 6-34 includes the evaluation ratings for the Recommendation Assistant given by the three user types. An overall analysis is given Figure 6-35, which includes combined averages for (a) Response Times (c) overall Evaluation Criteria and Level of Distraction ratings, and (d) Usability Ratings for individual evaluation criterion.

Response Times Analysis

For the ten system feedback alerts, Figure 6-32 shows a declining trend for all user types. It was 'not unexpected' that in the beginning, users would take longer to respond to the system feedback given by the Recommendation Assistant, and as they became familiar with the required action, the Response Time reduced.

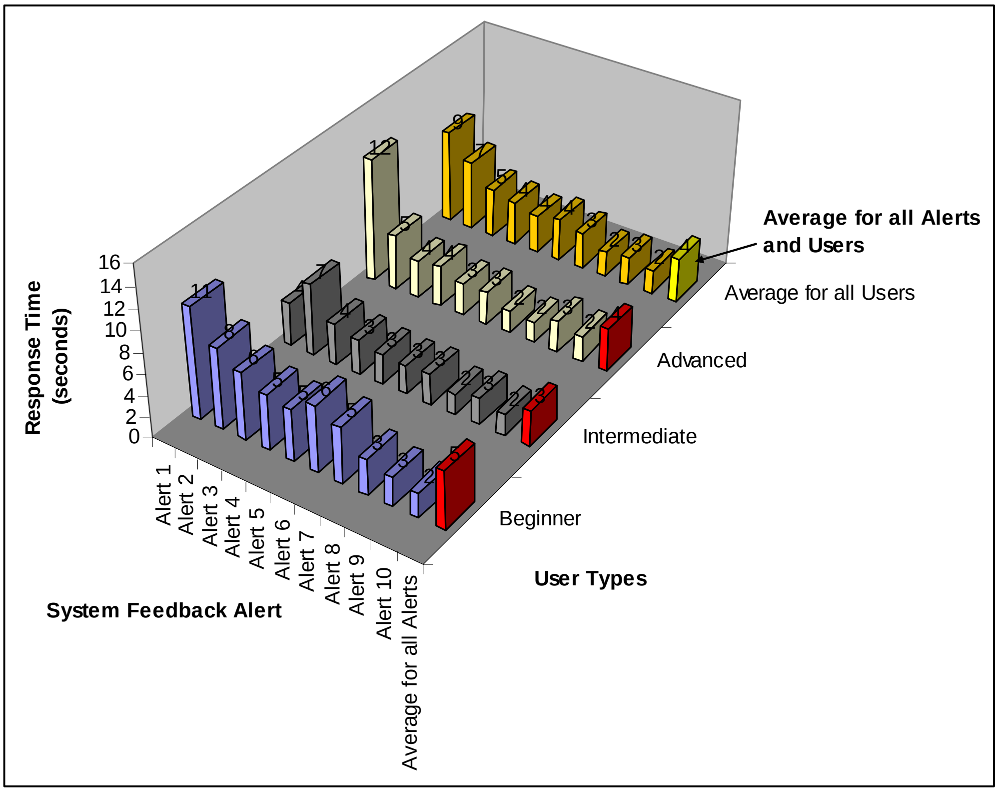


Figure 6-32: Usability Study G: Average Participant Response Times, For the Ten Alerts and Three User Types

Beginner and Advanced users recorded highest Response Times for the system feedback 'Alert-1', followed by a decrease in time for 'Alert-2'. This was not the case for Intermediate users, as they recorded a highest time for 'Alert-2', and then a second highest time for 'Alert-1'. This unexpected behaviour can be explained by the following observation made by the facilitator.

The facilitator observed that at the beginning of the experiment, Beginner users took longer as they were not confident in the actions they were required to perform. Advanced users were overconfident during briefing, in which they did not pay too much attention to instructions and then felt confused when commencing the experiment. Intermediate users did not show any sign of hesitation or overconfidence.

On average, Beginner users recorded the highest Response Times, while Advanced users recorded moderate times, and Intermediate responded the quickest.

Level of Distraction Analysis

Figure 6-33 shows the Level of Distraction ratings given by the three user types, and the overall average.

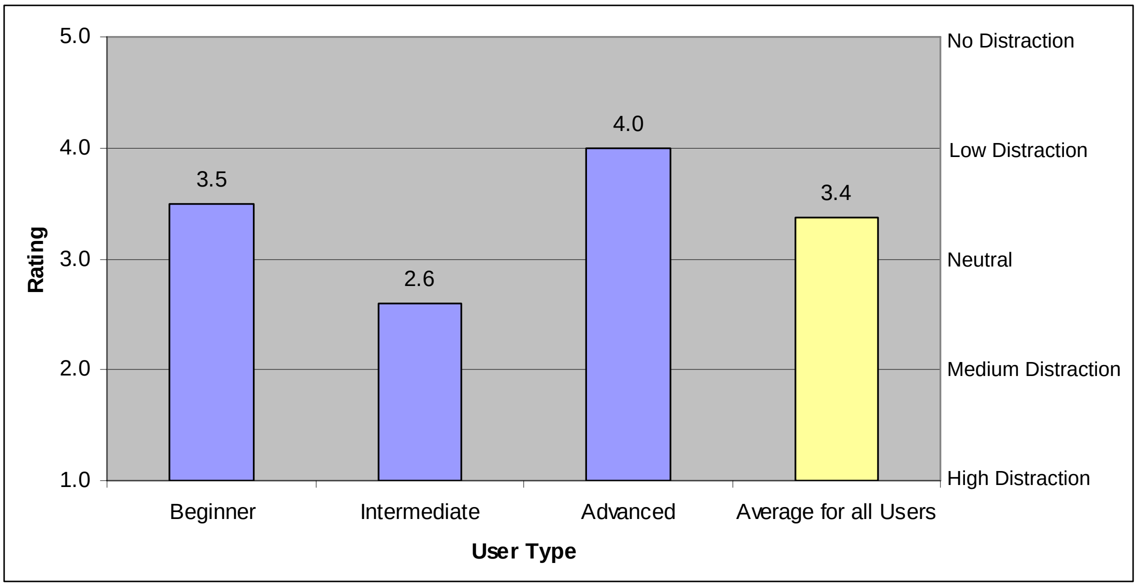


Figure 6-33: Usability Study G: Level of Distraction Ratings, For the Recommendation Assistant

Advanced users gave high ratings for the Recommendation Assistant, implying that they did not find it too distracting for carrying out System-Initiated QoS Re-negotiation in real-time. Beginner users provided moderate ratings, while Intermediate users gave low ratings. Overall for all users, the Level of Distraction averaged to a rating of 3.4, which implies a neutral-to-a-low-distraction. This means that all users perceived the Recommendation Assistant to be a little distracting but not significant enough to disrupt their communication session.

Usability Ratings Analysis

Figure 6-34 shows that there are no clear trends in the ratings given for the three user types. On average, all user types gave similar positive ratings for the Recommendation Assistant. Intermediate users gave the highest ratings; Beginner users gave moderate ratings, while Advanced users gave the lowest ratings.

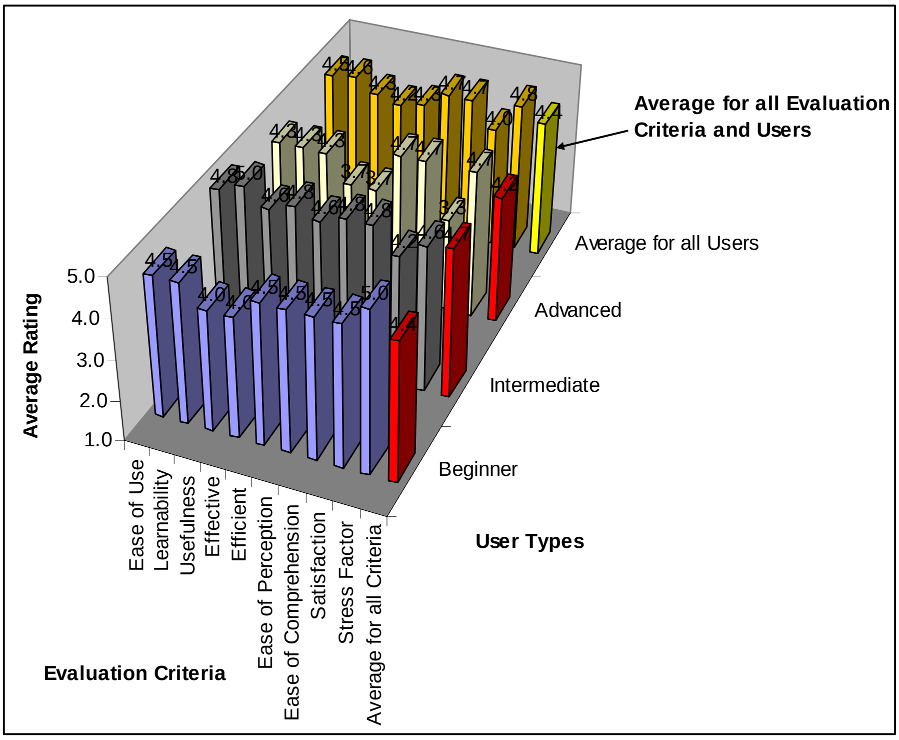


Figure 6-34: Usability Study G: Average Usability Ratings, For the Recommendation Assistant

Advanced users gave low ratings for effectiveness, efficiency and satisfaction – implying that they were not too satisfied with the usability and the functionality of the Recommendation Assistant. Overall, the combined ratings given by all user types average to a value of around 4.4 on the Likert scale of 5; this indicates that the Recommendation Assistant is efficacious for performing System-Initiated QoS Re-negotiation.

Combined Averages for Usability Study G

Figure 6-35a presents the combined average Response Times for the three user types. Intermediate users recorded the lowest response times, followed by Advanced users, and Beginner users took the longest.

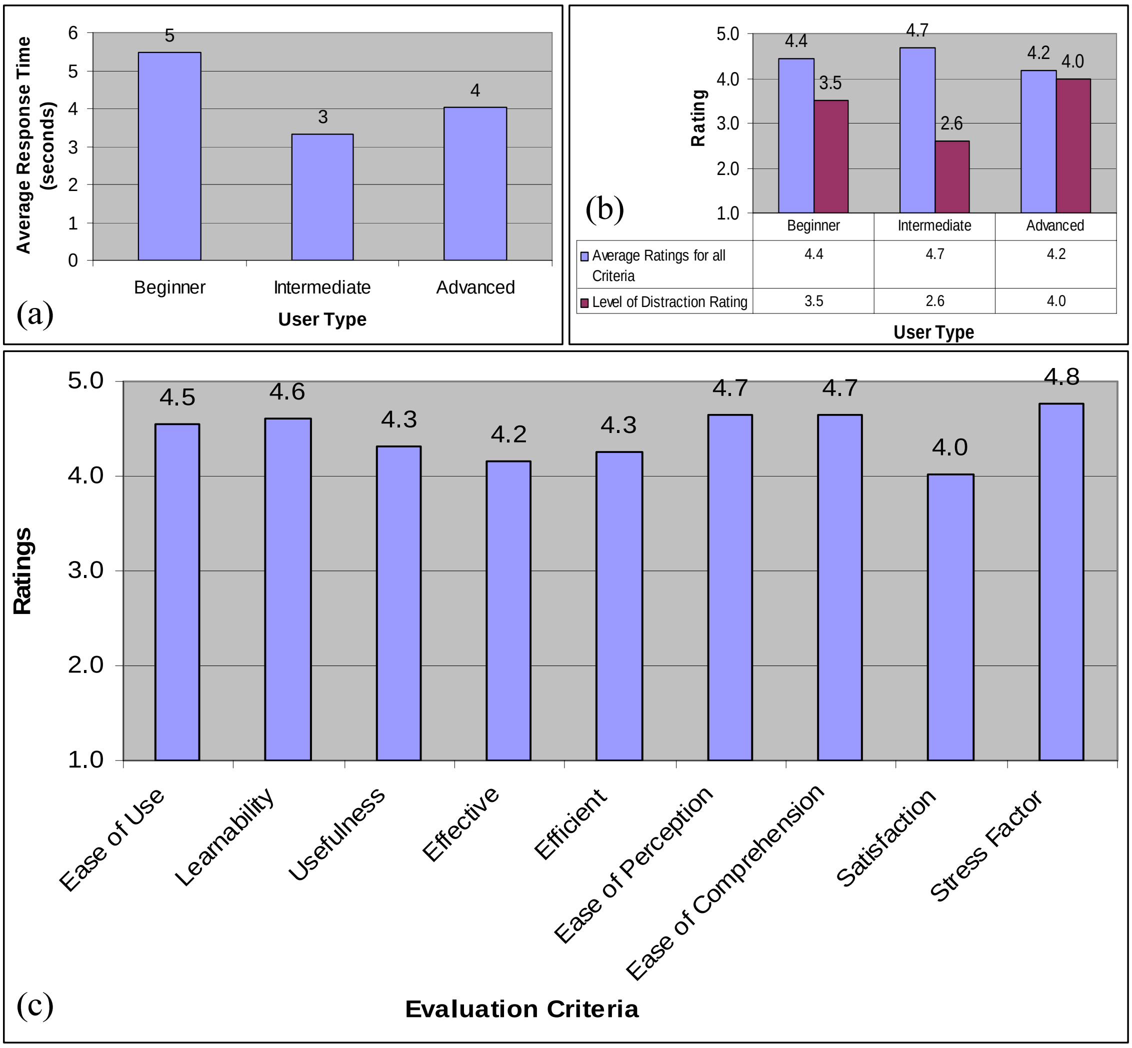


Figure 6-35: Usability Study G: Overall Usability Analysis,

For Recommendation Assistant: (a) Average Response Times, (b) Combined Usability and Level of Distraction Ratings, (d) Usability Ratings for each Evaluation Criterion.

Figure 6-35b shows the combined usability ratings, and the Level of Distraction ratings for the three user types. It clearly shows that Intermediate users gave the highest usability ratings, followed by Beginner users and then Advanced users. However, for the Level of Distraction, Intermediate users gave the lowest ratings, followed by

Beginner users and then Advanced users giving the highest ratings. Interestingly, these results show that Intermediate users gave the highest usability ratings and they also found the Recommendation Assistant rather distracting. This trend is similar to that for the Beginner users, but with less severity. Conversely, Advanced users gave similar high ratings for usability and for Level of Distraction, which implies that they found the interface useable and at the same time not very distracting.

Figure 6-35d shows an overall positive feedback given by all participants for each individual criterion, indicating good usability for the Recommendation Assistant.

Participant Comments

Participants enjoyed using the user interface to complete System-Initiated QoS Re-negotiation. Participants acknowledged the need for an interface that provides real-time feedback to the user for managing QoS. The idea of using the Recommendation Assistant for System-Initiated QoS Re-negotiation intrigued Beginner and Intermediate users, while Advanced users were less excited. Participants suggested that the coloured circle, which appeared in the QCTT interface to immediately disappear when the Pivot-Point was positioned on top of it. The intentional three second delay gave them the impression that the interface was not responding and not operating as intended.

6.8.5 Conclusion

It can be deduced that the Recommendation Assistant is suitable for performing System-Initiated QoS Re-negotiation in real-time. This conclusion is made based on the following observations.

- *Response Times* – Beginner, Intermediate and Advanced users improved on the time taken to respond to the system feedback alerts, thus indicating that they were able to re-negotiate with the system efficiently.

- *Level of Distraction* – All three user types gave positive feedback for the Level of Distraction, indicating that the Recommendation Assistant did not cause much distraction during the communication session.

- *Participant Ratings* – High positive feedback was given by all user types for the usability of the Recommendation Assistant to perform System-Initiated QoS Re-negotiation.

Thus far, this study has revealed that the Recommendation Assistant is useful for providing real-time system feedback for System-Initiated QoS Re-negotiations on Desktop Systems. Further research has been conducted in using the interface presented in this study for performing System-Initiated QoS Re-negotiation on Portable Systems; this investigation is presented in section 6.10.

6.9 Usability Study X: User Initiated QoS Re-negotiation – Portable Devices

6.9.1 Overview

This study evaluates the usability of the QCTT Interface enhanced to perform User-Initiated QoS Re-negotiation on Portable Devices. As Portable Devices and Mobile Phones in conjunction with wireless communications becomes prevalent, studying the usability of these systems is essential.

Current Portable Digital Assistants (PDAs) include processors speeds from 200MHz – 624MHz or higher, 32 – 128MB of RAM, Flash Memory storage of 1GB or more, Bluetooth and/or WiFi networking technology. Mobile operating systems such as PalmOS and Windows Mobile for Pocket PC include software that supports multimedia applications. Although these devices support multimedia, the performance of multimedia applications (such as Video on Demand, Video Conference) is still not very good. However, consistent improvement in the performance of Portable Devices is inevitable. Wireless communication technologies introduce new QoS issues, such as: communication dropouts, limited bandwidth, increased error rates etc. These issues adversely affect the presentation of the multimedia information.

These issues necessitate the need to adapt the QCTT Interface to allow re-negotiation of QoS in real-time for Portable Devices. As the screen size for portable devices is much smaller than that of Desktop Systems, the precision and accuracy for specifying QoS reduces. For this reason, the QCTT triangle was partitioned into fractal zones, where each zone defines a set of parameters.

Another aspect that requires investigation is the size of the QCTT interface. In viewing a video presentation, or conducting a video conference, users would want to use most of the screen for the application, predicating a smaller QCTT interface.

Consequently, three different versions of the QCTT interface were implemented and tested, namely: *Large QCTT Quad Fractals, Small QCTT Quad Fractals, Large QCTT 10,000 (10k) Fractals*. These interfaces were tested on a Personal Digital Assistant (PDA), an emulated PDA, and an emulated Mobile Phone.

At the time of this study, Mobile Phones that were available on the market did not provide adequate hardware and software support to use the QCTT Interface that was developed using Macromedia Flash version 7. Therefore, the functionality of a Mobile Phone was emulated on a Desktop Computer. To determine whether the performance of the PDA impacted the usability for the QCTT Interface, a real PDA was compared to that of an emulated PDA.

6.9.2 User Interface Design

Based on the issues raised earlier, the following three user interfaces were developed and enhanced for execution on Portable Devices.

- *Large QCTT Quad Fractals* interface comprised four fractal zones within the QCTT, where each fractal zone was assigned preset parameter values for the three performance aspects.

- *Small QCTT Quad Fractals* interface is similar to the Large QCTT Quad Fractals interface; however, the size of each fractal was reduced for better screen utilisation.

- *Large QCTT 10,000 (10k) Fractals* interface uses 10,000 fractals for much higher accuracy in specifying QoS as compared to the previous two interfaces.

The theory for implementing these interfaces is given in Appendix F.

User Interface for Mobile and Portable Devices

The Large QCTT Quad Fractals, Small QCTT Quad Fractals, and Large QCTT 10,000 (10K) Fractals implementations were integrated into a Video on Demand (VoD) user interface that was enhanced for Portable Devices, as shown in Figure 6-36. The user interface inherits most of the features of the QCTT Interface presented in Usability Study-F, section 6.7. These include the Pivot-Point Displacement, and Buttons interaction methods to control the Pivot-Point. Three Sliders method was not included as initial tests revealed this method to be cumbersome for use on Portable Devices. However, the Provision Ring and Threshold Lines interaction methods were included in this interface.

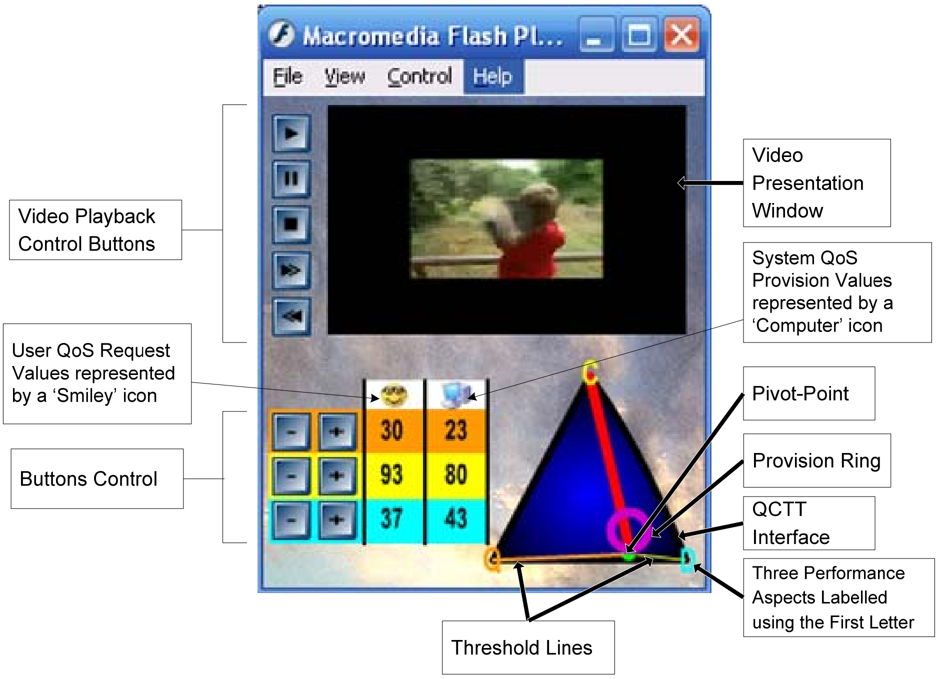


Figure 6-36: Usability Study X: Enhanced User Interface for Portable Devices

To better utilise the small screen on PDAs and Mobile Phones, text labels are replaced by icons or letter symbols, and a colour coded scheme is used to represent the three performance aspects. A 'smiley' icon is used to represent the *User QoS Request Values* for the three performance aspects. A 'computer' icon is used to represent the System QoS Provision Values for the three performance aspects. For example, in Figure 6-36, the user requests a value of 93 for the Cost Factor aspect, and the system provides a value of 80 for this performance aspect.

Previous studies indicated that general users were not familiar with the term 'Temporal', and did not know its function unless explained. Therefore, 'Temporal' was substituted with a more widely used term 'Delay'. Each performance aspect point on the QCTT was represented by the first letter of the performance aspect: Quality – Q, Cost – C, and Delay – D. The label for each performance aspect was colour matched to the user interface element that controlled it: Quality – Orange, Cost – Yellow, and Delay – Light Blue.

This user interface emulated a Video on Demand (VoD) system operating on a Portable Device and connected to a video server via a wireless communication connection. The quality of the video that would be seen on a real system was emulated in this user interface for a video clip that was locally stored on the Portable Device. When the Pivot-Point was moved within the QCTT, the following effects were emulated with respect to the three performance aspects:

- *Quality Resolution* – This adjusted the frame size of the video output; increasing the Quality Resolution value increased the video frame size, and vice versa.
- *Cost Factor* – This did not have any affect on the video presentation.
- *Delay* – This was emulated by adding random bursts of delays and jitter in the output of the audio and video. Increasing the Delay value increased the frequency of these random delays, and decreasing it performed the opposite.

Storage capacity was an issue when running a video on the PDA. Therefore it was necessary to use a short video clip set on continuous loop play for the duration of the experiment.

Controlling the User Interface

The three user interfaces were tested on the three devices: PDA, Emulated PDA, and Emulated Mobile Phone – these are collectively called Test Devices henceforth. The interaction method used for each Test Device is as follows:

PDA – This device comprises a touch screen, which is operated with a stylus (a pen-like pointing device).

Emulated PDA **–** The emulated PDA used a touchpad connected to the Desktop Computer via the USB port.

Emulated Mobile Phone **–** Mobile Phones of today use a keypad that includes the up, down, left, and right arrow keys, and an activate button to navigate the interface. The numeric keypad on a standard computer keyboard was used to emulate the mobile phone keypad.

The arrow keys were used to scroll through the different objects in the user interface, while the '5' button activated these objects. Selected objects were highlighted with a yellow box. To use the Pivot-Point Displacement method, selecting the QCTT interface and then clicking '5' on the keypad enables a mode in which the arrow keys directly control the Pivot-Point's movement.

6.9.3 Experiment Design

Aims:

This study investigates the usability of the three user interfaces: (a) Large QCTT Quad Fractals, (b) Small QCTT Quad Fractals, (c) Large QCTT 10,000 (10k) Fractals for User-Initiated QoS Re-negotiation on the three Test Devices: Emulated PDA, Emulated Mobile Phone, and PDA.

Objectives:

The deliverable objectives for this study are:

- Evaluate the usability for the three interfaces operating on the three Test Devices and using the methods: (i) QCTT Only, (ii) Threshold Lines, (iii) Provision Ring, and (iv) QCTT plus video feedback.
- Provide a Task Performance analysis for the three interfaces operating on the three Test Devices.

Participants:

Twelve participants were recruited and ranked based on the user ranking model presented in section 5.7. This study used two Advanced, six Intermediate, and four Beginner users.

Evaluation Criteria and Task Performance

Table 6-8 shows that the entire evaluation criteria set and task performance measures were used for each interface operating on the three portable devices.

Table 6-8: Usability Study X: Evaluation Criteria and Task Performance Measures

	Evaluation Criteria									Task Performance	
User Interface	Ease of Use	Learnability	Usefulness	Effectiveness	Efficiency	Ease of Perception	Ease of Comprehension	Satisfaction	Stress Factor	Task Completion Time	Task Error Count
1. Large QCTT Quad Fractals	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2. Small QCTT Quad Fractals	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3. Large QCTT 10,000 (10k) Fractals	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Test Scenario

The test scenario comprised of four tasks for each interface. Each task included a set of exercises that included qualitative descriptions instructing users to achieve a desired QoS goal.

Task 1: Participants were required to achieve three desired QoS goals by repositioning the Pivot-Point within the QCTT interface without any system feedback.

Task 2: Participants were required to achieve three new QoS goals while taking into consideration the feedback given by the Threshold Lines.

Task 3: Participants were required to achieve a new set of three QoS goals while taking into consideration the feedback given by the Provision Ring.

Task 4: Participants were required to achieve three QoS goals based on the video's Quality of Presentation (QoP).

6.9.4 Analysis

The analysis for Usability Study-X presents combined average results where clear trends have been identified in the recorded data. Detailed results are included in Appendix G. Figure 6-37 presents the combined average Task Completion Times for the three interfaces: (a) Large QCTT Quad Fractals, (b) Small QCTT Quad Fractals, (c) Large QCTT 10,000 (10k) Fractals and the three portable devices: (i) Emulated PDA, (ii) Emulated Mobile Phone and (iii) PDA. Figure 6-38 shows the combined average Task Error Count for these interfaces on the three Test Devices, and Figure 6-39 presents the combined average Usability Ratings for the same.

Task Completion Time Analysis

Figure 6-37 shows a declining trend in the Task Completion Times for the three user interfaces for all Test Devices. On average all users completed User-Initiated QoS Re-negotiations the fastest using the Large QCTT 10,000 (10k) Fractals interface.

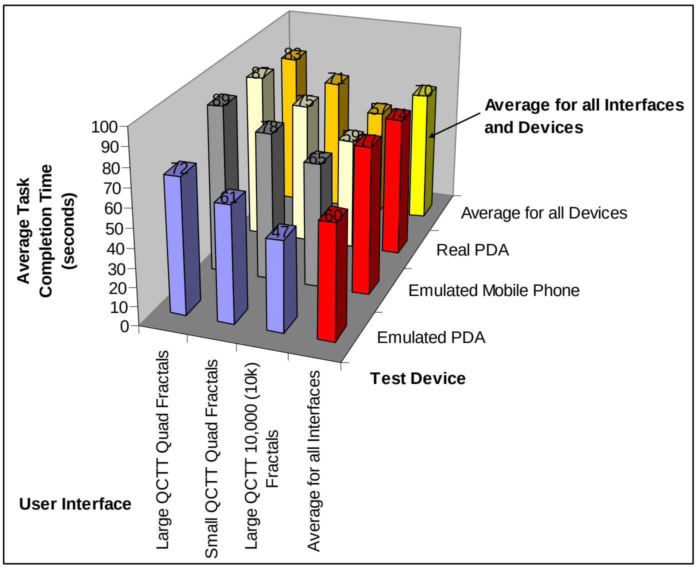


Figure 6-37: Usability Study X: Combined Average Task Completion Times, For the Three Interfaces used on an Emulated PDA, Emulated Mobile Phone, and Real PDA

A moderate time was recorded for the Small QCTT Quad Fractals interface, and Large QCTT Quad Fractals took the longest. For all interfaces, users took the longest to re-negotiate QoS using the Emulated Mobile Phone. The Emulated PDA was the quickest, and the real PDA recorded a moderate Task Completion Time.

The facilitator observed that for the Emulated Mobile Phone, users found it cumbersome using the numeric keypad to interact with the interface – thus taking users longer to complete the tasks. It was also observed that the real PDA recorded a higher time than the Emulated PDA as it required more effort to precisely adjust the QoS using the Pivot-Point. Furthermore, the lower performance of the real PDA, due to its hardware limitation, slightly increased the Task Completion Time.

Task Error Count Analysis

Figure 6-38 shows an even distribution for the average number of errors made for each user interface and for all Test Devices. A slight increase is shown for the Large QCTT Quad Fractals interface used on the Emulated Mobile Phone. Overall, the number of errors is rather low, indicating that users had little difficulty in using these interfaces and Test Devices.

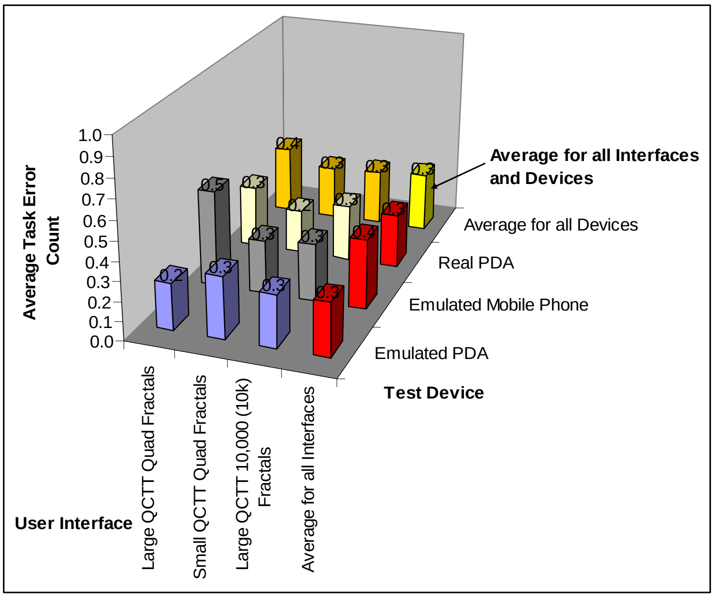


Figure 6-38: Usability Study X: Combined Average Task Error Count, For the Three Interfaces used on an Emulated PDA, Emulated Mobile Phone, and Real PDA

Usability Ratings

Figure 6-39 does not show any clear trend in the average ratings given for the three interfaces and the three Test Devices. However, it is clearly shown that on average all users gave high ratings for each interface and Test Device, which signifies good usability.

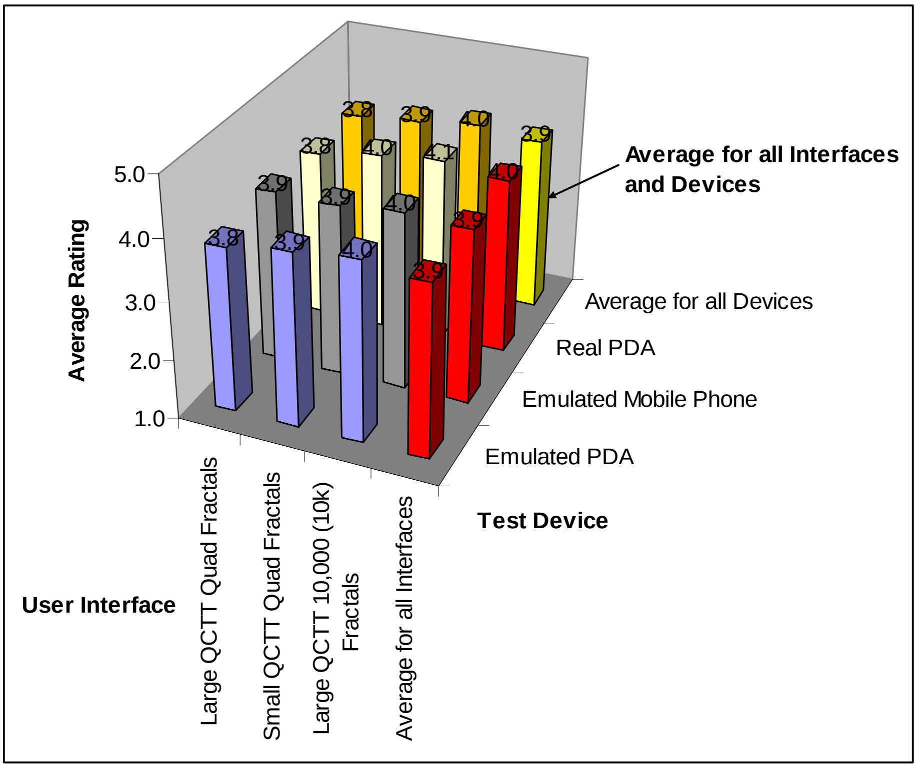


Figure 6-39: Usability Study X: Combined Average Ratings,

For the Three Interfaces used on an Emulated PDA, Emulated Mobile Phone, and Real PDA

Summary Analysis for using the QCTT plus Video Feedback method

Participants did not show any significant difficulties in using the QCTT interface to achieve the different goals of QoP for the video. Participants were easily able to observe the emulated QoP effects for the video, and make the correct adjustments to achieve the specified goals. Low average Task Completion Times, low Task Error Counts, and high usability ratings were recorded for this method. This shows that participants understood the operation of the QCTT interface, thus indicating good usability for the same. Detailed results for this interaction method are presented in Appendix G.

Participant Comments

Some participants commented that they found the PDA not very responsive. This was due to the low hardware specification of the PDA procured in 2004. For the Small QCTT Quad Fractals interface, participants suggested that of the Pivot-Point and the Provision Ring should be smaller to allow better precision. Some participants commented that it was difficult to see the colour change in the Threshold Lines as the screen size was too small. Participants did not clearly indicate any preference for any of the three interfaces used on the three Test Devices. However, participants liked the idea of being able to re-negotiate QoS in real-time using the QCTT interface.

6.9.5 Conclusion

In conclusion, this study has revealed that the three user interfaces are equally suitable for performing User-Initiated QoS Re-negotiations on the three Test Devices: Emulated PDA, Emulated Mobile Phone, and PDA. This decision is made based on the following observations:

- *Task Completion Times* – All three interfaces received low average Task Completion Times for all Test Devices, indicating that all users were able to effectively re-negotiate QoS.
- *Task Error Counts* – All three interfaces recorded low Task Error Counts when used on the three Test Devices. This implies that users did not find it cumbersome when using these interfaces.
- *Usability Ratings* – Equally high ratings were given for all three interfaces used on the three Test Devices. These ratings averaged to a value of around 4 on the Likert scale of 5, thus implying that participants found the usability for these interfaces to be reasonably good.

The performance of portable devices plays an important role in usability. For this study, the performance of the older model PDA caused some delay. However, this difficulty will get resolved as the hardware specification of PDAs keeps improving on the lines of Moore’s Law [96].

This study showed that the use of fractal zones for the Large and Small Quad Fractals interfaces made it easier to accurately specify QoS, regardless of the size of the QCTT Interface. Therefore, it can be deduced that different fractal sizes can be used for different sizes of the QCTT interface. To determine the most appropriate fractal size for each QCTT triangle size further research is required.

