

Dublin City University

Faculty of Engineering and Computing

School of Electronic Engineering

**Quality of Experience Aware
Adaptive Hypermedia System**

Submitted for the fulfilment of the requirements for the degree of

Doctor in Philosophy (Ph.D.)

Cristina Hava Muntean

Supervisor: Dr. Jennifer McManis

April

2005

DECLARATION

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work and has not been taken from the work of others save to the extent that such work has been cited and acknowledged within the text of my work.

ID No.: _____ 50162004 _____

Signed: _____ 

Date: _____ 11/08/2005 _____

To my dear parents, my sister and to my lovely husband

Acknowledgements

I especially want to thank my supervisors Dr. Jennifer McManis and Dr. Liam Murphy for accepting me in the PEL research group and for their constant support and confidence that I can do research, encouraging me to complete this project. I also thank them for allowing me the freedom to choose the topics that caught my imagination. With their open mind, they helped me a lot when I was faced with the challenge of choosing between many possible research directions.

I am equally indebted to Dr. John Murphy who helped me with his broad knowledge, great kindness, optimism and joviality since I started my research. He provided extremely important help for creating a wonderful work environment here at Performance Engineering Laboratory (PEL) and for my financial support while I was only an exchange student with Dublin City University. He is always a great source of encouragement for all PEL research members.

Together, the three of them have been a very good advising team as any post-graduate student would wish to have. Without their constant support, motivation and useful suggestions, this project would never have taken its present shape.

I want to thank to Prince, Vali, Ola and all PEL members, both from Dublin City University and University College Dublin, Ireland for their cheerful presence, without which the labs would not have been the same.

I also thank to my special Irish friends dear Eileen McEvoy who has warmly accepted me and my husband in her life and soul, making her home another home to us, McEvoy and Thomas families for being very supportive not only when we needed the most but also good friends and Clare Grogan White who always gives us a hand and great advices.

Finally, words cannot describe the love, support, advice and encouragement that I have received from all the members of my extended family. My husband, Gabriel, has always been a constant source of energy in my life. I thank him for initially encouraging me to pursue the path towards PhD, for his valuable professional support throughout, for his boundless patience in reading the draft of this thesis and finally but not the least for his unwavering and enduring love. As for my parents, their love, support and motivation have been the very backbone of my existence. All that I have achieved today is a result of the innumerable silent sacrifices they have made over the past decades.

Cristina Hava Muntean

Dublin, April 2005

A b s t r a c t

The research reported in this thesis proposes, designs and tests a novel Quality of Experience Layer (QoE-layer) for the classic Adaptive Hypermedia Systems (AHS) architecture. Its goal is to improve the end-user perceived Quality of Service in different operational environments suitable for residential users. While the AHS' main role of delivering personalised content is not altered, its functionality and performance is improved and thus the user satisfaction with the service provided.

The QoE Layer takes into account multiple factors that affect Quality of Experience (QoE), such as Web components and network connection. It uses a novel Perceived Performance Model that takes into consideration a variety of performance metrics, in order to learn about the Web user operational environment characteristics, about changes in network connection and the consequences of these changes on the user's quality of experience. This model also considers the user's subjective opinion about his/her QoE, increasing its effectiveness and suggests strategies for tailoring Web content in order to improve QoE. The user related information is modelled using a stereotype-based technique that makes use of probability and distribution theory.

The QoE-Layer has been assessed through both simulations and qualitative evaluation in the educational area (mainly distance learning), when users interact with the system in a low bit rate operational environment.

The simulations have assessed "learning" and "adaptability" behaviour of the proposed layer in different and variable home connections when a learning task is performed. The correctness of Perceived Performance Model (PPM) suggestions, access time of the learning process and quantity of transmitted data were analysed. The results show that the QoE layer significantly improves the performance in terms of the access time of the learning process with a reduction in the quantity of data sent by using image compression and/or elimination. A visual quality assessment confirmed that this image quality reduction

does not significantly affect the viewers' perceived quality that was close to "good" perceptual level.

For qualitative evaluation the QoE layer has been deployed on the open-source AHA! system. The goal of this evaluation was to compare the learning outcome, system usability and user satisfaction when AHA! and QoE-aware AHA systems were used. The assessment was performed in terms of learner achievement, learning performance and usability assessment. The results indicate that QoE-aware AHA system did not affect the learning outcome (the students have similar-learning achievements) but the learning performance was improved in terms of study time. Most significantly, QoE-aware AHA provides an important improvement in system usability as indicated by users' opinion about their satisfaction related to QoE.

List of Publications

Cristina Hava Muntean, Jennifer McManis, “*Adaptive E-learning Systems: Evaluation Issues*”, Transactions on Automatic Control and Computer Science Journal, Vol. 49 (63), pp. 193-198, ISSN 1224-600X, 2004

Cristina Hava Muntean, Jennifer McManis, “*End-User Quality of Experience Layer for Adaptive Hypermedia Systems*”, 3rd International Conference on Adaptive Hypermedia and Adaptive Web-based systems, Workshop on Individual Differences in Adaptive Hypermedia, Eindhoven, The Netherlands, pp. 87-96, 2004

Cristina Hava Muntean, Jennifer McManis, “*QoS AHA: A Performance Oriented Learning System*”, AACE World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA 2004), Lugano, Switzerland, 2004

Cristina Hava Muntean, Jennifer McManis, “*A QoS-Aware Adaptive Web-based System*”, IEEE International Conference on Communications (ICC 2004) Paris, France, 2004

Cristina Hava Muntean, Jennifer McManis, John Murphy, Liam Murphy, “*A Client's Perception-Based Adaptive Web Server Application*”, W3C International World Wide Web Conference, Poster Session, Honolulu, Hawaii, USA, 2002

Cristina Hava Muntean, Jennifer McManis, John Murphy, “*The Influence of Web Page Images on the Performance of Web Servers*”, IEEE International Conference on Networking (ICN'01), Colmar, France, Lecture Notes in Computer Science 2093, Springer-Verlag, pp. 821-828, 2001

Cristina Hava Muntean, Jennifer McManis, John Murphy, Liam Murphy, “*A New Dynamic Web Server*”, IEEE International Conference on Telecommunication (ICT'01), Bucharest, Romania, pp. 505-509, 2001

Cristina Hava Muntean, Jennifer McManis, John Murphy, Liam Murphy, “*A Client – Orientated Dynamic Web Server*”, IEE 17th UK Teletraffic Symposium, Dublin, Ireland, pp. 14/1-14/6, 2001

Contents

ACKNOWLEDGEMENTS	I
ABSTRACT	III
LIST OF PUBLICATIONS	V
TABLE OF CONTENTS.....	1
LIST OF FIGURE CAPTIONS	7
LIST OF TABLE CAPTIONS	11
1 INTRODUCTION	14
1.1 INTRODUCTION	14
1.2 MOTIVATION	16
1.3 PROBLEM AND GOAL.....	18
1.4 SOLUTION	19
1.5 CONTRIBUTIONS	20
1.6 OUTLINE OF THE THESIS.....	21
1.7 CHAPTER SUMMARY	22
2 ADAPTIVE HYPERMEDIA BACKGROUND	23
2.1 CHAPTER INTRODUCTION.....	23
2.2 HYPertext AND HYPERMEDIA.....	24
2.2.1 A Brief History of Hypertext and Hypermedia.....	24
2.2.2 Definitions.....	25
2.3 ADAPTIVE HYPERMEDIA	26
2.3.1 A Brief History of Adaptive Hypermedia	27
2.3.2 Definitions.....	28
2.3.3 Objectives and Benefits of Adaptive Hypermedia Systems.....	28
2.4 METHODS AND TECHNIQUES USED FOR ADAPTATION	29
2.4.1 Adaptive Presentation (Content-Level Adaptation).....	30
2.4.1.1 Adaptive Presentation Methods	31

2.4.1.2 Adaptive Presentation Techniques.....	32
2.4.2 Adaptive Navigation Support (Link-Level Adaptation)	33
2.4.2.1 Adaptive Navigation Support Methods.....	34
2.4.2.2 Adaptive Navigation Support Techniques	35
2.5 REFERENCE MODELS FOR HYPERTEXT/ HYPERMEDIA SYSTEMS	37
2.5.1 HAM Model.....	37
2.5.2 Dexter Hypertext Reference Model	38
2.5.3 Adaptive Hypermedia Application Model (AHAM)	39
2.5.4 Munich Reference Model.....	41
2.5.5 LAOS Model.....	43
2.6 USER CHARACTERISTICS ANALYSED BY ADAPTIVE HYPERMEDIA	44
2.6.1 User Data (User Characteristics).....	44
2.6.1.1 Demographic Data	44
2.6.1.2 User's Knowledge.....	45
2.6.1.3 User's Goals.....	45
2.6.1.4 Background and Experience	45
2.6.1.5 User's Skills and Capabilities	46
2.6.1.6 User's Interest.....	46
2.6.1.7 User's Preferences	46
2.6.1.8 User's Individual Traits	46
2.6.2 Usage Data	47
2.6.3 Environment Data	47
2.7 USER MODELS AND USER MODELING.....	48
2.7.1 Information Acquisition.....	49
2.7.2 User Modelling Methods	50
2.8 APPLICATION AREAS FOR ADAPTIVE HYPERMEDIA SYSTEMS	52
2.8.1 Educational Hypermedia.....	52
2.8.1.1 ELM-ART	53
2.8.1.2 INTERBOOK	54
2.8.1.3 AHA!	55
2.8.1.4 INSPIRE	55
2.8.1.5 JointZone	57
2.8.2 On-Line Information	57
2.8.3 On-Line Help Systems	59
2.8.4 Information Retrieval Hypermedia	59

2.8.5 Institutional Information	60
2.8.6 Personalized Views	61
2.9 CHAPTER SUMMARY	62
3 WEB QUALITY OF SERVICE BACKGROUND	63
3.1 CHAPTER INTRODUCTION.....	63
3.2 QUALITY OF SERVICE (QoS) DEFINITION	63
3.3 SOLUTIONS FOR IMPROVING END-TO-END WEB PERFORMANCE	65
3.3.1 Server - Side Performance Solutions	65
3.3.1.1 Reducing Server Overload.....	65
3.3.1.2 Web Servers' Replication	66
3.3.1.3 Server - Side Caching	67
3.3.2 Infrastructure Based Performance Solutions.....	68
3.3.2.1 Intelligent Web Switching	68
3.3.2.2 Intelligent Routers.....	69
3.3.2.3 Cache Appliances.....	69
3.3.3 Client - Side Performance Solutions	69
3.3.3.1 Client - Side Caching.....	70
3.3.3.2 Client - Side Pre-Fetching.....	70
3.4 QUALITY OF EXPERIENCE (QOE)	71
3.4.1 QoE in World Wide Web.....	72
3.5 CHAPTER SUMMARY	75
4 QUALITY OF EXPERIENCE LAYER FOR ADAPTIVE HYPERMEDIA SYSTEMS	77
4.1 CHAPTER INTRODUCTION.....	77
4.2 AHAM MODEL	78
4.2.1 Domain Model	79
4.2.1.1 Concepts.....	79
4.2.1.2 Concept Relationships	80
4.2.2 User Model.....	81
4.2.3 Adaptation Model	82
4.3 QUALITY OF EXPERIENCE (QOE) LAYER	82
4.3.1 Perceived Performance Model	85
4.3.1.1 Stereotype - Based Structure.....	86
4.3.1.2 Determination of Probabilistic Values Associated with Linguistic Terms .	88
4.3.1.3 User Classification.....	92

4.3.1.4 Suggestion Determination.....	94
4.3.1.5 Initialisation Phase.....	95
4.3.1.6 Update Phase.....	95
4.3.2 Illustrative Example of the Perceived Performance Model	98
4.3.3 Performance Monitor	107
4.3.4 Extended Adaptation Model	110
4.3.4.1 Adaptation Algorithm.....	111
4.4 CHAPTER SUMMARY	117
5 QOE LAYER VALIDATION AND ANALYSIS	119
5.1 CHAPTER INTRODUCTION.....	119
5.2 QOE LAYER VALIDATION AND ANALYSIS STRATEGY	120
5.3 OBJECTIVE OF THE SIMULATIONS	121
5.4 NETWORK SIMULATOR VERSION 2 (NS-2)	122
5.5 SIMULATION TESTS	123
5.5.1 Set-up conditions.....	123
5.5.1.1 Scenario 1: Simulation of a Browsing Session to a Virtual Web Site	124
5.5.1.2 Scenario 2: Simulation of a Learning Session on the AHA! Tutorial.....	125
5.5.2 QoE Layer Assessment in Constant Low Bit Operational Environment	125
5.5.2.1 Simulation Results for Scenario 1.....	126
5.5.2.1.1 Connection Type 1: up to 28 kbps	127
5.5.2.1.2 Connection Type 2: between 28 kbps and 42 kbps.....	130
5.5.2.1.3 Connection Type 3: between 42 kbps and 56 kbps.....	132
5.5.2.1.4 Connection Type 4: between 56 kbps and 64 kbps.....	134
5.5.2.1.5 Connection Type 5: between 64 kbps and 96 kbps.....	136
5.5.2.1.6 Connection Type 6: between 96 kbps and 128 kbps.....	138
5.5.2.1.7 Summary of the Results.....	139
5.5.2.2 Simulation Results for Scenario 1 with a Different Set of Web Pages....	143
5.5.2.3 Simulation Results for Scenario 2.....	145
5.5.2.4 Summary of the Results	149
5.5.3 QoE Layer Assessment in Variable Low Bit Operational Environment.....	150
5.5.3.1 Step-Wise Changeable Environment from 56 kbps to 96 kbps	151
5.5.3.2 Step-Wise Changeable Environment from 96 kbps to 56 kbps	155
5.5.3.3 Summary of the Results	159
5.6 CHAPTER SUMMARY	160
6 QOE LAYER EVALUATION IN EDUCATIONAL AREA	161

6.1 CHAPTER INTRODUCTION.....	161
6.2 EVALUATION ISSUES IN ADAPTIVE E-LEARNING SYSTEMS.....	162
6.2.1 Current Evaluation Approaches	162
6.2.1.1 System Evaluation “As a Whole”	163
6.2.1.2 Layer-Based System Evaluation	163
6.2.2 Evaluation Tests of Adaptive Educational Systems.....	165
6.2.2.1 Usability Evaluation Tests	165
6.2.2.2 Learner Achievement Evaluation.....	166
6.2.2.3 Learning Performance Evaluation	168
6.2.2.4 Assessment of Evaluation Results	168
6.3 EVALUATION OF THE QOE LAYER: QOECHA SYSTEM	169
6.3.1 AHA! System.....	170
6.3.1.1 AHA! Architecture.....	170
6.3.1.2 AHA! Functionality	171
6.3.1.3 Authoring Interface.....	172
6.3.1.3.1 Concept Editor Tool.....	173
6.3.1.3.2 Graph Author Tool.....	173
6.3.2 QoEAHA Evaluation Strategy	174
6.3.2.1 Objectives of the Evaluation Experiment	175
6.3.2.2 Experimental Setup.....	176
6.3.2.2.1 Scenario 1: Interactive Study Session.....	178
6.3.2.2.2 Scenario 2: Visual Quality Assessment	179
6.3.3 QoEAHA Evaluation Tests.....	180
6.3.3.1 Learner Achievement.....	180
6.3.3.2 Learning Performance.....	182
6.3.3.2.1 Learning Task	182
6.3.3.2.2 Search for a Term Task.....	189
6.3.3.3 Usability Assessment	190
6.3.3.4 Visual Quality Assessment	193
6.3.3.5 Correlation Between Individual Student Performance and Judgment on System Usability	194
6.3.3.6 Correlation Between Study Session Time and Individual Student Performance	196
6.3.4 Reflections on the Evaluation Results.....	197
6.4 CHAPTER SUMMARY	200

7 CONCLUSIONS AND FURTHER WORK	201
7.1 CHAPTER INTRODUCTION.....	201
7.2 MAIN ACHIEVEMENTS OF THE RESEARCH	201
7.3 CONCLUSIONS DRAWN FROM THE RESEARCH	203
7.3.1 Simulation Testing Conclusions	203
7.3.2 Qualitative Testing Conclusions	204
7.4 FUTURE WORK	206
7.4.1 Wider Range of End-User Operational Environments.....	206
7.4.2 Optimisation of the QoE Layer	206
7.4.3 QoE E-Learning System that Provides Personalised Multimedia Content	207
7.4.4 Performance Oriented Open Adaptive Hypermedia Systems	208
7.5 CHAPTER SUMMARY	209
APPENDIX A - STATISTICAL ANALYSIS	210
A.1 AVERAGE.....	210
A.2 VARIANCE	210
A.3 STANDARD DEVIATION	211
A.4 T-TEST.....	212
A.5 ANOVA TEST	214
A.6 F-TEST	215
A.7 Q-TEST	216
A.8 CORRELATION TEST	217
A.8.1 Pearson Correlation.....	218
A.8.2 Spearman Correlation.....	218
APPENDIX B – POST TEST EVALUATION.....	220
APPENDIX C – PRE TEST EVALUATION	224
APPENDIX D – USABILITY EVALUATION	226
REFERENCES	230

List of Figure Captions

Figure 1-1 The Future of the E-Learning Environment	17
Figure 2-1 Brusilovsky Taxonomy for Classification of Adaptive Hypermedia Systems	30
Figure 2-2 Layers of the Dexter Hypertext Reference Model.....	39
Figure 2-3 Adaptive Hypermedia Application Model.....	40
Figure 2-4 UML-based Munich Reference Model [39]	41
Figure 2-5 LAOS - Five LayerBased Model [46]	43
Figure 4-1 Adaptive Hypermedia Application Model.....	79
Figure 4-2 Concept Hierarchy from Domain Model.....	80
Figure 4-3 A Generic Architecture for Adaptive Hypermedia Systems	83
Figure 4-4 QoE-aware Adaptive Hypermedia System Architecture	84
Figure 4-5 Poisson Distribution for Different Mean Values “u”.....	89
Figure 4-6 Poisson Distribution for Five Stereotypes.....	91
Figure 4-7 Probabilistic Values associated with q=5 Linguistic Terms of a Feature F_i from the Stereotype T3	92
Figure 4-8 PPM Behaviour for Two Users	106
Figure 4-9 HTTP Client-Server Communication Process.....	107
Figure 4-10 Adaptation Algorithm.....	112
Figure 5-1 NS-2 Simulation Topology.....	123

Figure 5-2 Access Time per Page and Web Page Size Reduction for a Browsing Session Over a 28 kbps Connection Speed When Content Constraints Suggested by the QoE Layer were Applied.....	128
Figure 5-3 Access Time per Page and Total Web Page Size Reduction for a Browsing Session Over a 42 kbps Connection When Content Constraints Suggested by the QoE Layer were Applied.....	130
Figure 5-4 Access Time per Page and Total Web Page Size Reduction for a Browsing Session Over a 56 kbps Connection When Content Constraints Suggested by the QoE Layer were Applied.....	132
Figure 5-5 Access Time per Page and Total Web Page Size Reduction for a Browsing Session Over a 64 kbps Connection When Content Constraints Suggested by the QoE Layer were Applied.....	134
Figure 5-6 Access Time per Page and Total Web Page Size Reduction for a Browsing Session Over a 96 kbps Connection When Content Constraints Suggested by the QoE Layer were Applied.....	136
Figure 5-7 Access Time per Page and Total Web Page Size Reduction for a Browsing Session Over a 128 kbps Connection Speed for the Case When Content Constraints Suggested by the QoE Layer were Applied	138
Figure 5-8 Analysis on the Execution Time of a Learning Task and the Total Quantity of Data Transferred during the Learning Session Versus Different Connection Types when the QoE Suggestions Were Taken into Consideration Before Delivering a Web Page	140
Figure 5-9 PPM Suggestions on the Average Size of the Based-Web page and of the Embedded Objects as well as Average Total Size of the Delivered Web Page for a Learning Session over Different Network Conditions	141
Figure 5-10 PPM Suggestions on the Optimal Number of Embedded Objects in a Web Page Transmitted for a Learning Session over Different Network Conditions	141
Figure 5-11 Access Time per Page and Web Page Size Reduction for a Browsing Session Over a 28 kbps Connection Speed When Content Constraints Suggested by the QoE Layer were Applied.....	144