6.10 Usability Study Y: System Initiated QoS Re-negotiation – Portable Systems

6.10.1 Overview

This study measured the usability of the Recommendation Assistant used on Portable Devices to perform System-Initiated QoS Re-negotiation. Portable Devices are becoming more popular due to their support for multimedia applications and communications. Consequently, there is a need for the system to provide feedback for re-negotiating QoS in real-time on Portable Devices. Usability Study-X investigated three interfaces for User-Initiated QoS Re-negotiation. Therefore, it would be logical to determine whether the Recommendation Assistant can also be used on Portable Devices as results presented in Usability Study-G proved it to be useful for Desktop Systems.

In this study, the Large QCTT 10,000 (10k) Fractals interface is enhanced with the addition of the Recommendation Assistant for performing System-Initiated QoS Re-negotiation. This interface was tested on the three Test Devices: Emulated PDA, Emulated Mobile Phone, and PDA.

6.10.2 User Interface Design

The Large QCTT 10,000 (10k) Fractals interface, shown in Figure 6-40, was enhanced with the Recommendation Assistant.

In this interface, the Recommendation Assistant inherits some of the functionality from the interface presented in Usability Study-G, subsection 6.8.2. The system provides feedback to the user suggesting a QoS, and moving the Pivot-Point to the recommended position within the triangle prevents the multimedia session from being interrupted. Due to the small screen size of Portable Devices, the Recommendation Assistant does not include textual feedback. Instead, feedback is only given using the Red and Yellow circles appearing in the QCTT interface, which indicate the recommended position for the Pivot-Point. The Yellow circle implies advisory feedback, and the Red circle indicates critical feedback. The user is expected to position the Pivot-Point on top of the feedback circle to make the recommended adjustments. The feedback circle disappears after 1 – 2 seconds.

Figure 6-40: Usability Study Y: Enhanced User Interface for Portable Devices, Showing the Recommendation Assistant

As this interface does not have textual feedback elements, a guided tour may be required by the user to understand its operation. However, for this study the users were briefed by the facilitator.

6.10.3 Experiment Design

Aims:

This study investigates the usability for the Recommendation Assistant to perform System-Initiated QoS Re-negotiation on the three Test Devices: Emulated PDA, Emulated Mobile Phone, and PDA.

Objectives:

The deliverable objectives for this study are:

- Evaluate the usability for the Recommendation Assistant to perform System-Initiated QoS Re-negotiation on the three Test Devices.
- Evaluate the Level of Distraction in using the Recommendation Assistant on the three Test Devices.

Participants:

The same twelve participants recruited for Usability Study X, participated for the usability tests in this study; Two Advanced, six Intermediate, and four Beginner users were used.

Evaluation Criteria and Task Performance

Table 6-9 shows that the complete set of evaluation criteria was used in this experiment. *Level of Distraction* measured the degree at which the user felt distracted in using the Recommendation Assistant in real-time. Response Times were recorded to measure the time it took participants to adjust the QoS for each system feedback alert. Task Error Counts were not recorded as this study relates to the users' response to a prompt; as in Usability Study-G

Table 6-9: Usability Study Y: Evaluation Criteria and Task Performance Measures

	Evaluation Criteria										**Task Performance**
Interaction Method	Ease of Use	Learnability	Usefulness	Effectiveness	Efficiency	Ease of Perception	Ease of Comprehension	Satisfaction	Stress Factor	Level of Distraction	Response Time
1. Recommendation Assistant	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Test Scenario

This study comprised one task for using the Recommendation Assistant on the three Test Devices. Participants were presented with a sample video to simulate a Video on Demand session. Ten system feedback alerts, comprising a mixture of advisory and critical feedback were presented to the user at random intervals. For each alert, participants were required to take necessary actions to adjust the QoS as recommended by the system.

6.10.4 Analysis

The analysis for Usability Study-Y presents combined average results where clear trends have been identified in the recorded data. Detailed results are given in Appendix H. Figure 6-41 presents the average Response Times for ten system feedback alerts, for the three Test Devices and for Desktop Systems – which was recorded in Usability Study-G. Figure 6-42 presents the average Response Times for the three Test Devices and the three User Types. Figure 6-43 shows the Level of Distraction ratings for using the Recommendation Assistant to re-negotiate QoS. Figure 6-44 shows combined usability ratings for the three Test Devices and the three User Types. Figure 6-45 presents combined usability and Level of Distraction ratings showing the relationship between them.

Response Times Analysis

Figure 6-41 shows that the average Response Times for the Emulated Mobile Phone is rather high compared to the Emulated PDA and real PDA Test Devices. The high Response Time recorded for the Emulated Mobile Phone is due to the use of the arrow keys for the Pivot-Point movement; whereas with the Emulated PDA and the real PDA, the Pivot-Point is pre-positioned with a single tap of the stylus.

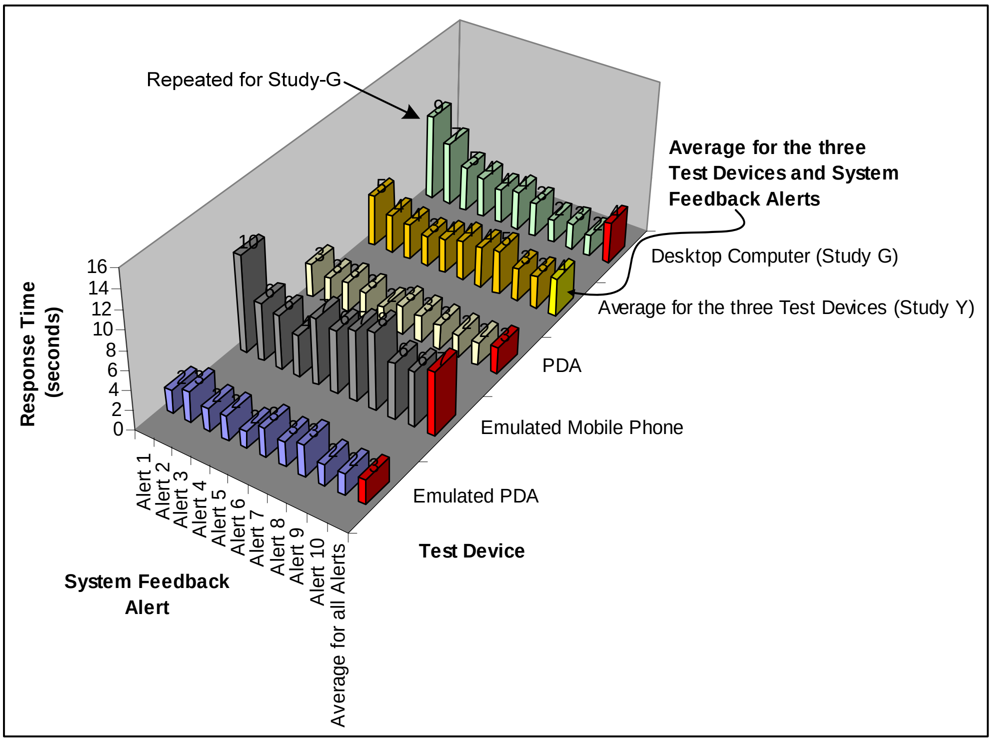


**Figure 6-41: Usability Study Y: Average Response Times,
For the Three Test Devices and the Ten Feedback Alerts**

For the ten system feedback alerts, Figure 6-41 shows little variation in the Response times for the three Test Devices. The Recommendation Assistant used in this study (Usability Study-Y) did not include any textual feedback. A previous study (Usability Study-G) indicated that the textual feedback used in the interfaces for Desktop Systems initially increased the Response Times, and then decreased as users became familiar with each feedback alert – shown as the last row in Figure 6-41.

By excluding textual feedback in Usability Study-Y, users were able to respond to each feedback alert equally fast. This indicates that by reducing the amount of information used in system feedback, less attention is required from the user, which consequently can reduce the Level of Distraction. However, presenting some textual information to the user may be beneficial to their understanding of the system's prompts. Further research is required to determine the optimal amount of information used for system feedback during QoS re-negotiation for real-time multimedia systems.

Figure 6-42 shows that the Response Times are rather high for the Emulated Mobile Phone for all three User Types. This confirms that the keypad interaction method added further delays. Unexpectedly, Advanced users recorded the highest Response Time for using the Emulated Mobile Phone.

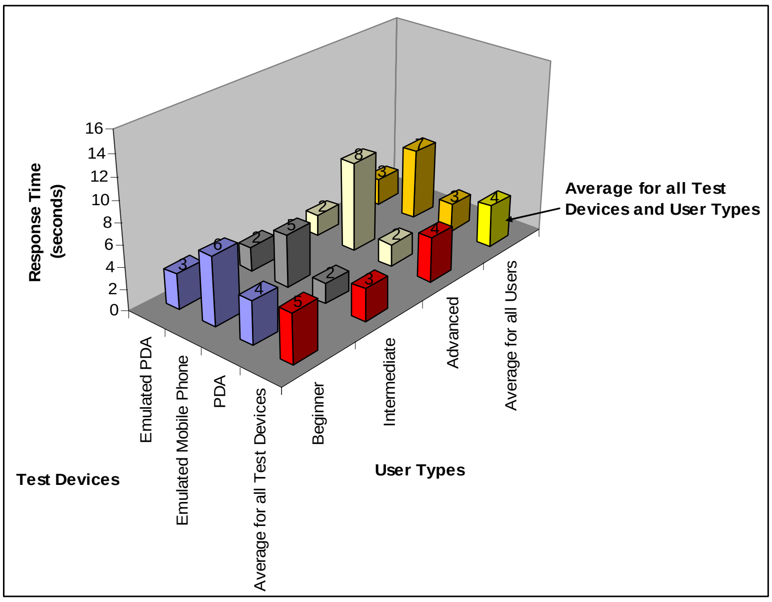


Figure 6-42: Usability Study Y: Combined Average Response Times, For the Three Test Devices and the Three User Types

The Facilitator observed that, in general, Advanced users were continuously re-positioning the Pivot-Point between alerts to pass time. A declining trend is shown for the Emulated PDA, and real PDA for all User Types, i.e. Beginner users recorded the highest Response Times, followed by Intermediate and Advanced users. Overall, there is not much difference in the Response Times between Test Devices and User Types; indicating good usability for the Recommendation Assistant.

Level of Distraction Analysis

Figure 6-43 shows Level of Distraction ratings given by the three User Types for using the Recommendation Assistant on the three Test Devices. As mentioned in Usability Study-G, a high rating implies ‘No Distraction’, while a Low rating indicates ‘High Distraction’, and a rating of 3 being ‘Neutral’.

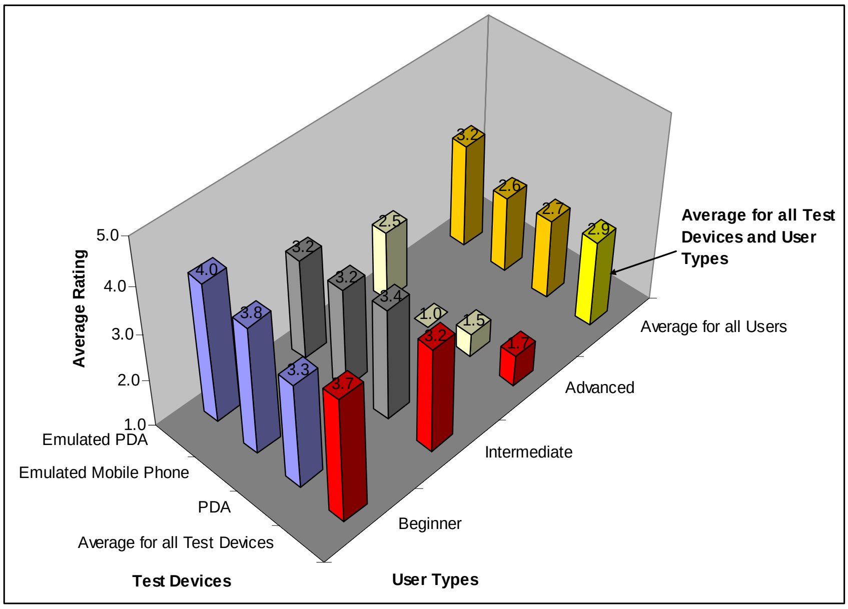


Figure 6-43: Usability Study Y: Level of Distraction Ratings For the Three Test Devices and the Three User Types

Figure 6-43 shows that for all Test Devices, Beginner users rated the Recommendation Assistant to be mildly distracting and Intermediate users gave slightly lower ratings indicating moderate distraction. Advanced users gave rather low ratings for all Test Devices, indicating high distraction. Comments made by some Advanced users indicated they did not like to be told what to do by an ‘inanimate device’, i.e. the computer.

Usability Ratings Analysis

Figure 6-44 shows that the Recommendation Assistant was rated reasonably high by all three User Types for its use on the three Test Devices. These high ratings average to a value of 3.8 on the Likert scale of 5, which indicates good usability for this interface.

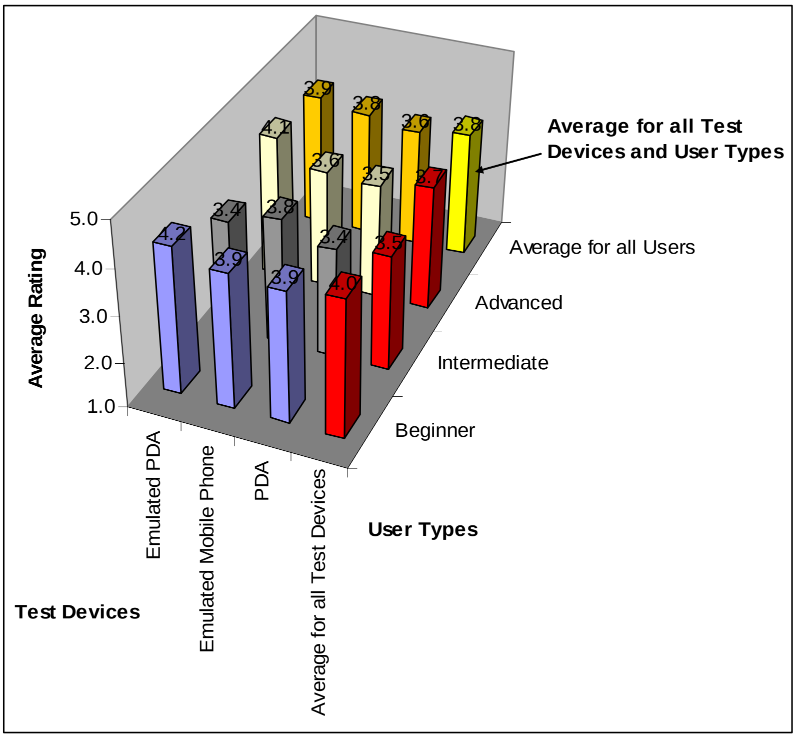
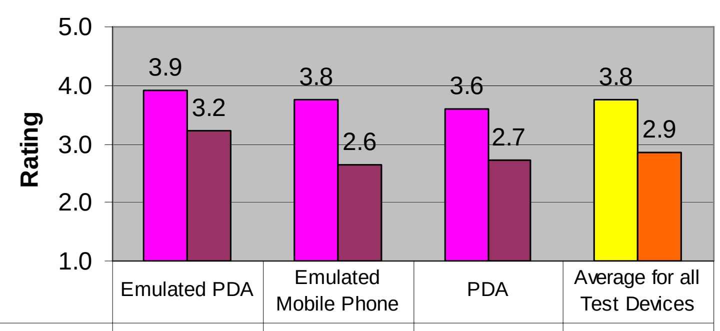


Figure 6-44: Usability Study Y: Combined Usability Ratings For the Three Test Devices and the Three User Types

Combined Average Ratings for Usability and Level of Distraction

Figure 6-45 shows the combined ratings for Usability and Level of Distraction for the three Test Devices. Results show that the Recommendation Assistant received high usability and Level of Distraction ratings for the Emulated PDA. The Emulated Mobile Phone received moderate Usability Ratings, and the lowest ratings for Level of Distraction. The PDA received the lowest Usability Ratings and a moderate rating for Level of Distraction. It can be deduced that the Recommendation Assistant worked best for the Emulated PDA, however, as real PDAs match the performance of Desktop Systems; these will be well suited for networked multimedia applications.



	Emulated PDA	Emulated Mobile Phone	PDA	Average for all Test Devices
Average Ratings for all Criteria	3.9	3.8	3.6	3.8
Level of Distraction Rating	3.2	2.6	2.7	2.9

Figure 6-45: Usability Study Y: Combined Usability Ratings For Evaluation Criteria and Level of Distraction

Overall, the Recommendation Assistant received higher ratings for usability than for the Level of Distraction for all Test Devices. Techniques for further improvement in the Level of Distraction are recommended for future research.

Participant Comments

Participants enjoyed interacting with the interface for completing System-Initiated QoS Re-negotiation. Beginner and Intermediate users expressed a positive attitude towards the operation of the interface, while some Advanced users found it distracting. Some users suggested increasing the speed of the Pivot-Points movement for the Emulated Mobile Phone, as the current slow movement caused them to become impatient.

6.10.5 Conclusion

In conclusion the Recommendation Assistant proved to be suitable for performing System-Initiated QoS Re-negotiation in real-time on an Emulated PDA, Emulated Mobile Phone and PDA. This conclusion was made based on the following observations.

- *Response Times* – Low Response Times were recorded for all three Test Devices. The Response Times were slightly higher for the Emulated Mobile Phone; however this was due to the keypad used for positioning the Pivot-Point.

- *Level of Distraction* – Ratings indicate that the Recommendation Assistant causes some distraction. However, this distraction was not significant enough to disrupt the multimedia session.

- *Usability Ratings* – The Recommendation Assistant received high ratings for its use on all three Test Devices, thus implying good usability for the same.

At the time of this study, the performance of mobile phones was not good enough to test this interface on a ‘real’ Mobile Phone; it is, therefore, recommended as future research.

6.11 Usability Study Z: Memorability Analysis

6.11.1 Overview

One of the major aspects to consider in usability is ‘memorability’, so that a casual user is able to return to the system after some period without having to learn everything all over again [64]. Improving the learnability of a user interface makes it easier to remember how to use it. For good learnability, the interface should be intuitive, simple, easy to use, and produce the required outcomes consistently. It is also important for that the interface is pleasant to use for first time users, this also improves its memorability, increasing the chance that they continue to use the system. If returning users find it difficult to use the interface, and are required to re-learn its operation, then they will avoid using it unless it is crucial [64].

There are various methods to test memorability for an interface. One method is to request the participant to re-use the interface after a certain time. Another method is to request the participant to complete a memorability questionnaire after using the interface.

In this study, participants were asked to complete a memorability questionnaire after using the interfaces tested in Usability Studies X and Y on the three Test Devices.

6.11.2 Memorability Questionnaire Design

Aims:

This study measures the memorability for the different features of the QCTT Interface to perform User-Initiated and System-Initiated QoS Re-negotiation.

Objectives:

The deliverable objectives for this study are:

- Measure the memorability for the five Interface Features: (a) QCTT Model/Interface, (b) Recommendation Assistant, (c) Threshold Lines, (d) Provision Ring, and (e) Pivot-Point Control Methods to re-negotiate QoS.
- Perform a memorability analysis for Beginner, Intermediate and Advanced users.

Participants:

The same twelve participants recruited in Usability Studies X and Y filled this memorability questionnaire. These studies used two Advanced, six Intermediate, and four Beginner users.

Memorability Measures

The memorability questionnaire comprised twenty five questions, which included a combination of ‘multiple choice’ answers and ‘true or false’ statements. The questionnaire measured the user’s memorability of the five Interface Features vis-à-vis the visual representation of an interface element and how it relates to its operation.

Memorability was measured as the sum of the number of questions answered correctly. The memorability questionnaire used in this study is given in Appendix I.

Test Conditions

A study carried out by Ebbinghaus (1913) in retention of memory [97], revealed that people retain in their short term memory 60% of the material presented to them up to twenty minutes; and thereafter their retention decreased progressively [98]. Ebbinghaus defines this as the ‘forgetting curve’.

It was therefore decided that participants complete the memorability questionnaire twenty minutes after completing the experiments for Usability Study X and Y. To obtain feedback that closely matches the memorability and behavioural conditions to the participants’ ‘natural’ environment, it was decided to inform them of the memorability questionnaire only at the end of the experiment. This prevented the user’s from preparing for the questionnaire, thus affecting their performance, and the memorability results. On being informed of the memorability questionnaire, participants were surprised and some were mildly bemused, however; none of them objected to answering the questionnaire.

6.11.3 Analysis

Figure 6-46 presents the memorability questionnaire results for the five Interface Features and the three user types as a percent score. These results show that, on average, the Threshold Line received the lowest memorability score, followed by the Recommendation Assistant, for all three user types.

The Provision Ring received the highest score, followed by the QCTT Model/Interface and then the Pivot-Point Control Methods.

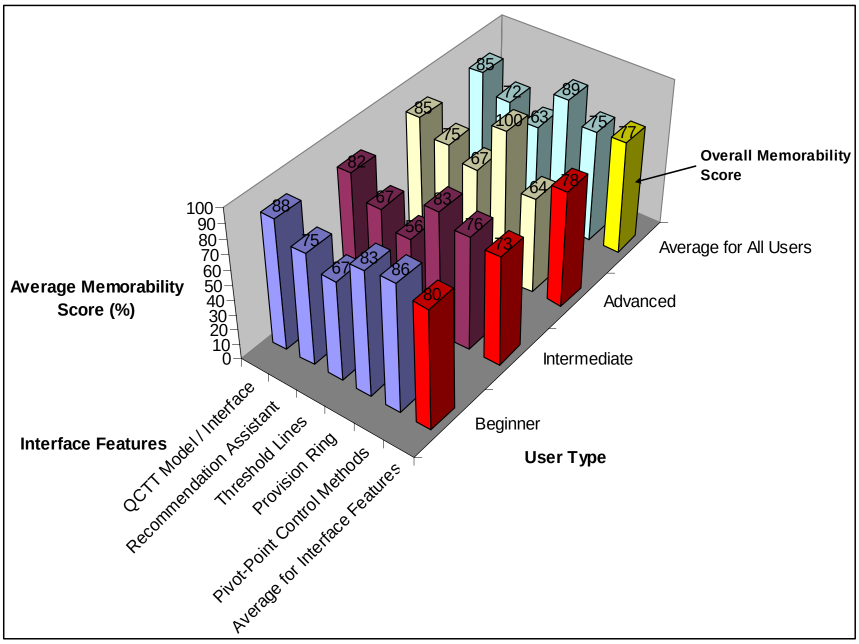


Figure 6-46: Usability Study Z: Memorability Questionnaire Results For the Five Interface Features and the Three User Types

Overall, Beginner users had the highest memorability score, followed by Advanced and Intermediate users; which was unexpected. However, this can be explained heuristically from the fact that Beginner users tend to pay more attention to the interface features, as they are apprehensive of their computing ability. The exact cause for this phenomenon would require further research into the behavioural differences between the three user types and their attitude towards graphical user interfaces. This is beyond the scope of the current research as it delves into Cognitive Psychology.

The overall memorability score for this interface is 77%, which is higher than the 60% memory retention score indicated by Ebbinghaus. This also confirms good usability for the entire interface.

6.11.4 Conclusion

In conclusion the QCTT Model/Interface, Recommendation Assistant, Threshold Lines, Provision Ring, and Control Methods implemented in a user interface for re-negotiating QoS has proven to be quite suitable. The high memorability score for each feature and the entire interface has revealed that both technical and non-technical users were able to remember how to use each feature for re-negotiating QoS. As the aim for this interface was to provide the ability for different types of users to re-negotiate QoS in real-time, this study has revealed that this aim has been successfully achieved.

6.12 Chapter Conclusion

This chapter presented individual usability studies for different user interfaces for Static and Dynamic QoS Management on Desktop Systems, and Dynamic QoS Management for Portable Systems.

Usability Study A investigated the usability of different Physical User Interface (PUI) devices for interacting with various Graphical User Interface (GUI) elements; aiming to determine the suitable interaction method that would be used for the QoS management interfaces. This study revealed the Pen & Touchpad and the Mouse & Keyboard to be most suitable. It was decided to use the Mouse & Keyboard combination for Desktop Computers, and the Pen & Touchpad combination for Portable Devices; being widely available for these devices.

Usability Study B investigated the suitability of different methods to provide system feedback for real-time multimedia systems; it was discovered that feedback methods that combine visual and audio alert were most preferred.

Usability Study C investigated different methods and related user interfaces for Static QoS Specification on Desktop Systems. It was discovered that the Profile with Example method was most suitable.

Usability Study D evaluated the usability of an interface and a related process for performing User-Initiated QoS Negotiation on a Desktop System. This study revealed good usability and learnability for the interface, making it suitable for Static QoS Negotiation. However, some refinements for the user interaction and interface design are required.

Usability Study E provided a usability comparison of the QCTT interface for the two implementation methods: Cartesian Coordinate System (CCS) and Triangular Fractal System (TFS). It was discovered that the TFS method was the better option for implementing the QCTT model as an interface for Dynamic QoS Management.

Usability Study F investigated the usability of the Provision Ring and the Threshold Lines feedback elements, implemented in the QCTT interface, for performing User-Initiated QoS Re-negotiation on a Desktop System. It was discovered that the two new feedback elements did not make much difference to the usability of the QCTT interface; however, the Provision Ring reduced the Task Completion Time substantially.

Usability Study G investigated the usability of the Recommendation Assistant to perform System-Initiated QoS Re-negotiation on a Desktop System. It was discovered that the Recommendation Assistant was rather useful for providing real-time QoS recommendations and it did not cause much distraction to the user.

Usability Study X investigated the usability of three user interfaces comprising different fractal sizes for the QCTT, for performing User-Initiated QoS Re-negotiation on the three Test Devices: Emulated PDA, Emulated Mobile Phone, and PDA. It was discovered that the use of fractal zones for the Large QCTT Quad Fractals interface and Small QCTT Quad Fractals interface made it easier to accurately specify QoS, regardless of the size of the QCTT Interface.

Usability Study Y revealed good usability for the Recommendation Assistant for performing System-Initiated QoS Re-negotiation on the three Test Devices: Emulated PDA, Emulated Mobile Phone, and PDA. It was discovered that Response Times and Level of Distraction ratings improved by using visual feedback only and excluding textual information for the system prompts.

Usability Study Z measured the memorability for the QCTT Model/Interface, Recommendation Assistant, Threshold Lines, Provision Ring, and Control methods implemented in the QCTT interface for re-negotiating QoS. The overall memorability score of 77% achieved for this interface is higher than the 60% memory retention score indicated by Ebbinghaus, thus confirming good usability and learnability for the entire interface.

Chapter 7
Conclusions & Further Research

Summary

This thesis presented a framework for developing a user-centred QoS management system. Research focused on usability studies of innovative interaction methods and interfaces for QoS management at the User Perspective. Analysis of research data confirmed the need for user interfaces to request, negotiate and re-negotiate QoS.

7.1 Introduction

Recent advancements in communication technologies have led to the development of networked multimedia systems that can now operate in fixed and mobile environments. Consequently, these technologies have been adopted by society and are used in various areas such as: Communication, Entertainment, Education, Edutainment, Health and Medicine, Marketing and Research.

As people are becoming dependant on these technologies, it is important that QoS guarantees are provided to ensure that users can effectively carryout their duties. As users require different quality guarantees for different multimedia applications, it is necessary for the users to be able to request QoS based on their needs, and negotiate a cost they are willing to pay. However, current communication and networking technologies predominantly support QoS guarantees at a technical level and do not provide such facilities for a layperson to negotiate QoS.

This research aimed to:

- Survey and identify the shortcomings of current technologies that support QoS guarantees for the delivery of multimedia information.

- Provide a framework for developing user-centred QoS management systems for networked multimedia applications.

- Investigate novel interaction methods and interfaces for technical and non-technical users to request and negotiate QoS prior to initiating a multimedia session, and re-negotiate QoS in real-time while the multimedia session is in progress.

- Study the usability of these novel user interfaces and their features for QoS management on Desktop Systems and Portable Devices.

This chapter concludes the thesis and presents major findings of this research. Section 7.2 addresses the aims and presents the conclusions drawn based on the outcomes of this research. Section 7.3 presents recommendations for future research.

7.2 Major Findings & Conclusions

This research project identified that current communication protocols and networking technologies have not been successful in providing integrated QoS, i.e. they do not include mechanisms for users to request and negotiate QoS. This issue has been addressed in this research by the development of a holistic QoS framework, and related models, that provide the foundation for developing a QoS management system. This thesis focused on the development of interfaces that enable users to request and negotiate QoS. Usability studies for these user interfaces were conducted on Desktop Systems and Portable Devices. The development and the findings of this research have been reported in eight papers given in the '*Publications Reporting Outcomes of this Research*' section, which is presented at the beginning of the thesis. Furthermore, industry application of the models and techniques developed in this research have been investigated and reported in [99], also given in the '*Publications Reporting Outcomes of this Research*' section.

The aims for this research, presented in Chapter 1, have been addressed and the following conclusions can be dawn.

Current QoS Models

Networked Multimedia Systems of today require QoS guarantees as they are being used in many aspects of our daily lives. Many organisations, businesses, companies, institutions and individuals are becoming reliant upon these technologies, requiring QoS guarantees. Failures in services for the delivery of multimedia information could lead to annoyance at the least, and to some catastrophic outcomes at the worst.

Consequently, a wide range of QoS communication protocols, QoS architectures and QoS negotiation techniques have been developed for QoS management. However, these technologies manage QoS primarily within the Transmission and Application Perspectives.

There is a need to develop a framework that integrates the User, Application and Transmission Perspectives, which takes into consideration the user's requirements for the provision of QoS. Development of such a system requires novel interaction methods and user interfaces that enable users to request and negotiate QoS. Usability studies for these interfaces are imperative to make certain that user's requirements have been met, vis-à-vis a user-centred QoS management system.

A User-Centred QoS Management Model

The Three Layer QoS (TRAQS) model integrates the User, Application, and Transmission Perspectives for the management of QoS, thus providing a framework for developing novel 'user-centric' QoS management systems.

The QoS Parameter Taxonomy and its related models provide a formal method for categorising parameters into various 'taxas', which assist the process of mapping user requests to parameter values. This facilitates the process for developing a profile system for the specification and negotiation of QoS parameters.

The Application Taxonomy provides a foundation for configuring QoS parameter values for different multimedia applications using an Application Profile Model (APM). The Application Classification Model (ACM) facilitates the process of providing correct advisory feedback to the user, based on the hard or soft requirements of a particular multimedia application.

Three performance aspects for the provision of QoS affect the Quality of Presentation for a given application, namely: quality, cost and time. The relationship between these performance aspects has an inherent triangular dependency; which can be represented as a Quality, Cost, and Temporal Triangle (QCTT). It is possible to implement this model as an interface to provide users the ability to manage QoS in real-time.

Novel Interaction Methods and Interfaces for QoS Management

QoS management takes place in two stages; Static QoS Management, and Dynamic QoS Management. Static QoS Management involves requesting and negotiating QoS prior to initiating a multimedia session, whereas Dynamic QoS Management entails re-negotiating QoS in real-time while the multimedia session is in progress.

To allow technical and non-technical users to request and negotiate QoS, novel interaction methods and interfaces for Static and Dynamic QoS Management are required. This research focused on the User Perspective Layer of the TRAQS model, where interaction methods and interfaces were developed, and tested for usability.

A prototype user interface, made up of panes and widgets, was developed based on the models and concepts devised in this research for the management of QoS. This interface presented a general overview of how various user interface elements can operate in a multimedia application (shown as an *Application Pane*) for QoS management. The following three panes were devised for specifying, negotiating and re-negotiating QoS.

- *QoS Specification Pane* presented an example of three methods implemented as widgets for specifying QoS, namely: Specification by Profile, Specification by Example, and Specification by Wizard.
- *QoS Negotiation Pane* presented an interface for negotiating QoS via the use of the three widgets: System Response, Cost, and User Action.
- *Dynamic QoS Management Pane* demonstrated an interface for re-negotiating QoS using the QCTT model; where the Real-Time QoS Control widget enables one to adjust and re-negotiate QoS in real-time, while system feedback is given using a Real-Time System Feedback widget.

This research covered the development of interfaces for Static and Dynamic QoS Management based on the specifications given in the prototype interface. Specific usability studies were conducted for each pane and widget, for managing QoS on Desktop Systems and Portable Devices.

To conduct these usability studies, re-engineering of current usability testing processes and re-configuration of the usability testing facility was required. This led to the development of a streamlined usability testing process that proved more efficient; it reduced the time taken to complete an experiment without affecting the experiment results.

Usability of Interfaces for QoS Management

Preliminary research confirmed that the most suitable user interaction device for Desktop Systems is the Mouse & Keyboard combination, and that the Pen & Touchpad combination is most suitable for Portable Devices.

Multimedia information includes audio and visual content; the relative importance of a particular content type depends upon the multimedia application. It is important that the system feedback methods do not distract the user from the multimedia session. Usability studies revealed that a combination of visual and audio feedback is necessary for system feedback. This outcome informed the development of the interfaces for QoS management. It was decided to use system feedback using colour schemes, textual description, and audio (where appropriate).

As there are no formal methods that allow technical as well as non-technical users to request QoS; a profile system was adopted and investigated. By defining a User Profile, it is possible to limit the technical level of specification. Application and QoS Profiles configured QoS parameters based on the QoS and Application Taxonomy. The Profile with Example interface was most suitable for QoS specification on Desktop Systems. User profiles gave good flexibility for different users to specify QoS based on their technical knowledge. Application and QoS profiles enabled the user to request QoS efficaciously; especially when combined with the option to preview an example of the content with the requested quality.

For negotiating QoS, a user interface was developed based on a User-Initiated QoS Negotiation process. This negotiation process imitated the traditional barter process that people use when shopping, where the user initiates the negotiation by requesting a quality and then negotiating a cost. The interface comprised of a User Action Widget, System Feedback Widget, User Status Display, System Status Display and a Cost Box to facilitate the negotiation process. Usability studies revealed positive feedback by all users and acceptance of the interface and the negotiation process. This revealed that such facilities are required, and traditional negotiation processes can be applied for negotiating QoS on Desktop Systems.

The QCTT model implemented as a user interface was efficacious for re-negotiating QoS in real-time. Prior to the development of the interface, two implementation options were investigated. Usability studies revealed that the Triangular Fractal System (TFS) implementation was superior to the Cartesian Coordinate System (CCS); as it achieved better usability, was easier to implement, and offers better scalability.

The QCTT interface was enhanced with the addition of a Provision Ring and a Threshold Lines, to provide real-time system feedback when users initiated re-negotiation during a multimedia session. These features provide an innovative method for users to re-negotiate QoS. Usability studies for this interface operating on a Desktop System revealed that these features did not make much difference to the usability of the QCTT interface; however, the Provision Ring reduced the time taken to re-negotiate QoS.

A Recommendation Assistant was added to the QCTT interface to provide System-Initiated QoS Re-negotiation, allowing for the system to suggest a QoS for the user in order to avoid disrupting the multimedia communication session. This feature achieved good feedback for its usability on Desktop Systems.

With the ongoing improvements in Portable and Mobile devices, multimedia applications are now being developed for these devices. However, the QoS issues for networked multimedia systems used on these devices are far greater to that of Desktop Systems. Therefore, QoS management interfaces are also required for these devices.

Generally, Portable Devices are operated using menu driven interfaces, which are similar to those found on Desktop Computers. As testing the QoS specification and negotiation interfaces on Portable Devices using menus would not lead to any novel usability aspects, this testing was limited to Desktop Systems. On the other hand, the

QCTT interface introduced a novel interaction method for even Portable Devices; consequently, the QCTT interface was enhanced for Portable Devices, and studies were conducted.