Figure 5-12 Analysis on the Execution Time of a Learning Task and the Total Quantity of Data Transferred during the Learning Session on the AHA! Tutorial Versus Different Connection Types when the QoE Suggestions Were Taken Into Consideration Before Delivering a Web Page.....	147
Figure 5-13 Access Time and Total Size Reduction for Page 3 from the Chapter One AHA! Tutorial When the Web Page was Delivered Over Different type of low bit operational environments	147
Figure 5-14 PPM Suggestions on the Average Size of the Based-Web page and of the Embedded Objects as well as Average Total Size of the Delivered Web Page for a Learning Session on the AHA! Tutorial Over Different Network Conditions.....	148
Figure 5-15 PPM Suggestions on the Optimal Number of Embedded Objects in a Web Page Transmitted for a Learning Session on the AHA! Tutorial over Different Network Conditions	149
Figure 5-16 Network Properties During a Simulated Learning Session that Involves Ten Web Pages	151
Figure 5-17 Access Time per Page and Web Page Size Reduction for a Study Session that Involves Ten Web Pages in Changeable Network Environment from 56 kbps to 96 kbps	152
Figure 5-18 PPM Suggestions on the Optimal Size of the Based-Web Page and of the Embedded Objects as well as Total Size of the Web Page, during a Ten Page Learning Session, with the Step-Wise Improvement of the Network Conditions	154
Figure 5-19 PPM Suggestions on the Optimal Number of Embedded Objects during a Ten Page Learning Session with the Step-Wise Improvement of the Network Conditions	155
Figure 5-20 Network Properties During a Simulated Learning Session that Involves Ten Web Pages	156
Figure 5-21 Access Time per Page and Web Page Size Reduction for a Study Session that Involves Ten Web Pages in Step - Wise Changeable Network Environment from 96 kbps to 56 kbps.....	156

Figure 5-22 PPM Suggestions of the Optimal Size of the Based-Web Page and of the Embedded Objects as well as Total Size of the Web Page, during a Ten Pages Learning Session, with the Step – Wise Degradation in the Network Conditions	158
Figure 5-23 PPM Suggestions of the Optimal Number of Embedded Objects during a Ten Pages Learning Session with the Step – Wise Degradation in the Network Conditions	159
Figure 6-1 Laboratory Network Configuration for the Perceptual Testing	177
Figure 6-2 Distribution of the Study Time Performed by the Students Involved in a Learning Task	183
Figure 6-3 Students Distribution Based on Number of Visits to Page 1	188
Figure 6-4 Students Distribution Based on Number of Visits to Page 2	189
Figure 6-5 Search for a Term Average Time	190
Figure 6-6 Usability Evaluation Results on Questions that Assessed the End-User QoE....	191
Figure 6-7 Comparative Presentation of the Answers on Usability Questionare.....	192
Figure 6-8 Correlation Graph Between Students Performance and Usability When Using QoEAHA	195
Figure 6-9 Correlation Graph Between Students Performance and Usability When Using AHA!	195
Figure 6-10 Correlation Graph Between Students Study Session Time and Students Evaluation Marks Using QoEAHA	196
Figure 6-11 Correlation Graph Between Students Study Session Time and Students Evaluation Marks When Using AHA!	197

List of Table Captions

Table 4-1 Group of Features for a Stereotype	87
Table 4-2 Group of Suggestions for a Stereotype	88
Table 4-3 Probabilistic Values Of The Linguistic Terms Associated With Feature Fi From The Stereotype T3	92
Table 4-4 Group of Features for Stereotype "Bad"	99
Table 4-5 Group of Suggestions for Stereotype "Bad"	99
Table 4-6 Group of Features for Stereotype "Poor"	100
Table 4-7 Group of Suggestions for Stereotype "Poor"	100
Table 4-8 Group of Features for Stereotype "Normal"	100
Table 4-9 Group of Suggestions for Stereotype "Normal"	100
Table 4-10 Group of Features for Stereotype "Good"	101
Table 4-11 Group of Suggestions for Stereotype "Good"	101
Table 4-12 Group of Features for Stereotype "Excellent"	101
Table 4-13 Group of Suggestions for Stereotype "Excellent"	101
Table 4-14 Final Group of Suggestions	102
Table 4-15 Group of Features for Stereotype "Bad"	104
Table 4-16 Group of Features for Stereotype "Poor"	104
Table 4-17 Group of Features for Stereotype "Normal"	105

Table 4-18 Group of Features for Stereotype "Good".....	105
Table 4-19 Group of Features for Stereotype "Excellent"	105
Table 4-20 Final Group of Suggestions	106
Table 4-21Size of Images and User Non-Interest in the Images.....	115
Table 5-1 Parameters of Distribution Function of the Web Page Characteristics.....	125
Table 5-2 Characteristics Of The Generated Web Pages Used For The Study Session.....	127
Table 5-3 Web Content Related Constraints Suggested by the PPM and User Classification into Stereotypes After Each Requested Web Page Over a 28 kbps Connection	129
Table 5-4 Web Content Related Constraints Suggested by the PPM and User Classification into Stereotypes After Each Requested Web Page over a 42 kbps Connection Throughput.....	131
Table 5-5 Web Content Related Constraints Suggested by the PPM and User Classification into Stereotypes After Each Requested Web Page over a 56 kbps Connection.....	133
Table 5-6 Web Content Related Constraints Suggested by the PPM and User Classification into Stereotypes After Each Requested Web Page over a 64 kbps Connection.....	135
Table 5-7 Web Content Related Constraints Suggested by the PPM and User Classification Into Stereotypes After Each Requested Web Page over a 96 kbps Connection.....	137
Table 5-8 Web Content Related constraints Suggested by the PPM and User Classification into Stereotypes After Each Requested Web Page over a 128 kbps Connection	139
Table 5-9 Average Degree of Match Between User's Characteristics and Each Stereotype Class Determined by the PPM Model for a Learning Session Over Different Types of Non-Changeable Network Environments.....	142
Table 5-10 Characteristics Of The New Set of Ten Web Pages that Were Generated	143
Table 5-11 Web Content Related Constraints Suggested by the PPM and User Classification into Stereotypes After Each Requested Web Page Over a 28 kbps Connection	145

Table 5-12 Web Content Related Constraints Suggested by the PPM and User Classification Into Stereotypes After Each Requested Web Page over a Changeable Network Environment from 56 kbps to 96 kbps During a Learning Session.....	153
Table 5-13 Web Content Related Constraints Suggested by the PPM and User Classification Into Stereotypes After Each Requested Web Page over a Step – Wise Changeable Network Environment from 96 kbps to 56 kbps During a Learning Session.....	157
Table 6-1 Pre-Test Results	180
Table 6-2 Post-Test Results.....	181
Table 6-3 Post-Test Results for Questions with Answers Presented in the Embedded Images	182
Table 6-4 Study Time for the Learning Taks when AHA! and QoEAHA were Used.....	183
Table 6-5 Percentage of Students that Have Succeeded to Learn for Different Study Period Times.....	184
Table 6-6 Data Analysis for the Study Time Spent on Page 1 and Page 2	185
Table 6-7 T-Test Analysis on Study Time results for Page 1 and Page 2.....	185
Table 6-8 Number of Accesses to a Page Performed by a Person	186
Table 6-9 T-Test Analysis on Number of Accesses Per Person to Page 1 and Page 2	186
Table 6-10 F-Test Analysis on the variance of Group 1 and Group 2 Number of Accesses Samples.....	187
Table 6-11 Perceptual Quality Grading for the AHA and QoEAHA Systems	193
Table 0-1 Spearman Rank Order Conversion	219

Chapter I

Introduction

1.1 Introduction

The World Wide Web (WWW or Web) is one of the most important Internet services, and has been largely responsible for the increasing popularity of the Internet in recent years. The WWW offers its users access to a range of information-based services such as e-commerce, e-learning, news and entertainment, and is continually expanding in both size and content. There were recorded more than 60 million Web sites in March 2005 [1], compared to only 600 Web sites in 1993 [2]. The number of Internet users has grown from 100 million in 2000 to 930 million in 2004 and will top 1 billion in mid 2005 [3]. The growth of Internet users will continue but much of future Internet users growth will come from populous countries such as China, India, Brazil, Russia and Indonesia [3].

With the increase in the number of Web sites, in the number of persons who have access to the Internet and the diversification of their need for services, the presentation of services offered by Web servers has become increasingly sophisticated. Initially, Web pages were text-based with file sizes on the order of kilobytes. Nowadays, Web pages have become much more complex. Static and animated pictures, dynamically generated pages, and multimedia components have been included, increasing the typical total size of Web pages to hundreds of kilobytes. By doing so, Web pages have become more attractive for their clients, but also more resource-intensive to send and retrieve which in turn affects the user perception of quality of these services. Some common factors that influence user perceptual quality of Web sites include: *quality of information available* (e.g. whether the information desired by a user is available and of interest; quality of retrieved images, sound, video, and multimedia information) and *quality of delivery* (e.g. speed of connection establishment; download time).

There has also been a realization that users may have different characteristics, they may seek different information, or (depending on their capabilities and tasks) may have different expectations of delivery quality. The fact that users can be differentiated based on their characteristics has led to attempts to tailor Web service for specific clients or client categories. Two broad research approaches have been taken:

- Web Quality of Service
- Personalisation

Web Quality of Service (QoS) studies the components involved in transmission between Web users and service providers. These components might include: personal computers, Web servers, network, cables, routers, transmission types, etc. A major goal is to discover ways of improving delivery. One class of proposed solutions involves serving different content based on performance characteristics. Roughly speaking the philosophy behind these approaches is that the volume of data transmitted is responsible for performance problems – either through overloading the server, congestion in the network, or inability of the user device to process what is sent. The solution to this problem lies in reducing the volume sent, usually performed dynamically in response to observed conditions. Details about Web QoS related research are presented in Chapter 3.

Personalisation seeks to identify users and serve adapted content based on user characteristics. The content served can be tailored according to what the user wants, or potentially according to what information a vendor wishes certain users to access. There are different basic types of personalisation depending on the amount of control a user has over the adaptation process. The systems that allow the user to control the initialisation, selection and result of the adaptation by introducing a user profile are called *adaptable hypermedia systems*. These systems do not change the profile unless the user explicitly updates it. *Adaptive Hypermedia (AH)* research seek to develop systems capable of automatically identifying users and delivering differentiated content based on the user's characteristics and by observing the user's browsing behavior or the behavior of other users. Most user characteristics considered by the Adaptive Hypermedia Systems (AHS) include user knowledge, user skills and capabilities, user goals, user interests and preferences. Chapter 2 includes a more detailed presentation of related works in the area of Personalisation.

1.2 Motivation

Need For A More Performance Oriented Adaptive Hypermedia System

Analysing the research work performed in both *Web QoS* and *Personalisation* areas, it can be noticed that both approaches seek to *optimise user satisfaction* with Web sites from two different points of view:

- by improving the user perceived performance of the Web service
- by tailoring Web content according to user interests.

By concentrating strictly on either performance or content issues, a realistic view of user satisfaction is not being considered.

Specifically, *Web QoS* does not address the issue of how and what content modification solutions should be performed (e.g. what Web page components should be reduced in size or eliminated in order to minimise the negative impact on user's satisfaction related to the content). Meanwhile, *personalisation* places very little emphasis on performance issues. Although AH research has demonstrated the benefit of providing personalized content and navigation support for specific users or users categories based on the users inferred interests, little consideration is given to QoS-oriented factors (e.g. size of the Web page, access time of the requested page, quality of the delivered content), or how the impact on QoS will affect user satisfaction with the system (end-user Quality of Experience). It is not always feasible to deliver an unlimited amount of material given the variety of devices and network environments that might be used to access the material.

Therefore it seems significant to be able to merge these two directions and propose a solution that combines WebQoS and Personalisation benefits.

High Demand for E-learning Systems

Technological advancements have opened up new possibilities in the field of e-learning. Potentially, e-learning allows greater access to a more diverse student population and also enables lifelong learning – allowing people to acquire new skills, to develop and improve existing ones, and to update knowledge as the technology advances. E-learning may also be more flexible, efficient, faster, allows for a great deal of interactivity as well as being cost effective in comparison with conventional training. Currently, the Web-based e-learning market is a growing one worldwide both in the educational and corporate training

sector. It is estimated for example that distance education will account for 50 % of all post-secondary learning by 2010 and Internet delivered courses are playing a more central role in distance education or in supporting conventional delivery methods [4].

As the e-Learning area matures, there are growing expectations of the quality of the provided content and of its delivery performance. Thus, end-users of educational and training services expect not only high-quality and efficient educational material but also a perfect integration of this material with the day-to-day operational environment and network framework. Given the potential mobility of e-learners, they may move from a low bit rate home environment to a high quality school or work connection, and even onto public transportation with a connection of variable quality (Figure 1-1). In this context it may not even be possible a priori to assess the amount of material the student can access at any given time. Therefore there is a need for adaptive e-learning systems that are able to track both students' requirements and their current connectivity characteristics, and to deliver adaptive educational material that would best fit both students needs and delivery conditions.



Figure 1-1 The Future of the E-Learning Environment

Trends indicate an increasing interest in personalisation of e-learning content, with two thirds of developed adaptive Web-based hypermedia systems being applied in the educational area [5]. The goal of Adaptive Educational Hypermedia Systems (AEHS) is to

capture and analyse user related features such as knowledge level, skills, capabilities, interests and preferences and to personalise the educational material to the student's individual learning requirements in order to optimise student learning experience with their online course material.

1.3 Problem and Goal

The QoS-unaware approach taken by AEHS is perhaps unsuited to a general learning environment where mobility of e-learners, device properties and variations of the delivery network conditions are key factors that influence the learning process. Although the AEHS has selected material that would best match user's interest, it is not always feasible to deliver this material given the variety of devices and connections that might be used to access the material. Inappropriate material from the performance point of view may cause frustration for the end-user and may affect the learning performance and learning outcome for the student.

In this context it is significant to highlight a new problem faced by network-based education over the Internet: providing a good level of end-user perceived Quality of Service (QoS) or Quality of Experience (QoE) during the learning process. Therefore, the AEHS goal of delivering the best content to a student should be combined with providing the student with an adequate QoE, representing the end-user's satisfaction with end-to-end QoS.

Therefore for best results, an AHS and especially an AEHS should also take into consideration end-user QoE characteristics when the user profile is built and regularly monitor in real-time any change in the system that might indicate or even produce variations of QoE. These include changes in the user's operational environment and also modifications of user behaviour, which might possibly indicate dissatisfaction with service (such as an "abort action").

The research presented in this thesis *bridges the gap between the AH research and Web QoS* by enhancing the AHS with QoE awareness. A QoE enhanced AHS will provide adaptive Web content not only based on different user's features, but also on the current delivery environment conditions that exist between the user and the Web service during a navigation session and their impact on user's experience with the system.

This thesis *proposes, designs, and tests a novel Quality of Experience (QoE) Layer for Adaptive Hypermedia Systems (AHS)*. The QoE layer is an extension to classic AHS that

complements the advantages brought by personalization with performance analysis and decision making capabilities in order to improve end-user QoE. Its goal is to improve end-user QoE in different operational environments by combining performance-related content adaptations with the Web-based content personalisation typically provided by AHSs. The Web content considered in the testing consists of text and static images.

1.4 Solution

The proposed QoE layer takes into account multiple factors that affect Quality of Experience, such as Web content characteristics and network connectivity and provides server-side strategies for tailoring Web content in order to improve end-user's QoE. A variety of Web-based performance metrics are measured in real-time in order to learn about Web user's operational environment characteristics and changes in network connectivity that may appear, and the impact of these characteristics on content delivery and hence on the user's QoE is analysed.

The QoE layer also estimates users satisfaction related to their Web experience. For this estimation to be accurate, some Web-based performance metrics that have a substantial impact on the QoE, user behavior metrics and users' subjective opinions about their QoE were mapped into the *Perceived Performance Model (PPM)*. These metrics are measured and monitored in real time by a Performance Monitor module. PPM uses stereotype-based technique and statistical formulas to make correlations between these metrics values and Web page characteristics. Based on these correlations, PPM suggests optimal characteristics for Web pages such as the number of embedded objects, the dimension of the base Web page and the total dimension of the embedded components. If applied, these suggestions increase end-user QoE in a given operational environment.

Based on the PPM suggestions the content of a personalised Web page generated by a classic AHS is modified in order to improve the end-user QoE by minimising the negative impact on user satisfaction related to the delivered information. *An adaptation algorithm* that determines and applies the correct transformations on a Web page in order to match the PPM suggestions was proposed. The performed adaptations consist of elimination and/or property modification of some information fragments from the Web page.

The decision about which Web page fragments are affected is based on the information provided by the User Model (UM). The UM is part of the classic AHS and builds a user profile that represents the user knowledge, goals, preferences, navigation

history and other important user characteristics. Therefore the UM provides information related to the strength of user interest on Web page fragments. Hence, the proposed adaptation algorithm performs changes starting with information fragments the user is the least interested in. These changes are applied until the PPM suggestions are matched.

The proposed QoE layer has been *designed, modeled, implemented and tested through simulations and qualitative evaluation* in order to verify and validate its performance and to *highlight the benefits of the novel QoE-Layer*. Due to the high demand for e-learning, educational area was chosen for exemplification and testing purpose.

Therefore QoE layer was deployed on the AHA! 2.0 application, an open-source AHS, and tested in the educational area. Students that interact with a QoE-aware AHA in a home-like low bit rate operational environment through PC desktop devices were considered. It was very important to test the benefits brought by this layer for e-learning in terms of learning outcome and learning performance and user satisfaction with the adaptive e-learning system. In this context the *QoE-aware AHA was compared with the original AHA! system*.

The objectives of the QoE layer evaluation are the following:

- to assess the improvements brought by the QoE layer in terms of end-user perceived performance
- to investigate the impact of the QoE layer on student learning performance
- to measure the usability and effectiveness of an QoE-aware AEHS in comparison with a classic AEHS
- to assess the user opinion in relation to the QoE-aware AEHS

1.5 Contributions

This section highlights the contributions this thesis makes to research in the area of AHS.

- This research explores a new dimension of individual differences between Web users, that focuses on end-user Quality of Experience (QoE) which should be **taken into consideration in the personalization process** performed by AIIS. QoE is directly influenced by the operational environment through which the user

interacts with the AHS (bandwidth, delay, loss, device capabilities, etc) and by the subjective assessment of the user's perceived performance.

- This thesis proposes a new QoE-based enhancement for AHS as a means of increasing end-user QoE. It analyses key factors that influence user QoE such as network connectivity characteristics, wide range of Web components and user's behaviour. A novel Perceived Performance Model (PPM) makes correlations between the values of these factors and Web page characteristics using stereotype-based modeling and statistical formulas in order to suggest the optimal characteristics for Web pages that would provide the best QoE in given conditions.
- This thesis describes a novel adaptation algorithm that determines and applies performance-based adaptations to a Web page in order to match the PPM suggestions. The adaptations consist of elimination and/or modification of the properties of some information fragments from the Web page based on the strength of user interest in Web page fragments.
- The proposed QoE enhancement for AHS was comprehensively tested for educational applications. Tests indicate the usefulness of the system in distance learning environment, where users typically operate in a low bit-rate operational environment (e.g. home connection). In such environments, the tests showed that learning outcome was not affected (the students have similar-learning achievements) while the learning performance was improved in terms of execution of a learning task, access time per page and number of re-visits to a page. These improvements were due to the fact that the educational material was delivered faster and the students were constantly focused on their task. Long periods of waiting for a required page annoys people and disturbs their concentration on the task performed. Most significantly, QoE-aware AHS provides an important improvement in system usability as indicated by users' opinion about their satisfaction related to QoE.

1.6 Outline of the Thesis

This thesis is organised in seven chapters that present the research performed, the related works, the proposed solution and its testing, and conclusions drawn as well as possible future research directions.

This first chapter has mainly presented the motivation for the research, the problem to be solved, the goal, the solution and the thesis contributions.

The second and third chapters present related works in the area of Adaptive Hypermedia and Web QoS respectively.