The QCTT interface, that was used to perform User-Initiated QoS Re-negotiation on Desktop Systems, was enhanced for the three portable Test Devices: Emulated PDA, Emulated Mobile Phone, and PDA. Due to the small screen size of Portable Devices, three variations of the QCTT interface were investigated. It was discovered that the use of fractal zones for the Large QCTT Quad Fractals interface and Small QCTT Quad Fractals interface made it easier to accurately specify QoS, regardless of the size of the QCTT interface. The performance of the Portable Device also plays an important role in its usability.

The Recommendation Assistant was also investigated for performing System-Initiated QoS Re-negotiation on the same three Test Devices. Results indicated good usability for this feature. Using visual feedback only and excluding textual information for the system prompts causes less distraction to the user.

Memorability tests for the QCTT interface revealed good memorability scores for technical and non-technical users, implying that the aim for good learnability and usability of this interface was achieved.

Summation

Presently there are no interfaces developed for Static and Dynamic QoS Management for networked multimedia systems operating on Desktop Systems and Portable Devices. This research has presented innovative interaction methods and interfaces for QoS management. Usability studies for these interfaces show positive feedback from technical as well as non-technical users. Users agreed that there is a need for such facilities that enable them to request, negotiate and re-negotiate QoS.

7.3 Recommendations for Future Research

Usability studies for the novel interfaces developed in this research answered some, and raised many more, questions. Some limitations were encountered which point towards the need for further work; thus, many new ideas have been generated and are suggested here as opportunities for further research.

Recommended Further Research

Study System-Initiated QoS Negotiation **-** For negotiating QoS prior to initiating a multimedia session, two methods for QoS negotiation were identified, namely: User-Initiated QoS Negotiation and System-Initiated QoS Negotiation. The User-Initiated QoS Negotiation method requires the user to initially request QoS without knowing what the system can provide; and then negotiate a quality and cost; thus this process is more complicated.

The System-Initiated QoS Negotiation involves the system presenting a list of the available levels of QoS and their cost, and then the user selects an option; thus this process is less complicated. It was deemed that the User-Initiated QoS Negotiation method better modelled traditional negotiation processes used by people when ‘buying’ products or services when ‘shopping’. This method was therefore implemented in a user interface and then subjected to usability studies.

As the System-Initiated QoS Negotiation process was simple, it did not introduce any novel user interface aspects to be investigated. However, outcomes of a survey suggested that 50% of the participants preferred the User-Initiated QoS Negotiation method, and the remaining would like to try the System-Initiated QoS Negotiation method. Therefore, it would be useful to investigate the usability of an interface for the System-Initiated QoS Negotiation method.

Investigate Optimal Fractal Sizes for the QCTT Interface – This research investigated the usability of fractal zones for large and small QCTT (triangular) interfaces, both containing four (quad) fractals. Usability studies suggested that the use of fractal zones made it easier to specify QoS on small screens, regardless of the size of the QCTT interface. It was concluded that different fractal sizes would be used for different sizes of the triangle. Therefore it will be beneficial to investigate what is the most optimal fractal size for different multimedia applications, operating on different devices.

Improve Level of Distraction for System Feedback Methods – As Desktop Systems have a larger screen, it is possible to include textual and visual information for system feedback. However, Portable Devices comprise a much smaller screen size, thus do not afford the luxury of including both textual and visual feedback. Due to the small screen size, users would want to utilise most of the space for the multimedia application itself. Usability studies for the Recommendation Assistant used on Desktop Systems and Portable Devices revealed that excluding textual information in the system prompts for the Portable Device caused less distraction to the user. Further investigations should be conducted to determine the optimal balance between textual and visual information.

Conduct Usability Studies for the QCTT Interface on a Real Mobile Phone – At the time of when the usability studies were conducted for this research, the performance of mobile phone handsets was not good enough to test the QCTT interface on a Real Mobile Phone. Therefore, usability testing had to be limited to an Emulated Mobile Phone. Given their performance has now improved, the QCTT interface should be tested on a Real Mobile Phone.

Test Usability of Interfaces in Natural Operating Environments – Usability studies conducted for this research were carried out in a usability laboratory, where controlled experiments were performed for testing the user interaction and interfaces for QoS specification, negotiation, and re-negotiation. It is recommended that further usability studies are carried out in a natural operating environment, bringing them closer to their real application. It would also be beneficial for these interfaces to be tested for QoS management on different multimedia applications such as: Video on Demand (VoD), Video Conferencing, Collaborative Conferencing, and Education-on-Demand (EoD).

Benefits of Suggested Future Research

The above suggestions for future research will assist in the ongoing improvement of the user interaction methods and interfaces for requesting, negotiating, and re-negotiating QoS. This will reduce the gap between the user requirements and the system functionality, thus, facilitating the development of User-Centred QoS Management systems. Bringing greater focus to the user in developing new QoS management systems, this will lead to a new era, where even the average person can carry out multimedia communications to enhance the benefits obtained from applications such as: education, entertainment, and edutainment.

Bibliography

[1] W. Stalling, *Data and Computer Communications*, Fifth ed. New York: Prentice Hall, 1997.

[2] Nortel Networks, "Benefits of Quality of Service (QoS) in 3G Wireless Internet," 1997. [Online]. Available: http://www.3gweekly.net/html/whitepapers/nt_qos.zip. [Accessed: Nov. 27, 2002].

[3] W. Saabeel, T. M. Verduijn, L. Hagdorn, and K. Kumar, "A Model of Virtual Organisation: A Structure and Process Perspective," *Electronic Journal of Organizational Virtualness*, vol. 4, pp. 1-16, 2002.

[4] N. Sharda, *Multimedia Information Networking*. New Jersey: Prentice Hall, 1999.

[5] H. Ma and G. K. Shin, "Multicast Video on Demand Services," *ACM SIGCOMM Computer Communication Review*, vol. 32, pp. 31-43, 2002.

[6] M. Maybury, "News on Demand: Introduction," *Communications of the ACM*, vol. 43, pp. 32-34, 2002.

[7] O. S. Sweeney, F. Crestani, and T. Anastasios, "Mobile delivery of news using hierarchical query-biased summaries," presented at Proceedings of the 2002 ACM symposium on Applied computing, Madrid, Spain, 2002.

[8] L. Goodstein, " Location based services making a humble comeback, declares ABI Research," *Directions Magazine*, 2004. [Online]. Available: http://www.directionsmag.com/press.releases/index.php?duty=Show&id=9222&trv=1. [Accessed: Mar. 3, 2005].

[9] Woo-Jin Choi and S. Tekinay, "Location based services for next generation wireless mobile networks," presented at The 57th IEEE Semiannual Vehicular Technology Conference, Jeju, South Korea, 2003.

[10] R. Vertegaal, I. Weevers, C. Sohn, and C. Cheung, "GAZE-2: Conveying Eye Contact in Group Video Conferencing using Eye-controlled Camera Direction," presented at Proceedings of the SIGCHI Conference on Human Factors in Computing systems, Ft. Lauderdale, Florida, USA, 2003.

[11] T. Plagemann and V. Goebel, "Experiences with the Electronic Classroom - QoS Issues in an Advanced Teaching and Research Facility," presented at Proceedings of the 6th IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS '97), Tunis (Tunesia), 1997.

[12] J. M. Pullen, "Applicability of Internet Video in Distance, Education for Engineering," presented at Proceedings of the IEEE Frontiers in Education Conference, Reno, NV, 2001.

[13] J. Leonard, E. Riley, and E. M. Staman, "Classroom and support innovation using IP video and data collaboration techniques," presented at Proceedings of the 4th Conference on information Technology Curriculum, CITC4 '03, 2003.

[14] Microsoft Windows Server TechCenter, "What is QoS?," *Microsoft Corporation*, 2003. [Online]. Available: http://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/TechRef/1c1f53a6-da9e-496f-be84-b91e2763dbeb.mspx. [Accessed: Sept. 18, 2005].

[15] Cisco Systems Inc, *Internetworking Technologies Handbook*. Indianapolis: Cisco Press, 2003.

[16] Dictionary.com, "Quality of Service," *Lexico Publishing Group, LLC*, 1998. [Online]. Available: http://dictionary.reference.com/search?q=quality%20of%20service. [Accessed: Sept. 18, 2005].

[17] ISO/IEC JTC1/SC21, "Basic Reference Model of Open Distributed Processing, Part 2: Descriptive Model," ITU-T X.902 - ISO/IEC 10746-2, August, 1994.

[18] N. Wang, K. Parameswaran, M. Kurcher, and D. C. Schmidt, "Applying Reflective Middleware Techniques to Optimize a QoS Enabled CORBA Component Model Implementation," presented at The 24th International Computer Software and Applications Conference, COMPSAC 2000, Taipei, 2000.

[19] S. Jackowski, "Native ATM Support for ST2+," RFC1946, *Internet Engineering Task Force (IETF)*, 1996. [Online]. Available: http://www.ietf.org/rfc/rfc1946.txt. [Accessed: Sept. 23, 2005].

[20] F. Fluckiger, *Understanding Networked Multimedia*: Prentice Hall, Inc., 1995.

[21] A. Vogel, B. Kerhervé, G. V. Bochmann, and J. Gecsei, "Distributed Multimedia Applications and Quality of Service - A Survey," *IEEE Multimedia*, vol. 2, pp. 10-19, 1995.

[22] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification," RFC 2205, *IETF Working Group*, 1997. [Online]. Available: ftp://ftp.isi.edu/in-notes/rfc2205.txt. [Accessed: Aug. 3, 2002].

[23] F. Halsall, *Data Communications, Computer Networks and Open Systems*, Fourth ed. UK: Addison Wesley, 1996.

[24] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," RFC 3031, *IETF Working Group*, 1999. [Online]. Available: http://www.ietf.org/rfc/rfc3031.txt. [Accessed: Sept. 6, 2005].

[25] White Paper, "DiffServ – The Scalable End-to-End QoS Model," *Cisco Systems, Inc*, 2005. [Online]. Available: http://www.cisco.com/en/US/products/ps6610/products_white_paper09186a00800a3e2f.shtml. [Accessed: Sept. 20, 2005].

[26] R. Yavatkar, D. Hoffman, Y. Bernet, and F. Baker, "SBM (Subnet Bandwidth Manager): A Protocol for RSVP-based Admission Control over IEEE 802-style Networks," RFC2814, *IETF Working Group*, 2000. [Online]. Available: http://www.ietf.org/rfc/rfc2814.txt. [Accessed: Aug. 5, 2002].

[27] T. Campbell, C. Aurrecoechea, and L. Hauw, "A review of QoS Architectures," presented at Proceedings of the 4th International Workshop on Quality of Service, Paris, France, 1996.

[28] T. Campbell and G. Coulson, "QoS Adaptive Transports: Delivering Scalable Media to the Desktop," *IEEE Network Magazine*, vol. 11, pp. 18-27, 1997.

[29] D. J. Ecklund, V. Goebel, T. Plagemann, and E. F. Ecklund, "A Dynamically-Configured, Strategic QoS Management Hierarchy for Distributed Multimedia Systems," presented at Proceedings of the 2001 International Workshop on Multimedia Middleware, Ottawa, Ontario, Canada, 2001.

[30] B. Bauer and A. S. Patrick, "A Human Factors Extension to the Seven-Layer OSI Reference Model," 2004. [Online]. Available: http://www.andrewpatrick.ca/OSI/10layer.html. [Accessed: Jan. 15, 2005].

[31] C. Allison, H. Lawson, D. McKechan, and A. Ruddle, "A Holistic View of Quality of Service," *Interactive Learning Environments*, vol. 9, pp. 33-50, 2001.

[32] M. Dabrowski, G. Eichler, M. Fudala, D. Katzengruber, T. Kilkanen, N. Miettinen, H. Tarasiuk, and M. Titze, "Evaluation of the AQUILA Architecture: Trial Results for Signalling Performance, Network Services and User Acceptance," in *Lecture Notes in Computer Science*, vol. 2698. Berlin / Heidelberg: Springer, 2003, pp. 218-233.

[33] G. Ghinea, J. P. Thomas, and R. S. Fish, "Multimedia, Network Protocols and Users – Bridging the Gap," presented at Proceedings of the Seventh ACM International Conference on Multimedia, Orlando, Florida, United States, 1999.

[34] G. Ghinea and J. P. Thomas, "An Approach Towards Mapping Quality Of Perception To Quality Of Service In Multimedia Communications," presented at Proceedings of the IEEE Multimedia Signal Processing Workshop, Copenhagen, Denmark, 1999.

[35] D. Chalmers and M. Sloman, "A Survey of Quality of Service in Mobile Computing Environments," *IEEE Communications Surveys & Tutorials*, vol. 2, pp. 2-10, 1999.

[36] T. Plagemann, K. Shire, and V. Goebel, "Application Requirements and QoS Negotiation in Multimedia Systems," presented at Second International Workshop on Protocols for Multimedia Systems (PROMS 95), Salzburg, Austria, 1995.

[37] J. Wroclawski, "The Use of RSVP with IETF Integrated Services," RFC2211, *IETF Working Group*, 1997. [Online]. Available: http://www.ietf.org/rfc/rfc2211.txt. [Accessed: Sept. 9, 2003].

[38] White Paper, "QoS Protocols & Architectures – A White Paper," *Quality of Service Forum*, 1999. [Online]. Available: http://www.qosforum.com/white-papers/qosprot_v3.pdf. [Accessed: Sept. 14, 2002].

[39] Nortel Networks, "The Case for IPV6 Abstract," 1999. [Online]. Available: http://www.nortelnetworks.com/corporate/technology/ipv6/doclib.html. [Accessed: Sept. 14, 2003].

[40] R. Callon, N. Freedman, A. Fredette, G. Swallow, and A. Viswanathan, "A Framework for Multiprotocol Label Switching," *Internet Draft*, 1999. [Online]. Available: http://www.globecom.net/ietf/draft/draft-ietf-mpls-framework-02.html. [Accessed: Sept. 21, 2003].

[41] K. Nichols, V. Jacobson, Cisco, L. Zhang, and UCLA, "A Two-bit Differentiated Services Architecture for the Internet," RFC2638, *IETF Working Group*, 1999. [Online]. Available: http://tools.ietf.org/html/2638. [Accessed: Sept. 19, 2003].

[42] J. Ghanwani, W. Pace, V. Srinivasan, A. Smith, and M. Seaman, "A Framework for Integrated Services Over Shared and Switched IEEE 802 LAN Technologies," RFC2816, *IETF Working Group*, 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2816.txt. [Accessed: Sept. 4, 2003].

[43] D. B. Hehmann, R. G. Herrtwich, W. Schulz, T. Schuett, and R. Steinmetz, "Implementing HeiTS: Architecture and Implementation Strategy of the Heidelberg High Speed Transport System," presented at The 2nd International Workshop on Network and Operating System Support for Digital Audio and Video, Heidelberg, Germany, 1991.

[44] C. Sluman, "Quality of Service in Distributed Systems." British Standards Institution, UK, 1991.

[45] K. Nahrstedt and J. Smith, "Design, Implementation and Experience of the OMEGA End-point Architecture," *IEEE Journal on Selected Areas in Communications*, vol. 14, pp. 1263–1279, 1996.

[46] White Paper, "3G/UMTS Towards Mobile Broadband and Personal Internet," *UMTS Forum*, 2005. [Online]. Available: http://www.umts-forum.org/servlet/dycon/ztumts/umts/Live/en/umts/MultiMedia_PDFs_Papers_Towards-Mobile-Broadband-Oct05.pdf. [Accessed: Nov. 12, 2005].

[47] T. Savage, R. Meier, and V. Cahill, "Near and far: Exploring the Use of Broadband Technology to Overcome Segregation between Distance and Face-to-Face Students," presented at Proceedings of the 2nd International Conference on Technology in Teaching and Learning in Higher Education, Greece, 2001.

[48] S. D. Antonio, M. Esposito, S. P. Romano, and G. Ventre, "Assessing the Scalability of Component-Based Frameworks: The CADENUS Case Study," *SIGMETRICS Perform. Eval. Rev.*, vol. 32, pp. 34-43, 2004.

[49] M. Diaz, R. Canonico, L. Costa, S. Fdida, D. Hutchison, L. Mathy, A. Meissner, S. Owezarski, R. Vida, and L. C. Wolf, "GCAP: A New Multimedia Multicast Architecture for QoS," presented at Proceedings of the 6th international Conference on Protocols for Multimedia Systems, PROMS 2001, Enschede, The Netherlands, 2001.

[50] P. Arabas, M. Kamola, K. Malinowski, and M. Malowidzki, "Pricing for IP Networks and Services," *Information Knowledge System Management*, vol. 3, pp. 153-171, 2002.

[51] TEQUILA Project, "Traffic Engineering for Quality of Service in the Internet, at Large Scale," *IST TEQUILA*, 2004. [Online]. Available: http://www.ist-tequila.org/. [Accessed: Nov. 5, 2005].

[52] W. L. Bethea, "Adding Parametric Polymorphism to the Common Object Request Broker Architecture (CORBA)," presented at Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications, OOPSLA 2000, Minneapolis, Minnesota, United States, 2000.

[53] T. Engel, H. Granzer, K. B. Koch, M. Winter, P. Sampatakos, I. S. Venieris, H. Hussmann, F. Ricciato, and S. Salsano, "AQUILA Adaptive Resource Control for QoS Using an IP-Based Layered Architecture," *IEEE Communications Magazine*, vol. 41, pp. 46-53, 2003.

[54] G. Ghinea and G. D. Magoulas, "Quality of Service for Perceptual Considerations: An Integrated Perspective," presented at IEEE International Conference on Multimedia and Expo, ICME 2001, Tokyo, Japan, 2001.

[55] UsabilityNet, "International Standards for HCI and Usability," 2003. [Online]. Available: http://www.usabilitynet.org/tools/r_international.htm. [Accessed: Jul. 24, 2005].

[56] The Usability Company, "History of Usability," 2005. [Online]. Available: http://www.theusabilitycompany.com/resources/history.html. [Accessed: Jul. 24, 2005].

[57] N. Sharda, M. Georgievski, I. Ahmed, J. A. Leisa, M. Brogan, A. Woodrard, G. Kohli, and M. Clark, "Leading Edge Developments in Tourism ICT and Related Underlying Technologies: Key Findings and Future Directions. Project# 70153," *Sustainable Tourism CRC*, 2005. [Online]. Available: http://melba.vu.edu.au/STCRC/2004/2004FinalReport/STCRCReport2004V5.7.pdf. [Accessed: Nov. 5, 2005].

[58] M. Pearrow, *Web Site Usability Handbook*. Rockland, Massachusetts: Charles River Media, 2000.

[59] D. Thompson, P. Williams, D. Nicholas, and P. Huntington, "Accessibility and Usability of a Digital TV health Information Database," *Aslib Proceedings: New Information Perspectives*, vol. 54, pp. 294-308, 2002.

[60] L. K. Peng, C. K. Ramaiah, and S. Foo, "Heuristic-Based User Interface Evaluation at Nanyang Technology University in Singapore," *Program: Electronic Library & Information Systems*, vol. 38, pp. 42-59, 2004.

[61] C. Condos, A. James, P. Every, and T. Simpson, "Ten Usability Principles for the Development of Effective WAP and M-Commerce Services," *Aslib Proceedings: New Information Perspectives*, vol. 54, pp. 345-355, 2002.

[62] M. H. Blackmon, P. G. Polson, M. Kitajima, and C. Lewis, "Cognitive Walkthrough for the Web," presented at Proceedings of the on Human Factors in Computing Systems, Minneapolis, CHI 2002, Minnesota, USA, 2002.

[63] M. F. Costabile, "Usable Multimedia Applications," presented at IEEE International Conference on Multimedia Computing and Systems, 1999.

[64] J. Nielsen, *Usability Engineering*. Boston: Academic Press, 1993.

[65] J. Hom, "The Usability Methods Toolbox," 1998. [Online]. Available: http://jthom.best.vwh.net/usability/. [Accessed: Nov. 19, 2005].

[66] J. Preece, Y. Rogers, and H. Sharp, *Interaction Design: Beyond Human-Computer Interaction*, First ed. Yew York: John Wiley & Sons Inc, 2002.

[67] N. Sharda and M. Georgievski, "A Holistic Quality of Service Model for Multimedia Communications," presented at International Conference on Internet and Multimedia Systems and Applications, IMSA2002 Kaua'i, Hawaii, USA, 2002.

[68] M. Georgievski and N. Sharda, "A Taxonomy of QoS Parameters and Applications for Multimedia Communications," presented at International Conference on Internet and Multimedia Systems and Applications, IMSA2003 Kaua'i, Hawaii, USA, 2003.

[69] E. England and A. Finney, *Managing Multimedia - Project Management for Interactive Media*, Second ed. Harlow, England: Addison-Wesley, 1999.

[70] F. Paternò, "Tools for Task Modelling: Where we are, Where we are headed," presented at Proceedings of the First International Workshop on Task Models and Diagrams for User Interface Design, TAMODIA 2002, Bucharest, Romania, 2002.

[71] M. Georgievski and N. Sharda, "Task Modelling for a Holistic Quality of Service Model – TRAQS," presented at IEEE India Council Annual Convention and Exhibition 2003, ACE2003, Hotel Le Meridien, Pune, India, 2003.

[72] F. Alexander, "Introduction to Task Modelling," 2001. [Online]. Available: http://www.scenarioplus.org.uk/handwritten/introduction.html. [Accessed: Aug. 1, 2003].

[73] B. A. Myers, "A Brief History of Human Computer Interaction Technology," *ACM Interactions*, vol. 5, pp. 44-54, 1998.

[74] G. Mori, F. Paterno, and C. Santoro, "CTTE: Support for Developing and Analysing Task Models for Interactive System Design," *IEEE Transactions on Software Engineering*, vol. 28, pp. 797-813, 2002.

[75] G. W. Reynolds, *Ethics in Information Technology*, First ed. Canada: Course Technology, 2002.

[76] D. G. Johnson, *Computer Ethics*, Third ed. New Jersey: Prentice Hall, 2001.

[77] K. Ehrlich, Butler, M. B., and Pernice, K., "Getting the Whole Team into Usability Testing," *IEEE Software*, vol. 11, pp. 89-91, 1994.

[78] J. Kirakowski, "Questionnaires in Usability Engineering," *Human Factors Research Group*, 2000. [Online]. Available: http://www.ucc.ie/hfrg/resources/qfaq1.html. [Accessed: Dec. 8, 2005].

[79] K. Eason, *Information Technology and Organisational Change*. Bristol, PA, USA: Taylor & Francis, Hemisphere Publishing, 1988.

[80] I. Roth, "An Introduction to Object Perception," in *Perception Representation: A Cognitive Approach*, I. Roth and J. B. Frisby, Eds.: Milton Keynes, Open University, 1986.

[81] L. Wesson, "An Introduction to Usability Engineering," *Usability Engineering Seminar*, 2001. [Online]. Available: http://www.chi-sa.org.za/seminar%5CDVanGreunen.ppt. [Accessed: Dec. 2, 2005].

[82] J. Mullin, M. Jackson, A. H. Anderson, L. Smallwood, M. A. Sasse, A. Watson, and G. Wilson, "The ETNA Taxonomy," *UCL Network and Multimedia Research Group*, 2002. [Online]. Available: http://www-mice.cs.ucl.ac.uk/multimedia/projects/etna/taxonomy.pdf. [Accessed: Nov. 14, 2003].

[83] P. M. Mullins and S. Treu, "Measurement of Stress to Gauge User Satisfaction with Features of the Computer Interface," *Behaviour and Information Technology - BIT*, vol. 10, pp. 325-343, 1991.

[84] J. Nielsen, "Heuristic Evaluation," in *Usability Inspection Methods*, J. Nielsen and R. L. Mack, Eds. New York, NY: John Wiley & Sons, 1994.

[85] Microsoft Corporation, "Usability Research, Usability Labs," 2002. [Online]. Available: http://www.microsoft.com/usability/tour.htm. [Accessed: Jul. 18, 2004].

[86] J. Kimber, M. Georgievski, and N. Sharda, "Developing Usability Testing Systems and Procedures for Mobile Tourism Services," presented at Annual Conference on Information Technology in the Hospitality Industry, HITA 2005 Hospitality Information Technology Association, Los Angeles, USA, 2005.

[87] W. Milszus, "Comparison of Usability Testing Methods.," *Project P807, JUPITER2 - Joint Usability, Performability and Interoperability Trials in Europe, European Institute for Research and Strategic Studies in Telecommunications, EURESCOM*, 1999. [Online]. Available: http://www.eurescom.de/~public-webspace/P800-series/P807/Results/Usability/R1/D2-T3-Usability-R1-TestMethods.pdf. [Accessed: Jul. 16, 2004].

[88] M. Georgievski and N. Sharda, "Re-Engineering the Usability Testing Process for Live Multimedia Systems," *Journal of Enterprise Information Management (EIMJ)*, vol. 19, pp. 223-233, 2006.

[89] M. Georgievski and N. Sharda, "Re-Engineering the Usability Testing Process for Live Multimedia Systems," presented at The 2004 International Research Conference on Innovations in Information Technology, IIT2004, Dubai, UAE, 2004.

[90] M. Georgievski and N. Sharda, "Usability Testing for Real-Time QoS Management," presented at The 24th IEEE International Real-Time Systems Symposium, RTSS2003, Cancun, Mexico, 2003.

[91] X. Gu, K. Nahrstedt, W. Yuan, D. Wichadakul, and D. Xu, "An XML-based QoS Enabling Language for the Web," *Journal of Visual Language and Computing, Special Issue on Multimedia Languages for the Web*, vol. 13, pp. 61-95, 2002.

[92] X. Gu, D. Wichadakul, and K. Nahrstedt, "Visual Quality of Service Programming Environment for Ubiquitous Multimedia Services," presented at Proceedings of IEEE International Conference on Multimedia and Expo 2001 (ICME2001), Tokyo, Japan, 2001.

[93] M. Georgievski and N. Sharda, "Enhancing User Experience for Networked Multimedia Systems," presented at The 4th International Conference on Information Systems Technology and its Applications, ISTA2005, Massey University, Palmerston North, New Zealand, 2005.

[94] T. Nelson, *Computer Lib/Dream Machines*. Redmond, WA: Microsoft Press, 1987.

[95] M. Georgievski and N. Sharda, "Implementation and Usability of User Interfaces for Quality of Service Management," presented at Tencon’05, Annual Technical Conference of IEEE Region 10, Melbourne, Australia, 2005.

[96] R. R. Schaller, "Moore's Law: Past, Present, and Future," *IEEE Spectrum*, vol. 34, pp. 52-59, 1997.

[97] H. Ebbinghaus, *Memory. A Contribution to Experimental Psychology*. New York: Teachers College, Columbia University (Reprinted Bristol: Thoemmes Press, 1999), 1913.

[98] J. T. Wixted and E. Ebbesen, "On the Form of Forgetting," *Psychological Science*, vol. 2, pp. 409-415, 1991.

[99] N. Sharda, M. Georgievski, and K. Lim, "Mapping the TRAQS Quality of Service Model to the m-Apps Middleware," presented at M2USIC 2005: MMU International Symposium on Information and Communications Technologies, PJ Hilton Hotel, Petaling Jaya, Malaysia, 2005.

Appendix A
Re-engineering Usability Testing Processes

This appendix presents an overview of a comparative study for the traditional usability testing process and the re-engineered usability testing process. Usability studies performed in this research employed the re-engineered usability testing process. This appendix expands on the summary of this study given in section 4.5.

This study has been published by Georgievski and Sharda, and presented in the following Journal and Conference:

- M. Georgievski and N. Sharda, "Re-Engineering the Usability Testing Process for Live Multimedia Systems," *Journal of Enterprise Information Management (EIMJ)*, vol. 19, pp. 223-233, 2006.

- M. Georgievski and N. Sharda, "Re-Engineering the Usability Testing Process for Live Multimedia Systems," presented at The 2004 International Research Conference on Innovations in Information Technology, IIT2004, Dubai, UAE, 2004.

Details for this investigation are included on the Thesis CD, which is attached to the back of this thesis.

Appendix B
Usability Test Scenario Example

This appendix presents an example of a test scenario used in this research project. Test scenarios were created in Microsoft PowerPoint and presented to participants on a computer screen.

The test scenarios created for all other usability studies were based on a similar structure and visual presentation, to the one presented in this appendix.

The usability test scenario example is included on the Thesis CD.

Appendix C
Pre-Experiment Questionnaire Example

This appendix presents an example of the Pre-experiment Questionnaire used in the usability studies to obtain participant background information. The Pre-Experiment Questionnaire was created in Microsoft Excel, where participants entered their feedback directly into the spreadsheet.

The Pre-Experiment Questionnaires produced for other usability studies were based on a similar structure and visual presentation to the one presented in this appendix.

An example of the Pre-Experiment Questionnaire used in this research is included on the Thesis CD.

Appendix D
Post-Experiment Questionnaire Example

This appendix presents an example of a Post-experiment Questionnaire used in this research project for obtaining feedback from participants. Post-Experiment Questionnaires were created in Microsoft Excel, where participants entered their feedback directly into the spreadsheet.

Different Post-Experiment Questionnaires were created for specific usability studies, which used a similar structure and visual presentation to the Post-Experiment Questionnaire presented in this appendix.

An example of Post-Experiment Questionnaire used in this research is included on the Thesis CD.

Appendix E
QCTT Implementation using the CCS & TFS Methods

This appendix presents the theory for implementing the QCTT model using the two methods: Cartesian Coordinate System (CCS) and Triangular Fractal System (TFS). This appendix expands on the user interface design presented in Usability Study E (section 6.6).

This theory has been published by Georgievski and Sharda, and presented in the following conference:

- M. Georgievski and N. Sharda, "Implementation and Usability of User Interfaces for Quality of Service Management," presented at Tencon'05, Annual Technical Conference of IEEE Region 10, Melbourne, Australia, 2005.

Details for the CCS and TFS implementation methods are included on the Thesis CD.

Appendix F
QCTT Interface Implementation for Portable Devices

This appendix presents the implementation theory for the three interfaces: (a) Large QCTT Quad Fractals, (b) Small QCTT Quad Fractals, and (c) Large 10,000 (10k) Fractals. Details are given for the application of Fractal Zones, which enhance the QCTT interface for use on Portable Devices.

This appendix expands on the user interface designs presented in Usability Study X (section 6.9).

Details for implementing the three interfaces are included on the Thesis CD.

Appendix G
Detailed Results for Usability Study X

This appendix includes detailed results for Usability Study X. (section 6.9). These results are included on the Thesis CD.

Appendix H
Detailed Results for Usability Study Y

This Appendix includes detailed results for Usability Study Y (section 6.10). These results are included on the Thesis CD.

Appendix I
Memorability Questionnaire

This appendix presents the Memorability Questionnaire used to measure the memorability for the QCTT interface that was enhanced for Portable Devices, as presented in Usability Study Z (section 6.11). The Memorability Questionnaire is included on the Thesis CD.

THESIS APPENDICES

Contents

List of Figures

List of Tables

Appendix A
Re-engineering Usability Testing Processes

This appendix presents an overview of a comparative study for the traditional usability testing process and the re-engineered usability testing process. Usability studies performed in this research employed the re-engineered usability testing process. This appendix expands on the summary of this study given in section 4.5.

This study has been published by Georgievski and Sharda, and presented in the following Journal and Conference:

- M. Georgievski and N. Sharda, "Re-Engineering the Usability Testing Process for Live Multimedia Systems," *Journal of Enterprise Information Management (EIMJ)*, vol. 19, pp. 223-233, 2006.

- M. Georgievski and N. Sharda, "Re-Engineering the Usability Testing Process for Live Multimedia Systems," presented at The 2004 International Research Conference on Innovations in Information Technology, IIT2004, Dubai, UAE, 2004.

A.1 Introduction

To efficiently carry out usability studies, the usability test facility must be configured according to the type of tests performed. This also involves developing usability testing processes that are suited to the type of application to be tested. An investigation was conducted to evaluate current traditional usability testing processes and determine their suitability for the usability studies to be performed for this research. This investigation resulted in re-engineering current usability testing processes and re-configuring of the usability test facility.

A.2 Customising the Usability Test Facility

A.2.1 The Traditional Approach

The initial configuration of the usability testing facility was based the traditional approach. The observation room comprised an analog monitoring system that consisted of a four channel video multiplexer, giving four different camera views of the participant (Figure A-1); side profile, front face, controller view and screen capture view.

Figure A-1: Monitoring System Screen Capture

The monitoring system configuration included a video multiplexer that enables to switch different camera views, or show multiple camera views simultaneously. As shown in Figure A-2, each camera is connected to the video multiplexer. The multiplexer outputs the video signal to a Video Cassette Recorder (VCR) and from this point, video signals are sent in parallel to the TV monitor and video switch. The video switch sends the signal to a video capture card, which is installed on a desktop computer. This enables to digitally record the information, and if required, allows streaming of this information onto a website for demonstration purposes. The architecture shown in Figure A-2 is defined as a hybrid monitoring system that supports analog and digital recording.

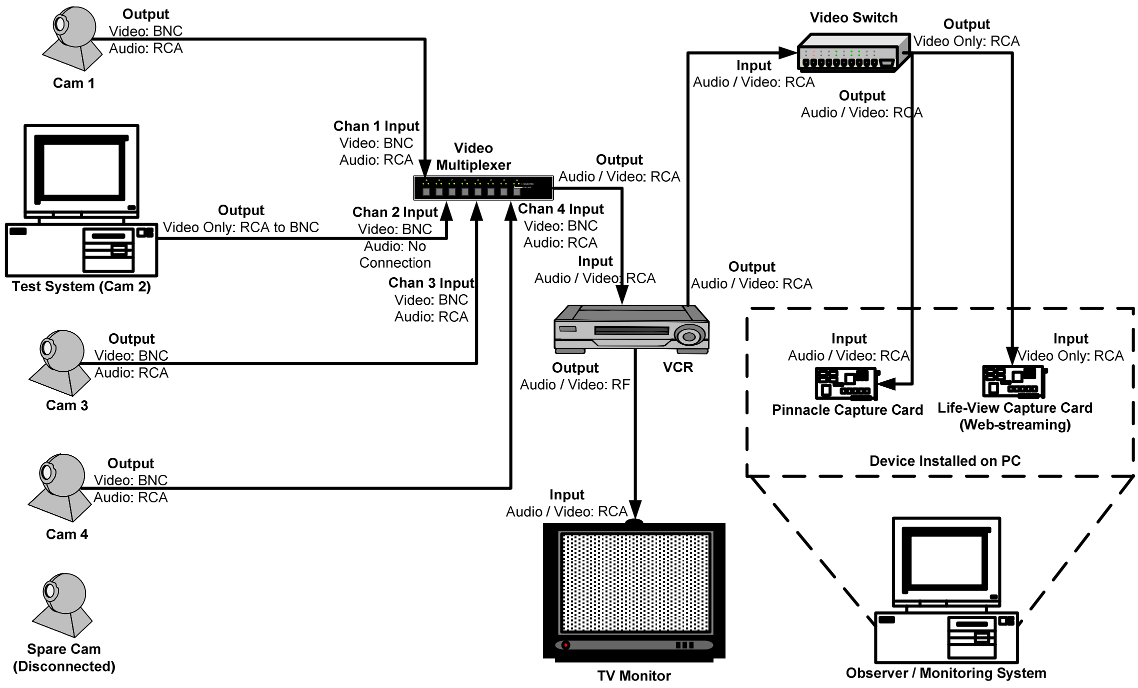


Figure A-2: Usability Laboratory Monitoring System Circuit Diagram

The observation room was used to observe the participant and record the experiment on the VCR. The observer manually recorded the Task Completion Times on hard copy. The participant performed the experiment on one of the three test systems situated in the participant room, while reading the scenario from a hard copy. Pre-experiment and post-experiment questionnaires were filled-out by the participant on hard copies. The facilitator manually recorded the Error Count per task on paper.