The fourth chapter focuses on a detailed presentation of the proposed server-side QoE Layer extension for AHS and includes a description of a generic AHS architecture that incorporates the QoE layer and a detailed presentation of the functionality of the QoE layer components and the principle of the performance-based adaptation algorithm.

The fifth chapter describes a large set of simulation tests that analyse and validate the performance improvements due to the QoE layer in different low bit rate network environments.

The sixth chapter presents qualitative test results when the QoE layer was deployed on the AHA! system, an open-source AHS tested in the educational area. The goal of these experimental tests was to investigate the usability and effectiveness of the proposed QoE layer for online learning by analyzing the students' learning process, learning outcome and their opinion about the system when the QoE-aware AHA was used.

The seventh chapter draws some conclusions and highlights possible future work directions. The list of references and the appendixes end this thesis.

1.7 Chapter Summary

This chapter starts with a presentation of two main research directions in the area of World Wide Web (Web QoS and Personalisation). These directions seek to optimise end-user's experience with Web services that have become increasingly sophisticated, complex and resource-intensive. It also presents their drawbacks and raises problems that are still to be addressed. E-learning is one of the application areas that has benefited from personalisation of educational material, individualisation of the learning process, high levels of learner control and highly flexible education option, brought by research on Web personalisation. With recent technological advancements new possibilities in the field of e-learning were opened such as mobility of e-learners, variety of access devices, etc. In this context this chapter introduces the motivation for the research presented in this PhD thesis, the problem to be solved and the research goals and it ends with a description of the proposed solution and of the significant contributions made.

Chapter II

Adaptive Hypermedia

Background

2.1 Chapter Introduction

This chapter focuses on the presentation of related research work in Adaptive Hypermedia. Section 2.2 centers on the field of hypertext and hypermedia, presents a brief history of the research in this area and provides different definitions proposed in the literature for the terms: hypertext and hypermedia. The following sections introduce Adaptive Hypermedia and provide information on different aspects of this field. Thus, section 2.3 gives a brief introduction to Adaptive Hypermedia, and details on techniques and methods used for the adaptation of both content and navigation support are provided in section 2.4.

As this work focuses on proposing, implementing and testing an extension to the classic Adaptive Hypermedia Systems (AHS), section 2.5 looks at different models and frameworks proposed for Hypertext/Hypermedia systems. Of major interest are those models that were extended in order to allow for description of new adaptive systems developed in the AH field.

Since the main goal of AH research is to find solutions for adapting the content and navigation support of the hypermedia according to some particular characteristics of the user, sections 2.6 and 2.7 center on different aspects of the user's characteristics that are taken into consideration by AHS. This user related information is captured by a User Model and makes use of different modeling methods described in section 2.7.

The last section of this chapter presents an overview of application-based AHS and the areas in which these systems were applied. The educational area has attracted huge

interest and the most important systems deployed in education are described in details while systems from the other areas are only briefly introduced.

2.2 Hypertext and Hypermedia

2.2.1 A Brief History of Hypertext and Hypermedia

Looking back in time into the history of hypermedia we can notice that hypermedia has evolved through multiple stages. The first stage started in 1945, with Vannevar Bush's visionary ideas [6] that describe an archive in which associative connections between documents are possible. These connections would allow quick access to documents' content. Vannevar Bush's ideas are seen as the first step toward the idea of hypertext [7].

Later on, in 1965, Ted Nelson coined the term "*hypertext*" in the context of the Xanadu [8] project. This project was part of the first generation of hypermedia systems. Since these systems were mainframe-based text-only, they were named *hypertext systems*. They were primarily designed for authoring purposes, they offered very limited navigational capabilities and they did not provide any mechanism to customise the content to user needs.

The beginning of the 80s brought a new stage in the evolution of hypermedia. Systems such as Intermedia [9], Symbolics Document Examiner [10], NoteCards [11] that represented the second generation of hypermedia systems and they made the transition from hypertext to hypermedia. These systems are similar to the first generation systems but additionally, they offer support for other forms of information such as graphics, sound, animation, video graphics, and advanced user interfaces. In the following years, as different media types became more and more accessible, hypertext systems evolved into *hypermedia systems*. Ted Nelson also introduced the term "*hypermedia*". Nowadays the two terms, hypertext and hypermedia are used interchangeably.

A new step forward in hypermedia evolution was made in 1994 by Tim Berners-Lee who introduced the concept of a simple hypertext client-server approach applied on a global scale, known today as World Wide Web (WWW), or more simply the Web. The main goal of the development of Web was to allow collaborators in remote sites to share their ideas and knowledge. Like any hypermedia system, the Web consists of documents, nodes or pages interconnected by a set of navigational hyperlinks (also called links). When a user reads a document, a set of links are provided to the user for redirection to new documents containing

related information. This associative relation between information is an essential component of all hypermedia systems [12].

In the last ten years, the Web's popularity increased exponentially and it became one of the most important Internet services. With the increase in the number of the Web sites and in the number of the persons who have access to the Internet, the WWW is continually expanding in both size and content. This expansion has raised some important issues for hypertext research such as: information overload and the "lost in hyperspace problem". In order to solve these problems, a hypermedia systems should guide users through the information space and offer them documents that contain relevant and useful data. This task requires understanding the users, discovering their goals, preferences and interests. Therefore the last stage in the evolution of hypermedia consists of the addition of adaptiveness, creating *adaptive hypermedia systems* that bring a certain level of intelligence to hypermedia systems by delivering the information appropriate to users needs. More details about these systems are presented in section 2.3.

2.2.2 Definitions

More detailed definitions for the terms of hypertext and hypermedia introduced in the previous paragraph, as the hypertext/hypermedia researchers proposed them are provided next.

Hypertext Definitions

The term "hypertext" was introduced in 1965 by the pioneer of hypertext, Ted Nelson who defined it in his self-published "Literary Machines" [13] as "*non-sequential writing*".

"I mean non-sequential writing – text that branches and allows choices to the reader,...this is a series of text chunks connected by links which offer the reader different pathways..." [14].

For the scope of this thesis a very general definition of the hypertext term as it is provided in a dictionary is presented next.

Definition 1 (hypertext): "A collection of documents (or "nodes") containing cross-references or "links" which, with the aid of an interactive browser program, allow the reader to move easily from one document to another" [15].

More definitions were provided in time by different researchers from the hypertext area.

Definition 2 (hypertext): "*a database that has active cross-references and allows the reader to "jump" to other parts of the database as desired*" [16].

Definition 3 (hypertext): "*Information is linked and cross-referenced in many different ways and is widely available to end-users*" [17].

Hypermedia Definitions

Hypermedia, a term derived from hypertext, extends the notion of the hypertext link to include links among any set of multimedia objects, including sound, motion video, and virtual reality. Apparently Ted Nelson was the first to use this term, too.

Bieber et al [18] has provided a definition of hypermedia and he also stated that the terms hypertext and hypermedia can be used interchangeably.

Definition 2 (hypermedia): "*A concept that encourages authors to structure information as an associative network of nodes and interrelating links*" [18].

Definition 3 (hypermedia systems): "*An interactive system that allows users to navigate a network of linked hypermedia objects*" [19].

2.3 Adaptive Hypermedia

Adaptive Hypermedia (AH) represents a relatively new research direction at the crossroads of hypertext/hypermedia and user modeling, within the area of user-adaptive systems [20]. An Adaptive Hypermedia System (AHS) enlarges the functionality of a hypermedia system by tailoring the content delivered to the users and improving the relationship with the users. Whereas traditional hypermedia systems present the same content and links to all users, adaptive hypermedia systems maintain a repository of knowledge about their users (called user model) and use this information in order to adapt the presentation and navigation structures of the hypermedia system. Therefore, each user **has an individual view** and navigational possibility when interacting with the hypermedia system. The repository of knowledge about users maintained by the system is used for

limiting the number of available links in a hypermedia system and the hyperspace accessible to the user. In this way, AHSs attempt to solve the problems of information overload and the “lost in hyperspace” problem mentioned in paragraph 2.2.1.

Adaptive hypermedia systems can be useful in any application area where the systems is expected to be used by people with different goals and knowledge and where the hyperspace is reasonable big [20]. More details about the application area of AHS are presented in section 2.8.

2.3.1 A Brief History of Adaptive Hypermedia

Adaptive hypermedia research can be tracked back to the early 1990s, when the two parent areas, Hypertext/Hypermedia and User Modeling achieved a level of maturity that would allow the exchange of ideas between them [5]. At that time, the hypermedia research community had identified the main problems of the hypermedia systems and they started to explore different solutions in order to adapt the output and behavior of the system to individual users. With the support of the user modeling research community they promoted adaptive hypermedia as an independent research direction in user modeling. Their work led to Adaptive Hypermedia Systems that combine ideas from each of these fields. By that time, several adaptive hypermedia systems had been built, making use of different novel adaptive techniques. Peter Brusilovsky gave an overview on adaptive hypermedia systems, methods and techniques used by these adaptive systems in 1996 [20]. A brief description on adaptation techniques and methods used by adaptive hypermedia systems will be provided in paragraph 2.4.

The year 1996 can be considered a turning point in adaptive hypermedia research [5]. Before 1996, a few isolated teams had performed research in this area and most of the developed systems were not Web-based. After 1996, adaptive hypermedia has gone through a period of rapid growth and adoption of Web technology. The main factors were the exponential increase in the use of the World Wide Web and the consolidation of the research in this field. Due to the large numbers of the adaptive hypermedia systems developed since 1996, Brusilovsky updated his view of the state of the art in adaptive hypermedia system in 2001 [5].

While at the beginning of 90s the focus of the adaptive hypermedia systems was student education, nowadays it has moved away, covering a wider range of topics such as

information systems, online help systems, personalized news and TV guide. More information on the applicability areas of these systems will be provided in section 2.8.

2.3.2 Definitions

Since this area of research is quite new and the concept of adaptive hypermedia systems has not been clearly defined yet, many definitions exist within the community. Peter Brusilovsky provided in his review on adaptive hypermedia the following definition:

Definition 1 (adaptive hypermedia systems): “*By adaptive hypermedia systems we mean all hypertext and hypermedia systems which reflect some features of the user in the user model and apply this model to adapt various visible aspects of the system to the user. In other words, the system should satisfy three criteria: it should be a hypertext or hypermedia system; it should have a user model; it should be able to adapt the hypermedia using this model.*”[20]

A clear distinction should be made between *adaptable* and *adaptive* hypermedia systems. In both systems the user has an important role and the main goal is to offer personalized content and navigation support. However there is a major difference.

Definition 2 (adaptable hypermedia systems): *A system that allows the user to control the initialization of the system and the selection of the adaptation. The user provides a profile, and the contents are delivered with respect to the selected profile.*

Therefore, *adaptable* hypermedia systems do not change the profile unless the user explicitly updates it. In contrast, *adaptive* hypermedia systems update the user profile automatically by observing the user’s behavior or the behavior of other users.

2.3.3 Objectives and Benefits of Adaptive Hypermedia Systems

While the objective of hypermedia systems is to make information easily accessible to the users, adaptive hypermedia systems improve this objective with a higher flexibility and comfort provided to each individual user, enhancing the user learning process and the user satisfaction with the systems. This goal is achieved by learning about the users while they interact with the systems and by tailoring the content and navigation support according to the users’ interests.

The main benefits of adaptive systems are that a wider group of users can use them as the system adapts in order to fit each user's interest and competence [20]. The systems are easier to use and motivate the users to continue to use them.

Another important benefit of these systems is that they reduce the risk for users of being lost in hyperspace by reducing the navigation space, eliminating links that are not of interest to the user and by offering some guidance and help to the users when needed [20].

Although these systems bring many benefits there are some problems noticed by different research teams. The main problem is to build the adaptive functionality. This is a complex, difficult, time-consuming and expansive job. Another problem is to build the user model and to determine the correct individual differences of the users.

2.4 Methods and Techniques Used For Adaptation

As it was mentioned in section 2.2, hypermedia consists of a collection of *documents* that are connected by *links*. Therefore, there are two aspects that can be adapted: the content and the links.

In Brusilovsky's reviews of the work performed in adaptive hypermedia [5, 20], a set of criteria was defined. A taxonomy (see Figure 2-1) was proposed in order to identify and classify AHS based on the methods and techniques used for the adaptation process. Two main levels of adaptation were defined depending on what could be adapted: **content-level adaptation** and **link-level adaptation**. As consequence, two different classes of hypermedia adaptations were created and labeled as: *Adaptive Presentation* and *Adaptive Navigation Support*. For each class different methods and techniques were identified. A *method* is defined as a notion of adaptation that can be presented at conceptual level. A *technique* is a solution to implement a specific method. Different techniques could implement the same method. A technique could also be used to implement more than one method.

In the following subsections, different methods and techniques for adaptive presentation and adaptive navigation support will be presented.

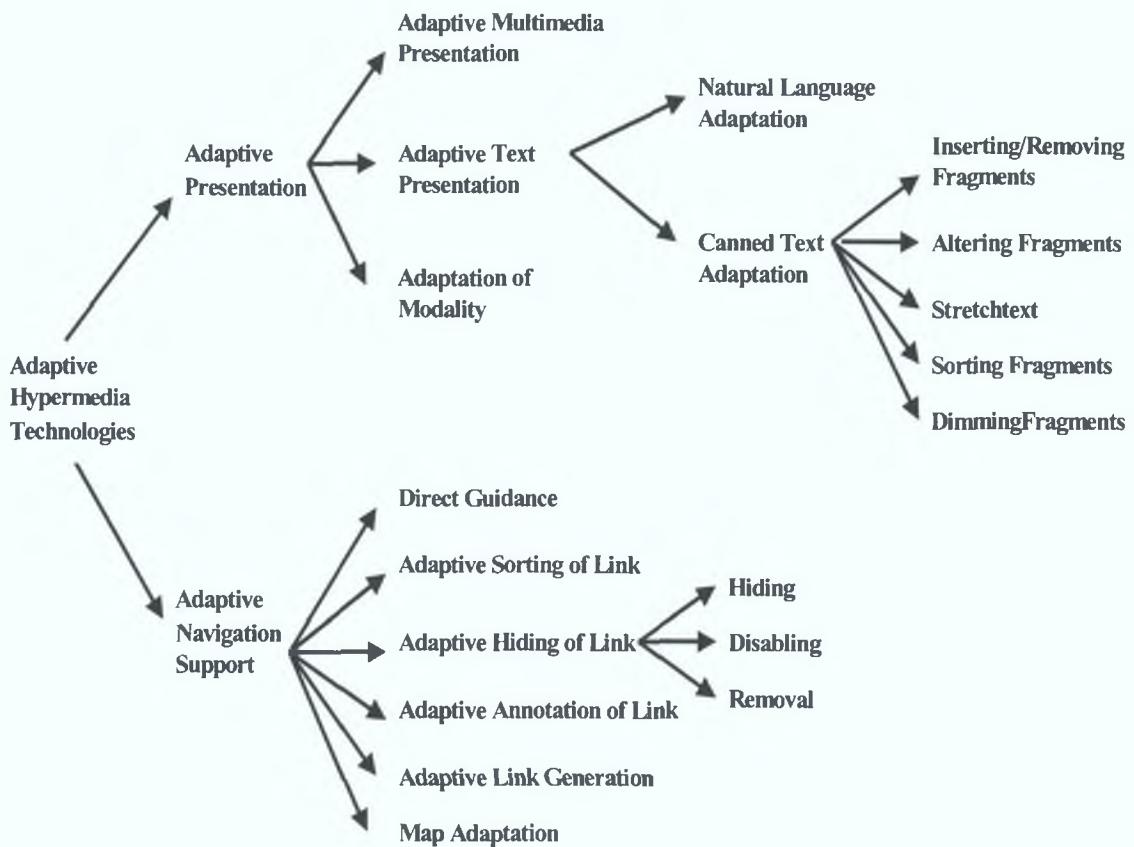


Figure 2-1 Brusilovsky Taxonomy for Classification of Adaptive Hypermedia Systems

2.4.1 Adaptive Presentation (Content-Level Adaptation)

The main goal of content-level adaptation is to increase the usability of the application for a wide group of users that have different preferences, knowledge, goals or background. Therefore, information on a certain topic is presented in different ways according to user's characteristics. For example, the system may decide to provide additional explanations as well as to hide content for a novice user, while an expert user in the domain will receive more specific and more detailed information. The information presented to the user may consist of not only text, but also various items of different types of media. However the current AHS offer only adaptive presentation for text and selection of multimedia items. Adaptation of the multimedia content is not provided at this stage.

The main benefit of the adaptive presentation is that it tries to reduce the amount of information available to the most relevant information for a particular user, solving the “overload” problem of the classic hypermedia systems.

2.4.1.1 Adaptive Presentation Methods

According to [20] the following main adaptive presentation methods are considered:

- Additional explanations
- Prerequisite explanations
- Comparative explanations
- Explanation variants
- Sorting

Additional explanation is the most used method. The goal of this method is to provide more information, examples, and illustrations to those users (e.g. novice users) in order to understand the presented concept. Thus, apart from the basic presentation, additional information is provided for a certain category of users. At the same time, the system will hide these explanations for users from other categories (e.g. expert users) who do not need these explanations anymore.

Prerequisite explanation is a special case of the first method. Before presenting a concept, the system decides to insert explanations of all its prerequisite concepts, which are not sufficiently known to the user, in order to compensate for the lack of prerequisite knowledge.

The idea of the *comparative explanations* is to explain new concepts by stressing their relations to known concepts. This additional information on how the current concept relates to other concepts is presented only when the user knows these other concepts.

Explanation variants extend the method of prerequisite explanations. It is used when most of the users need the same explanations, but presented in a different ways. For example some versions of the explanations may use different media types or specific technical terms.

Sorting is applied when the concept presented to the user consists of independent fragments that can be presented in any order. According to the user's background and knowledge the system sorts these fragments from most relevant to least relevant.

Adaptive hypermedia systems such as MetaDoc [21], KN-AHS [22], Anatom-Tutor [23], have used the explanations methods, while HypadAPTER [24] system has used the sorting method.

2.4.1.2 Adaptive Presentation Techniques

In [5, 20] Brusilovsky distinguished the following techniques for the implementation of adaptation methods presented in the previous section:

- Inserting/removing fragments (conditional text)
- Fragments variants and page variants (altering fragments)
- Stretchtext
- Sorting fragments
- Dimming fragments
- Frame-based

Inserting/removing fragment is a simple and effective technique. Information related to a concept is broken in several fragments. Each fragment is associated with a condition or elements from the user model. When information related to a concept is presented, the system selects only those fragments for which the condition is true. This technique can be used to implement methods such as additional, prerequisite and comparative explanations.

Fragments variants and *Page variants* are the simplest adaptive presentation techniques. They consist of keeping two or more alternative pages (or fragments) with different presentations of the same content. Each variant is created for one group of users such as beginner, intermediate and expert groups for example. The system will select the variant to display based on the user model. These techniques can be used for the implementation of the explanation variants method.

Stretchtext technique involves replacing keywords from the document with a longer description. Based on the user's actual knowledge, the system decides which keywords are stretched when displaying a page. This technique is useful to implement additional, prerequisite and comparative explanations.

For the implementation of the sorting method the *sorting fragment technique* is mainly used. The idea is to present a set of fragments (part of a concept) to the user, ordered from the most relevant to the least relevant, according to some criteria based on user's knowledge.

Dimming fragments is a technique used to shade a fragment in order to indicate that it is not relevant for a particular user.

The *frame-based* approach includes all related information on a particular concept in a frame. Slots of a frame may contain explanation variants of the concept and links to other frames. The frames are shown, hidden, presented alternatively or ordered according to some rules. These rules refer to the user's knowledge level or other features of the user. The frame-based technique can be employed for the implementation of additional explanations and explanation variants methods.

Adaptive Hypermedia Systems that provide adaptive presentation and make use of some of the presented techniques include:

- AHA! [25], which makes use of insert and remove fragments, fragments and page variants and dimming fragments.
- HypadAPTER [24], which uses alters, inserts and removes fragments and frame-based technique.
- Anatom-Tutor [23], which applies page and fragments variants; PUSH [26] that uses frame-base and stretchtext techniques.
- SaD system [27], which implements fragment dimming technique.

2.4.2 Adaptive Navigation Support (Link-Level Adaptation)

The basic idea of the link-level adaptation is to adapt the link structure in order to guide the user towards relevant and interesting information for him/her and to prevent the user to follow navigation paths that are irrelevant with his/her tasks or goals. While adaptive presentation changes the content of the documents, adaptive navigation support changes the link structure between the documents and how this navigation structure is presented to the user.

The main benefit of adaptive navigation support is that it simplifies the rich link structure and solves the “*lost in hyperspace*” problem, while maintaining the navigation freedom specific to all hypermedia systems.

2.4.2.1 Adaptive Navigation Support Methods

Next a list of major adaptive navigation support methods that were mentioned in the literature [5, 20] is presented:

- Global guidance
- Local guidance
- Global orientation support
- Local orientation support
- Personalised views

Global guidance method is used in hypermedia applications where the user has a “global” information goal. The information (s)he wants to access is contained in one or more nodes from the hyperspace. The objective of this method is to assist the user in finding the shortest navigation path to the information, by suggesting navigation paths on a global scale. This method is very useful in educational hypermedia systems where the students have a global goal: to learn about a certain topic. In this context, the system suggests a set of pages to read together with an indication on the reading order.

Local guidance method is used to assist the user in just one navigation step, by suggesting the most relevant links to be followed from the current document. The suggestions are made according to the preferences and knowledge of the user.

The objective of the **global orientation support** is to provide an overview of the whole structure of the hyperspace, to help the user to understand the structure and to indicate his/her position in it. The adaptive system also indicates in the structure relevant parts to be followed, parts that were already visited before and parts that are to be avoided (due to the current lack of knowledge of the user).

Local orientation supports allows the user to understand what nodes (documents) are available from the current node and helps the user to follow the appropriate link. This

method shows a small part of the hyperspace, such as one or two levels up or down from the current page.

Another method that protects the users from the complexity of the hyperspace is *personalised views*. This method allows the users to organize an electronic workspace (view) with reasonable small parts of the hyperspace. Each view may have a list of links to parts of the hyperspace that are relevant for a particular working goal.

Many adaptive hypermedia systems made use of these methods in order to improve the navigational support and to help the users to achieve their goals. For example, WebWatcher [28], HyperMan [29] and Hyperflex [30] make use of global and local guidance in order to suggest to the user at each step of browsing which of the links from the given node to follow. Systems like PUSH [26], Hyperadapter [24] and ISIS-Tutor [31] have used local and global orientation support. Bazar [32] is an example of a system that provides adaptive management of the personalised views. It uses agents to collect and maintain a set of links relevant to one of the user's goals.

2.4.2.2 Adaptive Navigation Support Techniques

Several different techniques that implement the methods mentioned above can be used individually or combined in order to provide navigational support. Following [5, 20, 33, 34] the following techniques that manipulate anchors and links with the purpose of adapting the navigation support to the current user's characteristics can be distinguished:

- Direct guidance
- Link annotation
- Link hiding
- Link removal
- Link disabling
- Link sorting
- Map-adaptation
- Link augmentation

With the *direct guidance* technique the user sees only one option to continue from the current page. “The best next link” for the user is determined by the AHS based on the information from the user model.

Link annotation is a very powerful technique that changes the link’s text, style and appearance. For example the anchors of a link can have a different colour, bullet or text to show the relevance of the destination. If the link’s destination is undesirable for the user, the link can be rendered in the same style as the surrounded text. Bigger font and brighter colour for the anchor may indicate the importance and usefulness of the link.

Link hiding is a very useful method to simplify the navigation space and to hide irrelevant pages by changing the colour of the anchors to that of the normal text.

The idea of the *link removing* technique is that links that the system considers inappropriate are removed and they do not exist anymore. The anchors of these links are replaced with normal text.

The objective of the *link disabling* is to remove the link functionality. This technique is combined with link hiding or link annotating.

Link sorting is a common technique used by many AHS. It consists of ordering all the links from a page according to some goal-oriented criteria. Thus the links are presented in decreasing order of the relevance to the user. This technique is typical for information retrieval systems.

Another technique described in the literature is *map-adaptation*. The objective is that the content and presentation of the map of the link structure of the hyperspace is adapted. It consists of a combination of the other techniques such as link removal, annotations, sorting applied to a graphical visualisation of the navigation structure.

A new technique not mentioned in [5,20] recently used in several adaptive hypermedia systems [33, 34] is *link augmentation*. This technique involves the insertion of additional links to useful or related information, apart of the existing hyperlinks found in the source document. Thus the user has access to the page’s original hyperlinks but also to the additional links, which are embedded in the text of the page.

These adaptive navigation support techniques were used stand-alone or combined for an optimal navigation support by many AHS systems. Interbook [35] and ELM-ART II

[36] make use of link sorting and link annotation. AHA! [37] system implements link removal, hiding, disabling and annotation.

2.5 Reference Models for Hypertext/ Hypermedia Systems

Although a wide range of applications have been produced by the Adaptive Hypermedia (AH) community, there are very few reference models for AH.

A **reference model** describes, at a conceptual level, the components and functionality of the hypermedia system in a manner independent from implementation [38]. Different formal, semiformal and informal techniques are used for the specification of the reference model. Formal techniques consist of descriptive languages for which syntax, semantics and manipulation rules are defined. Semiformal techniques make use of diagram and tabular techniques that present the information in a structured form. Informal techniques make use of natural languages.

Next, five major models that have been developed and referred in the literature will be presented:

- HAM (Hypertext Abstract Machine) proposed by Campbell and Goodman in 1987.
- Dexter Hypertext Reference Model formalized by Halasz and Schwartz in a specification language Z, in 1988.
- AHAM (Adaptive Hypermedia Application Model) proposed by De Bra from Eindhoven University in 1999 and the model is based on Dexter Model.
- Munich Reference Model, an object-oriented specification written in UML (Unified Modeling Language) proposed in 2001.
- LAOS, a 5 layer adaptive authoring model for adaptive hypermedia systems, proposed in 2003

2.5.1 HAM Model

The HAM is an informal – semiformal model published by Campbell and Goodman for the first time in 1987 [39]. It represented an important step forward in the development of hypertext research due to the fact that it was the first attempt to define a reference model.

HAM is a general purpose, transaction based, and multi user server for a hypertext storage system. It provided a general and flexible model that could be used in several different hypertext applications. The HAM storage model is based on five objects: graphs, contexts, nodes, links and attributes. The following operations could be performed on HAM objects: create, delete, destroy, change, get, filter, and special. The HAM storage model has been successfully tested against systems such as Guide, Intermedia, and NoteCards.

2.5.2 Dexter Hypertext Reference Model

Dexter Model has appeared as a result of intensive discussions at a hypertext workshop organised at the Dexter Inn in New Hampshire. The goal of the Dexter Hypertext Reference Model was to produce a model in order to capture both formally and informally, the important abstractions found in a wide range of existing and future Hypermedia systems, as well as to provide a common set of terms for describing and comparing hypermedia systems. As the model has been formally specified in the language Z [40], it is considered a formal model.

The model is divided into three layers: Run-Time Layer, Storage Layer and Within-Component Layer. The Run-Time Layer describes mechanisms supporting the user interaction with the hypermedia and presents the hypertext to the user. The Storage Layer describes the network of nodes and links. The Within-Component Layer covers the content and structures within hypermedia nodes. Between the three layers there are two interface mechanisms: the presentation specification mechanism that acts as an interface between run-time and storage layers and the anchoring mechanism that acts as an interface between storage layer and within-component layer. Figure 2-2 shows these layers as presented by Halasz and Schwartz [41].

It should be noticed that this separation of contents, structure and presentation aspects of hypermedia systems is the basis of the most of design methods. There is no doubt that the Dexter Reference Model is one of the important milestones in the history of hypermedia development. It uses the word hypertext to refer to both text-only and multimedia systems.

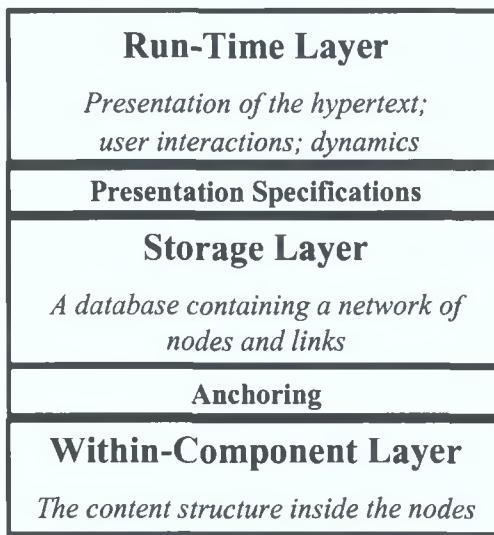


Figure 2-2 Layers of the Dexter Hypertext Reference Model

2.5.3 Adaptive Hypermedia Application Model (AHAM)

While Dexter Model could successfully describe most of the hypertext systems of that time, in 1996 the authors of the AHAM have realised that the Dexter Model needs to be extended in order to describe the new systems developed in the Adaptive Hypermedia field.

Adaptive Hypermedia Application Model (AHAM) presented by Wu, Houben and De Bra in [42] is a semiformal model defined with tuples. It divides adaptive hypermedia systems into the same three layers as the Dexter Model does for hypermedia systems. However AHAM modifies the storage layer, which is a network of nodes, and links in order to include a User Model (UM), a Domain Model (DM) and an Adaptation Model (AM) as presented in Figure 2-3.

The Domain Model represents the author's view of the application domain and describes the structure of an adaptive hypermedia system as a finite set of concept components. Two types of concept components are considered: concepts and concepts relationships. A concept represents an abstract information item from the application domain and it can be either an atomic concept or a composite concept that consists of a set of atomic concepts. Concepts relationships are used to describe the relations between concepts.

The User Model originally referred as the Teaching Model, in AHAM expresses individual user data (e.g. preferences, age) and user's knowledge about the concepts from the

Domain Model. It consists of named entities for which attribute-value pairs are stored. A table representation is used as a conceptual representation of the user model.

The Adaptation Model provides the adaptive functionality of the AHS. It consists of a set of adaptation rules (originally mentioned as pedagogical rules) that combines information from User Model and Domain Model and determines how the information is changed in the User Model and which information will be presented to the user.

It is considered that AHS also has an Adaptation Engine (AE) that is a software environment. While DM, UM and AM describe the information and adaptation at conceptual level (implementation independent level) the adaptation engine provides the implementation-dependent aspects.

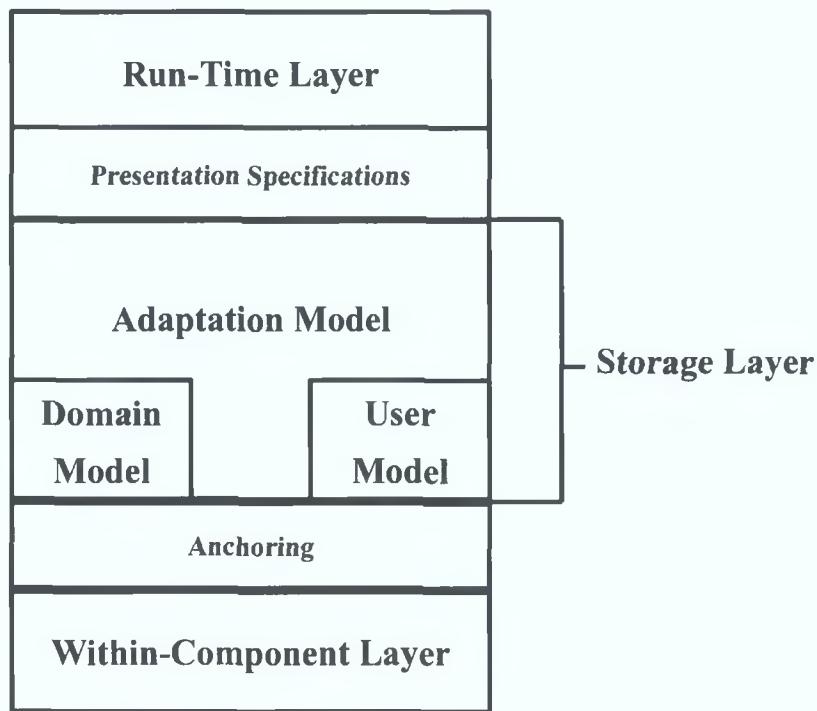


Figure 2-3 Adaptive Hypermedia Application Model

Using all the components mentioned above, any adaptive hypermedia system can be defined in AHAM as a 4-triple: $AHS = \{DM, UM, AM, AE\}$.

More details about the AHAM are provided in the Chapter 4.

AHAM was used later on for the development of the AHA! [25] system, a general-purpose adaptive hypermedia system. AHA! was first deployed and used in the educational area as an adaptive hypermedia courseware application.

2.5.4 Munich Reference Model

AHAM is not the only model based on the Dexter Model. The Munich Reference Model proposed by Koch and Wirsing in 2001 [43, 44] is a semiformal-formal model and follows the same approach as AHAM by including user modeling aspects and rule-based adaptation mechanisms. The main difference between the two models is that AHAM defines an adaptation rule language while the Munich model makes use of an object-oriented specification written in UML (Unified Modeling Language), which provides both an intuitive semi-formal visual representation of the reference model and a formal unambiguous specification in OCL (Object Constraint Language).

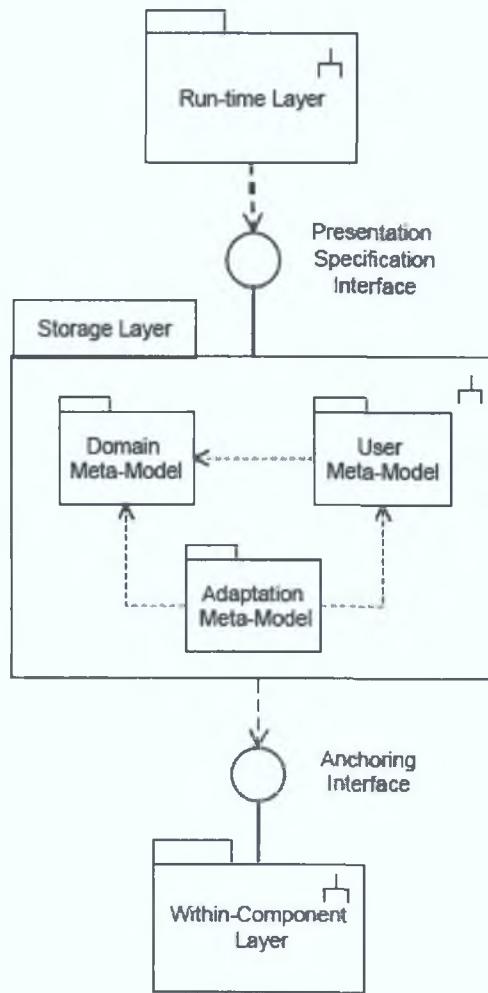


Figure 2-4 UML-based Munich Reference Model [39]

The Munich Reference Model constitutes the basic for the UML-based Web Engineering approach that focuses on development of adaptive hypermedia applications

[44]. Figure 2-4 illustrates the three-layer structure of the Munich Model where the Run-time Layer, Storage Layer and Within-Component Layer are presented as UM subsystems.

The *Run-Time Layer* contains the description of the presentation of the nodes and links. It is responsible for user interaction, acquisition of user behavior and management of the sessions.

The *Storage Layer* is divided into three sub-models:

- *Domain Meta-Model* that manages the network structure of the hypermedia system in terms of mechanisms by which the links and nodes are related and navigated.
- *User Meta-Model* that manages a set of users represented by their user attributes with the objective to personalize the application.
- *Adaptation Meta-Model* that consists of a set of rules that implement the adaptive functionality.

Within-Component Layer consists of content and structure within hypermedia nodes.

Three types of operations were considered in order to specify the functionality of adaptive hypermedia systems:

- Authoring operations used to update components, create a link or a composite component, to create, update or delete rules and to add or delete user attributes from the User Model.
- Retrieval operations defined to access the hypermedia domain structure and the User Model in order to get a component or to get the rules triggered by the user's behavior
- Adaptation operations used to dynamically adapt the User Model content according to the user's behavior.

The Munich Reference Model was used for the development of the SmexWeb [45], a framework for adaptive Web learning applications.

2.5.5 LAOS Model

LAOS [46] represents another model that extends the AHAM by constructing a more general five layer-based model (Figure 2-5) for adaptive hypermedia authoring. This model introduces two new layers to the AHAM:

- Goal and Constraints Model (GM) between the Domain Model (DM) and User Model (UM).
- Presentation Model (PM) below the Adaptation Model (AM).

GM is a multiple sub-layers model that allows the author to define goals to give a focused presentation, and constraints to limit the space of the search. PM adds adaptive features regarding presentation means such as variable page lengths, variables for figure display, formats, synchronizations points, etc.

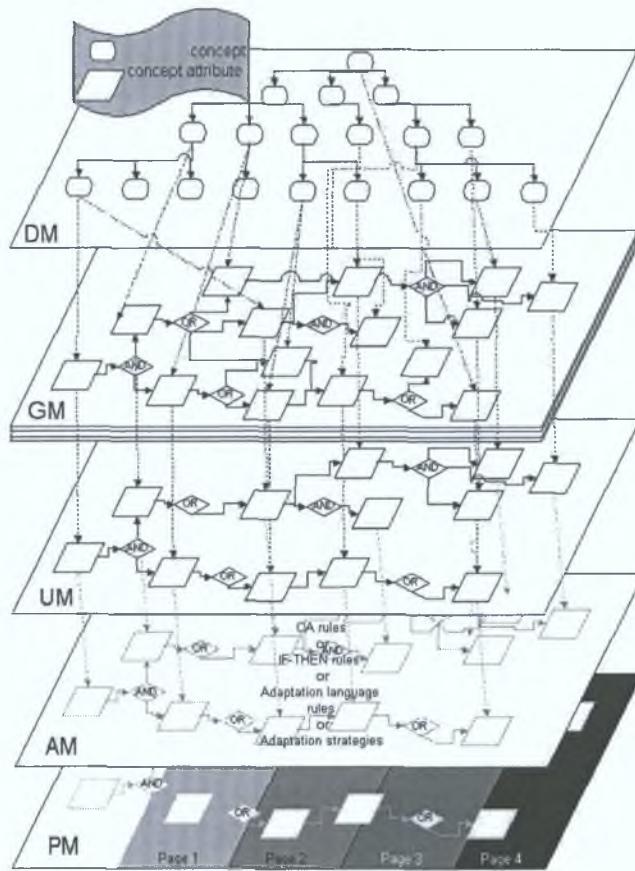


Figure 2-5 LAOS - Five LayerBased Model [46]

For the exemplification of the LAOS model, MOT system [47], an adaptive authoring system for adaptive hypermedia was developed. MOT implements the ideas and definitions

introduced by LAOS and it demonstrates the usefulness of separating DM and GM, as well as of the automatic authoring.

2.6 User Characteristics Analysed by Adaptive Hypermedia

The main function of the AHS is to adapt the content and navigational support of the hypermedia according to some particular characteristics of the user. Therefore, the system has to collect information about the users in order to make assumptions about them.

A main question that rises is what aspects of the user working with the system should be taken into account when providing adaptation? Kobsa et al. [19] distinguished three main categories of user related data:

- user data,
- usage data
- environment data.

User data refers to information about personal characteristics of the user, while *usage data* is related to information on user's interactive behavior that cannot be resolved as user characteristics. *Environment data* reflects aspects of the users' environment that are not directly related to users' themselves.

Adaptive hypermedia systems built before 1996 dealt exclusively with user characteristics (user data) in the adaptation process. Currently, a large number of Web-based adaptive systems are able adapt to the other user related information, and not only to user personal characteristics.

2.6.1 User Data (User Characteristics)

Based on the Brusilovsky and Kobsa reviews [5, 19, 20] that have discussed different user features used by adaptive hypermedia systems, the following major categories of user data were identified and are presented next.

2.6.1.1 Demographic Data

Demographic data about the user are objective facts related to that user and consists of record data (e.g. name, phone number), geographic data (e.g. address, area code, city,

country), user characteristics (e.g. age, sex, income, education), customer qualifying data (e.g. frequency of product/service usage) registration for information offerings, etc

2.6.1.2 User's Knowledge

User's knowledge is the most important information used by existing adaptive hypermedia systems during the adaptation process. Almost all adaptive techniques use user's knowledge with regard to the domain of the application system as a source of the adaptation. The changing of the knowledge state of the user is also a critical part for adaptation. Thus, the system always has to update its estimation about the user's knowledge and the adaptation module has to take into account the current knowledge state.