A.2.2 The Re-engineered Approach

The *re-engineered usability test facility* comprised an overhauled architecture. As shown in Figure A-3, the test system was augmented by a Scenario Presentation System, and a Data Collection System.

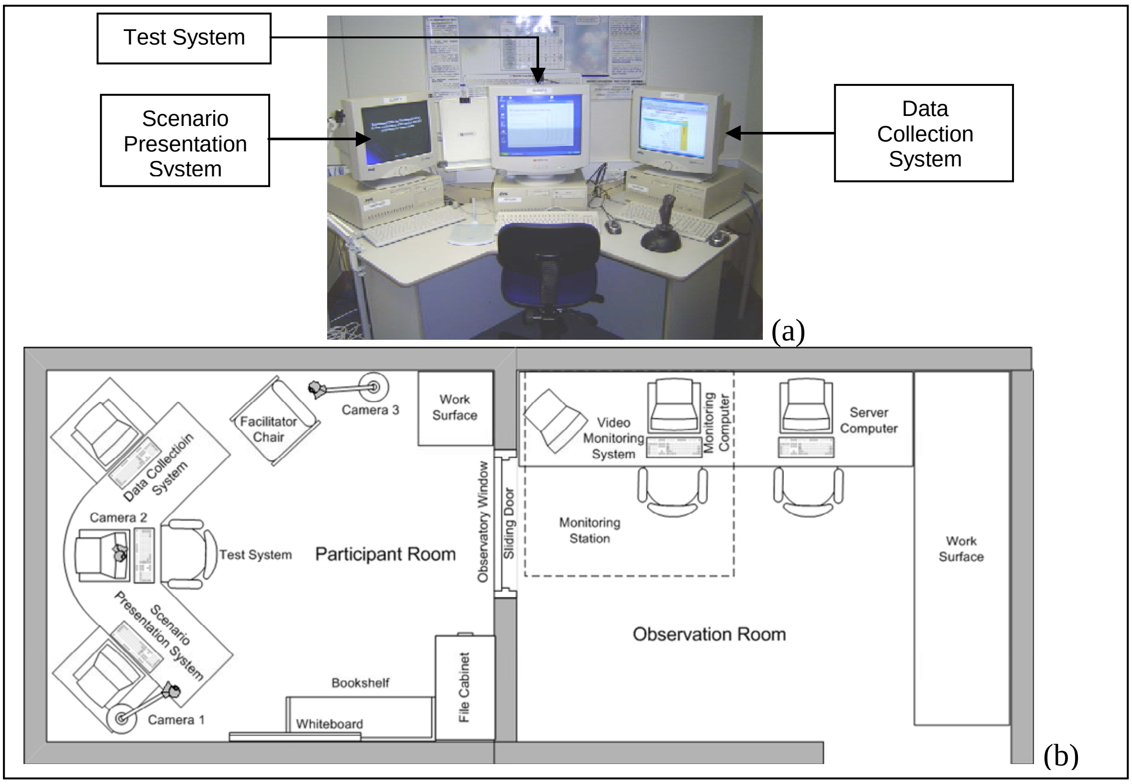


Figure A-3: Usability Laboratory Setup,

(a) Photograph of the Participant Room, (b) Laboratory configuration.

This usability testing facility comprised the following hardware and software configuration.

Hardware Configuration

- *Test System* – user interfaces and applications were loaded onto the test system. The participant interacted with them on the this same system

- *Scenario Presentation System* – used to present the scenarios and tasks to be performed in the experiment.

- *Data Collection System* – comprised software to collect feedback from the participant via questionnaires.

- *Monitoring Computer* – served the purpose of recording and monitoring the experiment progress, which includes recording Task Completion Times, etc.

- *Monitoring System* – included the same configuration that was used in the traditional usability testing facility.

Software Configuration

- *Presentation Software* – Microsoft PowerPoint was used to present the scenarios and tasks for the experiment.

- *Feedback Collection Software* – Questionnaires were developed in Microsoft Excel and used to gather feedback from participants.

- *Task Performance Monitoring* – a stop watch application was used on the Monitoring Computer to record the Task Completion Times for the experiments.

The facilitator was positioned in such a way that he/she could remotely view the scenario displayed on the Scenario Presentation System, and the Test System screen. This made it easier for the facilitator to coordinate the experiment and record error counts.

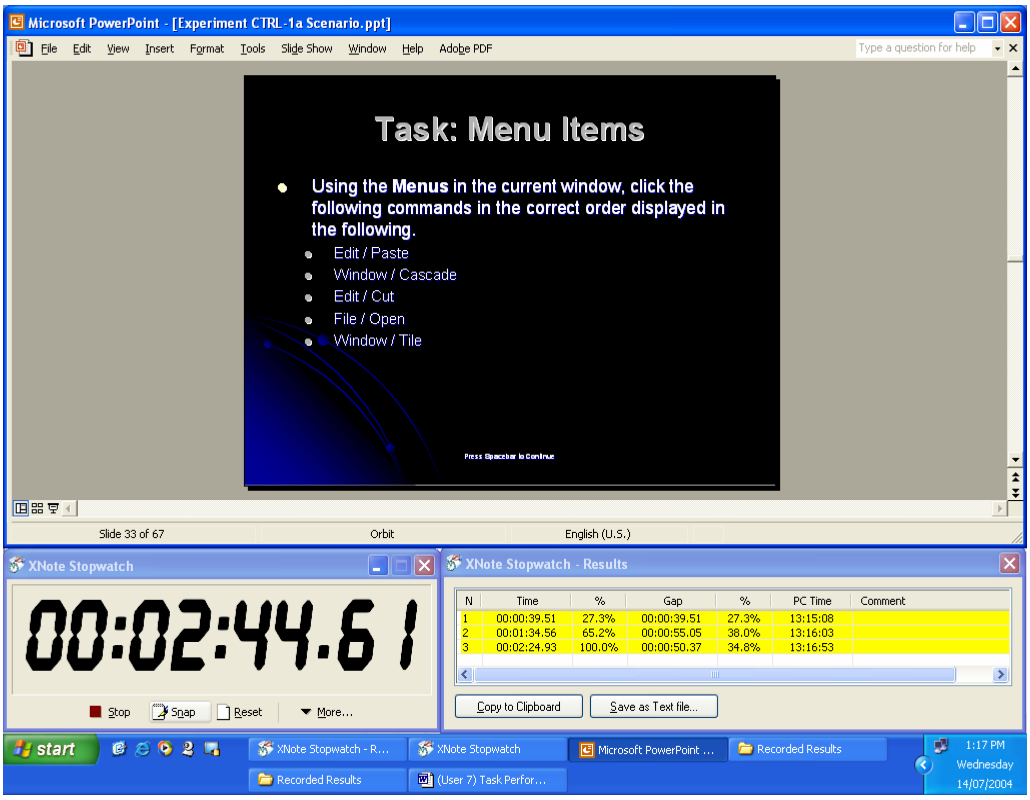


Figure A-4: Task Completion Time Logging Screen

Re-engineering the usability test facility resulted in developing an improved usability testing process.

A.3 Improving the Usability Testing Process

A.3.1 The Traditional Approach

The *traditional usability testing process* relied on using hardcopy:

- *Scenarios and tasks* – Multiple copies were required to be made to cater for each participating member in the experiment.

- *Feedback Questionnaires* – Experiment questionnaires were used to gather feedback from participants. Feedback was then transcribed manually into Microsoft Excel.

- *Task Completion Time* – The observer recorded Task Completion Times using a stop watch and then noted the time down on paper. Task Completion Times were then manually transcribed into Microsoft Excel.

- *Error Count* – The facilitator recorded Error Counts on paper for each task. This was then manually transcribed into Microsoft Excel.

As all data was recorded on hard copy, it proved to be very time consuming to collate and transcribe it later in Microsoft Excel for analysis.

A.3.2 The Re-engineered Approach

The *re-engineered usability testing process* relied on electronically collecting and managing experiment data.

- *Scenarios and Tasks* – Microsoft PowerPoint was used to present the test scenarios and test tasks on the Scenario Presentation System and the Monitoring System. This made it possible for the participant, facilitator and observer to have a synchronised view of the scenarios and tasks.

- *Feedback Questionnaires* – Experiment questionnaires were developed in Microsoft Excel and loaded onto the Data Collection System. Participants could directly input their feedback into these spreadsheets.

- *Task Completion Time* – A software 'stop-watch timer' was used on the Monitoring Computer. The observer was required to start and stop the timer for each task. Task Completion Times were exported from the stop-watch timer software to a Microsoft Excel Spreadsheet.

- *Error Count* – The facilitator recorded Error Counts on hard copy, which were then transcribed into Microsoft Excel. To enhance this process, using a Tablet PC would enable to record this data directly into electronic form.

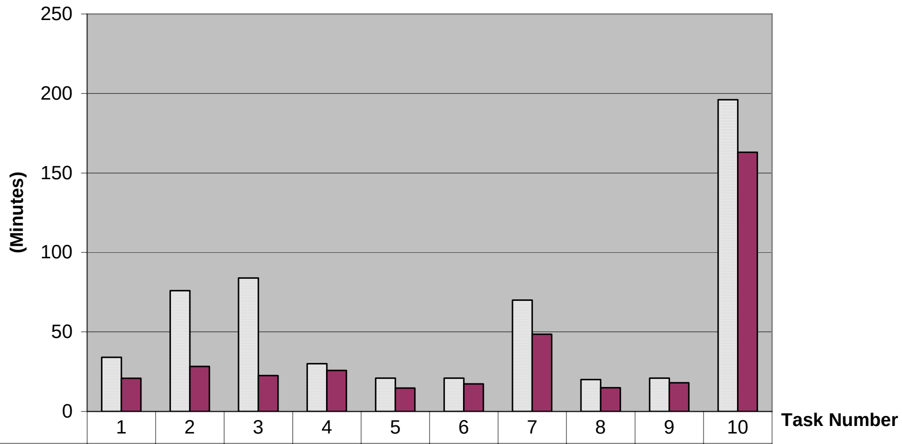
The re-engineered usability testing process made it efficient to collect, manage and analyse data. Usability experiments were completed much faster as this process was more systematic.

A.3.3 Comparative Analysis

To verify whether the re-engineered usability testing process improved efficiency, a usability test was conducted using both methods. This test comprised a scenario of ten tasks, where each task required the participant to use a joystick to interact with various user interface elements such as, buttons, sliders, and menu items etc. The participant then filled out a post-experiment feedback questionnaire. An 'error' was considered as an action that the participant performed, which was not given in the scenario of tasks. To obtain a well-founded comparison, the same participant (a third year student currently studying a Computer Science Bachelor degree at Victoria University with good technical skills) performed the experiment using the traditional and re-engineered usability testing processes.

Figure A-5 shows that, overall, the participant took much longer to perform each task using the *traditional usability testing process*. It took 16 minutes and 53 seconds to complete the entire experiment using the traditional approach, and only 10 minutes and 54 seconds using the *re-engineered process*.

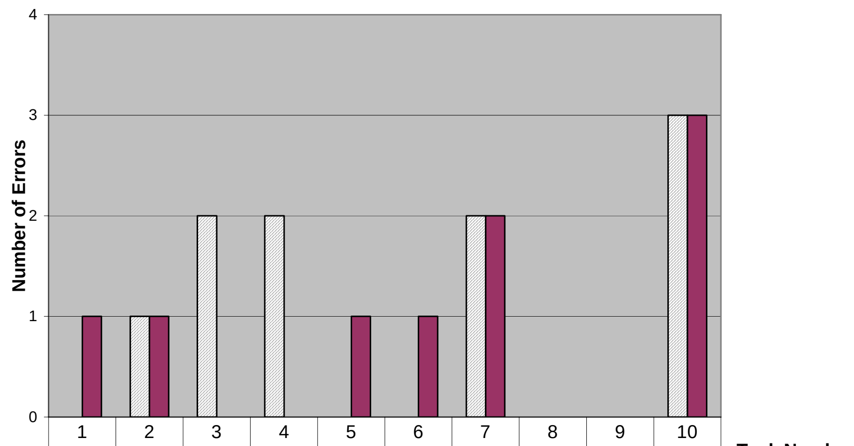
Figure A-6 shows that there were a total of 10 errors made using the *traditional usability testing approach*. The *re-engineered process* recorded 9 errors. This indicates that the new process is not affecting the test as it had no affect on the number of errors the participant made.



	1	2	3	4	5	6	7	8	9	10
Traditional Approach	34	76	84	30	21	21	70	20	21	196
Re-engineered Approach	20.9	28.28	22.63	25.84	14.75	17.42	48.57	14.95	18.06	163.05

Figure A-5: Comparison of the Task Completion Times,

For the Traditional and Re-engineered Usability Testing Approach



	1	2	3	4	5	6	7	8	9	10
Traditional Approach Error Count	0	1	2	2	0	0	2	0	0	3
Re-engineered Approach Error Count	1	1	0	0	1	1	2	0	0	3

Figure A-6: Comparison of the Task Error Count

For the Traditional and Re-engineered Usability Testing Approach

This study has revealed that customising the usability test facility and processes to suit the type of application has improved the efficiency of performing the usability study.

Appendix B
Usability Test Scenario Example

This appendix presents an example of a test scenario used in this research project. Test scenarios were created in Microsoft PowerPoint and presented to participants on a computer screen.

The test scenarios created for all other usability studies were based on a similar structure and visual presentation, to the one presented in this appendix.

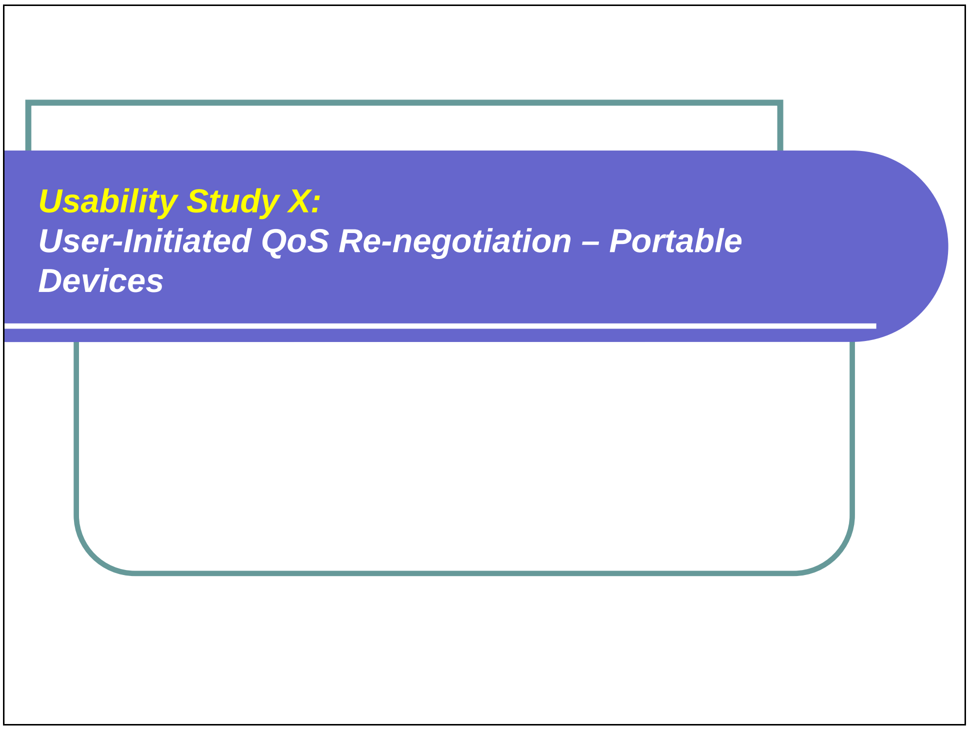


Figure B-1: Title of Usability Study

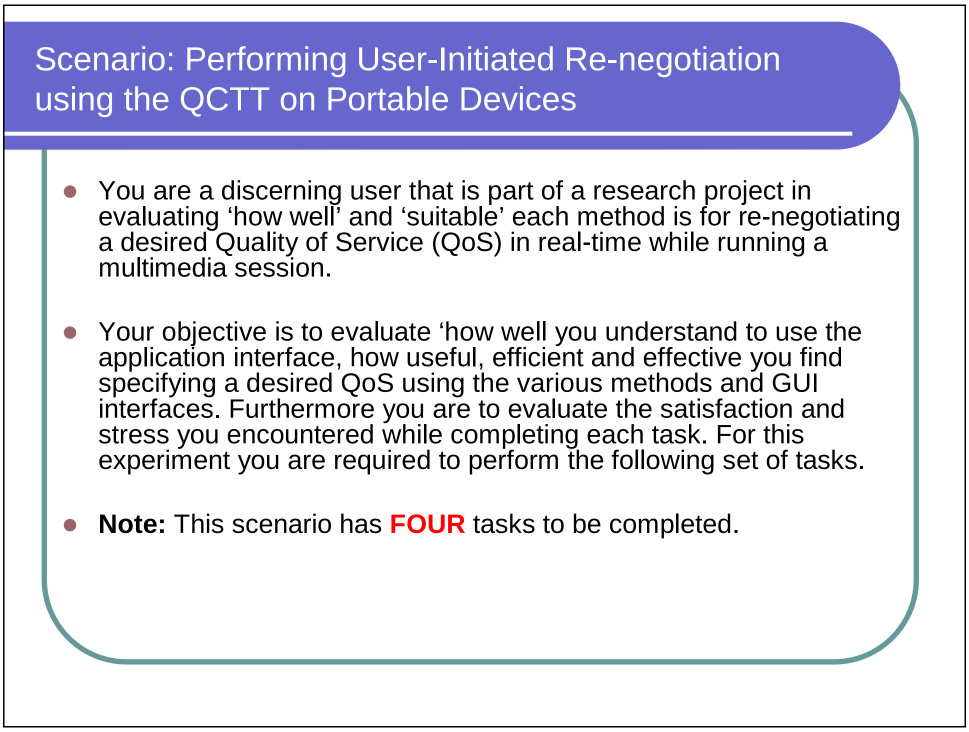


Figure B-2: Scenario Overview

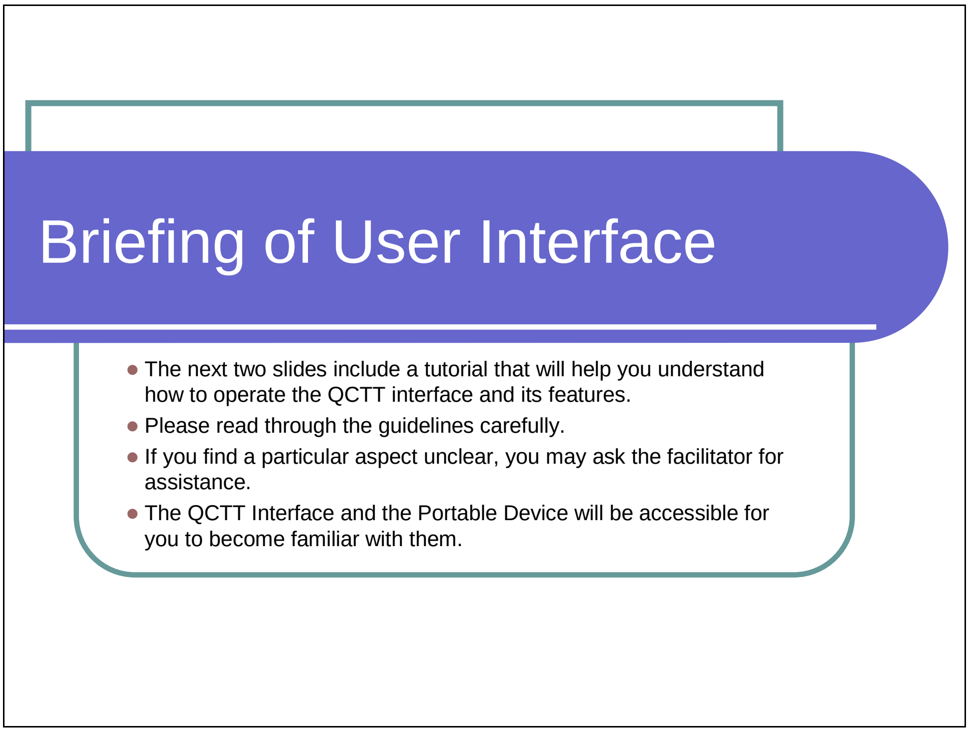


Figure B-3: Experiment Briefing Slide 1

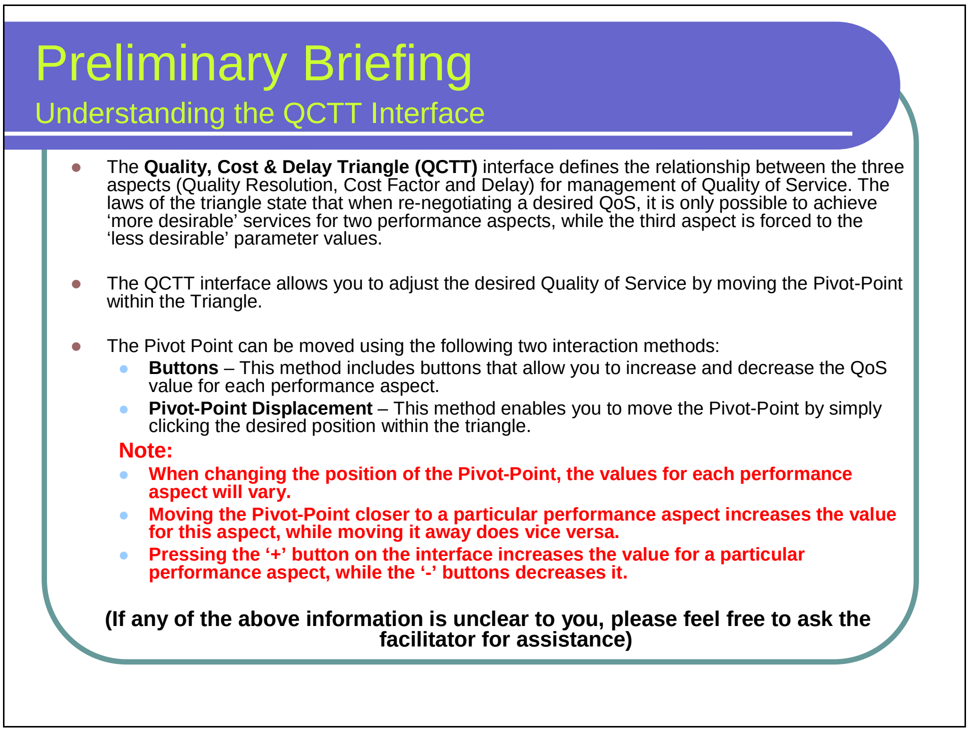


Figure B-4: Experiment Briefing Slide 2

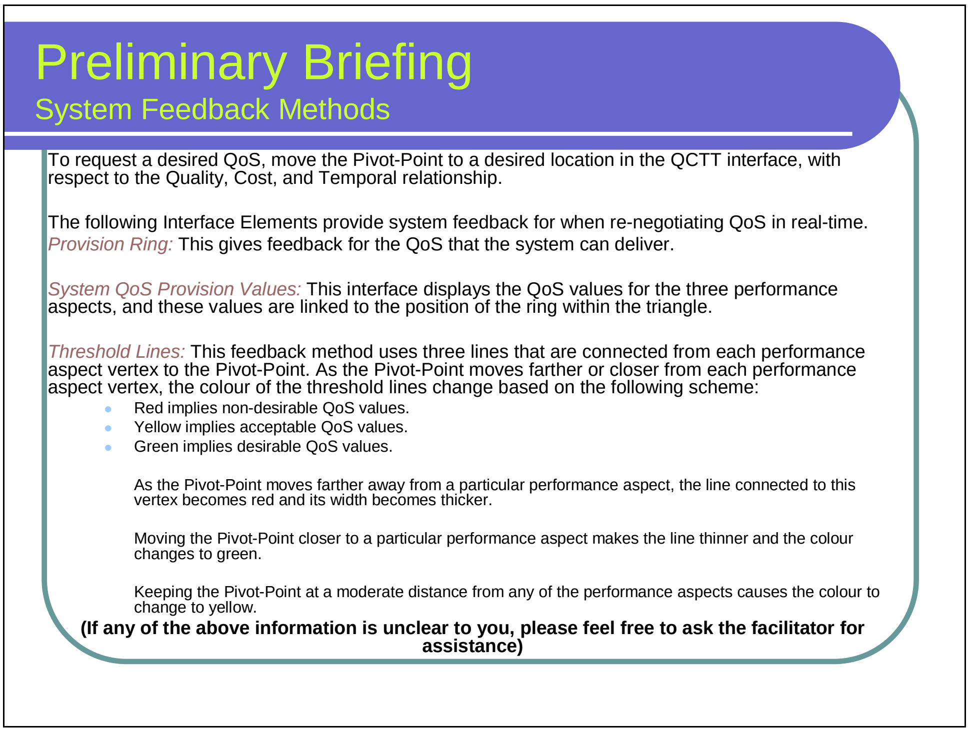


Figure B-5: Experiment Briefing Slide 3

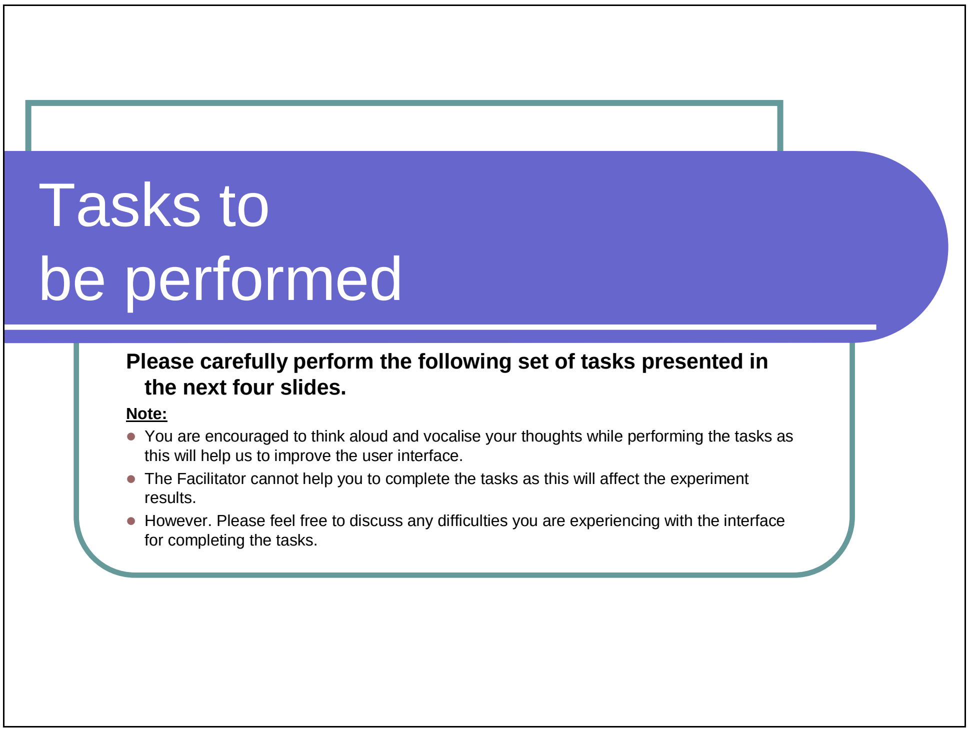


Figure B-6: Instructions Prior to Commencing Experiment

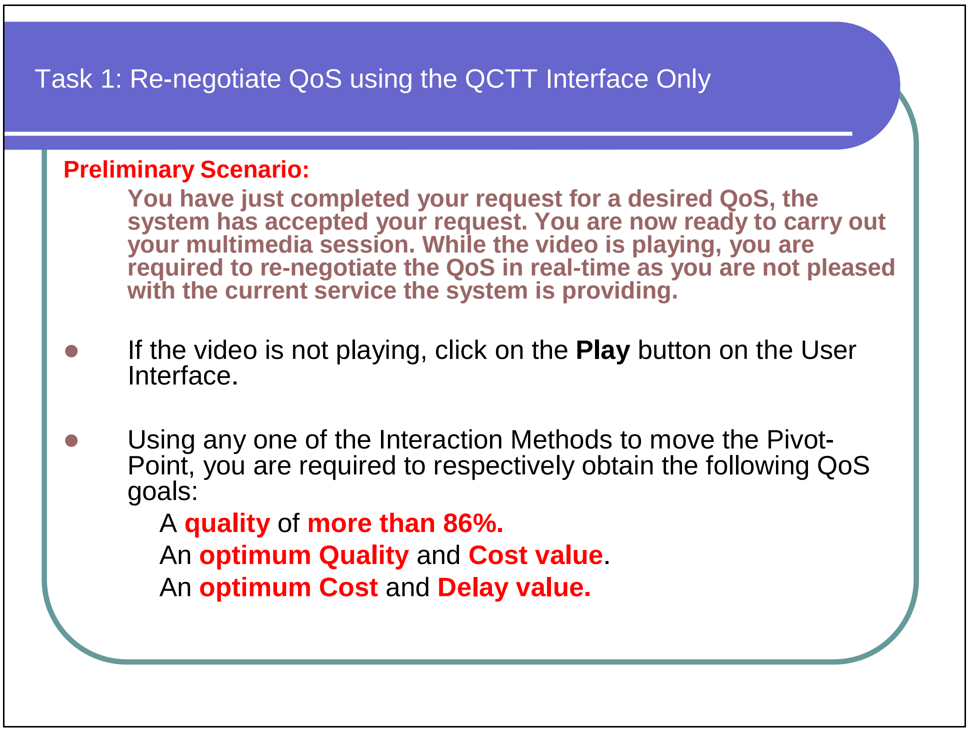


Figure B-7: Task 1

Task 2: Re-negotiate QoS while considering feedback given by the Threshold Lines

Preliminary Scenario:

You have just completed your request for a desired QoS, the system has accepted your request. You are now ready to carry out your multimedia session. While the video is playing, you are required to re-negotiate the QoS in real-time as you are not pleased with the current service the system is providing.

- If the video is not playing, click on the **Play** button on the User Interface.
- Using any one of the Interaction Methods to move the Pivot-Point, you are required to respectively achieve the following QoS goals while taking consideration of the system feedback given by the Threshold Lines:
- An **optimum Quality** and **Delay** value while **not allowing** the **Cost QCDT Threshold line** to exceed into the **red zone.**
- An **optimum Cost** and **Delay** value while **not allowing** the **Quality QCDT Threshold line** to exceed into the **red zone.**
- An **optimum Delay** value while **not allowing** the **Cost** and **Quality QCDT Threshold line** to exceed into the **red zone.**

Figure B-8: Task 2

Task 3: Re-negotiate QoS while considering feedback given by the Provision

Preliminary Scenario:

You have just completed your request for a desired QoS, the system has accepted your request. You are now ready to carry out your multimedia session. While the video is playing, you are required to re-negotiate the QoS in real-time as you are not pleased with the current service the system is providing.

- If the video is not playing, click on the **Play** button on the User Interface.
- Using any one of the Interaction Methods to move the Pivot-Point, you are required to respectively achieve the following QoS goals while taking consideration of the system feedback given by the Provision Ring:
- A **best Quality** value that is **within range** of **what the system can provide.**
- A **best Quality** and **Delay** value that is **within range** of **what the system can provide.**
- A **best Cost** and **Delay** value that is **within range** of **what the system can provide.**

Figure B-9: Task 3

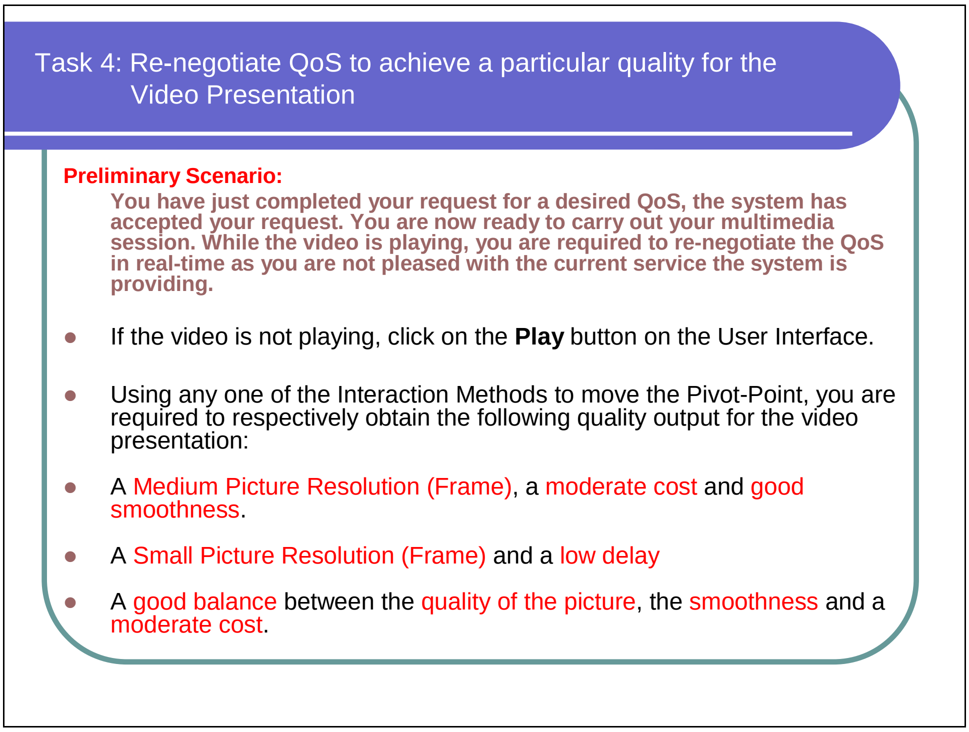


Figure B-10: Task 4

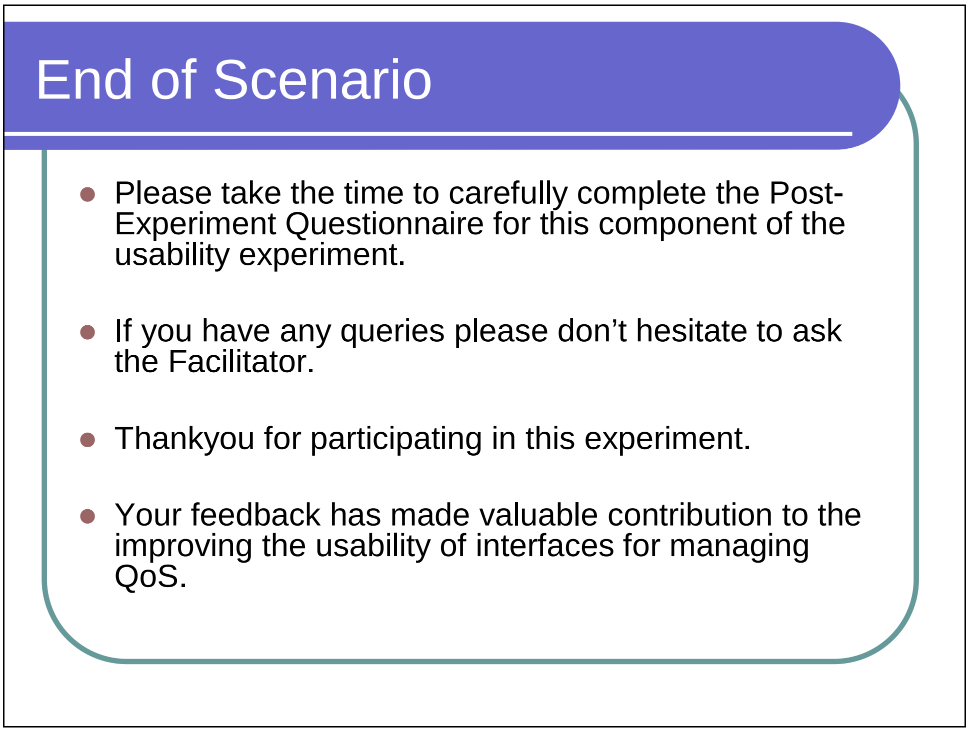


Figure B-11: End of Scenario

Appendix C
Pre-Experiment Questionnaire Example

This appendix presents an example of the Pre-experiment Questionnaire used in the usability studies to obtain participant background information. The Pre-Experiment Questionnaire was created in Microsoft Excel, where participants entered their feedback directly into the spreadsheet.