User's knowledge is most often represented as an overlay of the domain model. For each concept defined in the domain model a value that is an estimation of the user knowledge level on that concept is stored.

2.6.1.3 User's Goals

The goal of the user depends on his current activity with the hypermedia system. It shows what the user wants to achieve by using the system. Based on the type of the hypermedia system, there may have work a goal in application systems, a search goal in information retrieval systems or a learning goal in educational systems.

Another important characteristic of the user's goal is the fact that it is the most changeable user feature. User's goal can change from one navigation session to another, or even within the same session. Thus, some systems define *local goals*, which change often, and *general goals*, which are more stable. For example in education area a learning goal would be considered as a general goal while a problem-solving goal is a local goal.

The user goal is usually modeled by a set of possible user goals, not necessarily related to each other, that the system could recognize (PUSH [26], Hyperflax [30]). Advanced goal-based systems use a hierarchy of goals [48]. Currently user goals are represented by a set of goal-value pairs, where the value is the probability of that goal to be the user's current goal.

2.6.1.4 Background and Experience

The user's background describes all the information related to the user's previous relevant experience outside the subject of the hypermedia system (e.g. experience of could

belong to work, programming experience in C++ for an educational hypermedia system that presents Java programming language).

User's experience reflects how familiar is the user with the structure of the hyperspace and how easy the user can navigate through the hyperspace.

In general, these two user features are modeled by using stereotypes.

2.6.1.5 User's Skills and Capabilities

While user' knowledge feature is dealing with the question "what she/he knows", user's skills and capabilities feature tries to determine "how s/he knows". This feature has an important role in adapting to the user needs.

Kupper and Kobsa [49] distinguish between the actions a user is familiar with and the actions s/he is able to perform. For example a user might know how to perform a task, but is not able to do it due to the lack of required permissions or some physical handicap.

2.6.1.6 User's Interest

User interest can vary considerably between users of the same application and therefore information provided for a group of users may not be of interest to another group. In consequence, this feature is very important in the adaptation process. User's interest has been very often used by information retrieval systems. These systems combine user's interest and search goal in order to improve the information filtering and recommendations provided to the user.

2.6.1.7 User's Preferences

Apart from the interest the users may have different preferences. They may prefer different font types, navigation styles, pictures, colors or even different type of information or links. These characteristics cannot be estimated by the system without any input data from the user. Some adaptive systems assume that user's preferences are not often changeable.

2.6.1.8 User's Individual Traits

Individual trait is a group name for user features that together define a user as an individual [5] (e.g. personality factors, learning styles, cognitive factors). Individual traits are **stable features of the user and they are extracted** by psychological tests. Although these are important features the use of individual traits in the adaptation process has not been a success

yet. This is due to the fact that there is a little agreement on which features should be used and how to be used. Several systems from the educational area [50, 51] have tried to adapt to learning style but is not clear which aspects of the learning style should be modeled.

2.6.2 Usage Data

While User Data denotes information about personal characteristics of the user, Usage Data is related to a user's interactive behavior. Usage Data can be directly observed and recorded, or acquired by analyzing observable data [19].

There are many ways in which the users may interact with the system. Some of the interactions can be observed directly and used immediately in the adaptation process.

Examples of these types of interactions are: selective actions (e.g. clicking on a link, scrolling and enlarging operations for hypermedia objects, audio control operations), temporal viewing behavior rating (users are required to explicitly rate objects, links, web pages), purchases and purchase-related actions, confirmatory and disconfirmatory actions, etc.

In many cases some observable user interactions cannot directly lead to adaptation, further processing of the usage data being necessary. Examples of usage data that is acquired after the information was processed are: usage frequency, action sequences and situation-action correlations.

2.6.3 Environment Data

The range of different hardware and software used by the user of web-based systems is extremely wide. The spectrum is becoming even broader since the number of web-capable appliances with limited abilities (such as PDAs and mobile telephones, TabletPC) is rapidly increasing. Thus, adaptation to the user's environment has become an important issue for adaptive hypermedia researchers. A number of adaptive hypermedia systems have applied adaptation techniques that take into account user's location and user platform (e.g. hardware and software). GUIDE [52] system considers hand-held units as tools for navigation and display of the information, and the user current location in order to provide an adaptive tourist guide. INTRIGUE [53], a tourist information system that assists the user in the organization of a tour and provides personalized recommendations of tourist attractions that can be displayed on WAP phones.

The increase in popularity of digital Interactive TV (iTV) and the availability of hundreds of TV channels has attracted the interest of adaptive hypermedia research. Thus, Electronic Program Guides (EPGs) [54, 55] were proposed in order to support the TV viewer to easily find programs that match their interests and they would like to watch. Later on, personalized programs [56] and advertisements [57] have been introduced based on intelligent set-top boxes that run adaptive programs that monitor the viewer's behavior each time he/she watches TV, and acquire long-term user models. Personalization of advertisements in iTV refers to the delivery of advertisements tailored to the individual viewer's profile on the basis of knowledge about user preferences.

2.7 User Models and User Modeling

A **model** can be defined as an abstract representation of an entity from the real world. It contains only the significant features or aspects of the entity. The entity may be anything from a single item or object to a complete system of any size and complexity. The objective of the model is to offer a formalized or simplified representation of a class of phenomena which can be studied easily.

A **user model** is a representation of an user and consists of a collection of data related to the user. The information kept in the user model of the adaptive hypermedia systems consists of the following user properties: preferences, goals, tasks, experience, etc. For more details see paragraph 2.6.

Modeling refers to the process of generating a model as an abstract representation of some real world entity.

User modeling in adaptive hypermedia is defined as the process of acquiring knowledge about a user, constructing, updating, maintaining and exploiting an user model.

User Modeling has a very important role in adaptive hypermedia systems, mainly in the adaptation process that consists of three stages:

- gathering relevant information about the user
- processing the information in order to build and update the user model
- making use of the user model in order to provide adaptation

Different techniques for collecting information, constructing the user model and using it for adaptation have been proposed. Currently there are no standard techniques for user modeling. The proposed techniques were implemented and tested by different prototype systems. Their most significant limitation is that they are restricted to certain domains.

2.7.1 Information Acquisition

In order to populate the user model the system may use different acquisition techniques. Chin [58] has classified the acquisition techniques along several orthogonal dimensions as follows:

- Active or Passive techniques
- Automatic or User Initiated
- Direct or Indirect
- Explicit or Implicit
- Logical or Plausible
- Online or Offline

Active or Passive techniques involve the participation of the user in the acquisition process. Active techniques interact directly with the user by querying the user about personal characteristics (e.g. online-forms, questionnaires, ratings). Passive techniques are based on observations of the user actions and behavior (e.g. visited pages, click actions, time spent for reading the document, etc.).

Although there are discussions if the users should be interviewed or not (it may disrupt normal browsing behaviour) the active techniques where applied in many systems because they solve some problems related to user modelling in adaptive systems such as:

- user modelling is not completely reliable
- some components of the user model (e.g. background and user's preferences) can not be deduced at all and have to be provided directly by the user

Automatic or user initiated techniques are based on who initiates the acquisition process. User initiated techniques allow the user to change the user model, while automatic

techniques do not offer control to the user, the system taking the decisions to modify the user model.

With a *direct acquisition technique* the system updates the user model with information that is derived directly from the user feedback. *Indirect acquisition technique* builds upon the results of a direct technique using inference rules.

Explicit and implicit techniques are based on the type of the user feedback. The first one allows the user to consciously provide information, while the other technique is based on unobtrusive observation of the user behavior.

Logical and plausible techniques differ according to the results produced. Plausible techniques require the explicit representation of uncertainty in the user model (e.g. using Bayesian networks) and need mechanisms to maintain consistency in the user model.

Most of the adaptive systems would acquire information related to the user *online*, while the user interacts with the system. There are some situations when the user could be observed *offline* by extracting information about the user from databases. There are debates in the user modeling research area on the offline acquisition since the information presented in the database may not correspond to the real users of the adaptive system.

2.7.2 User Modelling Methods

In order to construct the user model, to analyze the user profile and to derive new facts about the user, different user modeling methods can be used. They include:

- Overlay method
- Stereotype method
- Bayesian methods
- Machine learning methods (neural networks rule induction algorithms, instance-based learning)
- Neural networks

The most frequently used methods are overlay, stereotype and Bayesian and they will be described next.

Overlay method is quite simple and widely used. In an overlay model the user's state of knowledge is described as a subset of the expert's knowledge of the domain.

When the user model represents the user knowledge on a subject, for each concept defined in the domain model, the overlay model stores a value. This value is an estimation of how well the user is familiar with this concept. The estimation can have discrete or probabilistic values. Thus, the overlay user model can be represented as a set of pairs "concept - value". There are many examples of the adaptive hypermedia systems that have used the overlay method for representing user's knowledge. The AHA! system [37] is a general adaptive hypermedia system for hypermedia courses delivery over the Web. The user model is based on the knowledge of concepts acquired by reading hypermedia pages and by solving tests. In AHM [59], a prototype system for developing adaptive hypermedia courseware, adaptation depends on user's level of expertise about known concepts, which is a subset of all domain concepts.

Other adaptive hypermedia systems [21, 60] use **stereotype model**. User knowledge is also represented as a set of concept-value pairs (as in overlay model), but the values are not completely independent. The user can be assigned to one or more classes (called stereotypes) and each class is identified by a fixed set of concept-value pairs. The user assigned to a stereotype inherits all these properties. Stereotype modeling is reliable enough and works well for system, which need to adapt to different classes of users.

In order to achieve better results some adaptive systems (e.g Anatom-Tutor [23], Arcade [61], InterBook [35]) have combined stereotype and overlay models. Stereotype modeling is used to determine the class of the user and to assign initial values for overlay model. Then overlay modeling is used to keep the model updated.

Bayesian Networks have become popular in the last decade for modelling the user's knowledge and goals and to identify the best actions to be taken under uncertainty. Bayesian networks are directed acyclical graphs where nodes correspond to random variables (user properties). The nodes are connected by directed arcs that represent links from parent nodes to their children [62]. Each node is associated with a conditional probability distribution that assigns a probability to each possible value of this node for each combination of the values of its parent nodes. Bayesian Networks are more flexible than other models because they provide a compact representation of any probability distribution, they explicitly represent causal relations, and they allow predictions to be made about a number of variables [63].

There are many systems that rely on Bayesian Networks for user modelling such as: KBS Hyperbook System [64] - adaptive hyperbook system for an introductory course on computer science, ANDES - Intelligent Tutoring System for Newtonian physics [65], Lumiere project [66] - providing assistance to computer software users.

2.8 Application Areas for Adaptive Hypermedia Systems

Adaptive hypermedia systems can be applied in any application area where the hyperspace is large enough and the system is used by heterogeneous groups of users that have different goals, knowledge, interests, preferences and tasks. Although there is no restriction on the applicability area, most of the AHS are academic research efforts and therefore the proposed systems were applied only in six major areas:

- Educational hypermedia
- On-line information
- On-line help systems
- Information retrieval hypermedia
- Institutional information
- Personalized views

Next the most important systems that have been developed in each of these areas will be briefly introduced. More details about them can be found in [5, 20].

2.8.1 Educational Hypermedia

Adaptive Educational Hypermedia Systems (AEHS) in general and mainly web-based AEHS have attracted considerable interest due to their potential to facilitate personalized learning. Systems developed for the educational area are also known as adaptive teaching, learning, training or e-learning systems. These systems are used by heterogeneous groups of students with different levels of knowledge on a particular subject. The goal of the students is to learn all the material or a reasonable part of it. These systems consider as the most important feature of the user the knowledge level of the subject being studied. In order to provide different content to different users and to the same user at

different knowledge stages, the system “watches” the students during their work or learning process.

Before 1996 very few AEHS were developed. These systems were mainly lab systems built to explore some new methods that used adaptivity in an educational context [5]. Examples include a hypertext-based system for teaching the C programming language [67], Anatom-Tutor [23] - an intelligent anatomy tutoring system for use at university level and ELM-PE [68] - an on-site intelligent learning environment that supports learning of the LISP programming language with example-based programming.

After 1996, with the increase in popularity of Internet, the Web started to have an important effect on teaching, mainly in higher education. Consequently, the choice of Web as a platform has become a standard [5]. Many online lecture notes or complex tutoring applications were distributed on the Web. The realisation that there is a need to address the heterogeneous audience of Web-based courses has led to the development of a large number of Web-based educational adaptive hypermedia systems. Among these systems, some are extended versions of old systems, others are newly developed systems.

ELM-ART [69] systems and its successor ELM-ART II [36], and INTERBOOK [70] are some of the first adaptive hypermedia systems which were used over the Web. They are updated versions of the systems developed before 1996.

2.8.1.1 ELM-ART

ELM-ART is an updated version of the earlier stand-alone system ELM-PE [68]. Since ELM-PE was being limited by the inherent platform-dependent user interface and the large processor requirements [71], the developers have translated the course material into Web pages and they ported the functionality of ELM-PE to the Web, forming ELM-ART.

The very first version of ELM-ART had provided live examples and intelligent diagnoses of problem solutions. Later on, new enhancements were added leading to the new ELM-ART II [36]. This system supports online exercises and tests, student - tutor communications via email and student - student communications via chat rooms. The exercises and tests results allowed the system to assess the student's knowledge more carefully and to infer the user's knowledge state. The user model (called student model) was also enhanced in a way that the annotation of links informed learners of whether they successfully worked at a page, whether the system inferred that the learner already possessed

the knowledge to be learned on a page, and whether the users already had visited the page [71].

In the next version of the ELM-ART, the multi-layered overlay model was introduced and the original system was further enhanced [72]. Apart from the knowledge states, now users were able to declare knowledge units as already known. The users could change their user model whenever they wanted or switch back to the original state without any loss of information. The system was also extended with two new communication tools: discussion list and user group. Thus users can either enter in a discussion list and post messages that can be read by all other users of the course, or the users can communicate in a user group and post and fetch documents among members of the group they belong to.

The latest version of ELM-ART has been combined with NetCoach, an authoring tool for developing web-based courses. With NetCoach [73], authors can create adaptive web-based courses that are based on the multi layered overlay model, that support different types of test items, and include all the communication tools mentioned above [72].

2.8.1.2 INTERBOOK

INTERBOOK is a system for authoring and delivering adaptive electronic textbooks in the Web. It is an environment in which structured textbooks could be presented in a multiply navigable interface. All *InterBook*-served electronic textbooks have a generated table of content, a glossary, and a search interface. In the same way as ELM-ART, the system uses coloured annotations to inform the user about the status of the node referred by a link.

INTERBOOK stores a domain model of concepts and their structure and an overlay model that helps the system to measure the user's knowledge in different topics. The model is build based on the visited pages. These models are used by the system to provide adaptive guidance, adaptive navigation support, and adaptive help.

The system also provides different options to the user such as direct guidance in the form of "Teach me" button for the most suitable nodes to be read, adaptively sorted links to nodes that present information about background concepts of the current section. The system also provides a glossary index of the concepts.

2.8.1.3 AHA!

While Interbook is a tool for developing and delivering adaptive courseware, AHA! [25] is a general-purpose server-side Web-based adaptive hypermedia system, specifically intended to serve many different purposes. Although the system can be applied in many areas, it was used only in the educational area for delivering adaptive courses at Eindhoven University. The first version was developed in 1998, based on the AHAM model [42], and since then the system has undergone several revisions [74].

In order to create an AHA! application, an author has to write a set of XHTML web pages which contain conditional blocks that enable adaptivity, to create “concept” documents for each page and to specify the required user knowledge for that page. The concept pages also contain the processing actions to apply to the user model once the page has been read. The adaptive content on new pages is provided by combining the concept rules, conditional statements and user data together.

AHA! system consists of a domain model and an adaptive engine, which maintains a user-model, based on knowledge about concepts [75]. Knowledge is generated while the users read pages and take tests.

The content adaptation is performed by using the fragment variants technique. Whereas InterBook uses only link annotation, in AHA! links adaptation is performed by using both link hiding and link annotation techniques. For example the link anchors are coloured in blue (ready and new), purple (ready, but not new), or black (not ready). The colour scheme can be configured by the author and overridden by the user. The user is also allowed to choose between link annotation and link hiding.

Since AHA! system is an open source and one of the latest introduced well-known adaptive hypermedia systems it was used for testing the new QoE layer proposed in this thesis. More details about the system will be provided in section 5.3.1.

2.8.1.4 INSPIRE

Recently, many researchers are trying to integrate learning styles in the design of their educational adaptive hypermedia applications, along with the classic learner's features such as goals/tasks, knowledge, background, hyperspace experience, preferences and interests. INSPIRE emphasizes the fact that learners perceive and process information in very different ways, and integrates ideas from theories of instructional design and learning

styles. In this context, INSPIRE is an adaptive hypermedia system that monitors learner's activity and dynamically adapts the generated lessons to accommodate diversity in learner's knowledge state and learning style [76].

With regards to the adaptive dimension of INSPIRE, the selection of the lesson contents and the provided navigation support are both based on the domain model of the system. The presentation of the educational material to the learners follows their studying attitudes, which are mainly determined by their learning style [77].

The domain model is represented in three hierarchical levels of knowledge abstraction: learning goals, concepts and educational material. Each learning goal is associated with a subset of concepts of the domain knowledge, which formulates a conceptual structure that represents all the concepts of a goal and the relationships between them. Each concept associated with educational material consists of knowledge modules, which constitute multiple external representations of the concept, such as theory presentations, questions introducing or assessing the concept, examples, exercises, definitions in the glossary, etc.

The system makes use of a learner model (user model) in order to exploit learners' knowledge level and individual traits (such as its dominant learning style) and to determine the appropriate instructional strategy. This strategy helps in the selection of lessons' contents, the presentation of the educational material, and the annotation of hyperlinks in the domain hyperspace.

Several levels of adaptation are supported: from full system-control to full learner-control. It offers learners the option to decide on the level of adaptation of the system by intervening in different stages of the lesson generation process and formulating the lesson contents and presentation. Thus, learners have always the option to access their learner model, reflect upon its contents and change it in order to guide system's instructional decisions [77].

Currently INSPIRE is used to support an introductory course on Computer Architecture offered by the Department of Informatics and Telecommunications of the University of Athens, Greece.

2.8.1.5 JointZone

JointZone is a web-based learning application in Rheumatology, for undergraduate medical students. It combines user modelling, domain modelling and adaptive hypermedia techniques in order to deliver a personalised web-based learning environment.

The idea of keyword indexing and the site layout structure was used to model the domain giving a conceptual and structural representation of the content. Keyword indexing provides an alternative to the conventional domain modeling methods. It also reduces the involvement of a domain expert in organizing and labeling the content.

The content of JointZone [78, 79] exists in the form of an online electronics textbook, which is illustrated with photo images and videos taken on various forms of rheumatic diseases. In an additional section, there are a total of 30 interactive case studies that simulate a variety of rheumatic clinical scenarios where students can actively engage in problem solving rather than being passive recipients of information. The cases are subdivided into three groups designated "Beginner", "Intermediate" and "Advanced". The layout of these cases differs according to the degree of expertise of the user [78].

In JointZone, the user model captures two aspects of the students' differences: individual browsing history and knowledge level in the Rheumatology domain. The model also involves the novel idea of using individual effective reading speed to better identify if a student has read a page. The user's knowledge level is initialized based on his/her first entry registration details. This knowledge value evolves through the user engagement with the application, based on student performance in the case study.

The adaptation is performed by using two adaptive techniques: link hiding and link annotation. Based on these techniques and the information from the user model different personalised features are provided such as: personalised reading room, personalised site map, and personalised topic map.

2.8.2 On-Line Information

Online Information is another popular area for adaptive hypermedia systems. Before 1996 several Online-Information systems have been proposed among which is MetaDoc [21]. The goal of these systems is to provide reference access to information for the users with different knowledge level of the subject [20]. Since 1996 the area has been extended and it was divided into many subgroups. Apart of the "classic" Online-Information

systems (e.g. Swan [80] – adaptive web server for online information about nautical publications), it also includes a number of specialized systems such as: e-commerce applications, recommendation systems, digital libraries, electronic catalogs, virtual museums, information kiosks, and electronic encyclopedias [5]. These specialized systems provide some enhancements by taking into account the type of user activity in an application area. For example, PEBA-II [81] - an electronic encyclopedia - provides enhancements by tracing user knowledge about different objects presented in the encyclopedia and providing adaptive comparisons.

Virtual museums and handheld museum guides have the ability to provide adaptive guided tours in their hyperspace and to support the user's exploration of the virtual or real museums with context-adapted narration. Handheld museum guides such as the Hyperaudio system [82], can also determine the user's location and behavior in the physical museum space and therefore the system can support user navigation in the museum. For example, walking near an object could indicate the user's interest in that objects and the system can trigger a relevant narration [82].

Information kiosks offer assistance for travelers and tourists through city guides and support the people to better satisfy their needs. This is performed by giving tailored information on the basis of users interests and preferences. In the tourist information field, several adaptive hypermedia systems have been developed.

GUIDE [52] provides city visitors with information tailored to both the environmental context (the major attractions in the city) and the visitor's personal context (e.g. user current location, the set of attractions already visited). The system was deployed for the tourists that explore the city of Lancaster in England, UK and view this information via a handheld unit (Tablet PC).

WebGuide [83] is intended for web-based and mobile tourist guides and provides personalized tour recommendations for the city of Heidelberg, Germany. These recommendations are made on the basis of geographical information, information about points of interest and selected means of transport, individual user's interests and preferences, and tour restrictions specified by the user.

Intrigue [84] is an information kiosk system that recommends sightseeing destinations and itineraries by taking into account the preferences of heterogeneous tourist groups. It provides an interactive agenda for scheduling tours based on user's choices and

time constraints generates a suitable itinerary. The users can access the information by using desktops and handset devices (WAP Phone).

2.8.3 On-Line Help Systems

Closely related to on-line information systems are **On-Line Help Systems**. The difference from the former category is that these systems have a small hyperspace and they are not independent as they are always attached to tools or other systems. The goal of these systems is to assist the user, presenting help information when s/he has difficulties with a tool or a system. Regularly they automatically detect when the user needs some help. The help information is provided based on the user's goal and is very much dependent on the context in which the user is working.

The research in this area is very limited and only a small number of systems were proposed. The ORIMUHS system [85] is a framework for adaptive on-line help systems providing adaptive presentation and navigation through variants of hypermedia pages, thus varying the explanation content with respect to the current level of user support. The intelligent user support is initiated and controlled by the user, but the actual presentation is decided by the system.

I-Help [86] is an internet-based peer-help system designed to assist learners as they engage in problem-solving activities. The objective of the system is to locate resources, both electronic materials and humans able to help, that are suited to a learner's help request. The system contains a variety of learning resources, most prominently, public discussion forums, on-line materials, and a chat-tool for private discussion between peer learners. I-Help is based on a multi-agent architecture, consisting of personal agents and application agents. Each agent has a model of the resources of the user or application it represents. Personal agents keep a model of the knowledge level of the learner about domain topics, as well as some individual features, like eagerness, helpfulness and class ranking. Application agents keep models of the topics addressed by the instructional materials belonging to an application [87]. The system was deployed at the University of Saskatchewan, Canada and was used by over 600 students.

2.8.4 Information Retrieval Hypermedia

Information Retrieval Hypermedia (IR) is a relatively new subgroup of adaptive hypermedia. It consists of hypermedia search engines that combine traditional retrieval systems with adaptive hypermedia features. The objective of IR systems is to offer a set of

responses represented by a list of links determined by the search engine, in form of a query submitted to an information hyperspace. Their main characteristic is that the hyperspace is very large and cannot be structured “by hand”. The main advantage offered by IR systems is that they limit the navigation choices and can suggest the most relevant links to be followed.

Two major groups of IR systems can be distinguished [5]:

- search-oriented systems
- browsing-oriented systems.

The goal of the ***search-oriented systems*** is to create a list of links to documents that satisfy the user’s current information request. In order to do this the adaptive search engine combines both key words specified in a search request and a model of user’s interest and preferences. Examples of these systems include SmartGuide [88], Syskill and Webert [89].

The goal of ***browsing-oriented systems*** is to support their users in the process of search-driven browsing by using adaptive navigation support technologies. Brusilovsky has identified in his review [5] different *sub-categories of browsing-oriented systems*.

Adaptive guidance systems mark one or more links on the current page that are most relevant to the user’s goal. For example WebWatcher [90] suggests hyperlinks in a Web page by comparing the user’s current interests with the hyperlinks’ description. Personal WebWatcher [91] based on the WebWatcher is an agent that follows the user’s browsing activity and suggests hyperlinks based on the user’s interests as inferred from the content of the documents that the user has visited in the past.

Adaptive annotation systems attach various visual cues to the links on the current page in order to help the user select the most relevant one (e.g. IFWeb [92]).

Adaptive recommendation systems attempt to deduce the user’s goals and interests from his/her browsing activity, and build a list of suggested links to nodes that usually cannot be reached directly from the current page, but are most relevant to that user (e.g. SurfLen [93]).

2.8.5 Institutional Information

Institutional Information is a relatively new area application domain for AHS. The objective of these systems is to serve on-line all the information required to support the work

of some institutions such as for example a hospital. They represent a medium for everyday work of many institution employees. These systems are work-oriented and the users need to access only a specific area of the hyperspace, relevant to their current work goal.

The main difference between Institutional Information, search-oriented IR and On-Line information systems is the working area of a user. Within the last two systems the working area of the user is the entire hyperspace.

Hynecosum [94] is an institutional information system developed for the use of medical staff in a hospital environment. The system builds a model of the user and modifies the interface and navigation support according to the experience level of the user. It employs task-hierarchies to restrict views of information and make certain links visible or hidden, based on the experience level and individual need and preferences.

2.8.6 Personalized Views

Systems for managing **Personalized Views** constitute another fairly new application area for AHS. These systems use adaptive hypermedia techniques to hide some of the complexity of the overall hyperspace and to allow users access to subsets of the hyperspace for their own use. Thus, by defining their personalized views the users protect themselves from the complexity of the overall hyperspace. Each defined view can be allocated to one of user's goals or interests. This application area is similar to institutional information because it offers to the user a convenient access to a subset of information from the hyperspace for everyday work.

In the context of Web there are at least two common mechanisms for managing personalized views: personalized site views (e.g. MyYahoo, MyNetscape) and bookmark organizers [5]. The majority of the personalized sites and bookmark organizers are adaptable but only a small number of bookmark organizers are adaptive such as: PowerBookmark [95] or WebTagger [96].

PowerBookmarks [95] is a tool for Web information organization, sharing, and management, which parses metadata from bookmarked URLs and uses it to index and classify the URLs. The system supports advanced query, classification, and navigation functionalities on collections of bookmarks. PowerBookmarks also monitors and utilizes user behavior in order to provide valuable personalized services such as automated URL bookmarking, document refreshing, and bookmark expiration. It also allows users to specify their preferences in bookmark management.

2.9 Chapter Summary

This chapter has briefly introduced hypertext and hypermedia terms, major achievements in the area and the motivation for the latest evolution towards a new research direction: Adaptive Hypermedia (AH). Although Adaptive Hypermedia is a young research field, after 1996 it experienced a period of rapid growth. A large number of adaptive hypermedia applications were reported and the application areas have expanded. This chapter has also investigated the current major application areas and the main systems that have been proposed in each of them. Among these areas, nowadays, educational adaptive hypermedia is an established leader. The research findings from the education area were used during the evaluation phase of the proposed QoE layer.

Also this chapter has introduced the most important reference models for the design of the AHS that were proposed. Among them, the AHAM model was used in this thesis in order to illustrate a generic architecture of a QoE-aware Adaptive Hypermedia System.

The main benefits provided by AH systems over basic hypermedia systems are very significant and were highlighted. These systems offer higher flexibility and comfort to each individual user; they solve the “information overload” and “lost in hyperspace” problems. Another important benefit is that a wider group of users can use the adaptive system as it adapts in order to fit each user’s interest and competence.

Although these systems bring many benefits they also have some problems. With the advance in computer and communications technology a variety of Internet access devices and networks have been launched on the market. The type and capacity of the access device, the bandwidth and the state of the network the device operates in, the complexity of the Web pages delivered over a given network all affect the Quality of Experience for the end-user. At the same time, the users expect not only high-quality and efficient tailored material but also a perfect integration of this material with the day-to-day operational environment and network framework. In this context it is significant to highlight a new problem faced by Web-based AHS: providing a good level of end-user perceived Quality of Service (QoS), also called Quality of Experience (QoE). Therefore AHS should tailor the Web content according to both user individual characteristics and user QoE and should regularly monitor any changes in the user’s operational environment that may trigger variations of QoE. This thesis focuses on providing a higher level of personalisation, including QoE-issues.

Chapter III

Web Quality of Service

Background

3.1 Chapter Introduction

As the number of Web users and the diversity and complexity of Web applications continue to exponentially increase, the “user experience” with a Web service has become an important issue.

While Adaptive Hypermedia research seeks to tailor the Web information served to the users according to their interests and goals, Web QoS seeks to reduce the load on system resources or the quantity of the data sent to the clients, bringing benefits in terms of user perceived QoS. Some of the proposed Web QoS solutions are presented in section 3.3.

A set of studies have also investigated how users define and perceive QoS and what are user expectations on the QoS delivered by a Web system. These studies analyse the relationship between objective QoS metrics and the user perception of quality and the impact of the values of these metrics on the behaviour of the users. Details about this research are presented in section 3.4.

3.2 Quality of Service (QoS) Definition

A deep analysis into the QoS research area has shown that QoS is being handled at multiple levels by the standard communities and thus, different definitions have been proposed for the term of QoS.

According to ITU-T QoS Study Group the term of QoS is defined as:

“collective effect of service performances that determine the degree of satisfaction by a user of the service” (ITU-T R. E.800) [97].

International Organization for Standardization has proposed another definition in ISO/IEC 10746-2 [98] for the term of QoS as follows:

“a set of qualities related to the collective behavior of one or more objects”

while the Internet Engineering Task Force (IETF) Network Working Group has proposed the following definition in RFC 2386 [99]:

“a set of service requirements to be met by the network while transporting a flow”

The ITU-T definition closely relates QoS to the users' perception and expectations related to a certain service whereas the ISO/IEC's is more general, but with direct applicability in networking. Detailing the QoS definition, both ISO/IEC and ITU-T's view is that QoS concerns characteristics like rate of information transfer, latency, probability of a communication being disrupted, probability of system failure, probability of storage failure, etc. They also mention possible constraints that may affect the QoS, which include temporal ones (e.g. deadlines), volume constraints (e.g. throughput) and dependability, involving aspects of availability, reliability, maintainability, security and safety (e.g. mean time between failures).

At the same time the industry leaders define QoS closer to their object of activity. For example for Cisco QoS “refers to the capability of a network to provide better service to selected network traffic over various technologies” [100], while for Microsoft QoS “refers to the ability of the network to handle the traffic such that it meets the service needs of certain applications” [101].

Summarising, the QoS concept is very complex and as there is not a widely accepted definition for QoS. However ITU-T E.800 provides the closest definition for QoS to the goals of this research. Also, there are not general solutions for assessing, providing and quantifying QoS. However different aspects of QoS are explored according to certain interests that have driven extensive research in one direction or another. Since the main interest of the research presented in this thesis is to provide high QoS levels from end-user perspective in World Wide Web area, details are given about research directions that have proposed solutions in this context.

3.3 Solutions for Improving End-to-End Web Performance

The Quality of Service (QoS) perceived by the users has become a dominant factor for the success of any Internet based Web service. The principal QoS attributes the users perceived include those related to the service “responsiveness” (i.e. the service availability and timeliness) [102]. Therefore the main issue is to minimise the response time perceived by the user (end-to end-response time). This is defined as the time between the generation of a request from the browser, for the retrieval of a page, and the rendering of that page at the browser’s site [102].

There are three ways to improve end-to end-response time:

- decrease the generation time of the page by applying server-side performance solutions
- increase the speed at which that the information reaches the user by applying infrastructure related performance solutions
- lower the amount of time it takes to render the information making use of browser related solutions.

3.3.1 Server - Side Performance Solutions

Solutions currently investigated range from server overload prevention and control to performance enhancement techniques that run into the end-systems to enhance the QoS perceived by the users (i.e. server-side caching, Web servers’ replication). Some proposed solutions are presented next.

3.3.1.1 Reducing Server Overload

Admission control represents a common solution used by the regular Web servers under extreme overloads. The aim of the admission control mechanism is to maintain an acceptable load in the server even when the arrival rate is above the site’s capacity and thus quality of service for exiting users is not degraded. The mechanism is based on two solutions: rejection of requests or content adaptation. In the first case, some requests are rejected while the server is overloaded. With the second solution, the site delivers less resource intensive content to the customers during overload period. However, a content adaptation mechanism must always be used in combination with a rejection mechanism [103].

The solution proposed in [104] makes use of the content adaptation mechanism. The novelty introduced is that one out of multiple versions of the requested Web pages (pre-processed a priori and organized in tree structures), differing in quality and size, is served based on a measure of the current degree of server utilization and QoS requirements. In order to decide the right tree content for each client, the application measures the current degree of server utilisation using a server load monitor. The last one measures the service times of all requests for an observation period and finally computes the system utilisation as a sum of the request rate and the delivered bandwidth. When the Web server is overloaded the adaptive content delivery is done using a client prioritisation mechanism, which degrades first the lower priority clients.

The admission control mechanism with rejection of requests can be request-based or session-based. Request-based techniques define an upper limit to the number of requests processed at the same time in the site, while session-based technique defines a limit the number of ongoing customer sessions.

Web2K [105] is a session-based solution that improves the usage of the Web site and offers differential performance to the users. Based on specified prioritisation criteria such as the identity of the user issuing the request, the IP address of the user, a QoS module classifies incoming requests into different classes of users. When the Web2K server detects overload it performs admission control to avoid over-committing its resources. Thus, user access is denied based on their classification class. Other Session-based schemes have also been developed by Cherkasova and Phaal [106] and Chen and Mohapatra [107].

An admission control scheme that uses processing delay as a system variable was proposed in [108]. Processing delay is defined as the time it takes for the site to process an HTTP request. The authors have investigated and compared request-based and session-based versions of the proposed control scheme. The results have proved that the session-based control scheme guarantees that admitted customers may complete their sessions with good QoS thereby minimizing the number of unsatisfied customers.

3.3.1.2 Web Servers' Replication

The main goal of the replication process is to increase data availability and to reduce the load on a server by balancing the service accesses among the replicated Web Servers [102]. Currently there are two approaches to deploy a replication of a Web service: Servers' clustering and geographical replication.

The first approach considers a set of servers (cluster of servers or farm of servers) placed together in a single location that represents a Web site. A front-end computer called dispatcher behaves as a proxy by intercepting all the incoming requests and forwarding to one of the servers. The goal of the strategy is to maximize the Web site throughput and ensure content availability with very little emphasis on minimizing user response time. Since all the servers are located in the same place, the transmission time of the information is not improved and in consequence the user response time is not decreased. Currently there are many products based on this strategy such as: Cisco LocalDirector [109], Alteon ACEdirector [110], IBM's Web Accelerator [111].

In geographical replication all the replicated Web servers are spread in multiple Internet locations. Each location might also consist of a cluster of servers. The goal of this strategy is to increase both the web service availability and user perceived response time. In order to bind a web user's request to one or more server locations two selection policies were proposed: server- side policy and client side policy. In the fist case, servers, routers or DNS make the choice whereas in the second case the selection is made into the client browser or client-side proxy. The client-side approach has the advantage that it has a view of the network congestion, close to the one perceived by the user. Different techniques (e.g. user response time minimization [112], fixed-size data blocks [113]), for selecting the best server from which the client downloads the data were proposed in the literature. Commercial products developed by Cisco and Nortel (e.g. Cisco DistributedDirector [114]) are some examples of server-side approach.

3.3.1.3 Server - Side Caching

Server-side caching aims to accelerate application response times by fulfilling requests on behalf of the origin servers, thus offloading the web servers and improving system scalability. This solution involves a web cache located within the Web site, near the server farm. The Web cache intercepts all the requests sent to the server. If the requested page is located in the cache it delivers it. Otherwise it forwards the request to the server. Currently both static and dynamic server-side caches exist and are focused on accelerating the generation of static and dynamic content. Storing copies of the dynamic components into the cache is very useful when consecutive requests for a particular dynamic page produce identical results. Although server-side caching solution brings benefits in term of performance, maintaining cache consistency is still an important issue that has to be addressed.

3.3.2 Infrastructure Based Performance Solutions

Apart from different software and hardware solutions on either the server or the client sides that try to improve the server performance and the client satisfaction, the developers have also upgraded the network infrastructure in order to deliver the information faster and thus to accelerate the download time of the Web pages. Therefore, important companies have deployed different “smart” routers and Web switches that use on-board intelligence to reduce network latency and other delays, and caching appliances.

3.3.2.1 Intelligent Web Switching

The Web switching represents the main important part of the intelligent Web transport. Unlike conventional packet switching, Web switching provides the ability to identify individual users or specific content being requested. Web content switches - also known as URL switches or Layer 7 switches – can provide the highest level of control over incoming Web traffic to a web site. These switches direct Web traffic based on the content, provide a front-end for Web server farms or cache clusters and maximise server resources and Web caches’ performance. By examining the HTTP header, Web content switches can make decisions on how individual Web pages and images get served from the site. This level of traffic control can be helpful if the Web servers are optimised for specific functions, such as image serving, SSL (Secure Sockets Layer) sessions or database transactions.

Different companies have deployed a large variety of Web switching products with different capabilities and facilities. ArrowPoint’s Content Smart Web switches, a product of ArrowPoint Communications, Inc. [115], optimise e-commerce and Internet content delivery by dynamically directing specific content requests to the best site and best server at that moment, avoiding busy or overloaded sites. ArrowPoint’s products can differentiate content based on URL and cookies and can provide policy-based prioritisation for premium content requests or preferred users and tailor quality of service accordingly.

With Alteon’s content-intelligent switching technology, (Alteon WebSystems [116]) the enterprises will be able to offer more efficient and customized services by intelligently directing user requests to the most appropriate content by examining information, such as URLs, browser types, and HTTP cookies, found in each packet. Cisco content switching products [117] and Lucent Web switch, products of Bell Labs [118] offer similar capabilities.

3.3.2.2 Intelligent Routers

A router is a device that operates at the network layer, or layer 3 of the OSI model [119]. When compared to hubs and switches, routers are more complex. Its main function consists of separating files such as Web pages into individual packets of data and sending them along the network to other machines. RouteScience [120] is one of several companies introducing products and services aimed at using smart routing or intelligent routing technology in order to pick the best route on the fly for customers. Intelligent Routing technology measures real-time Internet performance (e.g. latency, packet loss) for each link and makes routing decisions based on these performance metrics.

3.3.2.3 Cache Appliances

Caching products are divided in two categories: caching appliances and software-based proxy servers. Caching appliances are dedicated hardware devices that are easy to manage and offer better performance. Proxy servers are less expensive, and the hardware on which they run can be used for tasks other than caching. Vendors such as Microsoft [121], Novell Inc [122] and Netscape Communications Corp. [123] sell software-based proxy servers running on general-purpose hardware platforms. Dedicated hardware appliances come from suppliers that include IBM [124], Cisco Systems [114], CacheFlow [125], Network Appliance [126], etc.

Therefore, a Web cache appliance stores certain content from Web servers and serves it to end-users in place of the Web servers. Thus, end users experience faster and more reliable Web delivery.

Currently, caching appliances are deployed either in front of a content site's server farm to increase the number of users that a site can serve simultaneously or at regional or metropolitan area points acting as content accelerators moving the Web content closer to the users.

3.3.3 Client - Side Performance Solutions

Caching and pre-fetching are standard techniques often used over WWW. They aim at improving the user perceived response time by decreasing the Web page retrieval times. In most cases caching and pre-fetching schemes do not attempt to cache or pre-fetch a dynamic web page due to the fact that a dynamically constructed web page might constantly change.