The Pre-Experiment Questionnaires produced for other usability studies were based on a similar structure and visual presentation to the one presented in this appendix.

Pre-Experiment Questionnaire

Please read and answer all questions.

Please Note: No record of participants' names will be kept for this questionnaire

	User Identification No:		
		User Response	**Comments**
1. Age *(Tick one that applies)*	Between: **a.** 18 & 21 **b.** 22 & 25 **c.** 26 & 30 **d.** 31 & 35 **e.** 36 & 40 **f.** 41 & 45 **g.** 46 & 50 **h.** 51 & 55 **i.** 56 & 60 **j.** 61 & 65 **k.** 66 & 70		
2. Occupation	**Type here:**		
3. Work Status *(Tick one that applies)*	**a.** Full Time **b.** Part Time **c.** Casual **d.** Unemployed **e.** Other (please specify)		
4. Formal Education (Tick as many as apply)	**a.** Secondary School **b.** Diploma **c.** University Degree **d.** Post Graduate University Qualification **e.** Other (please specify)		

5. Do you own a computer at home?	Yes = 1 No = 0		
(Tick as many as apply)	If Yes, what Operating System do you use? **a.** Microsoft Windows XP/2000/NT/ME/98/95 **b.** Apple Macintosh (MacOS) **c.** Linux **d.** Unix **e.** OS/2 **f.** Other (please specify)		
6. Do you have an Internet Connection for your home computer?	Yes = 1 No = 0		
7. Computer Usage *(Tick one statement that applies)*	**a.** I have **never** used a computer before. **b.** I have used a computer a **few times before**. **c.** I use a computer a **few times a month**. **d.** I use a computer **every week**. **e.** I use a computer **almost every day**. **f.** I use a computer **on a daily basis**.		
8. Level of Experience *(Tick statements that applies)*	**a.** I use computers for **basic** use. **b.** I use computers for **advanced** use. **c.** I use computer to play **games.**		
9. Level of Knowledge in Networked Multimedia Systems *(Tick one statement that applies)*	**a.** I have **no** understanding of Networked Multimedia Systems. **b.** I have **little** understanding of Networked Multimedia Systems. **c.** I have **basic** understanding of Networked Multimedia Systems. **d.** I have **excellent** understanding of Networked Multimedia Systems. **e.** I am an **expert** in Networked Multimedia Systems.		
10. Level World Wide Web Experience *(Tick one statement that applies)*	**a.** I have **never** used the Web before. **b.** I have used the Web a **few times before**. **c.** I use the Web a **few times a month**. **d.** I use the Web **every week**. **e.** I use the Web **almost every day**. **f.** I use the Web **on a daily basis**.		

11. Video Conferencing Experience *(Tick one statement that applies)*	**a.** I have **never** used video conferencing before. **b.** I have used video conferencing a **few times before**. **c.** I use video conferencing a **few times a month**. **d.** I use video conferencing **every week**. **e.** I use video conferencing **almost every day**.		
12. What application do you use for carrying out Video Conferencing Sessions? *(Tick as many as apply)*	**a.** Microsoft NetMeeting **b.** MSN Messenger **c.** ICQ **d.** OnlineCall **e.** ThruCam **f.** Other (please specify)		
13. How often do/have you play(ed) games on a computer or any game console? *(Tick one statement that applies)*	**a.** I have **never** played any games on a computer or game console. **b.** I have played games on a computer or game console a **few times before**. **c.** I play games on a computer or a game console a **few times a month**. **d.** I play games on a computer or a game console **every week**. **e.** I play games on a computer or a game console **almost every day**. **f.** I play games on a computer or a game console **on a daily basis.**		
14. Which controller device do you prefer to use when playing games? *(List in order of preference)* *1 = Most Preferred.* *5 = Least Preferred*	**a.** Keyboard & Mouse **b.** Joystick **c.** Game Pad Controller **d.** Steering Wheel with Foot Pedals **e.** Pen & Touchpad		
15. Which controller/input device do you prefer to use when using the computer? *(List in order of preference)* *1 = Most Preferred.* *5 = Least Preferred*	**a.** Keyboard & Mouse **b.** Joystick **c.** Game Pad Controller **d.** Steering Wheel with Foot Pedals **e.** Pen & Touchpad		

16. Are you left handed or right handed?	**a.** Left **b.** Right		
17. Have you been involved in any Usability Testing prior to this research?	Yes = 1 No = 0		

Appendix D
Post-Experiment Questionnaire Example

This appendix presents an example of a Post-experiment Questionnaire used in this research project for obtaining feedback from participants. Post-Experiment Questionnaires were created in Microsoft Excel, where participants entered their feedback directly into the spreadsheet.

Different Post-Experiment Questionnaires were created for specific usability studies, which used a similar structure and visual presentation to the Post-Experiment Questionnaire presented in this appendix.

Usability Study X: Post Experiment Questionnaire (SAMPLE)

This questionnaire is designed to tell us how you feel about the software you used in the experiment today. Please read and answer all questions.

User Identification No:

1. Ease of Use

a. Please rate the **ease of using** each interaction method to re-negotiate QoS.	Very Easy	Easy	Neutral	Difficult	Very Difficult	**User Rating**
QCTT Only	2	1	0	-1	-2	
Threshold Lines	2	1	0	-1	-2	
Provision Ring	2	1	0	-1	-2	
QCTT plus Video Feedback	2	1	0	-1	-2	

2. Ease of Learning

a. Please rate the **learnability** of using each interaction method to re-negotiate QoS.

	Very Easy	*Easy*	*Neutral*	*Difficult*	*Very Difficult*	**User Rating**
QCTT Only	2	1	0	-1	-2	
Threshold Lines	2	1	0	-1	-2	
Provision Ring	2	1	0	-1	-2	
QCTT plus Video Feedback	2	1	0	-1	-2	

b. Was it obvious to you on **how to interact** and work with this interface for re-negotiating QoS?

Yes = 1	No = 0	**User Response**
1	0	

3. Usefulness, Effectiveness & Efficiency

(Before proceeding with the questionnaire, please read the following)

Understanding the distinction between useful, effective and efficient:

An item is useful for a task if it can accomplish the same; even if it is not effective or efficient in doing it. For example, a pencil is useful in writing on paper, but not for writing on a whiteboard. However, a pen is more effective for writing on paper, as it produces more striking effect than the grey lead pencil.

Something is more efficient if it takes less effort or time in accomplishing the task. You may be able to write on a whiteboard with a fine tip felt pen, however, the whiteboard marker is not only more effective but also more efficient. Because, with a fine tip pen you will have to overdraw many times to produce a thick line, where as, with a marker it will be done with one stroke.

Thus:

1. *A pencil is useful and efficient for writing on paper, but not effective. It is useless for writing on a whiteboard.*

2. *A fine tip felt pen is useful, effective and efficient for writing on paper. It is useful for writing on a whiteboard, but not effective or efficient.*

3. *A marker is useful, effective and efficient for writing on a white board.*

a. Please rate the **usefulness** of using each interaction method to re-negotiate QoS	*Completely Useless*	*Not Quite Useful*	*Neutral*	*Useful*	*Very Useful*	**User Rating**
QCTT Only	-2	-1	0	1	2	
Threshold Lines	-2	-1	0	1	2	
Provision Ring	-2	-1	0	1	2	
QCTT plus Video Feedback	-2	-1	0	1	2	

b. Please rate the **effectiveness** of using each interaction method to re-negotiate QoS	*Completely Useless*	*Not Quite Useful*	*Neutral*	*Useful*	*Very Useful*	**User Rating**
QCTT Only	-2	-1	0	1	2	
Threshold Lines	-2	-1	0	1	2	
Provision Ring	-2	-1	0	1	2	
QCTT plus Video Feedback	-2	-1	0	1	2	

a. Please rate the **efficiency** of using each interaction method to re-negotiate QoS	*Completely Useless*	*Not Quite Useful*	*Neutral*	*Useful*	*Very Useful*	**User Rating**
QCTT Only	-2	-1	0	1	2	
Threshold Lines	-2	-1	0	1	2	
Provision Ring	-2	-1	0	1	2	
QCTT plus Video Feedback	-2	-1	0	1	2	

4. Satisfaction

a. Upon completing QoS re-negotiation, please rate how **satisfied** you were for using each interaction method.	Very Poor	Poor	*Average*	*Good*	Excellent	**User Rating**
QCTT Only	-2	-1	0	1	2	
Threshold Lines	-2	-1	0	1	2	
Provision Ring	-2	-1	0	1	2	
QCTT plus Video Feedback	-2	-1	0	1	2	

5. Stress Factor

a. Please rate the **stress** caused by using each interaction method to re-negotiate QoS.

	Relaxed	*Not Stressful*	*Neutral*	*Stressful*	*Very Stressful*	**User Rating**
QCTT Only	2	1	0	-1	-2	
Threshold Lines	2	1	0	-1	-2	
Provision Ring	2	1	0	-1	-2	
QCTT plus Video Feedback	2	1	0	-1	-2	

6. Ease of Perception

a. Please rate how easy it was for you to **recognise** the system feedback for each interaction method.

	Very Easy	*Easy*	*Neutral*	*Difficult*	*Very Difficult*	**User Rating**
QCTT Only	2	1	0	-1	-2	
Threshold Lines	2	1	0	-1	-2	
Provision Ring	2	1	0	-1	-2	
QCTT plus Video Feedback	2	1	0	-1	-2	

7. Ease of Comprehension

a. Please rate how **easy** it was for you **to understand** the meaning of each system feedback alert.

	Very Easy	Easy	Neutral	Difficult	Very Difficult	**User Rating**
QCTT Only	2	1	0	-1	-2	
Threshold Lines	2	1	0	-1	-2	
Provision Ring	2	1	0	-1	-2	
QCTT plus Video Feedback	2	1	0	-1	-2	

8. Additional Comments and/or Suggestions

Please indicate in the following any suggestions or comments about this experiment.

Appendix E
QCTT Implementation using the CCS & TFS Methods

This appendix presents the theory for implementing the QCTT model using the two methods: Cartesian Coordinate System (CCS) and Triangular Fractal System (TFS). This appendix expands on the user interface design presented in Usability Study E (section 6.6).

This theory has been published by Georgievski and Sharda, and presented in the following conference:

- M. Georgievski and N. Sharda, "Implementation and Usability of User Interfaces for Quality of Service Management," presented at Tencon'05, Annual Technical Conference of IEEE Region 10, Melbourne, Australia, 2005.

E.1 QCTT Graphical User Interface Function

According to the QCTT model, the dependency between the three performance aspects is based on the following assertions; where High = H, Low = L, Cost Factor = CF, Temporal Facet = TF, Quality Resolution = QR.

$$HQR \;\&\; LTF \rightarrow HCF \qquad \text{(E-1)}$$

$$HQR \;\&\; LCF \rightarrow HTF \qquad \text{(E-2)}$$

$$LCF \;\&\; LTF \rightarrow LQR \qquad \text{(E-3)}$$

For predicate E-1, a high QR and low TF are more desirable, and a high CF is less desirable. For predicate E-2, a low CF and low TF are more desirable, and low QR is less desirable.

For the GUI implementation of the QCTT model, each performance aspect was assigned a QoS value that varied from 0% to 100%. This made it possible to make adjustments for the QoS value between the three performance aspects. This translates to the following predicates:

$$HQR = 100\% \ \& \ LTF = 0\% \rightarrow HCF = 100\% \qquad \text{(E-4)}$$

$$HQR = 100\% \ \& \ LCF = 0\% \rightarrow HTF = 100\% \qquad \text{(E-5)}$$

$$LCF = 0\% \ \& \ LTF = 0\% \rightarrow LQR = 0\% \qquad \text{(E-6)}$$

Using the GUI for the QCTT, to specify desirable QoS values for two performance aspects, the following predicates have been defined.

$$QR = 50\% \ \& \ CF = 50\% \rightarrow TF = 100\% \qquad \text{(E-7)}$$

$$QR = 50\% \ \& \ TF = 50\% \rightarrow CF = 100\% \qquad \text{(E-8)}$$

$$CF = 50\% \ \& \ TF = 50\% \rightarrow QR = 0\% \qquad \text{(E-9)}$$

Predicate E-7 implies that when QR is 50%, and CF is 50%, TF is forced to 100%. Predicate E-9 implies that when CF is 50% and TF is 50%, QR is forced to 0%. Ideally, the user would desire that QR is 100%, CF is 0% and TF is 0%, which in reality is not possible. The user can use the QCTT GUI to find a balance between the three performance aspects. In the user interface, QR is inverted, such that moving the Pivot-Point closer to QR increases its QoS value. Moving the Pivot-Point closer to CF or TF decreases their QoS value.

The QCTT model was implemented as an equilateral triangle to satisfy the predicates E-7, E-8, and E-9.

E.2 Cartesian Coordinate System (CCS) Interface Implementation

The Cartesian Coordinate System (CCS) method involved implementing the QCTT using coordinates for the three points in the triangle. The QoS value for each performance aspect was calculated based on its distance from the Pivot-Point position.

For example, referring to Figure E-1 for the Pivot-Point, *P1*, the distance for each performance aspect is: QR = 50%, CF = 50, and TF = 86.

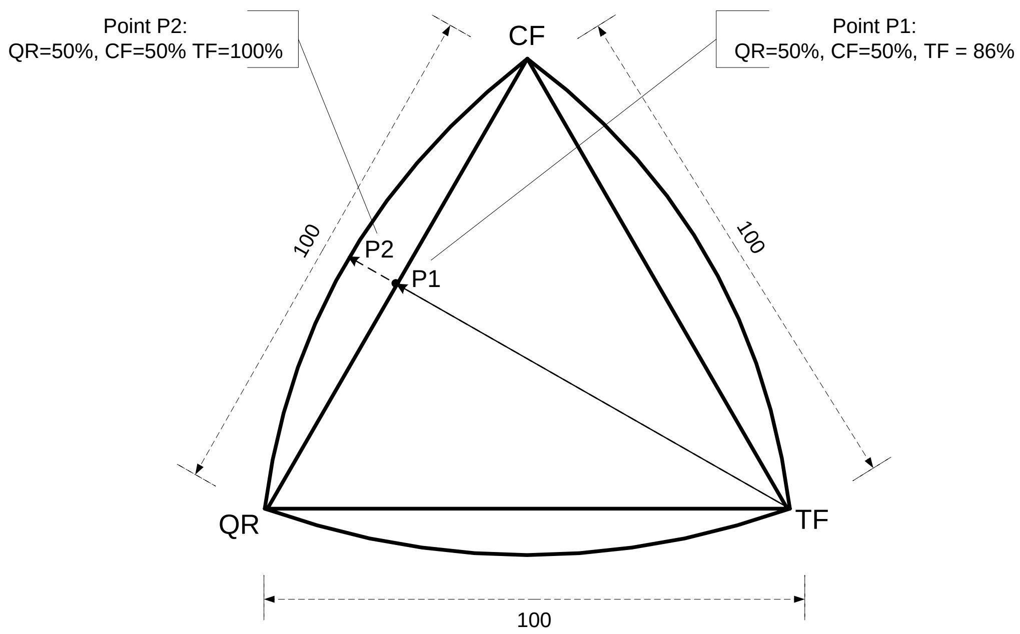


Figure E-1: Cartesian Coordinate System (CCS) Interface Implementation

Changing the value for each performance aspect also repositions the Pivot-Point within the triangle. Decreasing the value for TF moves the Pivot-Point closer to this point. Decreasing the value of CF performs the same function. This function implies reducing the cost and delays for the multimedia presentation. Increasing the value for QR moves the Pivot-Point closer to this point, which means quality is improved.

The CCS implementation deviated from the ideal conditions defined by the QCTT. For example, referring to Figure E-1, the distance from TF to the Pivot-Point P1 is:

QR = 50% & CF = 50% → TF = 86%

However, the conditions given in the QCTT model require the distance value to be:

QR = 50% & CF = 50% → TF = 100%

This limitation is due to the theoretical boundaries set by the CCS method. As a solution to this limitation, the Pivot-Point was enabled to exceed the boundary limits of the triangle by adding arcs from each polygon vertex between the other two vertices. This better modelled the predicates given for the QCTT. The visual representation of the QCTT model changed to a Reuleaux Triangle as shown in Figure E-1 and Figure E-2a.

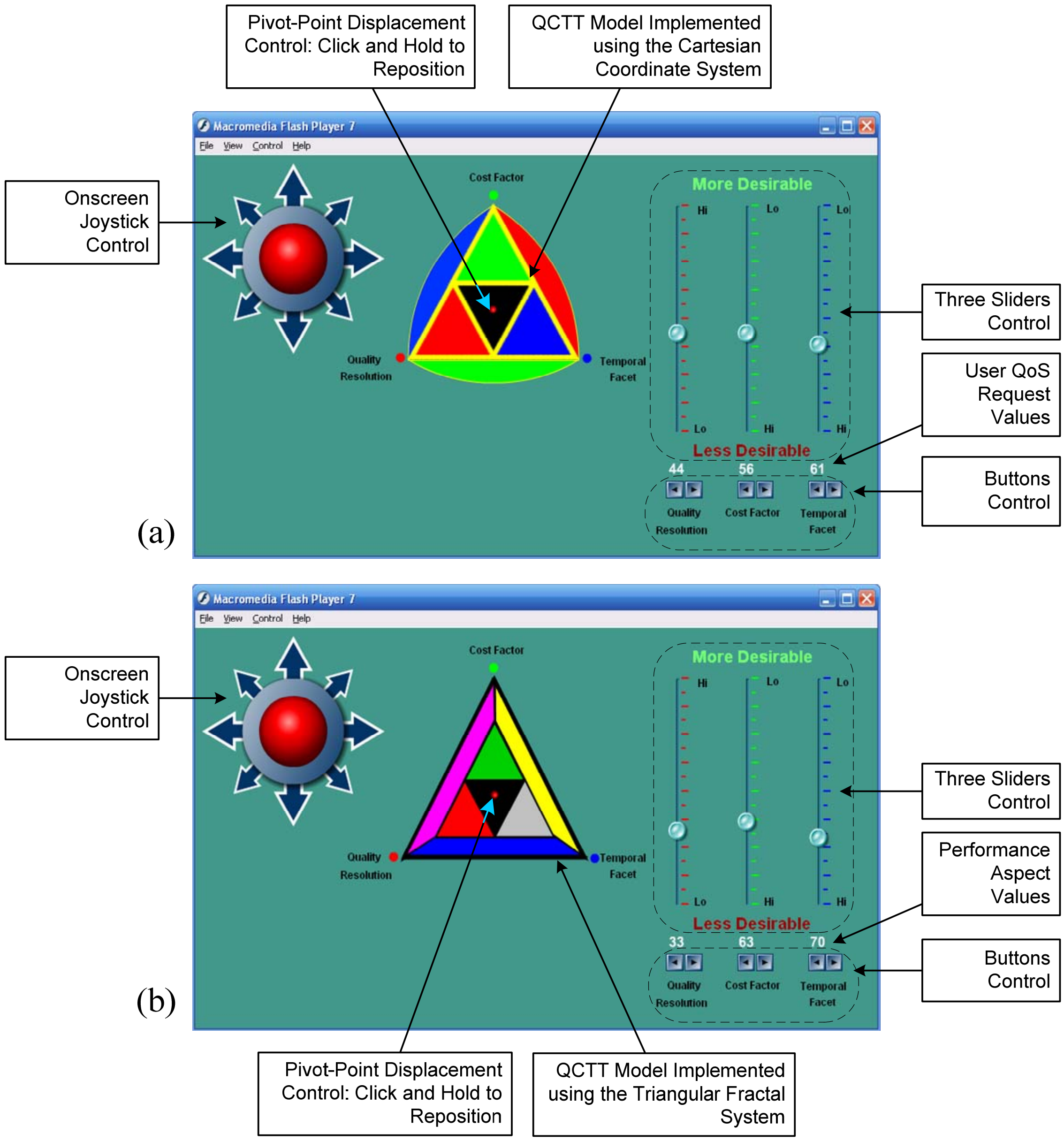


Figure E-2: QCTT Interface Implemented
Using the (a) Cartesian Coordinate System (CCS), and (b) Triangular Fractal System (TFS)

E.3 Triangular Fractal System (TFS) Interface Implementation

The Triangular Fractal System (TFS) method involved dividing the QCTT triangle into fractals. The triangle (QT) is divided into many *Triangular Fractals (TF)*. The size of the QT is defined in terms of the number of TFs *(n)* on each edge of the QT. As shown in Figure E-3: *AB = BC = CA = n.*

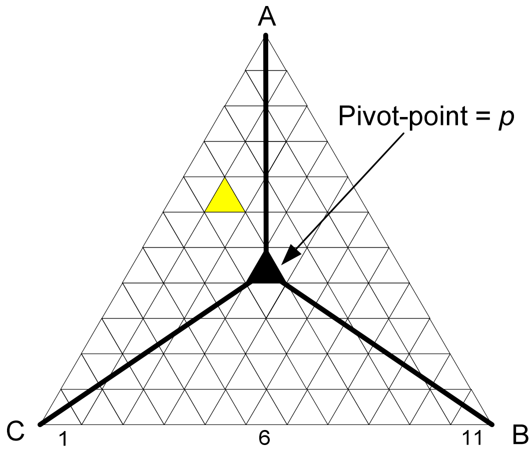


Figure E-3: Triangular Fractal System (TFS) Implementation

The Pivot-Point is represented by a small single triangle within the *Triangular Fractal Matrix (TFM)* of the main QT, and it can be positioned at any point. To determine the distance of the Pivot-Point *p* from any of the vertices (*A*, *B*, *C*) count the number of rows (r) of TFs perpendicular to the line joining the two points, excluding the row the Pivot-Point is in. For example, in Figure E-3: *AB = BC = CA = 11;* and *Ap = 6*, *Bp =7*, *Cp = 7.*

The TFS implementation gives greater control to the user in setting the position of the Pivot-Point. In the implementation of the user interface, *n* determines the fractal size of the triangle. By setting *n* to 200, it draws a reasonable size triangle where it is ergonomically easy to control the Pivot-Point in the QCTT triangle for QoS management.

The TFS method enabled to achieve predicates E-7, E-8, E-9 without the Pivot-Point exceeding the boundaries of the triangle. This also retained the original triangle representation of the QCTT model, as shown in Figure E-2b.

Appendix F
QCTT Interface Implementation for Portable Devices

This appendix presents the implementation theory for the three interfaces: (a) Large QCTT Quad Fractals, (b) Small QCTT Quad Fractals, and (c) Large 10,000 (10k) Fractals. Details are given for the application of Fractal Zones, which enhance the QCTT interface for use on Portable Devices.

This appendix expands on the user interface designs presented in Usability Study X (section 6.9).

F.1 Large QCTT Quad Fractals

The large QCTT Quad Fractals interface includes four fractal zones within the QCTT model. These fractal zones are made up of a group of fractals where each fractal zone configured parameter values for the three performance aspects; Quality Resolution, Cost Factor, and Temporal Facet. As shown in Figure F-1, each fractal zone specifies values based on a predefined parameter classes: High (H), Medium (M), and Low (L). Therefore, referring to the QCTT concept, these four fractal zones specify parameter values for each performance aspect, based on the following definitions where the four fractal zones are defined as F_1, F_2, F_3, and F_4:

$$F_1: QR = H, CF = H, TF = H \tag{F-1}$$

$$F_2: QR = L, CF = H, TF = L \tag{F-2}$$

$$F_3: QR = M, CF = M, TF = M \tag{F-3}$$

$$F_4: QR = L, CF = L, TF = H \tag{F-4}$$

To implement these fractal zones within the QCTT model, three new points were defined *d, e* and *f* that form the four fractal zones. As shown in Table F-1, the definition for the number of TFs *(n)* between each points *A, B, C, d, e, f* were defined.

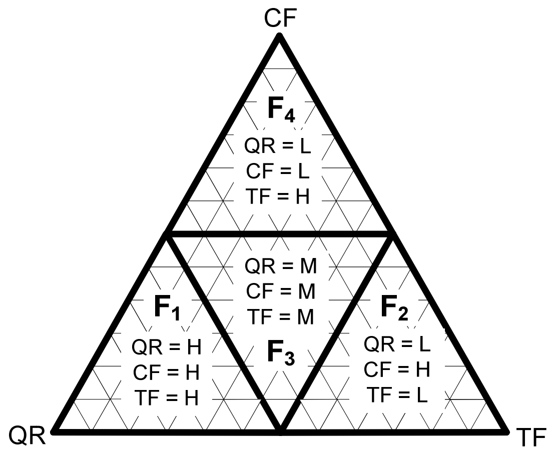


Figure F-1: QCTT Quad Fractals Performance Aspect Definition

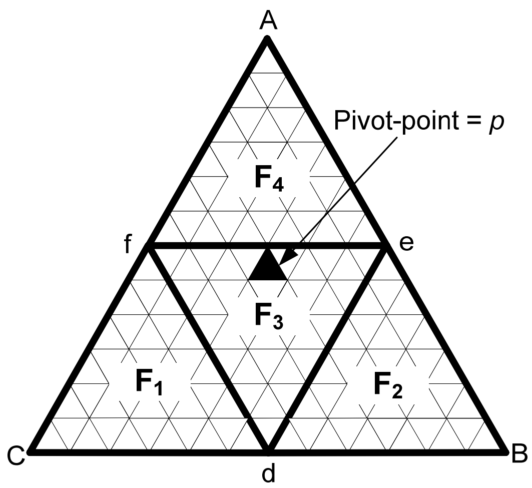


Figure F-2: TFS Quad Fractals

Table F-1: QCTT Fractal Zone Size Definition

	A	B	C
d	n	$\frac{n}{2}$	$\frac{n}{2}$
e	$\frac{n}{2}$	$\frac{n}{2}$	n
f	$\frac{n}{2}$	n	$\frac{n}{2}$

Based on Figure F-2 and the information presented in Table F-1, the size and position of each fractal zone F_1, F_2, F_3, and F_4 within the QCTT are defined as:

$$F_1 : cd = fd = cf = \frac{n}{2} \tag{F-5}$$

$$F_2 : dB = Be = de = \frac{n}{2} \tag{F-6}$$

$$F_3 : de = fe = df = \frac{n}{2} \tag{F-7}$$

$$F_4 : Ae = fe = Af = \frac{n}{2} \tag{F-8}$$

The number of fractals allocated to each fractal zone is determined by:

$$F_1 = F_2 = F_3 = F_4 = \frac{n^2}{4} \tag{F-9}$$

As the Pivot-Point is repositioned within the QCTT, it moves from one fractal zone to another. To determine which fractal zone the Pivot-Point is located in, the following conditions for each fractal zone were defined.

The Pivot-Point is located in fractal zone F_4, when the distance *(l)* from point *A* to Pivot-Point *(p)* satisfies the condition $F_4 : A_l < \frac{n}{2}$. The pivot point is located in fractal zone F_1, when the distance *(l)* from point *B* to Pivot-Point *(p)* satisfies the condition $F_1 : C_l < \frac{n}{2}$. The Pivot-Point is located in fractal zone F_2, when the distance *(l)* from *B* to pivot point *(p)* satisfies the condition $F_2 : B_l < \frac{n}{2}$. The pivot point is located in fractal zone F_3, when the distance *(l)* from *A*, *B*, and *C* to pivot point *(p)* satisfy the condition $F_3 : B_l < \frac{n}{2} \,\&\, A_l > \frac{n}{2} \,\&\, C_l > \frac{n}{2}$.

For the QCTT Interface, the size of the QCTT (triangle) *(n)* was set to 100 fractals. This produced a large size triangle for this user interface that enables to easily specify QoS on Portable Devices. The benefits of using the Large QCTT Quad Fractals is that it enhanced the accuracy of specifying QoS, as the small screen size of Portable Devices makes it difficult to precisely adjust the position of the Pivot-Point.

F.2 Small QCTT Quad Fractals

The implementation for the Small QCTT Quad Fractals interface is based on the same Large QCTT Quad Fractals interface. However, the size of the QCTT triangle *(n)* in this user interface is set to 50 fractals. The benefit of this interface is that it allows more screen space for the multimedia application, allowing better utilisation of the screen.

F.3 Large QCTT 10,000 (10K) Fractals

The Large QCTT 10,000 (10K) Fractals interface implementation is also based on the theory TFS theory. The QCTT size *(n)* is set to 100 fractals, where the number of fractals within the triangle is determined by n^2. Therefore $n^2 = 100^2 = 10{,}000$. This enabled 10,000 different parameter values for each performance aspect: QR, CF, and TF. Hence the name of this interface is derived from this concept. The benefit of this interface is it allows more precise specification of QoS due to the range of different parameter values that can be specified for each performance aspect. This concept is implemented in a large triangle (QCTT) interface as it enables a less restricted QoS specification. Implementing this concept in a small triangle (QCTT) interface would prove to be inefficacious; as it would be difficult to accurately specify QoS using the Pivot-Point Displacement method on the small screens included in Portable Devices.

Appendix G
Detailed Results for Usability Study X

This appendix includes detailed results for Usability Study X (section 6.9).

G.1 Emulated PDA

G.1.1 Large QCTT Quad Fractals Interface

Task Completion Times

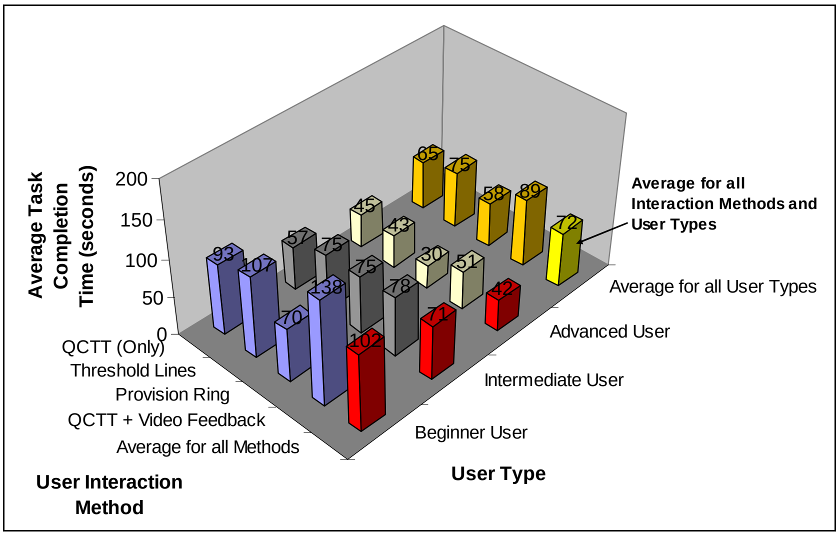


Figure G-1: Large QCTT Quad Fractals Interface used on an Emulated PDA, Task Completion Times for each User Type and Interaction Method

Task Error Count

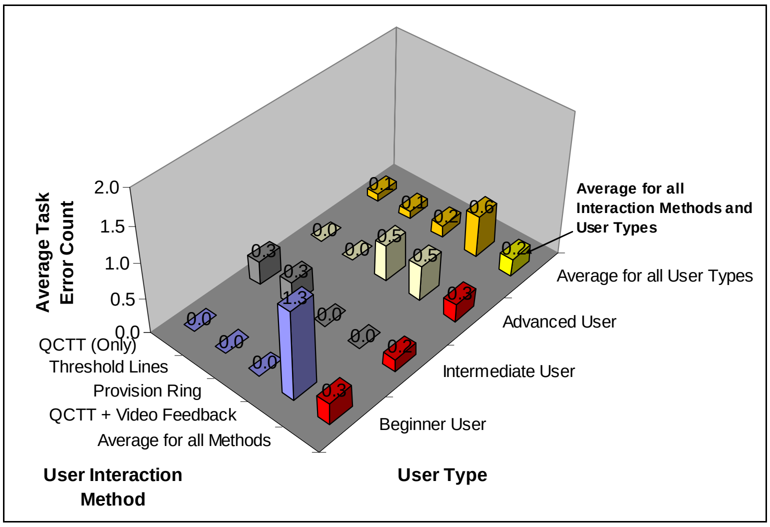


Figure G-2: Large QCTT Quad Fractals Interface used on an Emulated PDA, Task Error Count for each User Type and Interaction Method

Usability Ratings

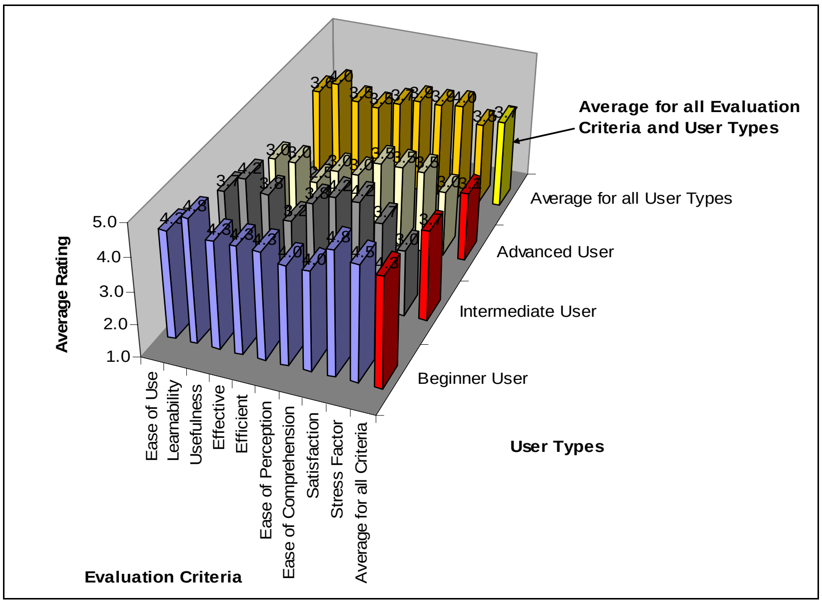


Figure G-3: Large QCTT Quad Fractals Interface used on an Emulated PDA, Usability Ratings for the *QCTT Only* Interaction Method

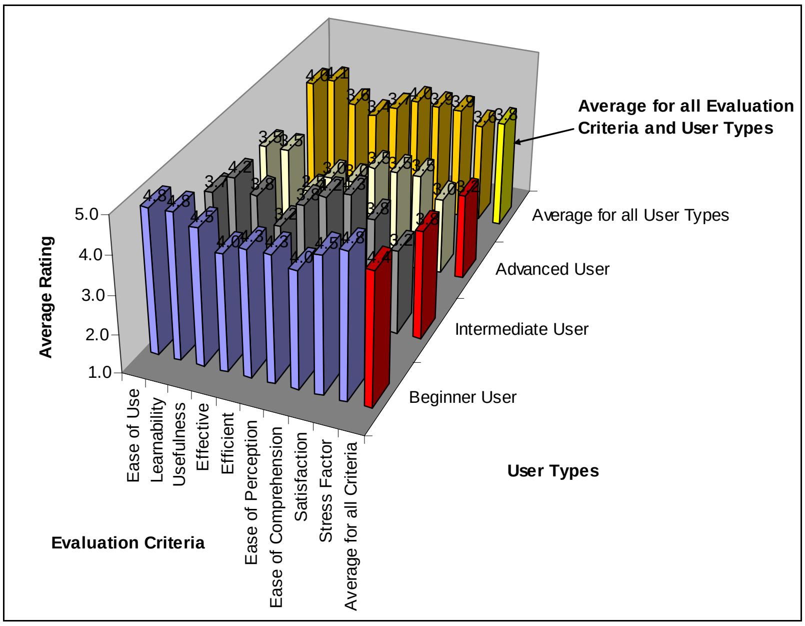


Figure G-4: Large QCTT Quad Fractals Interface used on an Emulated PDA, Usability Ratings for the *Threshold Lines* Interaction Method

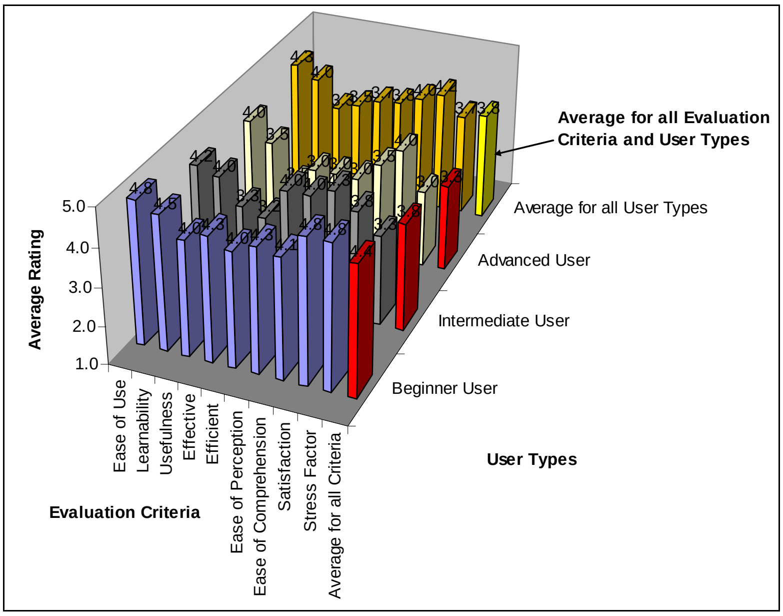


Figure G-5: Large QCTT Quad Fractals Interface used on an Emulated PDA, Usability Ratings for the *Provision Ring* Interaction Method

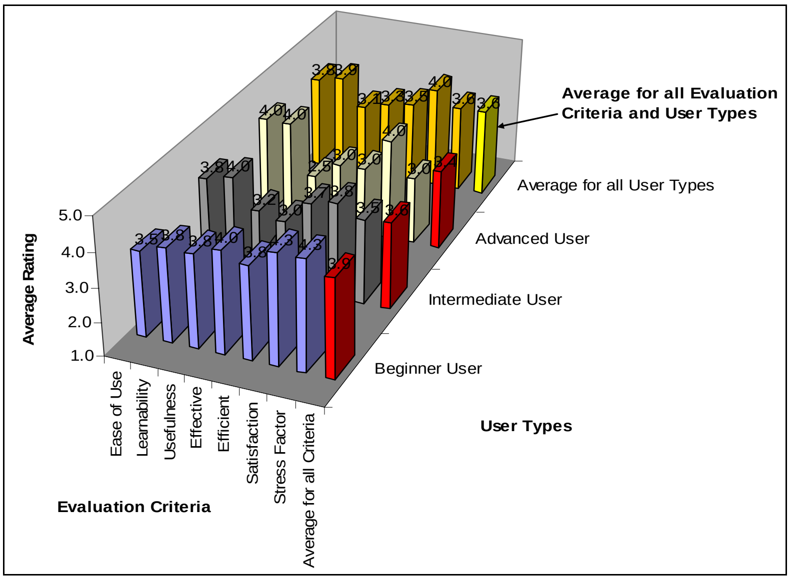


Figure G-6: Large QCTT Quad Fractals Interface used on an Emulated PDA, Usability Ratings for the *QCTT plus video feedback* Interaction Method

G.1.2 Small QCTT Quad Fractals Interface

Task Completion Times

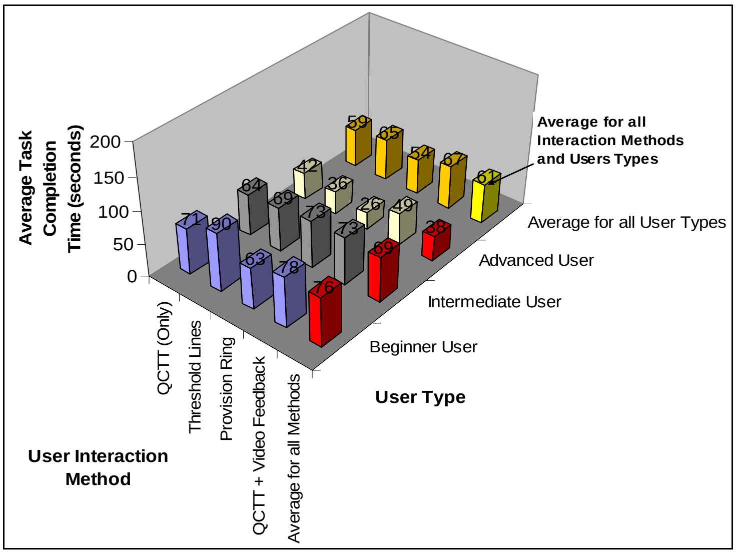


Figure G-7: Small QCTT Quad Fractals Interface used on an Emulated PDA, Task Completion Time for each User Type and Interaction Method

Task Error Count

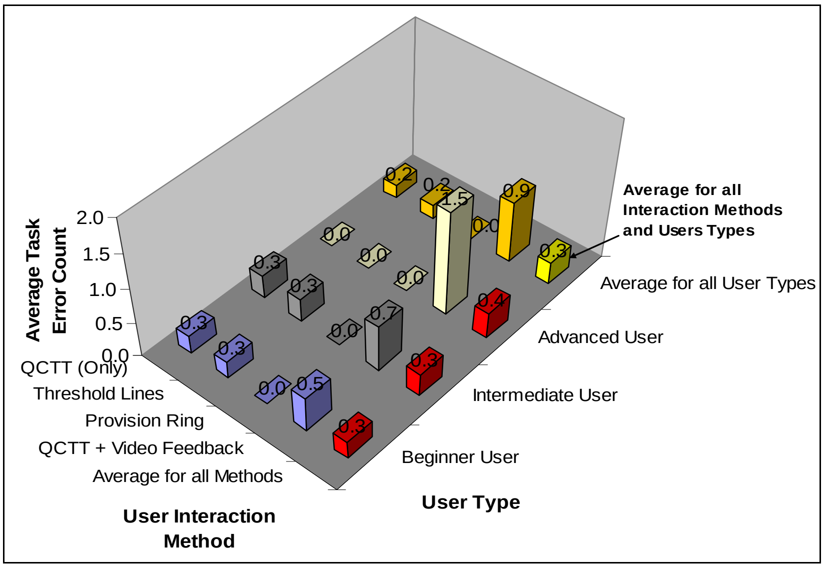


Figure G-8: Small QCTT Quad Fractals Interface used on an Emulated PDA, Task Error Count for each User Type and Interaction Method

Usability Ratings

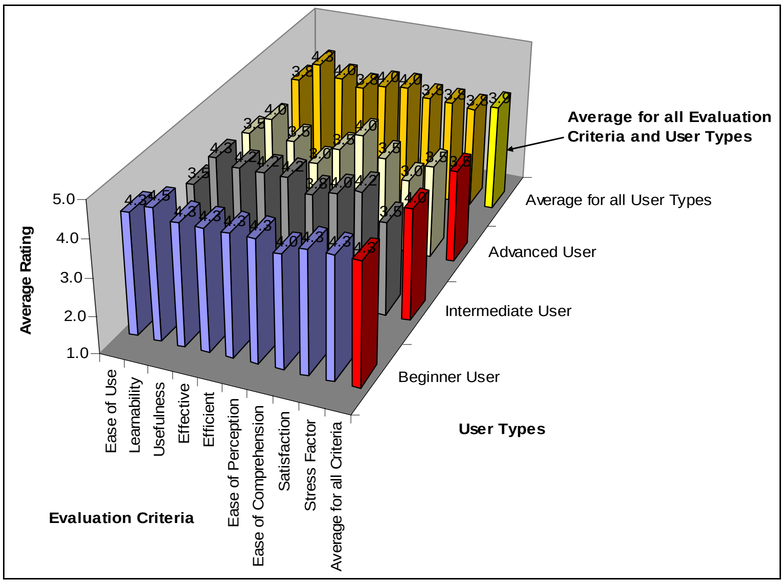


Figure G-9: Small QCTT Quad Fractals Interface used on an Emulated PDA, Usability Ratings for the *QCTT Only* Interaction Method

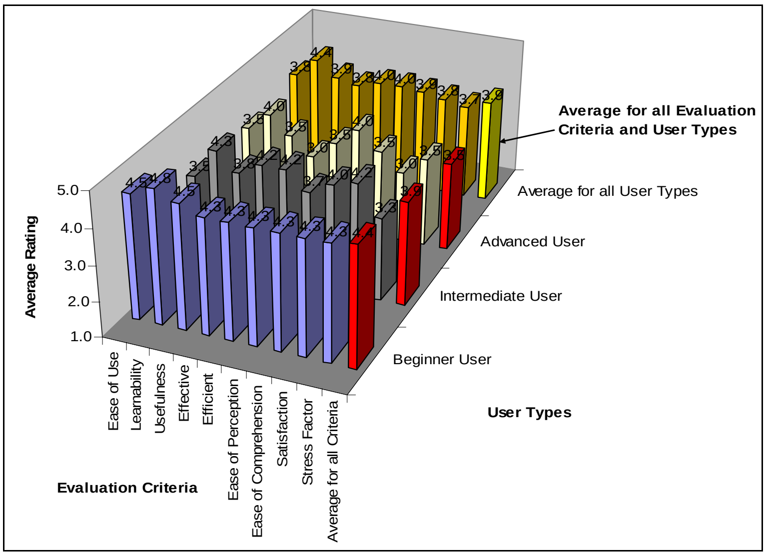


Figure G-10: Small QCTT Quad Fractals Interface used on an Emulated PDA, Usability Ratings for the *Threshold Lines* Interaction Method

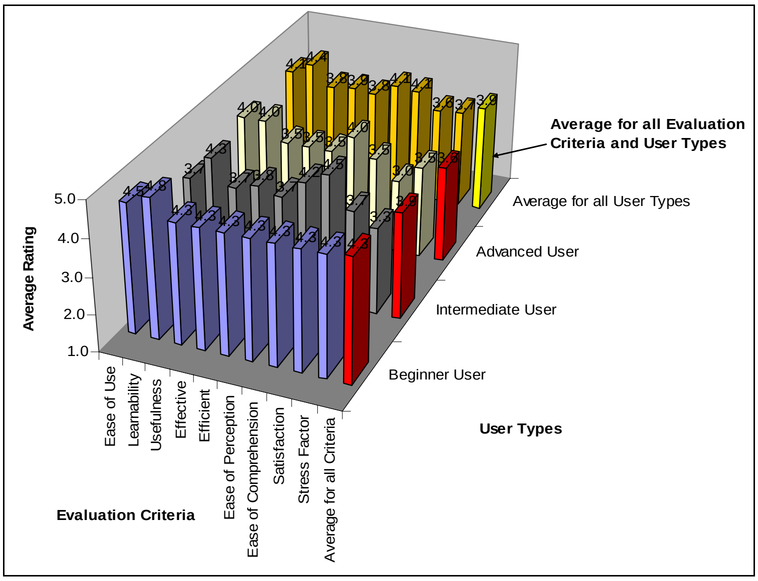


Figure G-11: Small QCTT Quad Fractals Interface used on an Emulated PDA, Usability Ratings for the *Provision Ring* Interaction Method

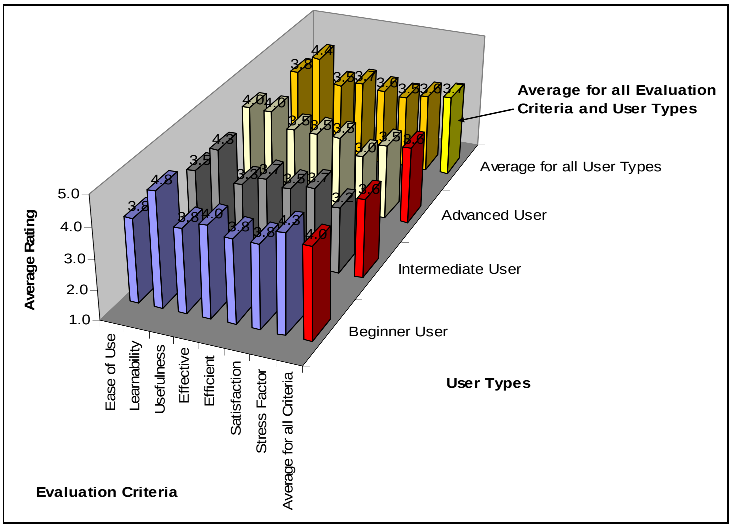


Figure G-12: Small QCTT Quad Fractals Interface used on an Emulated PDA, Usability Ratings for the *QCTT plus video feedback* Interaction Method

G.1.3 Large QCTT 10,000 (10k) Fractals Interface

Task Completion Times

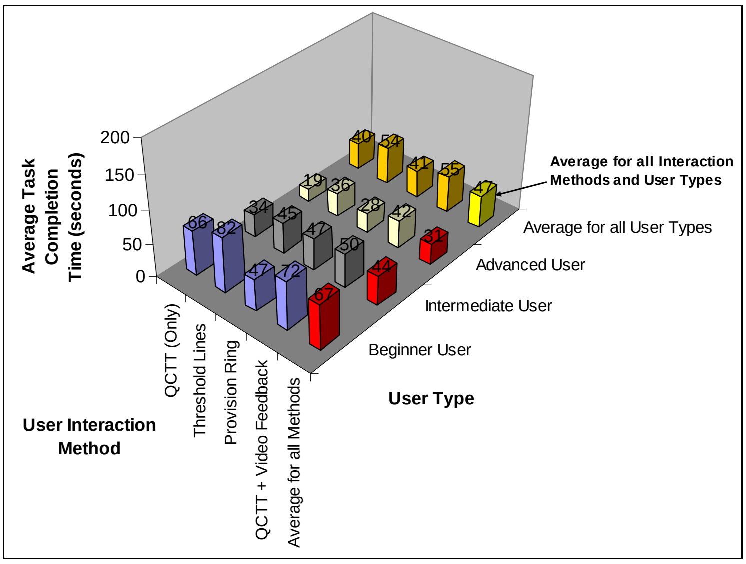


Figure G-13: Large QCTT 10,000 (10k) Fractals Interface used on an Emulated PDA, Task Completion Time for each User Type and Interaction Method

Task Error Count

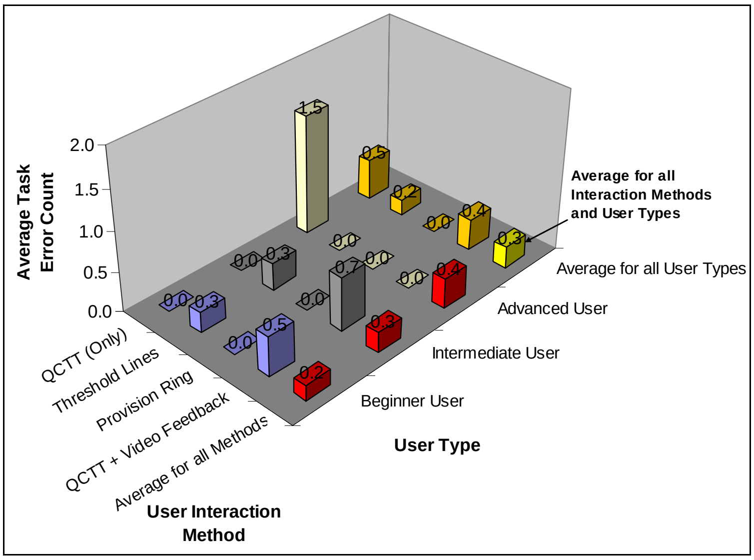


Figure G-14: Large QCTT 10,000 (10k) Fractals Interface used on an Emulated PDA, Task Error Count for each User Type and Interaction Method

Usability Ratings

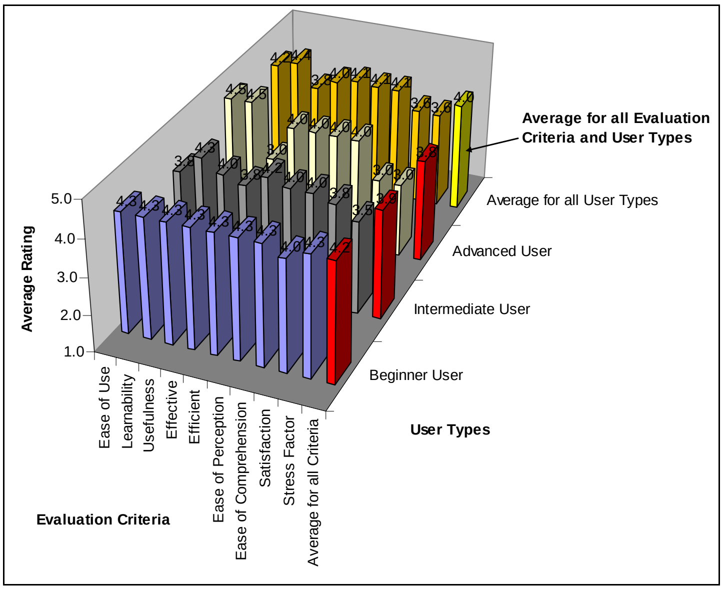


Figure G-15: Large QCTT 10,000 (10k) Fractals Interface used on an Emulated PDA, Usability Ratings for the *QCTT Only* Interaction Method

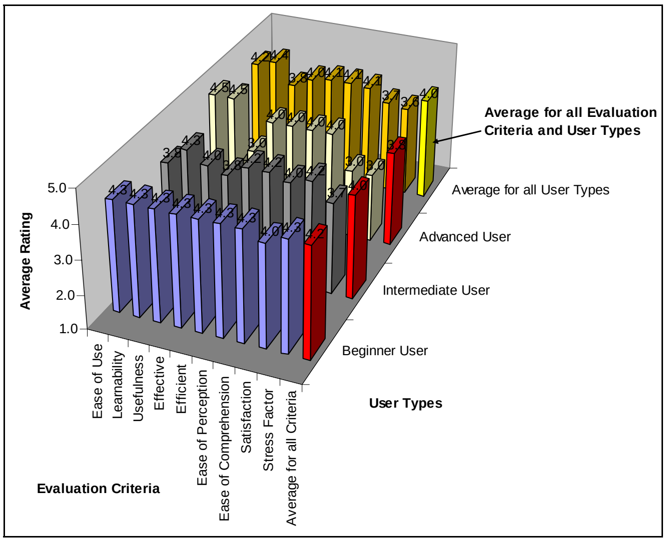


Figure G-16: Large QCTT 10,000 (10k) Fractals Interface used on an Emulated PDA, Usability Ratings for the *Threshold Lines* Interaction Method

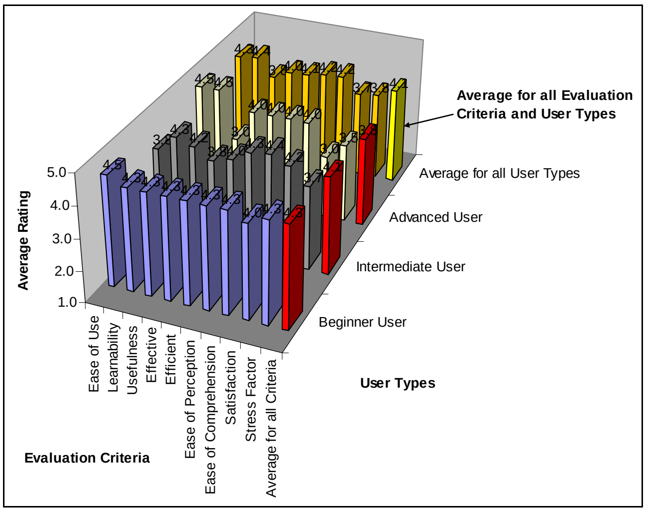


Figure G-17: Large QCTT 10,000 (10k) Fractals Interface used on an Emulated PDA, Usability Ratings for the *Provision Ring* Interaction Method

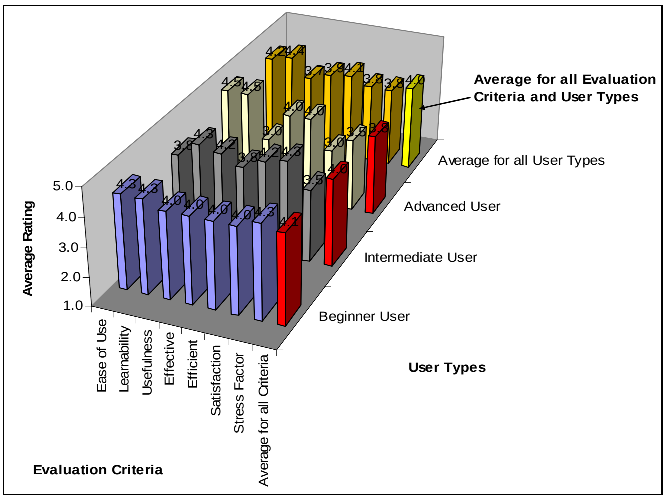


Figure G-18: Large QCTT 10,000 (10k) Fractals Interface used on an Emulated PDA, Usability Ratings for the *QCTT plus video feedback* Interaction Method

G.2 Emulated Mobile Phone

G.2.1 Large QCTT Quad Fractals Interface

Task Completion Times

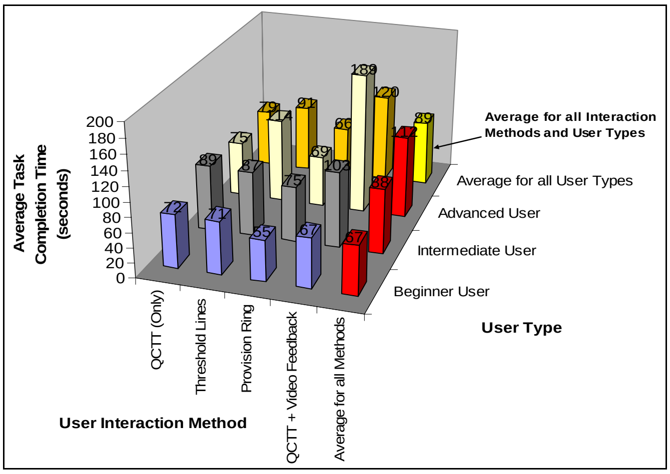


Figure G-19: Large QCTT Quad Fractals Interface used on an Emulated Mobile Phone, Task Completion Time for each User Type and Interaction Method

Task Error Count

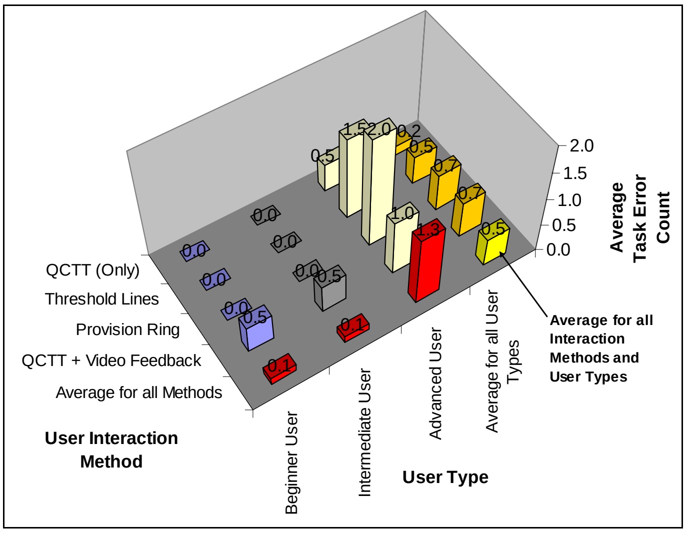


Figure G-20: Large QCTT Quad Fractals Interface used on an Emulated Mobile Phone, Task Error Count for each User Type and Interaction Method

Usability Ratings

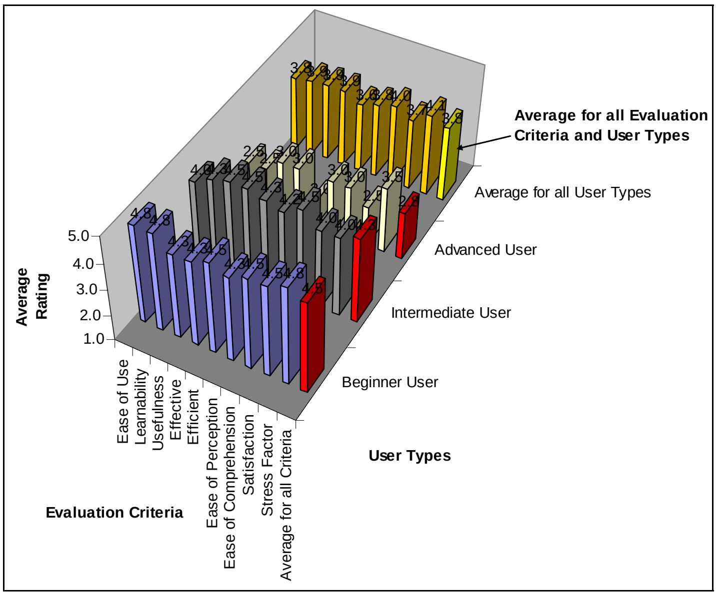


Figure G-21: Large QCTT Quad Fractals Interface used on an Emulated Mobile Phone, Usability Ratings for the *QCTT Only* Interaction Method

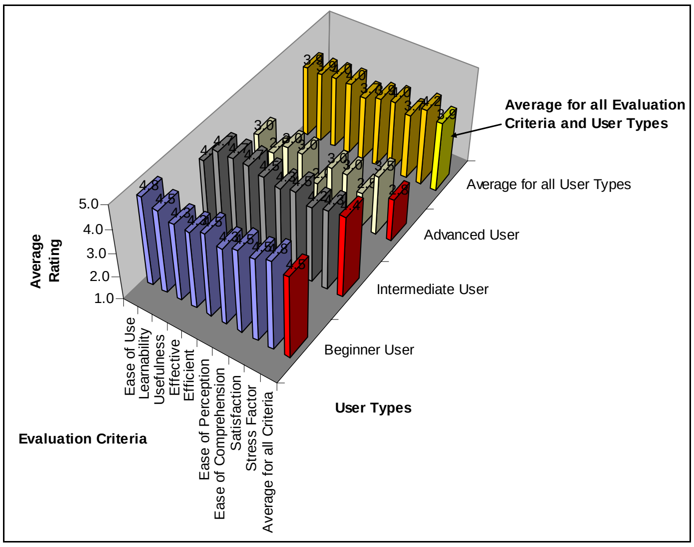


Figure G-22: Large QCTT Quad Fractals Interface used on an Emulated Mobile Phone, Usability Ratings for the *Threshold Lines* Interaction Method

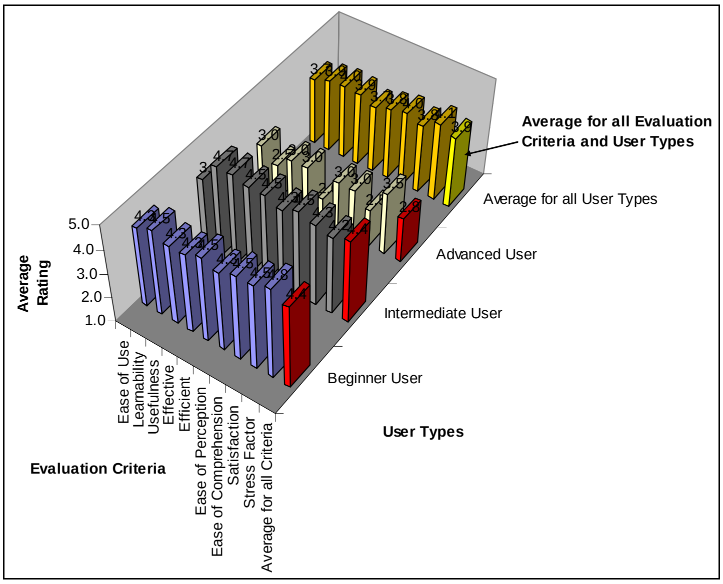


Figure G-23: Large QCTT Quad Fractals Interface used on an Emulated Mobile Phone, Usability Ratings for the *Provision Ring* Interaction Method

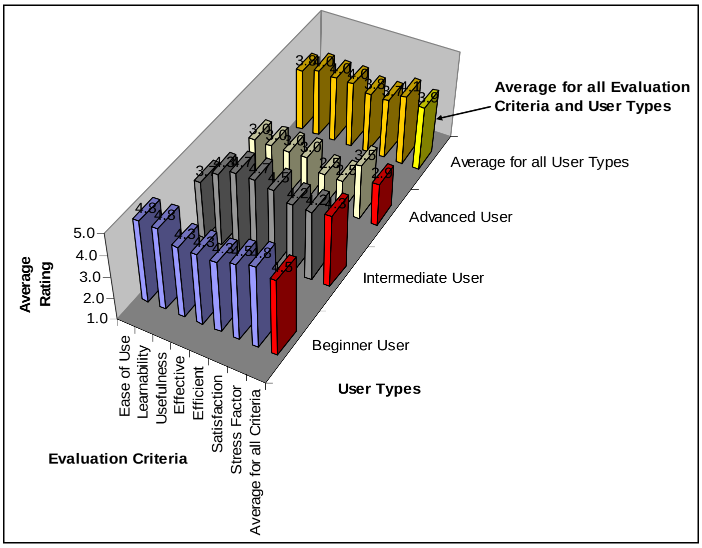


Figure G-24: Large QCTT Quad Fractals Interface used on an Emulated Mobile Phone, Usability Ratings for the *QCTT plus video feedback* Interaction Method

G.2.2 Small QCTT Quad Fractals Interface

Task Completion Times

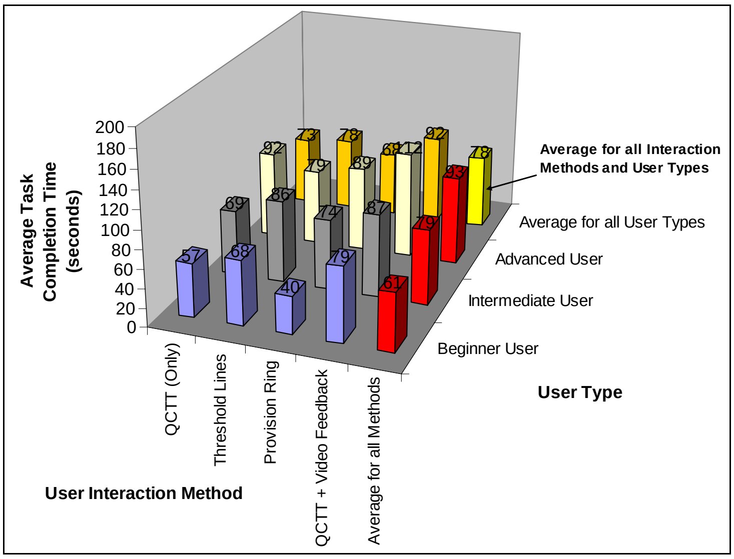


Figure G-25: Small QCTT Quad Fractals Interface used on an Emulated Mobile Phone, Task Completion Time for each User Type and Interaction Method

Task Error Count

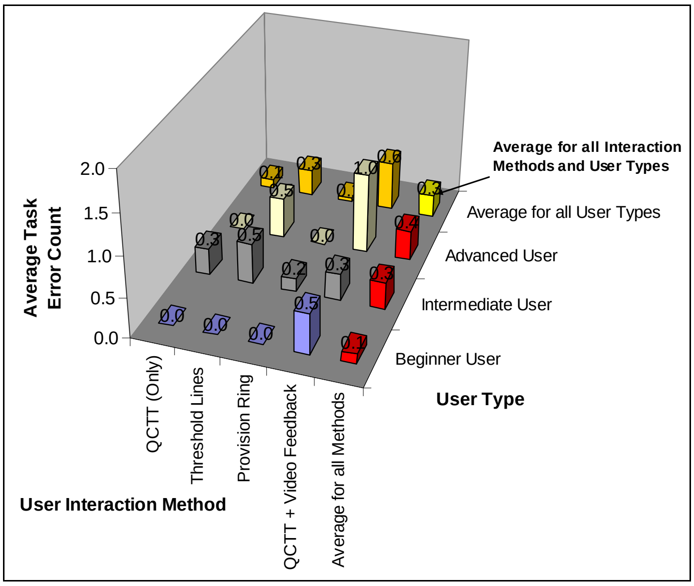


Figure G-26: Small QCTT Quad Fractals Interface used on an Emulated Mobile Phone, Task Error Count for each User Type and Interaction Method

Usability Ratings

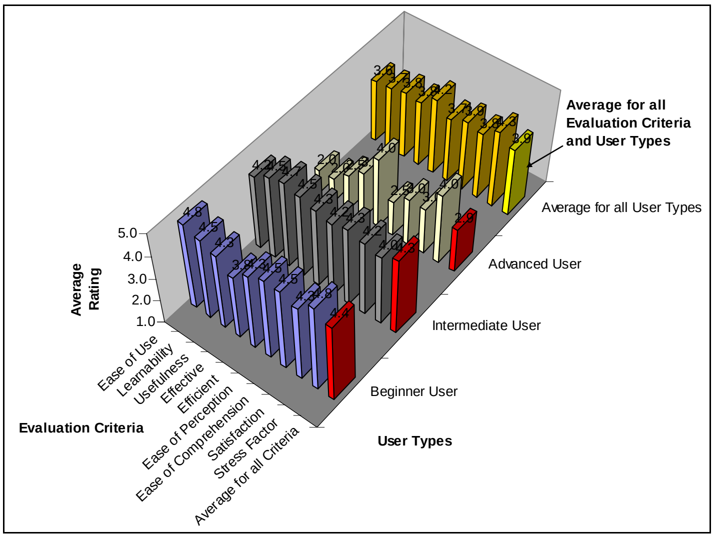


Figure G-27: Small QCTT Quad Fractals Interface used on an Emulated Mobile Phone Usability Ratings for the *QCTT Only* Interaction Method

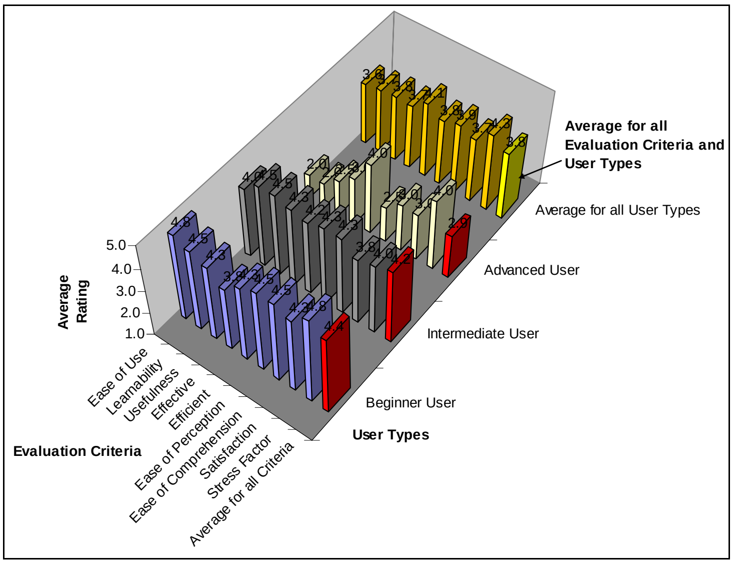


Figure G-28: Small QCTT Quad Fractals Interface used on an Emulated Mobile Phone, Usability Ratings for the *Threshold Lines* Interaction Method