3.3.3.1 Client - Side Caching

Client-side caching was the first solution early applied to the Web for improving the user response time. Currently, it is employed in most web browsers on the market and caching is activated by default.

Client side caching consists of storing static objects (i.e. HTML, GIF and JPEG files etc.) from requests made by the client, for a limited period of time on the local hard drive of the client. In order to improve its efficiency, the caching-proxy idea has been developed. Ideally, a proxy server is physically located closer to the clients and serves fewer clients than the Web server. Therefore caching documents close to clients reduces the number of server requests and the traffic associated with them. Different studies have proved that the benefits brought by the proxy caching are in term of cache's hit ratio and reduction in the client's perceived latency [127,128] when more than 8000 client are served. However as the number of dynamically constructed web pages increases, the benefit of employing a cache on a proxy server diminishes.

3.3.3.2 Client - Side Pre-Fetching

Pre-fetching has recently gained attention because of its potential performance benefits without requiring complicated changes or additions to Web servers. Pre-fetching complements the research in caching, since all benefits of pre-fetching are in addition to those of caching. Caching strategies attempt to provide fast access to a document from the second time it is accessed. Pre-fetching mechanism tries to provide fast access to a document the first time it is accessed.

Thus, the basic idea of pre-fetching mechanism comes from the following fact. After retrieving a page from a server, the user spends some time for processing the information from the Web page. During this period, the network link can be used to fetch some files in anticipation.

The research area of Web pre-fetching is quite new and currently under intensive investigation. Different pre-fetching algorithms for predicting the Web surfing hyperlink paths such as: pre-fetch by popularity of the documents [129, 130], pre-fetch by good-fetch that balances object access frequency and object update frequency [131] and pre-fetch by lifetime of the objects [132] have been investigated.

3.4 Quality of Experience (QoE)

The term Quality of Experience is relatively new and has mainly been used in company white papers. It focuses on the user and tries to understand end-user expectations for QoS. QoE is considered in [133] as a concept comprising all elements of a user's perception of the network and performance relative to his/her expectations. In order to deliver high QoE it is important to understand the factors contributing to the user's perception of the target services, and to apply that knowledge for defining the operating requirements.

The QoE concept may apply to any kind of network interaction applications such as Web navigation, multimedia streaming, Voice over IP, etc. According to the application it can mean different things. For example, for a voice over IP application a positive QoE relates to the sound fidelity and ability to smoothly take turns in a conversation. A remote multimedia streaming application has a high QoE if the video image is large and clear when presented to the user. For a Web surfer a good QoE means that Web content is retrieved fast enough before getting bored and clicking a link to another web site. In general QoE is expressed in human terminology rather than metrics. An experience can be excellent, very good, good, fair or poor.

Although QoE is very subjective in nature, it is very important that a strategy is devised to measure it as realistically as possible. According to the type of application the user interacts with, different QoE metrics that assess the user's experience with the system have been proposed. QoE metrics may have a subjective element to them and may be influenced by any sub-system between the service provider and the end-user. Some of the QoE metrics have a counterpart in the QoS metrics, but are different in that they take the customer angle. Therefore, the QoS performance metrics are mapped onto user perceptible QoE performance targets.

ITU-T Recommendation G.1010 [134] provides guidance on the key factors (e.g. delay, information loss) that influence Quality of Service (QoS) from the perspective of the end-user (i.e QoE) for a range of applications that involves voice, video, images and text. The Recommendation provides a list of performance parameters (e.g. one-way delay, delay variation, loss, lip-synch) that govern end-user satisfaction for these applications. An end-user QoS model that provides an indication of the upper and lower boundaries for each parameter in order the application to be perceived as essentially acceptable to the user was

proposed. For example exceeding of the upper boundary for loss or delay parameters suggests that the service will be considered unsatisfactory.

3.4.1 QoE in World Wide Web

In the area of World Wide Web, QoE has been referred as end-to-end QoS or end-user perceived QoS. Measuring end-to-end service performance, as it is perceived by end-users is a challenging task. Different research studies [135, 136, 137, 138] have tried to relate the objective network service conditions with human perception of the quality of service. The results have proved that many QoS parameters such as end-to-end response time (also called page download time or delivery time), network latency, perceived speed of download, successful download completion probability, user's tolerance for delay, delivery bandwidth and frequency of aborted connections factor into user perception of World Wide Web QoS. Measurement of these parameters may be used to assess the level of user satisfaction with the web-content delivery quality. The interpretation of these values is complex—varying from user to user and also according to the context of the user task.

Response Time

Response time (latency) metric quantifies how long the user must wait for a response to a query. For web-based requests, query response time also called download time, includes at least two elements: fetch latency – the time to load the page HTML into the browser, and render time – the time to receive all elements referenced by the page HTML, such as images, and display the page in the browser [139]. It is defined as the delay between a request for a Web page and the reception of that page in its entirety.

Different studies have analysed the “user experience” with a Web site and most of the emphasis has been on defining what constitutes “acceptable” download times. It is widely accepted that long delays cause user frustration leading to performance loss, distraction and difficulty to remember what the user was doing [140]. As a consequence, the user may find the content less interesting [141] and of a lower quality [142].

Many groups of researchers have set out to define how response time impacts on users' perception of Web sites. They also looked for an answer to the question: How long is too long to wait for a Web site?

Nielsen [143] suggested a 10-second limit but recent research [144, 145, 146] proved that under certain conditions users would accept download times significantly longer

than 10 seconds. For example, users who have little or no experience with high bandwidth connections and are used with dial-up connections are more patient [146]. Users' acceptance of page-response time also depends on the nature of the task they are involved in [144]. Users will wait longer in the service of completing important tasks. However, these same users report higher frustration levels in completing less important tasks although they experienced the same response time [145].

The studies outlined above converge with other findings to suggest that 10 seconds is not necessarily a hard rule. Willingness to wait more is also moderated by other factors. For instance, novice users and older individuals tend to be willing to wait longer for a computer to react [146, 147]. Users tend to be relatively more patient the first few times they visit a site [135].

Currently there is no universal standard limit for an acceptable download time. However, collectively, these findings suggest the users may be more tolerant than the 10-second rule suggested by Nielsen [143]. On average a download time higher than 12 seconds causes disruption and users start to lose their attention to the task. At the same time it is significant to mention that when the user is aware of his slow connection, he/she is willing to tolerate an average threshold of 15 seconds.

User's Tolerance for Delay

The research performed in [148] proposed a mapping of the values of the end-to-end response time in human perception space in order to express the user tolerance. Three zones of duration that represent how users feel were proposed: zone of satisfaction, zone of tolerance and zone of frustration. Based on a survey into a number of research studies the authors [148] concluded that a user is "satisfied" if the page is loaded in less than 10 to 12 seconds. The next zone begins when the page-load time exceeds the time limit from the zone of satisfaction. The user starts to become aware of the passage of the time, slowing building up into annoyance. It is believed that it is a wide band of time between when a user is no longer satisfied and when the user becomes frustrated. The zone of frustration starts when the user has reached the point when he/she is significantly unhappy with the service provided. According to a number of studies [135, 149, 150] different value between 30 and 41 seconds were considered as the critical point. The conclusion of the research was that 12 seconds is the upper limit for user satisfaction and 40 seconds is the limit at which performance becomes intolerable.

In a study of users' tolerance of delays on the World Wide Web [137] it is shown that the tolerance of delay is influenced by contextual factors such as duration of interaction, the type of task users are engaged in, and the page loading method. The phenomenon of the duration that causes the users to decrease their tolerance for delay, making them more critical on performance issues, was expressed through a "utility" of a session function. A negative value would indicate unacceptable performance. The authors [137] have also noticed a significant difference between having pages render either completely or incrementally. Participants in the incremental rendering group tolerated up to 6 times more delay. They suggest that incremental loading helps users keep their attention on the task.

Network Latency and Delivery Bandwidth

Bandwidth is defined as the amount of data passing through a connection over a given time and it is usually measured in bits per second (bps).

Network latency is measured as the time it takes for a packet of data to get from one designed point to another

The analysis performed in [138] has established that there is a non-linear relationship between QoS and QoE. The effect of two main network QoS parameters, namely network latency and network delivery speed on human satisfaction was investigated. The authors have concluded that network latency plays a less significant role on the level of user satisfaction when it is in the range of 50-500 msec than network bandwidth. However, higher values were not investigated. On the other hand, network bandwidth has a crucial role on the end-user satisfaction when it ranged between 0-100 Kbit/s and there is no gain in web browsing satisfaction for connection speed above 200kBit/s.

Frequency of Aborted Connections

The *frequency of aborted pages* is a QoE metric defined in [151] and reflects the client satisfaction in relation to the perceived QoS. The main idea behind this metric is that if the user gets impatient due to a slow perceived access time he/she will interrupt the transfer of the web page. A user can perform an interruption by clicking the 'stop' or "reload" button while a web page is loaded or by clicking a link from the page before the whole web page was loaded. Thus, only a subset of aborted actions reflect a poor perceived QoS while the others are caused by client-specific browsing patterns. In [151] only the aborted pages with a response time higher than a given threshold (7secs) were considered as reflecting a bad quality download.

3.5 Chapter Summary

With the increasing number and popularity of Web applications and services, Quality of Service (QoS) has become a significant factor in distinguishing the success of service providers. QoS determines the service usability and utility, both of which influence the popularity of the service.

The overall performance of a Web application depends on the application logic, network, and transport protocols (e.g. HTTP) that it uses. At the same time, the dynamic and unpredictable nature of the Web also affects the QoS. Therefore, providing acceptable QoS is really a challenging task.

High QoS is provided by using different approaches like caching, pre-fetching, load balancing of service requests, etc. Caching can be done at Web server level, network route level and at users level. This strategy attempts to provide fast access to a document from the second time it is accessed. Pre-fetching complements the caching by providing fast access to a document at the first time it is accessed. Load balancing, performed on the server side, prioritises various types of traffic and ensures that each request is treated appropriately to the business value it represents. Different solutions that make use of these techniques were described in this chapter. Performance issues such as caching, pre-fetching and server load are not subject of this thesis. Solutions that solve these issues can be used in conjunction with the work described here.

Although the Internet industry is spending significant amounts of money to provide networking resources and performance application level solutions (e.g. routing, scheduling) that would ensure high QoS, understanding and measuring the Web service performance perceived by the user, often expressed as Quality of Experience, is also a challenging task. While QoS considers factors that are objective, easily measured, and used to define and design systems, QoE makes use of subjective elements. Different research studies that have related QoS metrics such as response time, network latency, bandwidth, etc. with the human perception of the quality of service were presented in this chapter. These research findings were used for the design, exemplification and testing of the proposed QoE layer. The QoE layer modeled the performance-based metrics introduced in section 3.4.1 in order to provide a complete representation of the end-user QoE.

In conclusion, although Web QoS research has brought many benefits in terms of service usability, utility and performance, with a positive effect on the user perceived Web service's performance, it does not address the issue of user personal characteristics. More particularly, how and what content-related performance solutions should be performed in order to maximise the user's satisfaction with the delivered content. At the same time, caching and pre-fetching solutions performed close to the user side should also consider user's goal(s), interests, and knowledge when decisions on the web pages stored in the cache are taken or on the pre-fetched files.

Chapter IV

Quality of Experience Layer for Adaptive Hypermedia Systems

4.1 Chapter Introduction

No standardised architectural framework has yet emerged for adaptive hypermedia systems. However, commonalities do exist between already proposed and developed systems in the recent years. Most of the adaptive hypermedia systems follow the principles presented in the AHAM reference model that is briefly described in section 4.2. It divides the adaptation process into three components: User Model (UM), Domain Model (DM) and Adaptation Model (AM).

Starting from the generic architecture of an AHS that follows the AHAM principles, this thesis proposes a new end-user QoE layer that improves the adaptation process by providing a satisfactory end-user Quality of Experience (QoE). This is performed by taking into consideration different user-related performance features that have a substantial impact on the QoE.

Section 4.3 introduces the components of the QoE layer. The most important ones are: Performance Monitor (PM), Perceived Performance Model (PPM), and the extended Adaptation Model (AM). The PM monitors and measures in real time different user-perceived performance parameters that affect the QoE and delivers information to PPM (section 4.3.3). The PPM has the important function of providing a dynamical representation of user satisfaction related to the perceived QoE. It models performance-related information and generates constraints or suggestions on the content to be served. It makes use of the stereotype-based technique in order to construct and infer information about the user.

In order to apply the PPM content-related suggestions the classic AM had to be extended. Therefore an adaptation algorithm that determines and applies the correct

transformations on web page was proposed. This algorithm is described in section 4.3.4.1 and involves two types of transformations: modifications in the properties of the embedded components and/or elimination of some of the components. Since the web content provided by the AHA! tutorial was used during the evaluation of the QoE and it consists of text and images the adaptation algorithm performs transformations only on text and images.

4.2 AHAM Model

AHAM (Adaptive Hypermedia Application Model) is a general reference model that provides a framework to describe the adaptation functionality of adaptive hypermedia systems at an abstract level. Hongjing Wu and Paul de Bra from Eindhoven University (The Netherlands) introduced this model for the first time in [42] and more details were provided in [152]. AHAM is an extension of the Dexter reference model [41] for hypermedia applications. It consists of the same three layers as Dexter model but it extends the Storage Layer with three sub-models (Figure 4-1):

- Domain Model (DM) that describe the information domain
- User Model (UM) that offers a representation of the user profile and how it relates to the information content
- Adaptation Model (AM) that describes the adaptation process (how the AHS adapts the information to the user).

The division into a Domain Model, User Model and Adaptation Model provides a separation of the most important components and offers more flexibility when developing an adaptive hypermedia application. It makes adaptation in AHS easy to understand, independent of the information content and structure of a concrete application [152].

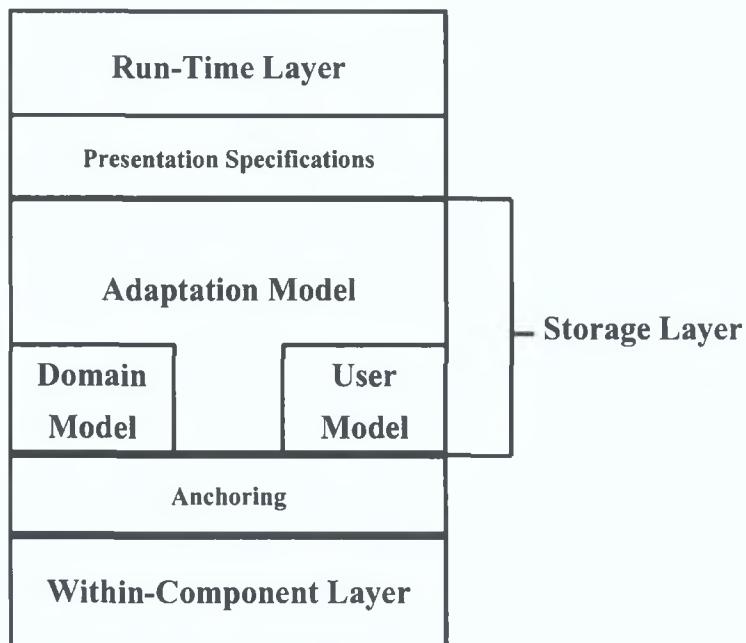


Figure 4-1 Adaptive Hypermedia Application Model

4.2.1 Domain Model

The Storage Layer defined in Dexter model describes the information domain at the abstract level. The Domain Model (DM) defined in AHAM extends the abstract presentation of the information domain by defining a set of *concepts* and *concept relationships*. Thus different levels of concepts, organized in a hierarchical structure, from very general abstract concepts to more specific and concrete concepts were introduced.

4.2.1.1 Concepts

A *concept* is defined as an abstract representation of an information item from the application domain. It can be an atomic concept or a (abstract) composite concept. An *atomic concept* corresponds to a *fragment of information* and it is a primitive in the model. A *composite concept* has a sequence of children that can be either atomic concepts or other composite concepts. A composite concept that has only atomic concepts as children is called *page*. Composite concepts, pages and fragments are organized in a hierarchical structure that must be an acyclic graph. No component can be a subcomponent of itself either directly or indirectly.

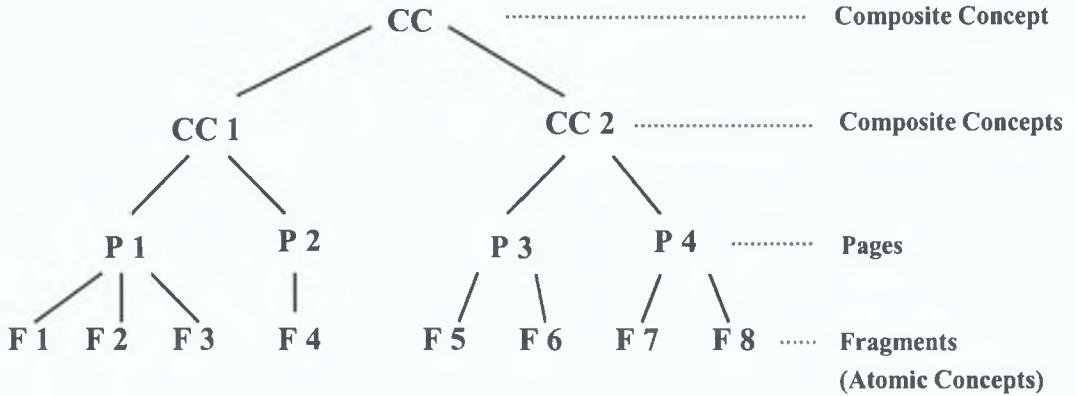


Figure 4-2 Concept Hierarchy from Domain Model

Figure 4-2 illustrates an example of concept hierarchy and shows the three levels of concepts defined in DM. The lowest level consists of fragments. The middle level consists of pages that represent the units of information presented to the user. The highest level is represented by abstract concepts.

4.2.1.2 Concept Relationships

Most AHS use relationships between concepts and concept properties in order to provide adaptive content and adaptive navigation support. AHAM provides a framework that allows the description of different types of concept relationships. In [153] a *concept relationship* is defined as an object with unique identifier and attribute-value pairs that relates a sequence of two or more concepts. The types of concept relationship considered in AHAM are:

- *Link* - suggests a navigational link between components
- *Prerequisite* - indicates a certain desirable reading order between concepts. For example concept C1 should be read before C2
- *Inhibitor* - describes an unusual type of desirable reading order. For example after visiting C4 it is no longer desirable to visit concept C1

AHAM also allows the system designers to define other types of relationships. For example “part-of” concept relationship could be used to represent the concept hierarchy.

Summarizing, the Domain Model includes concepts and concept relationships. In [152] the Domain Model of an adaptive hypermedia system is defined as a collection of

atomic concept components, page concept components, composite concept components and concept relationships.

4.2.2 User Model

The User Model (UM) in AHAM expresses individual user data such as user's preferences, background, age, and user's knowledge and interest in the concepts defined in the Domain Model. An instance of the UM is permanently stored and continuously updated by the adaptive system.

The UM is built as an overlay model of the DM. For each concept in the DM, a number of user-model attributes are defined. Therefore a UM is defined as a set of $\langle uid, uinfo \rangle$ where uid is a unique identifier of the DM concept and $uinfo$ is a set of attribute-values pairs [152]. A table or relational database is used to store the attribute values for each concept. The structure of this table is called ***user model scheme***. A table created for a specific user is called ***user model instance*** [153].

AHAM does not require the presence of a certain attributes in the UM allowing for flexibility of design. The most common attributes uses in AHS research are:

- *Knowledge* - indicates how much the user knows about a concept. Possible sets of knowledge values include: "not known", "learnt", "well-known", "well learned". The set of values for the attributes may also be represented as percentage with values between 0 and 100.
- *Read* – indicates if the user has read some information (a fragment, a page or a set of pages) related to the concept or not. The attribute values can be either Boolean values or a list of access times.
- *Ready-to-Read* – also called desirable or recommended, indicates if the user is ready or not to view information about that concept. The attribute values can be Boolean values.

The AHAM model offers to the system designer the flexibility to define his/her own set of attributes and attribute values.