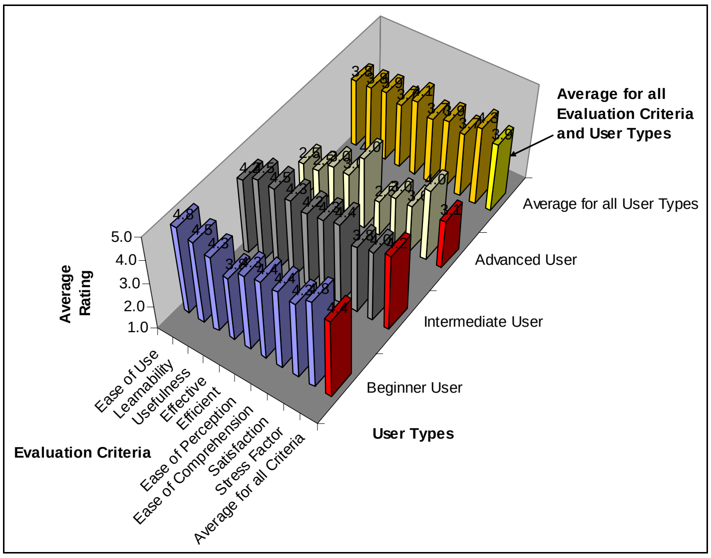


Figure G-29: Small QCTT Quad Fractals Interface used on an Emulated Mobile Phone, Usability Ratings for the *Provision Ring* Interaction Method

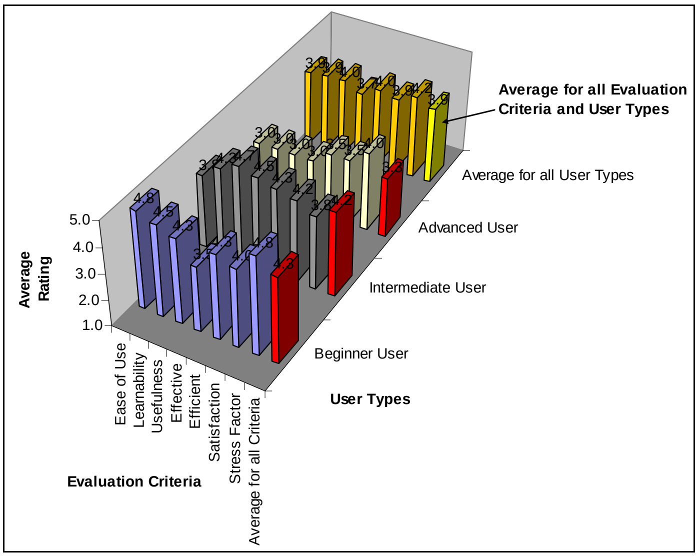


Figure G-30: Small QCTT Quad Fractals Interface used on an Emulated Mobile Phone, Usability Ratings for the *QCTT plus video feedback* Interaction Method

G.2.3 Large QCTT 10,000 (10k) Fractals Interface

Task Completion Times

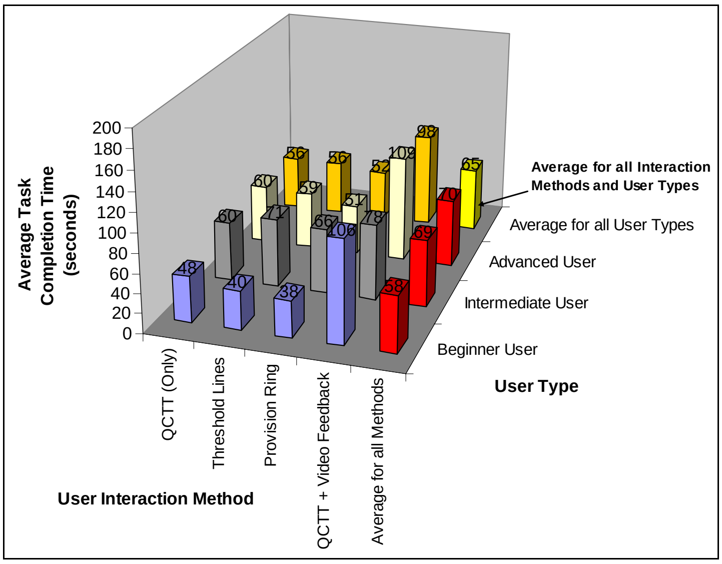


Figure G-31: Large QCTT 10,000 (10k) Fractals Interface used on an Emulated Mobile Phone, Task Completion Time for each User Type and Interaction Method

Task Error Count

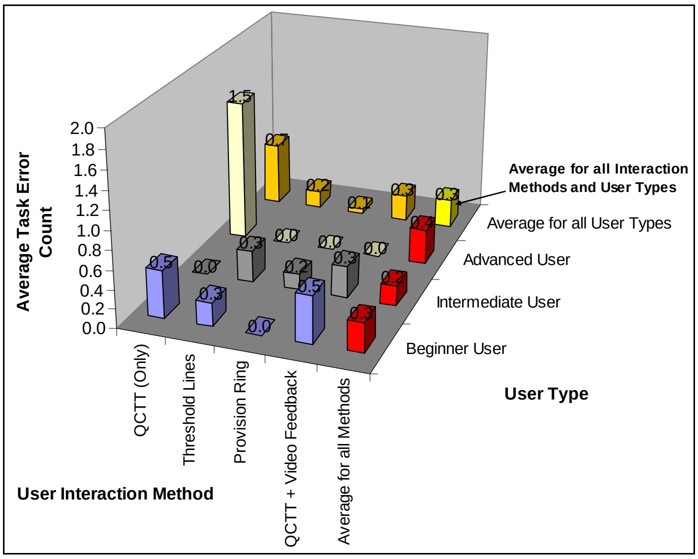


Figure G-32: Large QCTT 10,000 (10k) Fractals Interface used on an Emulated Mobile Phone, Task Error Count for each User Type and Interaction Method

Usability Ratings

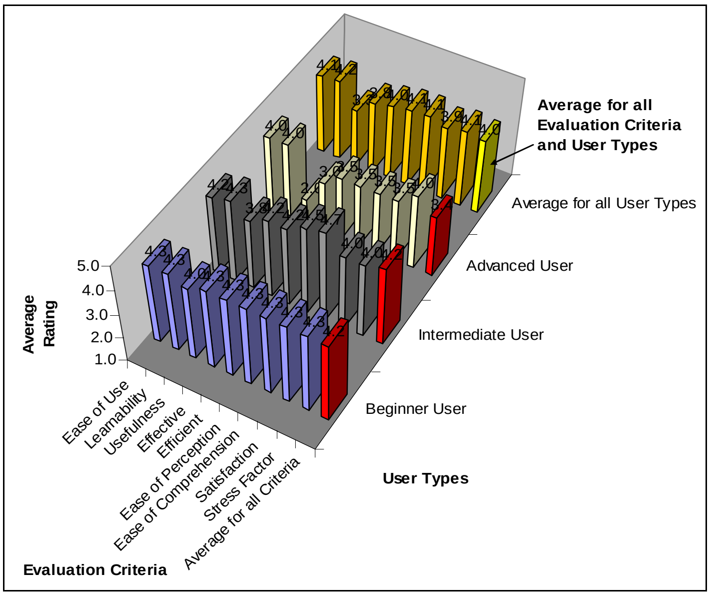


Figure G-33: Large QCTT 10,000 (10k) Fractals Interface used on an Emulated Mobile Phone, Usability Ratings for the *QCTT Only* Interaction Method

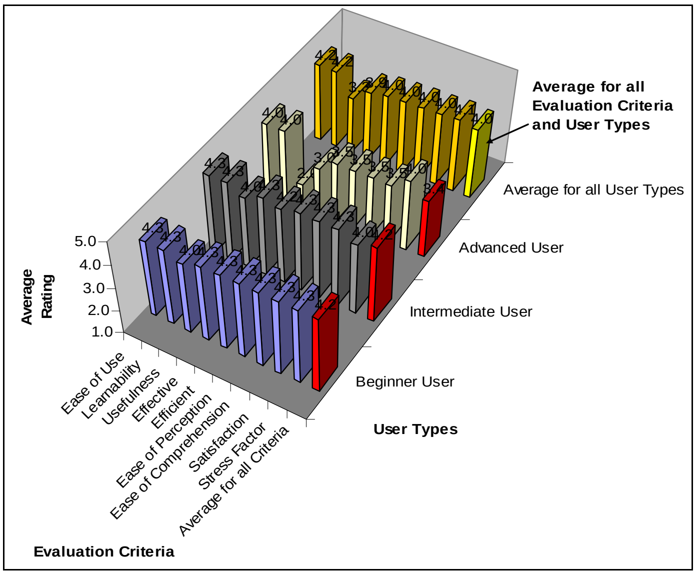


Figure G-34: Large QCTT 10,000 (10k) Fractals Interface used on an Emulated Mobile Phone, Usability Ratings for the *Threshold Lines* Interaction Method

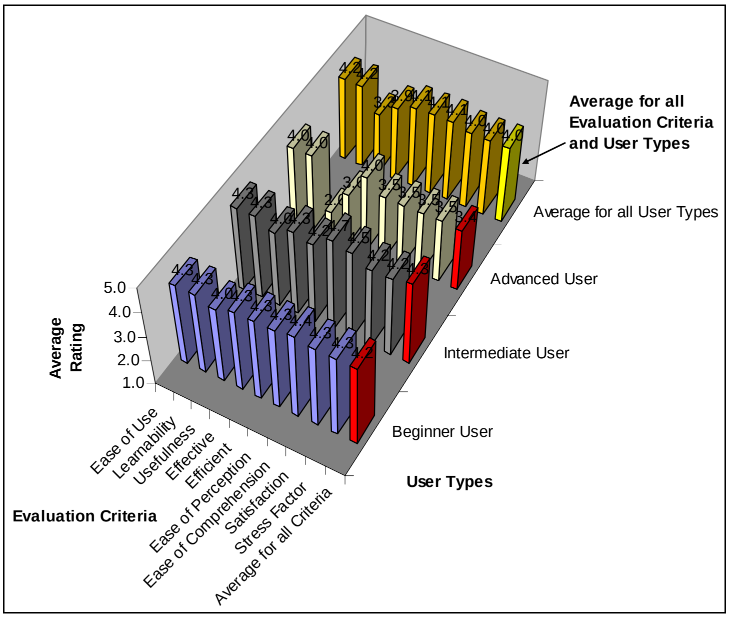


Figure G-35: Large QCTT 10,000 (10k) Fractals Interface used on an Emulated Mobile Phone, Usability Ratings for the *Provision Ring* Interaction Method

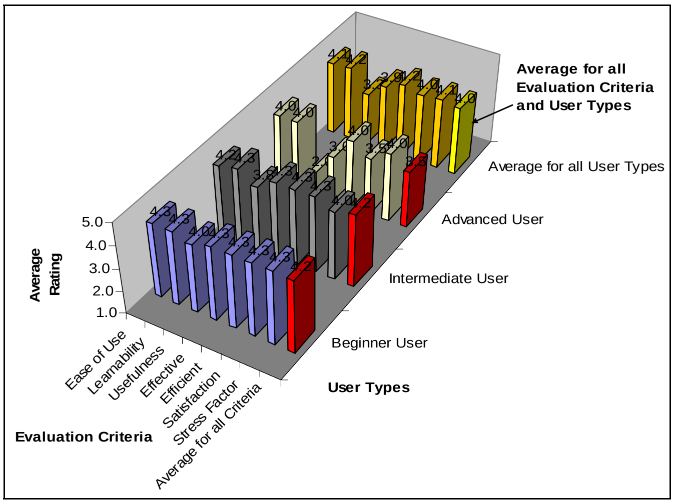


Figure G-36: Large QCTT 10,000 (10k) Fractals Interface used on an Emulated Mobile Phone, Usability Ratings for the *QCTT plus video feedback* Interaction Method

G.3 Real PDA

G.3.1 Large QCTT Quad Fractals Interface

Task Completion Times

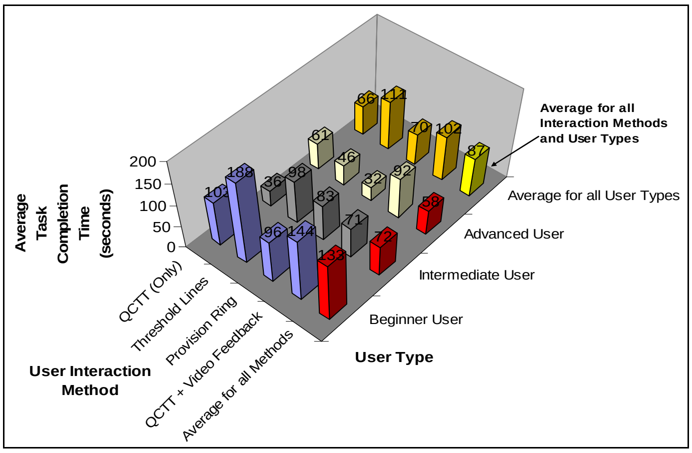


Figure G-37: Large QCTT Quad Fractals Interface used on a PDA, Task Completion Time for each User Type and Interaction Method

Task Error Count

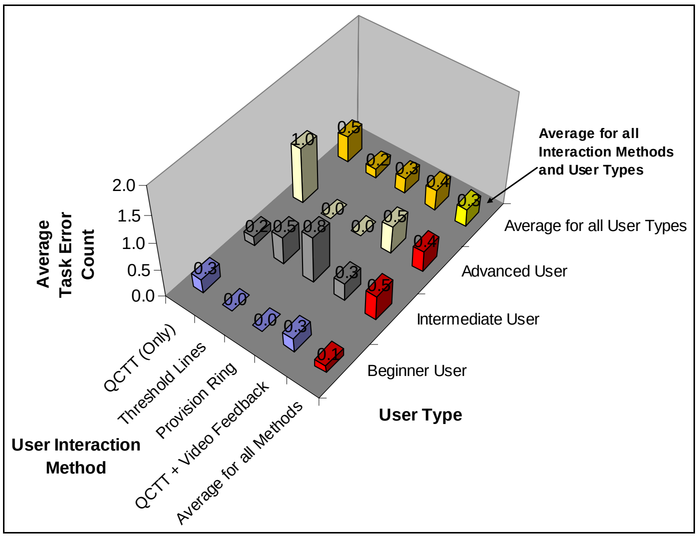


Figure G-38: Large QCTT Quad Fractals Interface used on a PDA, Task Error Count for each User Type and Interaction Method

Usability Ratings

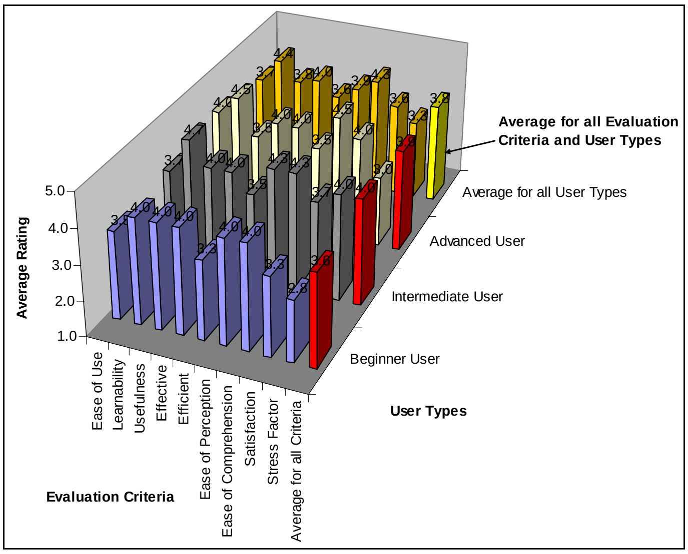


Figure G-39: Large QCTT Quad Fractals Interface used on a PDA, Usability Ratings for the *QCTT Only* Interaction Method

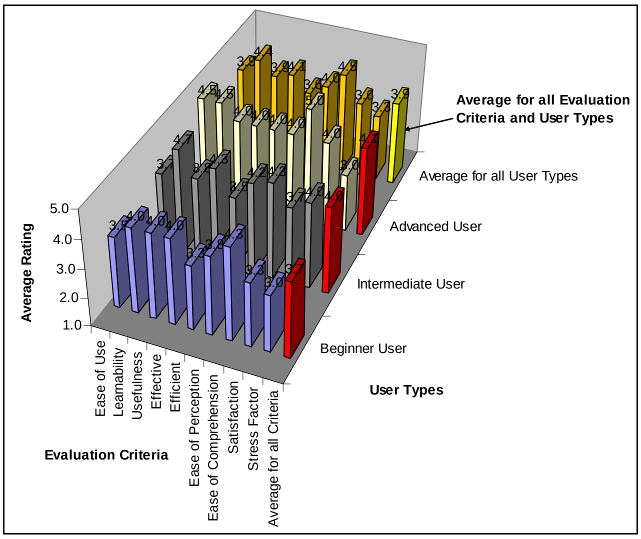


Figure G-40: Large QCTT Quad Fractals Interface used on a PDA, Usability Ratings for the *Threshold Lines* Interaction Method

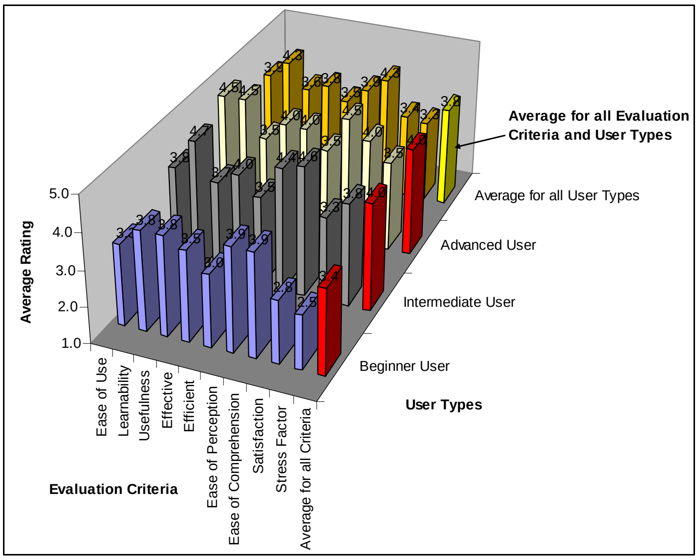


Figure G-41: Large QCTT Quad Fractals Interface used on a PDA, Usability Ratings for the *Provision Ring* Interaction Method

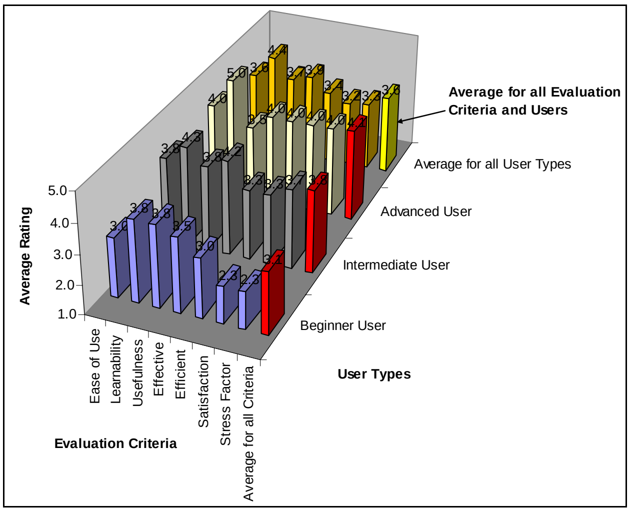


Figure G-42: Large QCTT Quad Fractals Interface used on a PDA, Usability Ratings for the *QCTT plus video feedback* Interaction Method

G.3.2 Small QCTT Quad Fractals Interface

Task Completion Times

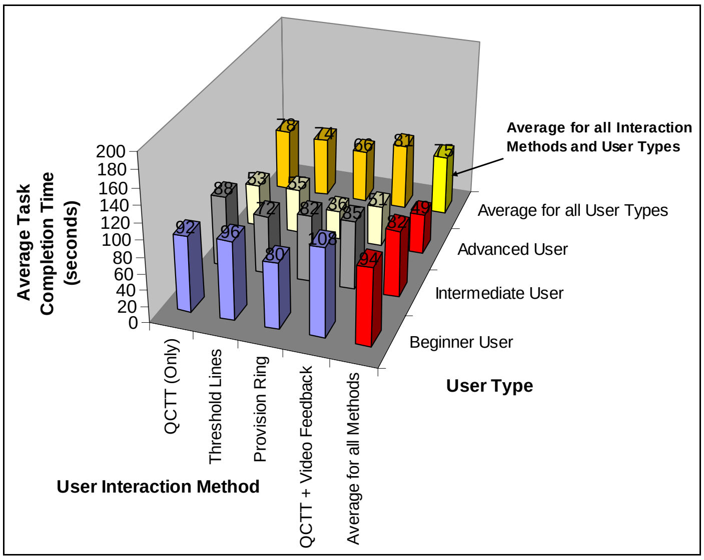


Figure G-43: Small QCTT Quad Fractals Interface used on a PDA, Task Completion Time for each User Type and Interaction Method

Task Error Count

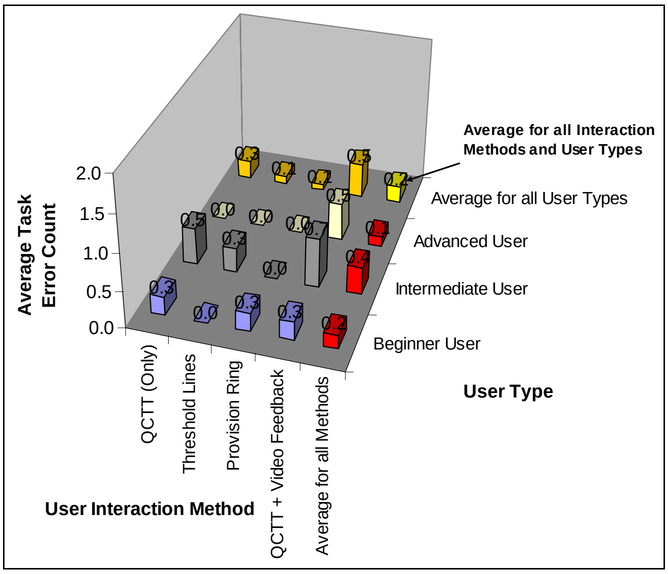


Figure G-44: Small QCTT Quad Fractals Interface used on a PDA, Task Error Count for each User Type and Interaction Method

Usability Ratings

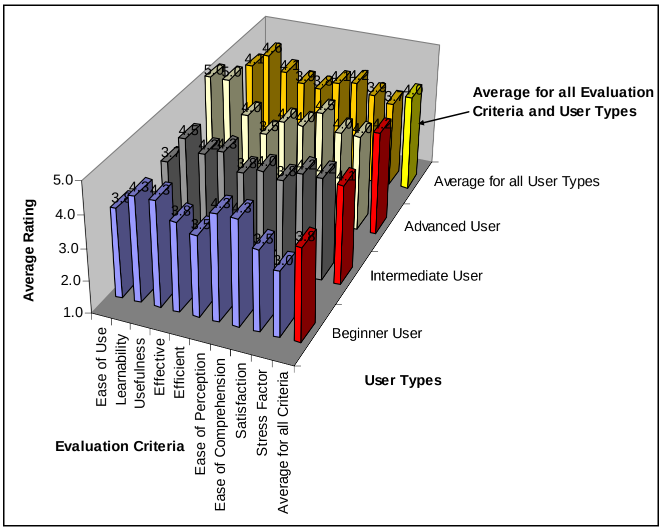


Figure G-45: Small QCTT Quad Fractals Interface used on a PDA, Usability Ratings for the *QCTT Only* Interaction Method

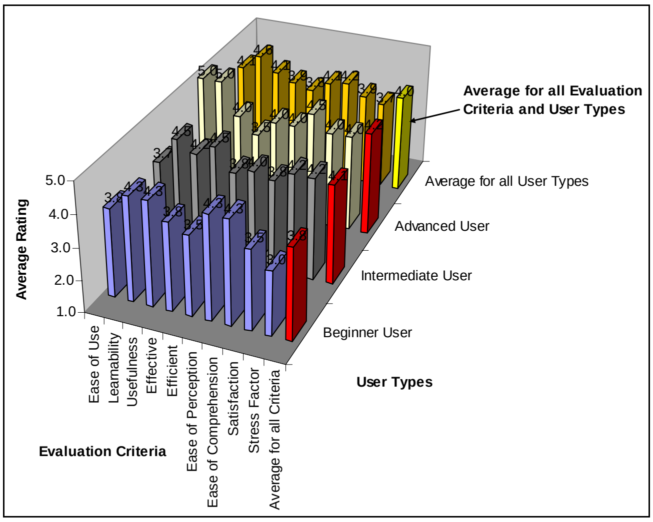


Figure G-46: Small QCTT Quad Fractals Interface used on a PDA, Usability Ratings for the *Threshold Lines* Interaction Method

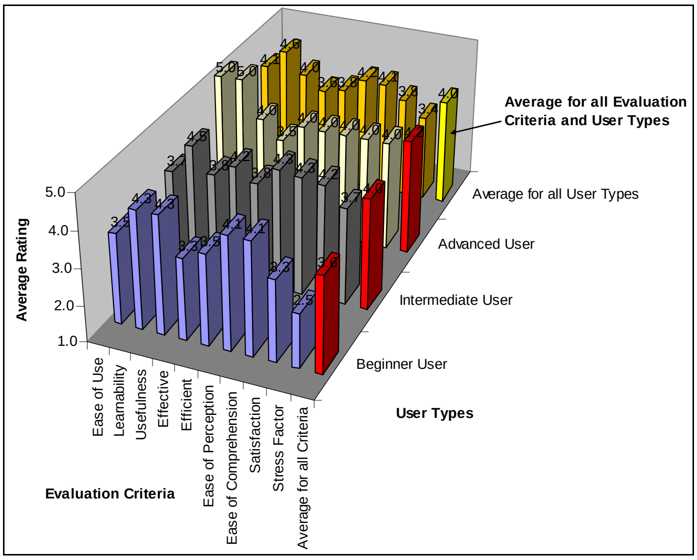


Figure G-47: Small QCTT Quad Fractals Interface used on a PDA, Usability Ratings for the *Provision Ring* Interaction Method

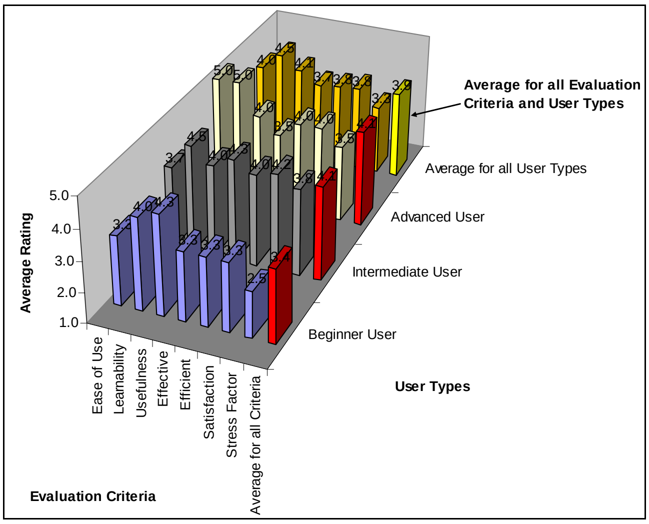


Figure G-48: Small QCTT Quad Fractals Interface used on a PDA, Usability Ratings for the *QCTT plus video feedback* Interaction Method

G.3.3 Large QCTT 10,000 (10k) Fractals Interface

Task Completion Times

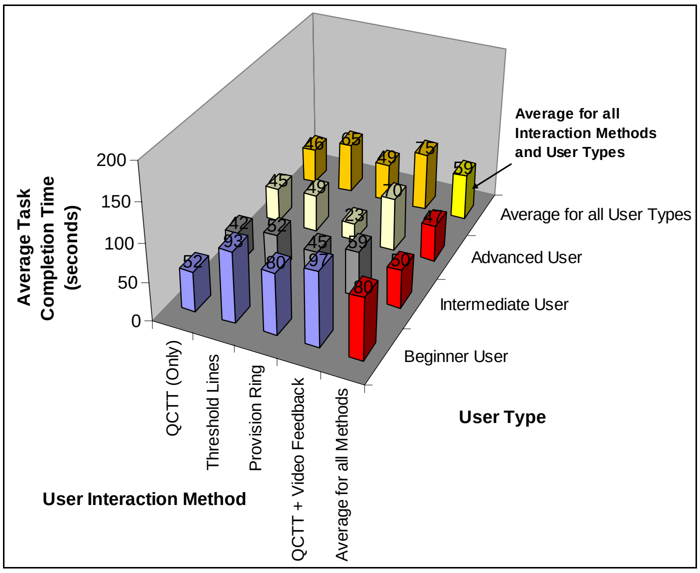


Figure G-49: Large QCTT 10,000 (10k) Fractals Interface used on a PDA, Task Completion Time for each User Type and Interaction Method

Task Error Count

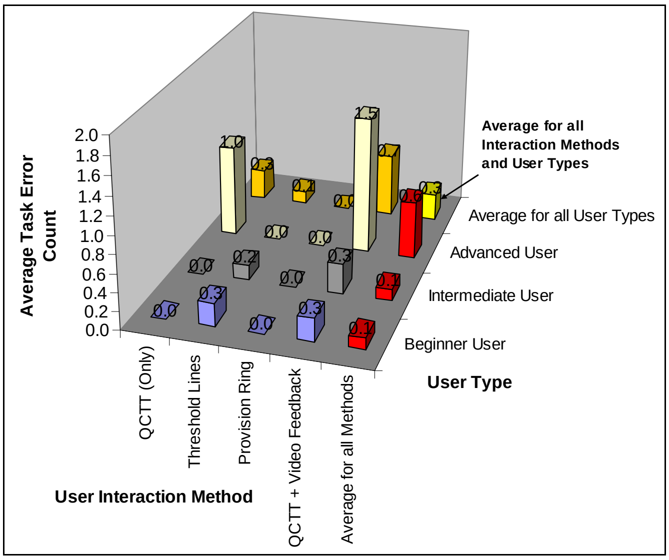


Figure G-50: Large QCTT 10,000 (10k) Fractals Interface used on a PDA, Task Error Count for each User Type and Interaction Method

Usability Ratings

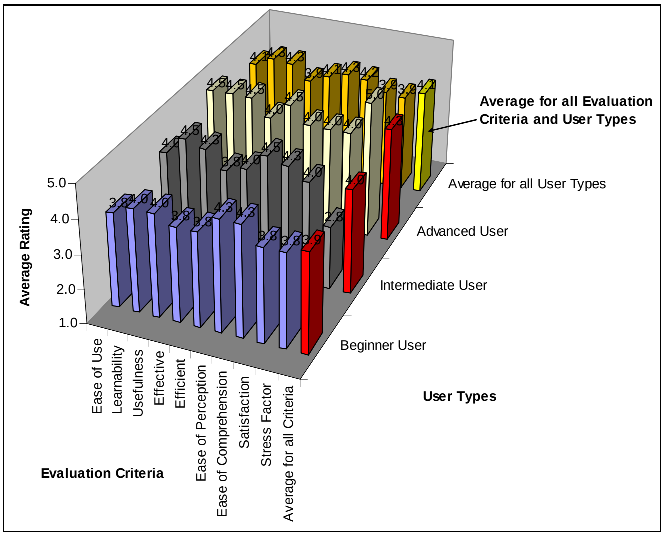


Figure G-51: Large QCTT 10,000 (10k) Fractals Interface used on a PDA, Usability Ratings for the *QCTT Only* Interaction Method

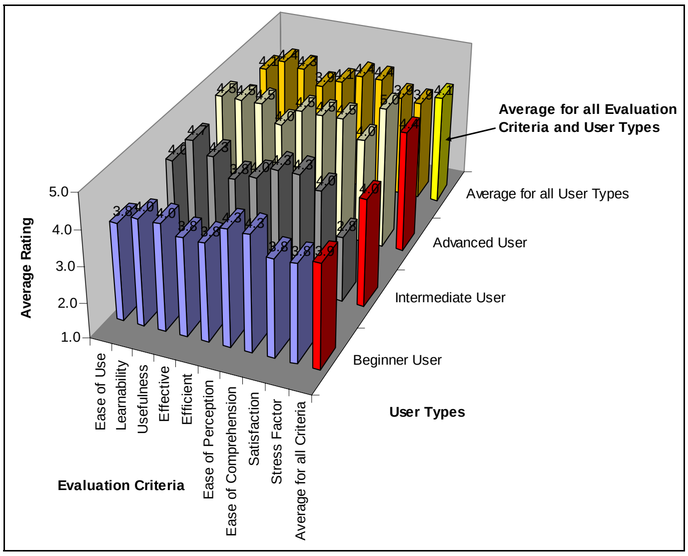


Figure G-52: Large QCTT 10,000 (10k) Fractals Interface used on a PDA, Usability Ratings for the *Threshold Lines* Interaction Method

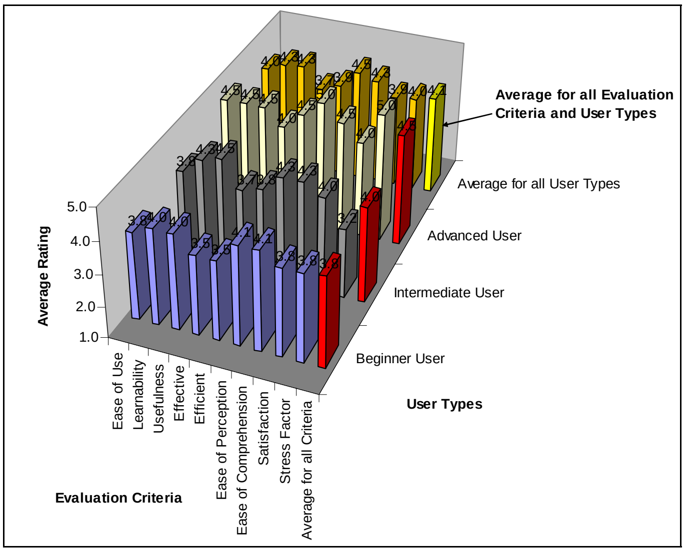


Figure G-53: Large QCTT 10,000 (10k) Fractals Interface used on a PDA, Usability Ratings for the *Provision Ring* Interaction Method

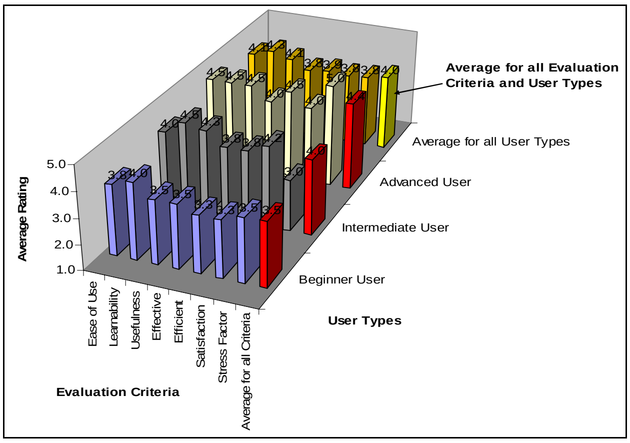


Figure G-54: Large QCTT 10,000 (10k) Fractals Interface used on a PDA, Usability Ratings for the *QCTT plus video feedback* Interaction Method

G.4 Combined Results

G.4.1 Task Completion Times

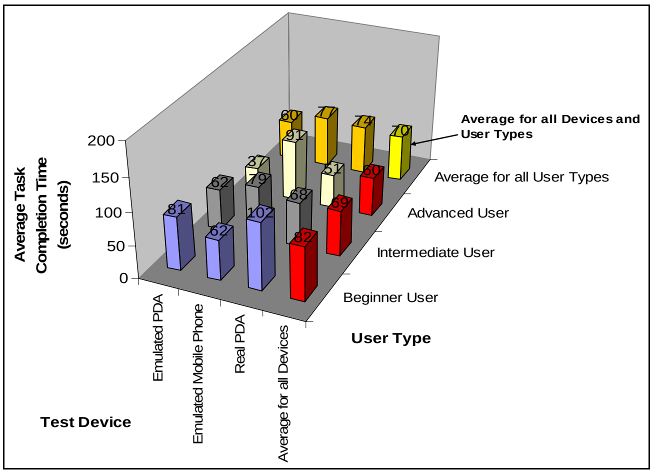


Figure G-55: Task Completion Times for each User Type and Test Device

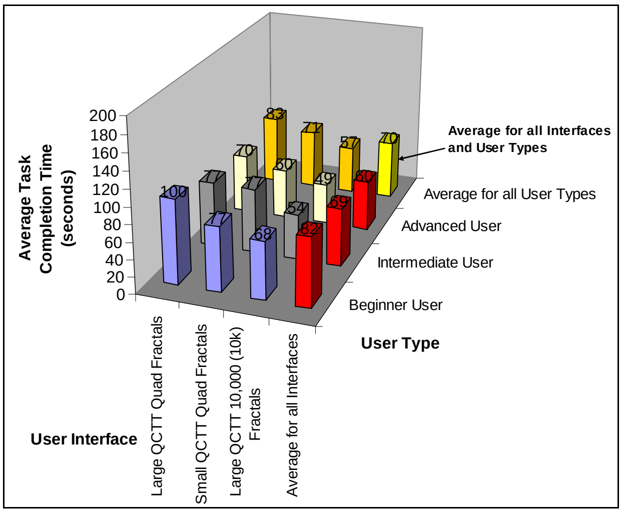


Figure G-56: Combined Task Completion Time for each User Type and Interface

G.4.2 Task Error Count

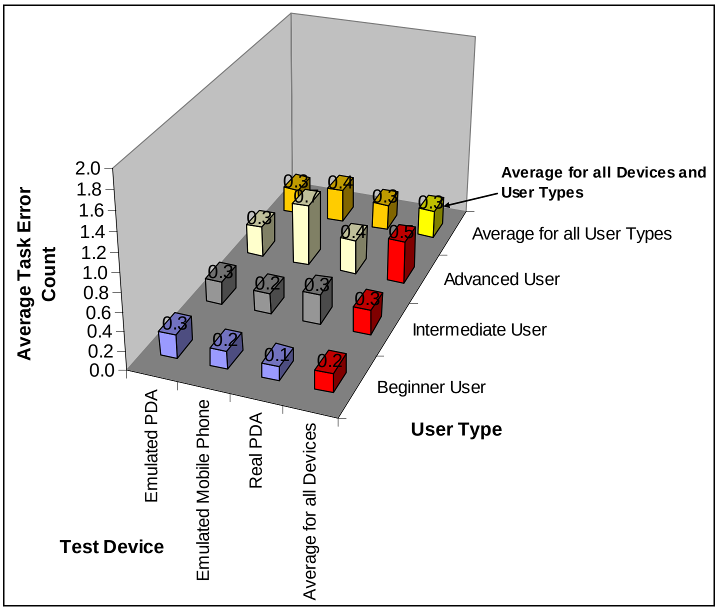


Figure G-57: Task Error Count for each User Type and Test Device

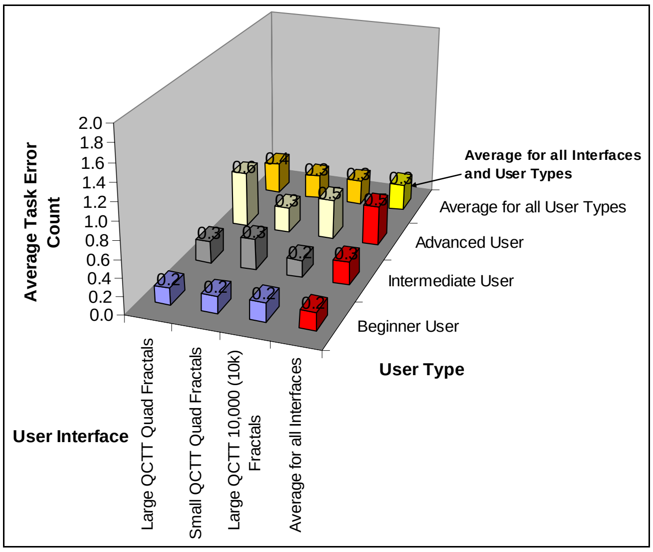


Figure G-58: Combined Task Error Count for each User Type and Interface

G.4.3 Usability Ratings

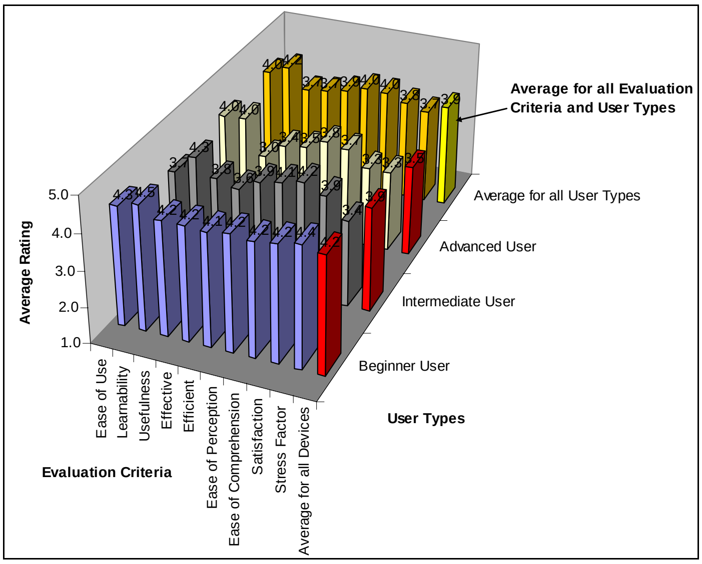


Figure G-59: Emulated PDA: Average Ratings for each User Type and Criterion

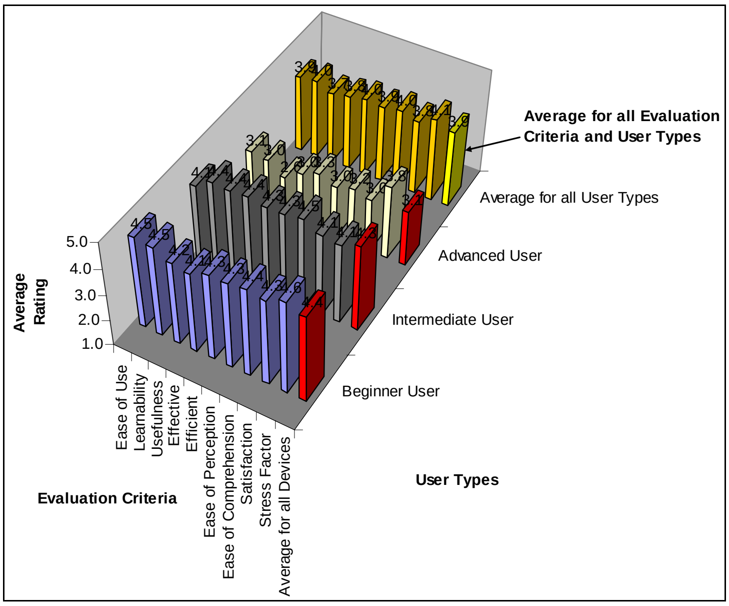


Figure G-60: Emulated Mobile Phone: Average Ratings for each User Type and Criterion

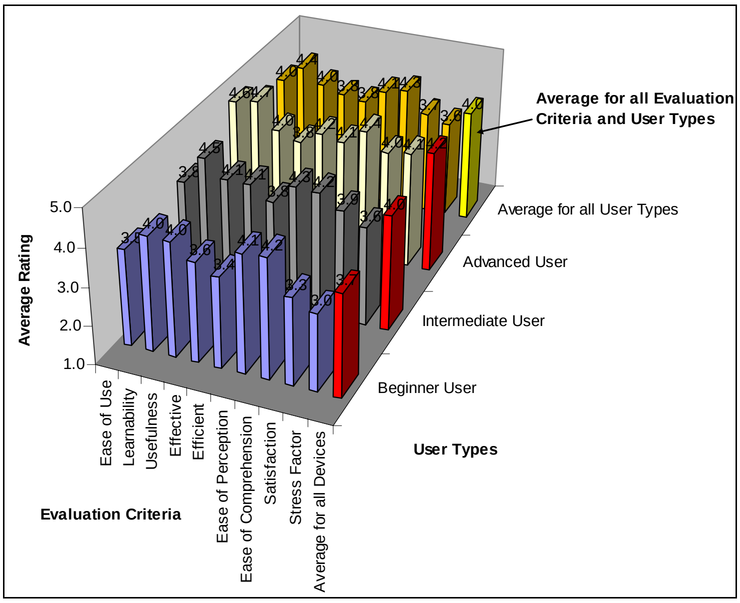


Figure G-61: PDA: Average Ratings for each User Type and Criterion

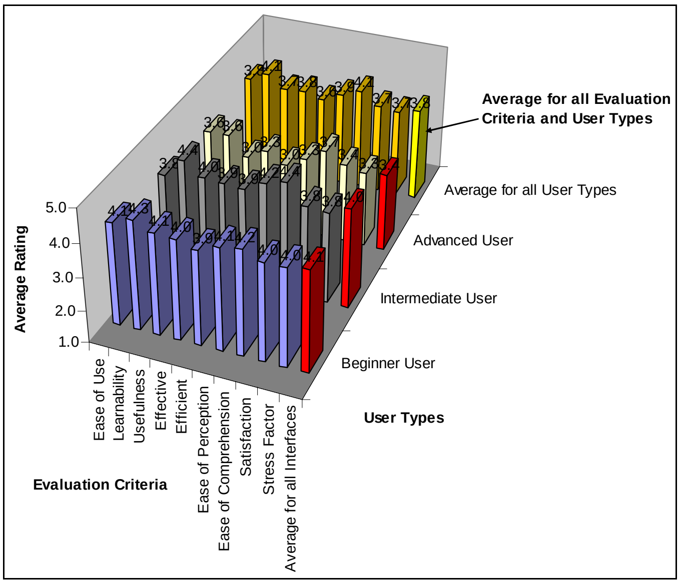


Figure G-62: Large QCTT Quad Fractals Interface, Average Ratings for each User Type and Criterion

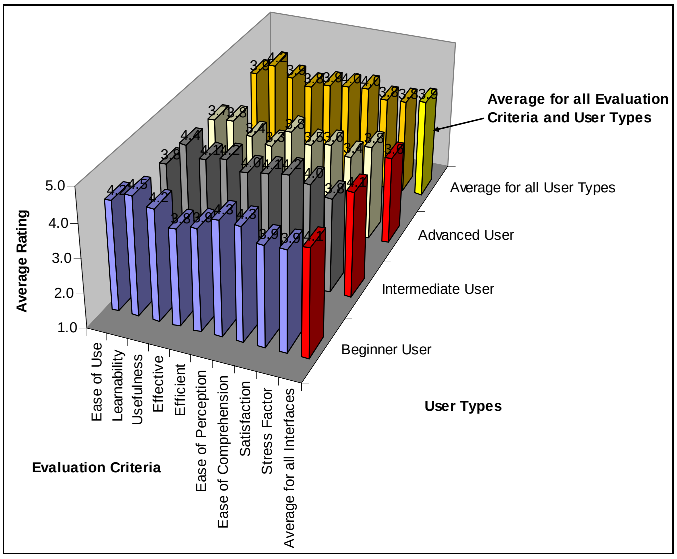


Figure G-63: Small QCTT Quad Fractals Interface, Average Ratings for each User Type and Criterion

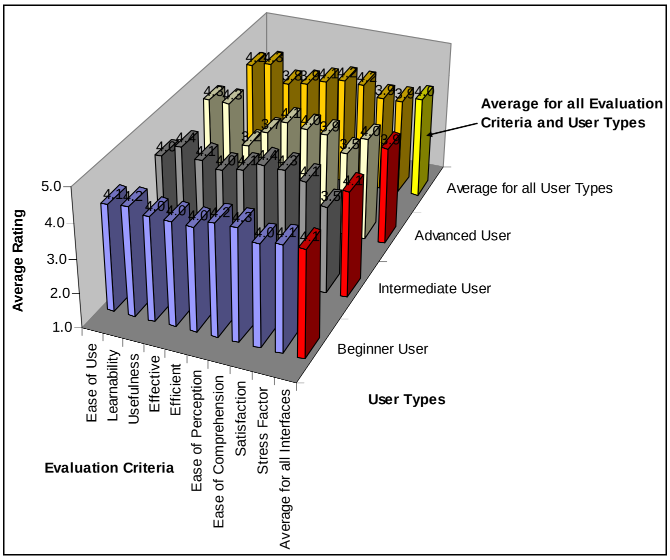


Figure G-64: Large QCTT 10,000 (10k) Fractals Interface, Average Ratings for each User Type and Criterion

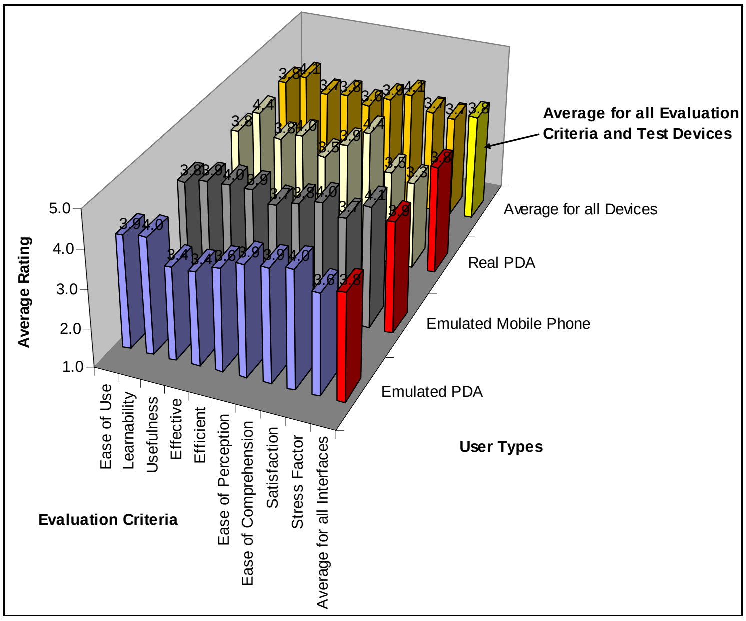


Figure G-65: Large QCTT Quad Fractals Interface, Average Ratings for each Test Device and Criterion

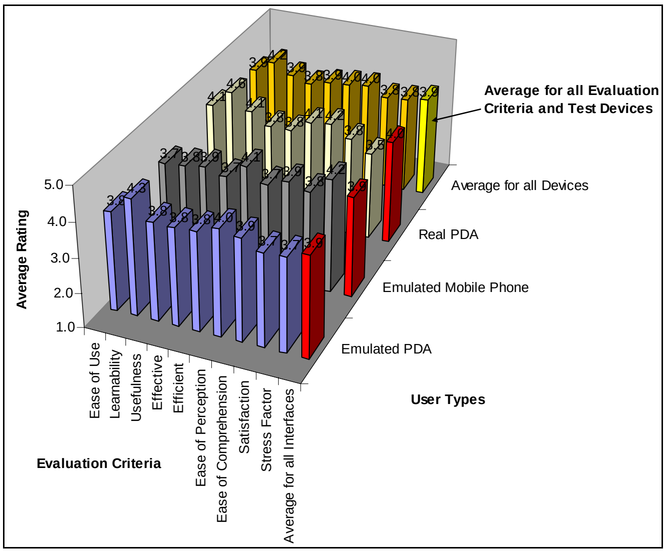


Figure G-66: Small QCTT Quad Fractals Interface, Average Ratings for each Test Device and Criterion

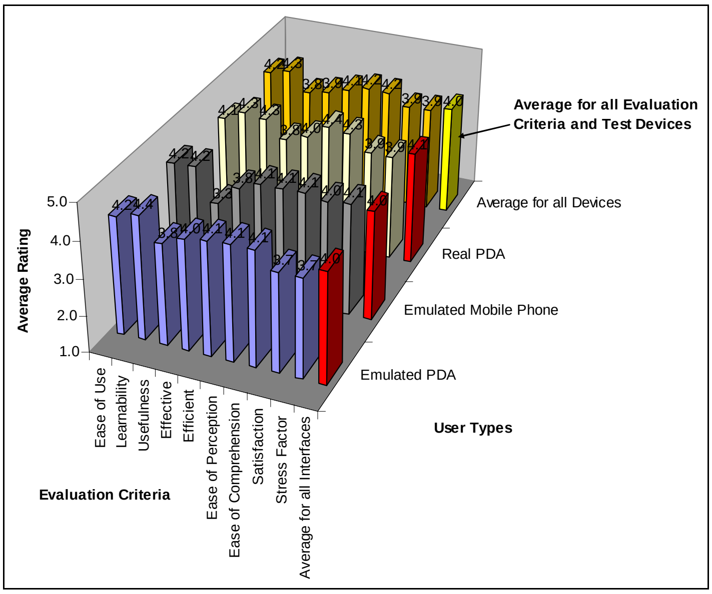


Figure G-67: Large QCTT 10,000 (10k) Fractals Interface, Average Ratings for each Test Device and Criterion

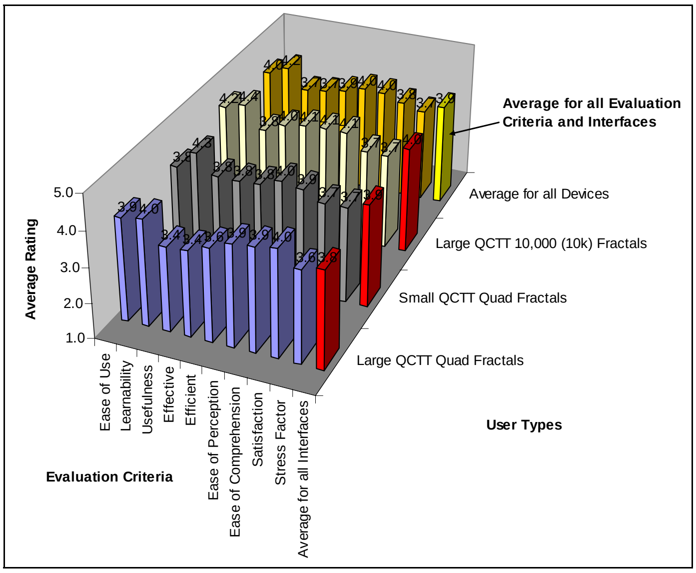


Figure G-68: Emulated PDA: Average Ratings for each Interface and Criterion

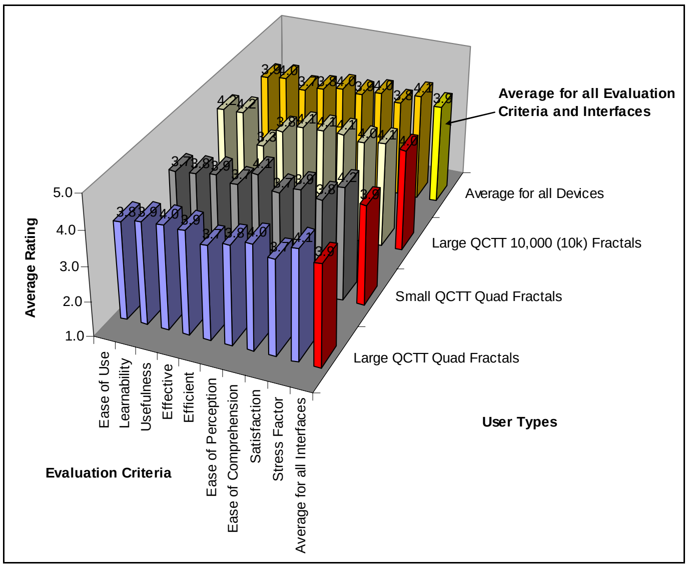


Figure G-69: Emulated Mobile Phone: Average Ratings for each Interface and Criterion

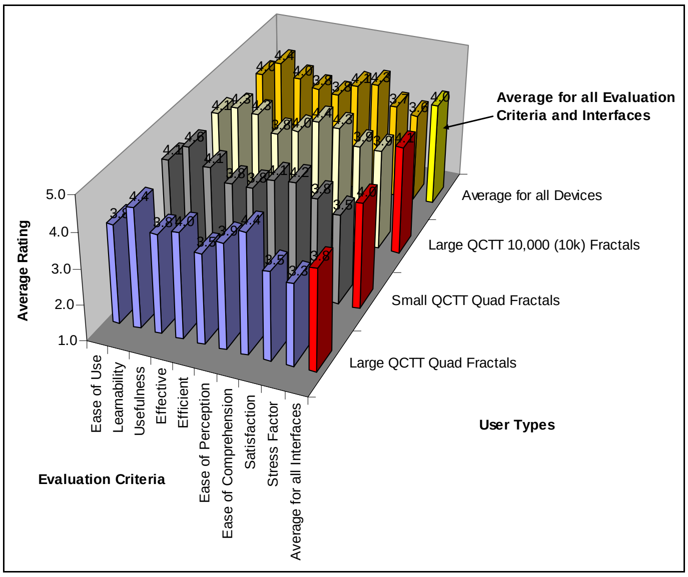


Figure G-70: PDA: Average Ratings for each Interface and Criterion

Appendix H

Detailed Results for Usability Study Y

This Appendix includes detailed results for Usability Study Y (section 6.10).

H.1 Response Times

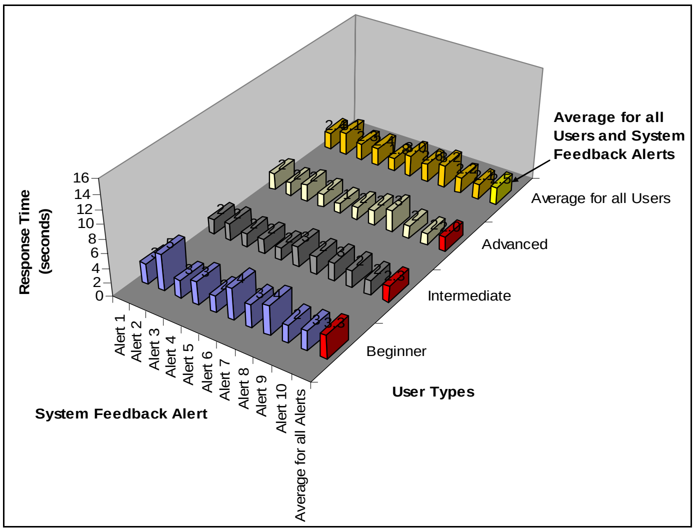


Figure H-1: Emulated PDA: Response Times for the Three User Types

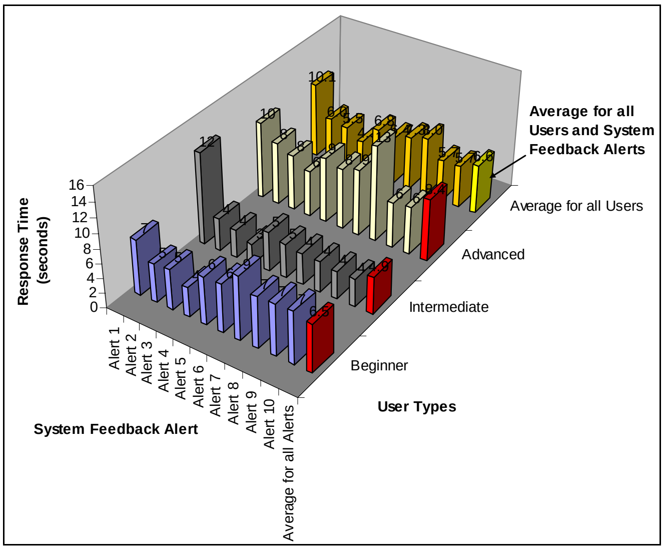


Figure H-2: Emulated Mobile Phone: Response Times for the Three User Types

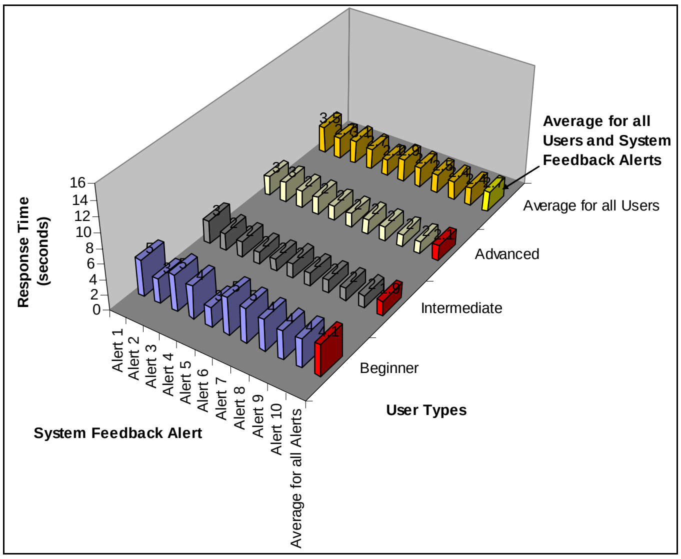


Figure H-3: PDA: Response Times for the Three User Types

H.2 Usability Ratings

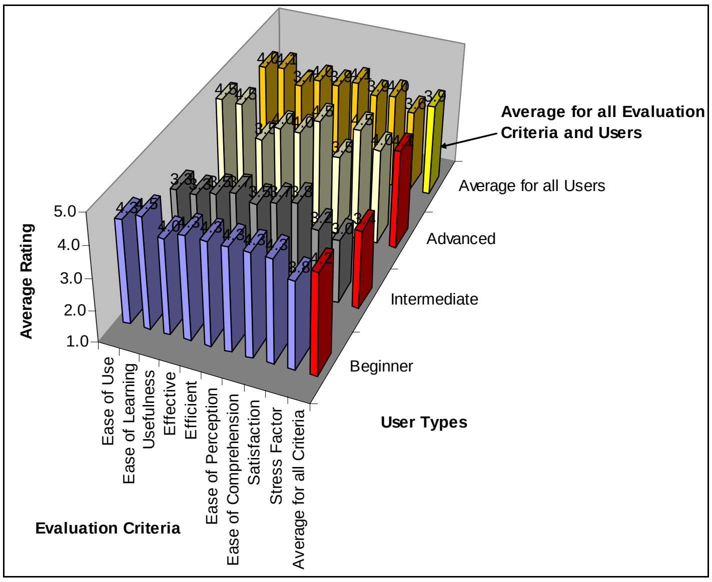


Figure H-4: Emulated PDA: Average Ratings for each User Type and Criterion

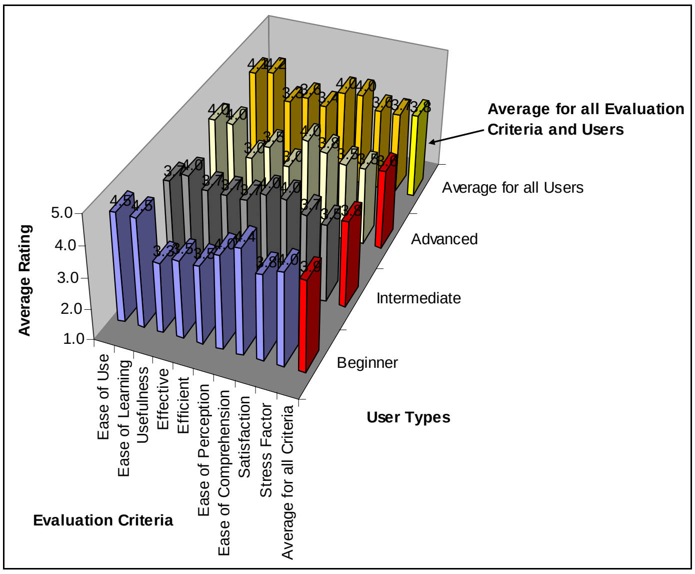


Figure H-5: Emulated Mobile Phone: Average Ratings for each User Type and Criterion

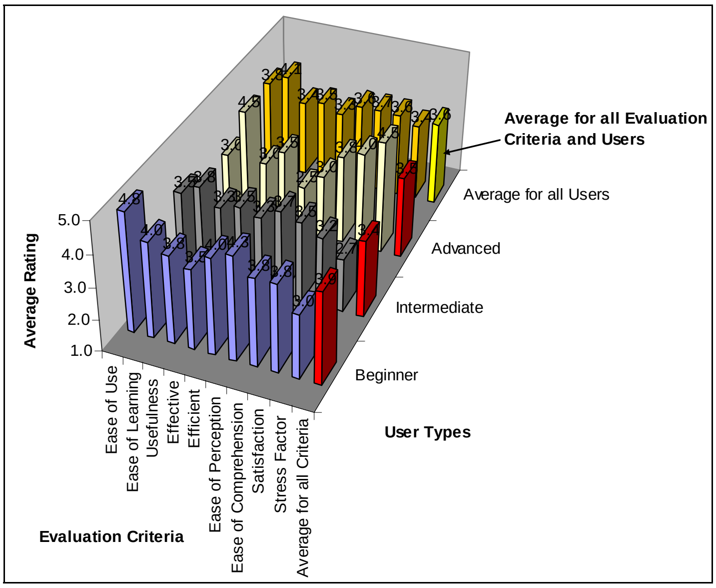


Figure H-6: PDA: Average Ratings for each User Type and Criterion

H.3 Combined Results

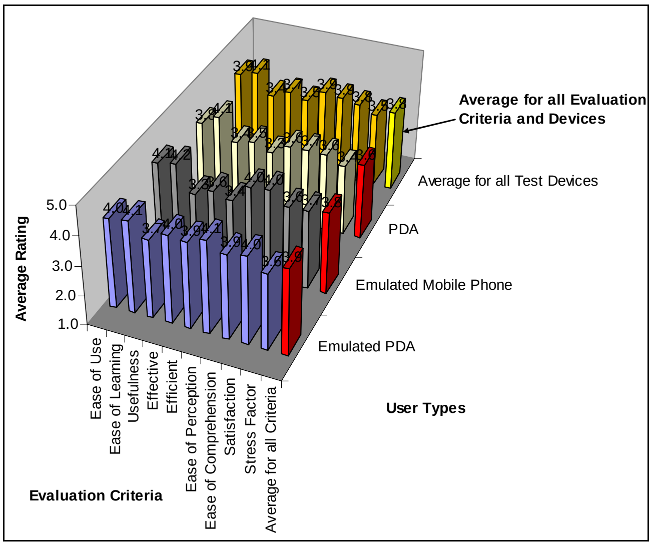


Figure H-7: Overall Average Ratings for each Test Device and Criterion

Appendix I
Memorability Questionnaire

This appendix presents the Memorability Questionnaire used to measure the memorability for the QCTT interface that was enhanced for Portable Devices, as presented in Usability Study Z (section 6.11).

Usability Study Z: Memorability Questionnaire for the QCTT Interface

Please complete the following memorability test twenty minutes after you have completed the experiment.

	User Identification Number:	
1	*A red circle displayed in the QCTT interface implies what type of given response?*	**Answer Here**
	a. Advisory b. Informative c. Critical	
2	*If you move the Pivot-Point closer to the Cost Factor (C) performance aspect, the cost will:*	**Answer Here**
	a. Increase b. Decrease c. Not Change d. Increase then Decrease e. None of a - d	
3	*The Purple Ring around the Pivot-Point is the:*	**Answer Here**
	a. User Request Ring b. Provision Ring c. Pivot Point Ring d. Video Ring e. None of a - d	
4	*When you move the Pivot-Point farther away from the Quality Resolution (Q) Performance Aspect, Quality Resolution will:*	**Answer Here**
	a. Increase b. Decrease c. Not Change d. Increase then Decrease e. None of a - d	
5	*A yellow circle displayed in the QCTT triangle implies what type of given response?*	**Answer Here**
	a. Advisory b. Critical c. Informative d. none of a - c	

6	*To increase the video frame size you should move the Pivot-Point farther away from the Quality Resolution (Q) Performance Aspect.*	**Answer Here**
	a. True b. False	
7	*An Orange Threshold Line for any given performance aspect means that the performance aspect is at a:*	**Answer Here**
	a. Desirable level b. Undesirable Level c. Less Desirable Level d. None of a - c	
8	*When you move the Pivot-Point closer towards the Delay (D) Performance Aspect, the Cost factor (C) will:*	**Answer Here**
	a. Increase b. Decrease c. Remain Static d. Decrease then Increase e. None of a - d	
9	*When the Pivot-Point moves outside of the Provision Ring, it means that the system cannot provide the required services for the particular performance aspect.*	**Answer Here**
	a. True b. False	
10	*At which three points of the triangle is the Quality Resolution (Q) located at:*	**Answer Here**
	a. Bottom Left b. Bottom Right c. Top Middle	
11	*A displayed Red Threshold Line for any given performance aspect means that performance aspect is at a:*	**Answer Here**
	a. Desirable level b. Undesirable Level c. Less Desirable Level d. None of a - c	
12	*The rules QCTT model states that it is possible to obtain the best services of one aspect, the better of the second and the least of the third aspect.*	**Answer Here**
	a. True b. False	
13	*If you press the "+" button for the Quality Resolution (Q) performance aspect, the value for this aspect will:*	**Answer Here**
	a. Increase b. Decrease c. Not Change d. Increase then Decrease e. None of a - d	
14	*If you press the "+" button for the Quality Resolution (Q) performance aspect, the Pivot-Point will move:*	**Answer Here**
	a. Closer to Q b. Farther from Q c. Closer to C d. a & c e. b & c	

15	*If you press the "-" button for the Cost factor (C) performance aspect, the Pivot-Point will move:*	**Answer Here**
	a. Closer to Q b. Closer to D c. Closer to C d. Farther from C e. None of a - d	
16	*In the User Interface, the values given under the 'Smiley Face' icon imply that these are the system provision values.*	**Answer Here**
	a. True b. False	
17	*At what point of the triangle is the Delay (D) located at:*	**Answer Here**
	a. Bottom Left b. Bottom Right c. Top Middle	
18	*The "+" and "-" buttons in the light blue coloured box row control the Delay (D) Performance Aspect.*	**Answer Here**
	a. True b. False	
19	*One way of moving the Pivot-Point is to simply click anywhere in the QCTT Triangle.*	**Answer Here**
	a. True b. False	
20	*The "+" and "-" buttons in the yellow coloured box row control the Quality Resolution (Q) Performance Aspect.*	**Answer Here**
	a. True b. False	
21	*In the User Interface, the values given under the 'Computer' Icon imply that these are the system provision values.*	**Answer Here**
	a. True b. False	
22	*To have a smooth running video display, it is required that you move the Pivot-Point closer to the Delay (D) performance aspect.*	**Answer Here**
	a. True b. False	

23	*At what point of the triangle is the Cost Factor (C) located at:*	**Answer Here**
	a. Bottom Left b. Bottom Right c. Top Middle	
24	*The Frame Size of the video will increase as the point moves away from the bottom left corner of the triangle*	**Answer Here**
	a. True b. False	
25	*A displayed Green Threshold Line for any given performance aspect means that performance aspect is at a(n):*	**Answer Here**
	a. Desirable level b. Undesirable Level c. Less Desirable Level d. None of a - c	