Some attribute values for the UM are recalculated on the fly according to the browsing behavior of the user or to different actions performed by the user (e.g. the user has performed a test). For other attributes the value is maintained. Therefore the user model may

consist of both persistent and volatile parts. In order to update the model, the system designer defines rules in the Adaptation Model that affect attribute values related to different concepts. More details about these rules are presented in the next section.

4.2.3 Adaptation Model

The Adaptation Model (AM) provides the adaptive functionality of the AHS. The main goal of the AM is to define how the content adaptation, navigation support adaptation and the update of the user model are performed. The adaptation process uses information from the Domain Model (DM) and the User Model (UM) and the user interaction with the system.

The AM is defined as a set of *generic and specific adaptation rules* that form the connection between the DM, UM and the information to be generated and displayed to the user. A generic rule applies to all concepts or all concept relationships of a certain type. A specific rule applies to a specific concept, set of concepts or a specific concept relationship.

Both generic and specific adaptation rules are Condition-Action (CA) type rules. Unlike Event Condition Action (ECA) rules that consist of three independent parts: event, condition and action, CA rules are simpler and consist of a condition and an action only. When a rule's condition becomes true, the rule's action is executed. CA rule is like an ECA rule where the fact that the condition becomes true is the event that triggers the rule. More details about the syntax of the CA-based adaptation rules are presented in [152].

An interpreter (*adaptation engine*), for adaptation rules was also proposed. The adaptation engine is a software environment that provides the implementation-depended aspects, while DM, UM and AM describe the information and adaptation at the conceptual, implementation-independent level [153].

An execution model that describes how the adaptation engine actually selects and executes the adaptation rules was defined and more details are presented in [152].

4.3 Quality of Experience (QoE) Layer

No architectural framework has been standardized for adaptive hypermedia systems yet. However, commonalities do exist between already proposed or developed systems in the last five years. Most of the adaptive hypermedia systems follow the principles and the structure presented in the AHAM. They construct and maintain a Domain Model (DM) that

structures the information of the application domain, build user profile for each person that interacts with the system based on a User Model (UM) and perform content and/or navigation support adaptation based on Adaptation Model (AM).

These three main components DM, UM and AM are complemented by a set of interfaces, part of the adaptation engine, that control and monitor user's interaction with the system. An *User Event Tracker* tracks different user events (e.g. mouse click, page scroll, task completion, request for help, etc.) during both registration and interaction session. These events provide an important feedback from the user to the system and based on this different statistics about the user's behavior or other user's characteristics may be computed and used to update the UM. A *Behaviour Monitor* module is in charge of this.

A *Registration* interface is used to gather personal information supplied explicitly by the user. The information is used for the initialisation of the UM. An *Information Delivery Interface* generates Web pages for a user, based on the conclusions of the Adaptation Model, combining atomic units selected from the Domain Model. All these components of a general Adaptive Hypermedia System (AHS) architecture and the cooperation between them are presented in Figure 4-3.

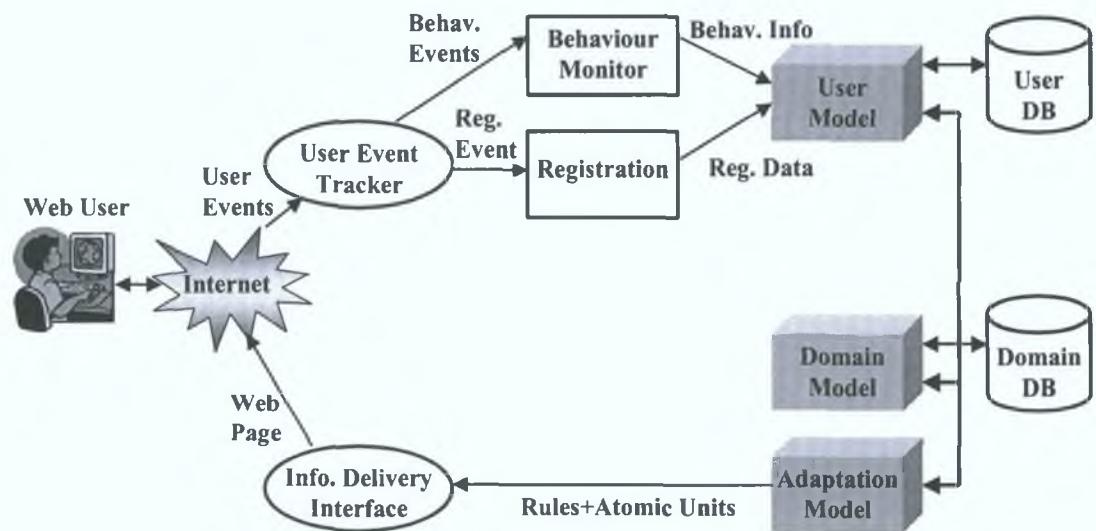


Figure 4-3 A Generic Architecture for Adaptive Hypermedia Systems

Unfortunately this generic AHS architecture does not take into consideration the end-user perceived performance features that have a substantial impact on user Quality of Experience (QoE). AHS brings many advantages in terms of easier access to the available information for a particular user by tailoring content and navigation support, increase in the user satisfaction and experience with the system and higher user motivation to continue to

enhance his knowledge. However, all these benefits are cancelled when the user operational environment and/or the network condition do not support the delivery of personalized content due to high load or eventual variations in traffic. These may significantly affect users' QoE in spite of the best selection of the delivered content.

Therefore, AHS should also both take into consideration QoE characteristics when building the user profile and regularly monitor in real-time any change in the system that might trigger modifications in QoE. These changes include variations in the user's operational environment and user behaviour that might possibly indicate dissatisfaction with service. Taking all these into account would allow for better Web content adaptation that suits varying delivery conditions.

In this context an enhancement of the generic AHS architecture with capabilities to analyse end-user QoE and possible factors that could affect it is desirable. Therefore the work presented in this thesis proposes a QoE Layer - an enhancement to the generic AHS architecture - that consists mainly of two new components: Performance Monitor (PM) and the Perceived Performance Model (PPM) (see Figure 4-4).

The *PM* monitors and measures in real-time different performance metrics that may either affect or indicate the user QoE such as end-to-end response time, round-trip time, throughput, user tolerance to delay and the user's behaviour. The latter could indicate approval or dissatisfaction with the service (e.g. abort actions).

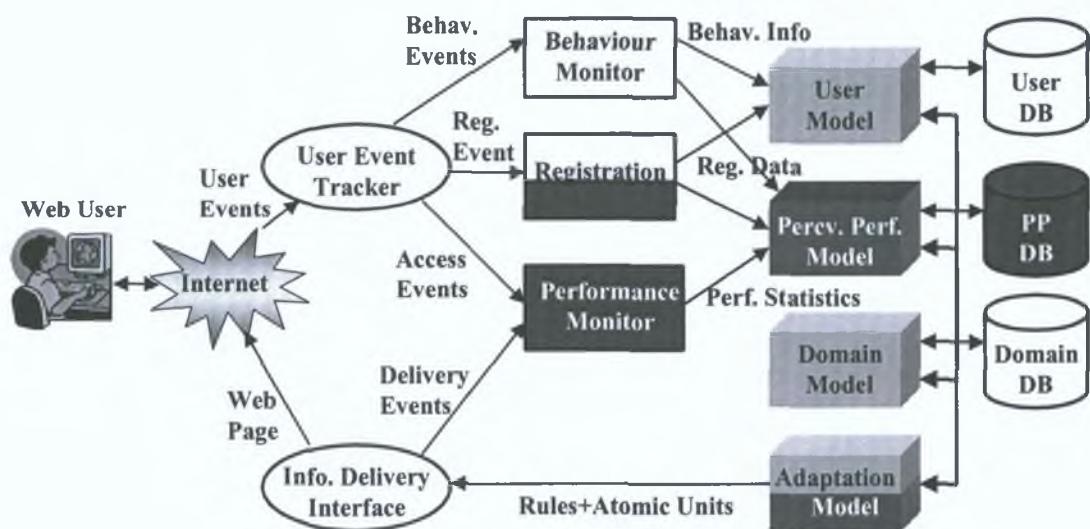


Figure 4-4 QoE-aware Adaptive Hypermedia System Architecture

The *PPM* provides a dynamic representation of user satisfaction related to the perceived QoE, during the delivery of Web content. The model is analogous to the User Model, storing information gathered about the user, but in this case, from the perceived performance point of view. It models these characteristics and generates constraints or suggestions related to the content to be served (e.g. number of images, size of images, overall number of bytes to be sent). The content related suggestions are based on the predicted impact on performance and on the estimation of the resulting effect on the user satisfaction. A stereotype-based technique is used to construct and infer information about the user.

For the initialisation of the PPM, the *Registration* process is enhanced with extra capabilities in order to retrieve from the user more information related to their subjective assessment of performance issues such as connectivity type, preferred level of embedded components, their quality, and the user's subjective perception of the registration process. The *Adaptation Model* is also enhanced for a better adaptation of the Web content integrating knowledge of user interests with performance constraints. Thus, the enhanced AM ensures the co-operation between the personalisation (the important feature of any AH system) and performance based adaptation.

More details about the QoE layer related components are provided in the next sections.

4.3.1 Perceived Performance Model

The Perceived Performance Model (PPM) has the important function of providing a dynamical representation of user satisfaction related to the perceived QoE. Even if the delivered information is tailored according to the user's interests, the user will not be satisfied if the information is delivered too slowly. The access time to the information is mainly affected by type of client's connection, backbone, network traffic etc.

The problem of user satisfaction related to the his/her Web experience has been studied by Human Computer Interface research while the problem of performance related factors and their impact on the end-to-end performance has been analysed as part of Web Quality of Service (QoS) research. Results of some of these studies were modelled, designed and incorporated in the proposed PPM. A stereotype-based technique was used to construct and infer information about the user.

Unlike the UM which focuses on users' interests, preferences and goals, the PPM deals with end-user perceived performance (QoE). Different user-related characteristics reflect the QoE such as: access (download) time, client connection type, transfer throughput etc. The PPM models these characteristics via performance metrics and suggests constraints on the characteristics of the content to be served (e.g. number of embedded objects in the Web page, dimension of the base-Web page without components and the total dimension of the embedded components). These suggestions are made based on the predicted impact on QoE and on the estimation of the resulting effect on the user satisfaction. The goal is to best meet the end-user expectation related to the provided QoS. These constraints are applied to the Web pages that have already been designed according to the user profile (based on the UM and AM). The PPM model also takes into consideration the users' subjective opinion about their QoE explicitly expressed by the users. This introduces a degree of subjective assessment, which is specific to each user.

PPM includes two phases:

- *Initialisation* of the model is performed during first user access to the Web server and uses information collected during the registration process.
- The *Update phase* is performed in real-time at every user access to the site, dynamically taking into consideration any variations in the user perceived performance as indicated by their behaviour and/or network-related characteristics.

For the design of the PPM a stereotype-based structure is used.

4.3.1.1 Stereotype - Based Structure

Stereotypes for maintaining models of users were first introduced in the seventies by Rich in the Grundy system [154] and they are still widely used by many AHS [23, 35, 60]. A stereotype contains assumptions about characteristics of a subgroup of users and is defined through a list of attributes, attributes values and value estimates.

The PPM uses a method based on stereotypes in order to build user profiles and to infer additional information about user perceived performance characteristics.

The PPM consists of a collection of stereotypes $T = (T_1, T_2, \dots, T_h, \dots, T_l)$. A stereotype characterizes a class of users with similar perceived performance features. The

collection of the stereotypes is increasingly ordered according to the quality of the perceived performance associated to these stereotypes. Each stereotype T_h consists of two components:

- *a group of features*¹ ($F = (F_1, F_2, \dots, F_i, \dots, F_n)$) that characterizes the stereotype
- *a group of suggestions*¹ ($S = (S_1, S_2, \dots, S_j, \dots, S_m)$) related to the web content parameters, that should be performed in order to optimise the client's perceived performance.

Each feature F_i has associated a pre-defined ordered list of linguistic terms (attributes values) $LF_i = (LF_{i1}, LF_{i2}, \dots, LF_{ik_i}, \dots, LF_{ip_i})$. Each linguistic term LF_{ik_i} has assigned a numeric value (value estimate) PF_{ik_i} , between 0 and 1, representing the probability that the feature F_i is equal to the linguistic term LF_{ik_i} for this stereotype T_h . (i.e. $PF_{ik_i} = P(F_i = LF_{ik_i} | T_h)$). For example, a feature F_i = “performance” may have associated the following list of pairs LinguisticTerm – ProbabilisticValue: [(bad, 0.2), (normal, 0.6), (excellent, 0.3)].

A similar structure is defined for each suggestion S_j , which has associated the linguistic terms $LS_j = (LS_{j1}, LS_{j2}, \dots, LS_{jr_j}, \dots, LS_{jq_j})$ and probabilistic values $PS_j = (PS_{j1}, PS_{j2}, \dots, PS_{jr_j}, \dots, PS_{jq_j})$. The structure of the stereotype T_h is presented in Table 4-1 (group of features) and Table 4-2 (group of suggestions).

Table 4-1 Group of Features for a Stereotype

FEATURE	LIST (LINGUISTIC TERM – PROBABILITY)
F_1	$(LF_{11}-PF_{11}), (LF_{12}-PF_{12}), \dots, (LF_{1k_1}-PF_{1k_1}), \dots, (LF_{1p_1}-PF_{1p_1})$
F_2	$(LF_{21}-PF_{21}), (LF_{22}-PF_{22}), \dots, (LF_{2k_2}-PF_{2k_2}), \dots, (LF_{2p_2}-PF_{2p_2})$
...	...
F_i	$(LF_{i1}-PF_{i1}), (LF_{i2}-PF_{i2}), \dots, (LF_{ik_i}-PF_{ik_i}), \dots, (LF_{ip_i}-PF_{ip_i})$
...	...
F_n	$(LF_{n1}-PF_{n1}), (LF_{n2}-PF_{n2}), \dots, (LF_{nk_n}-PF_{nk_n}), \dots, (LF_{np_n}-PF_{np_n})$

¹ Group of Features and Group of Suggestions are not ordered

Table 4-2 Group of Suggestions for a Stereotype

SUGGESTION	LIST (LINGUISTIC TERM – PROBABILITY)
S1	(LS11-PS11), (LS12-PS12), ..., (LS1r _j -PS1r _j), ..., (LS1q _l -PS1q _l)
S2	(LS21-PS21), (LS22-PS22), ..., (LS2r _j -PS2r _j), ..., (LS2q ₂ -PS2q ₂)
...	...
S _j	(LSj1-PSj1), (LSj2-PSj2), ..., (LSjr _j -PSjr _j), ..., (LSjq _j -PSjq _j)
...	...
S _m	(LSm1-PSm1), (LSm2-PSm2), ..., (LSmr _j -PSmr _j), ..., (LSmq _m -PSmq _m)

In the group of features, the probabilities PF_{ik_i} are used to determine the degree of match between the user's characteristics and the stereotype. In the group of suggestions, the $PSjr_j$ indicates the strength of the corresponding suggestion.

4.3.1.2 Determination of Probabilistic Values Associated with Linguistic Terms

The Poisson distribution $poisd(x, u)$ (Equation 4-1) for discrete events and its continuous version $pois(x, u)$ (Equation 4-2) are used for the determination of the probabilities values associated with the linguistic terms from the stereotypes classes associated with the PPM. Poisson function was selected since it provides the correct shape I was looking for the model.

The Poisson distribution models the number of events occurring within a given time interval. It is used to assign probabilities to a number of events. The "u" parameter is the shape parameter and it is the mean and the variance of the distribution in the given time interval. The "x" parameter identifies a particular event and is represented by an integer = 0, 1, 2, 3, ... n.

$$poisd(x, u) = \frac{u^x \cdot \exp(-u)}{x!}$$

Equation 4-1

Noting that $\gamma(x+1) = x!$, a continuous version of the Poisson distribution can be written as in Equation 4-2, where x is a positive real number.

$$pois(x, u) = \frac{\exp(x \log(u)) \cdot \exp(-u)}{gamma(x + 1)}$$

Equation 4-2

Gamma is a continuous function defined as in Equation 4-3.

$$gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt$$

Equation 4-3

The plot that involves the continuous version of the Poisson function $pois(x, u)$ function for different mean values $u=1, 3, 5, 7, 9, 11$ is shown in Figure 4-5.

Analysing the shape of the Poisson function, it can be noticed that for $u=7$ the Poisson function has almost a normal distribution across the $[0, 15]$ interval with a maximum value close to the middle of the interval ($pois(7, 7)=0.15$) and the minimum values (close to zero) for $x=0$ and $x=15$. In consequence, the interval $[0, 15]$ was considered for the computation of the Poisson function for the whole set of stereotypes.

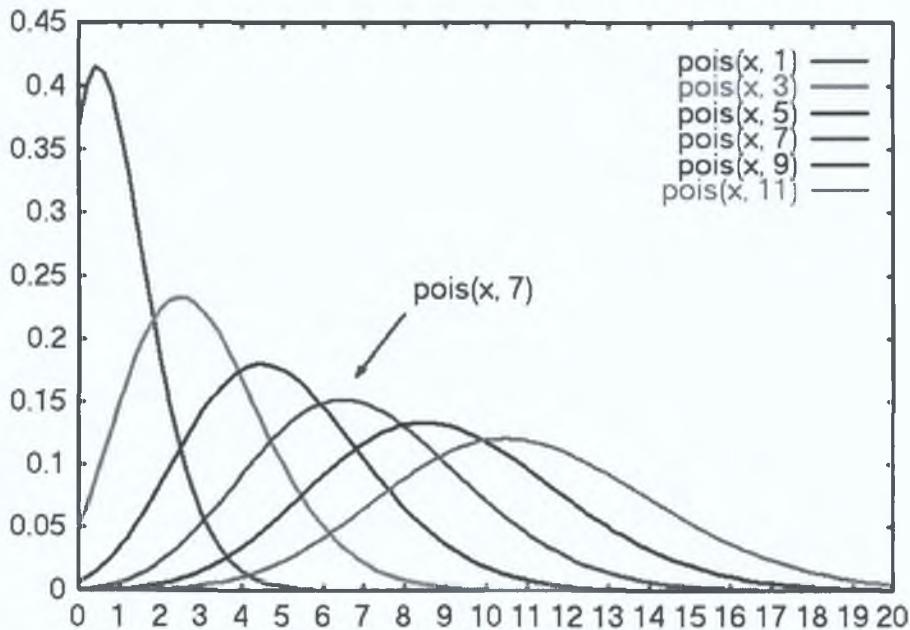


Figure 4-5 Poisson Distribution for Different Mean Values “u”

Let's consider that the PPM consists of a collection of Tl stereotypes where “ Tl ” is an even number. The probability values associated with the linguistic terms for the $Tl/2$

stereotype follow the $\text{pois}(x,7)$ distribution. The other stereotypes have associated probability values based on a Poisson distribution determined as in Equation 4-4 and Equation 4-5:

- for all stereotypes T_h , where $1 \leq h < \lceil l/2 \rceil$

$$\text{pois}_{T_h}(x, u_h) = \frac{\exp(x \log(u_h)) \cdot \exp(-u_h)}{\text{gamma}(x+1)}$$

$$u_h = \frac{15}{2l} + (h-1)\frac{15}{l}; \quad 1 \leq h < \lceil l/2 \rceil$$

Equation 4-4

- for all stereotypes T_k where $\lceil l/2 \rceil < k \leq l$

$$\text{pois}_{T_k}(15-x, u_k) = \frac{\exp((15-x) \log(u_k)) \cdot \exp(-u_k)}{\text{gamma}((15-x)+1)}$$

$$u_k = \frac{15}{2l} + (l-k)\frac{15}{l}; \quad \lceil l/2 \rceil < k \leq l$$

Equation 4-5

Figure 4-6 presents an example for the case when PPM consists of 5 stereotypes ($l=5$). One could notice that each of the stereotypes T_h , $h = \overline{1,5}$ has associated one Poisson distribution with a mean value u_k . This value is obtained by dividing the interval $[0,15]$ in five equal segments and considering their middle value. This computation exemplifies Equation 4-4 and Equation 4-5 for $l=5$. It can be observed that as u_k gets closer to zero the peak value of Poisson function increases. This increase is due to the behaviour of the Poisson function.

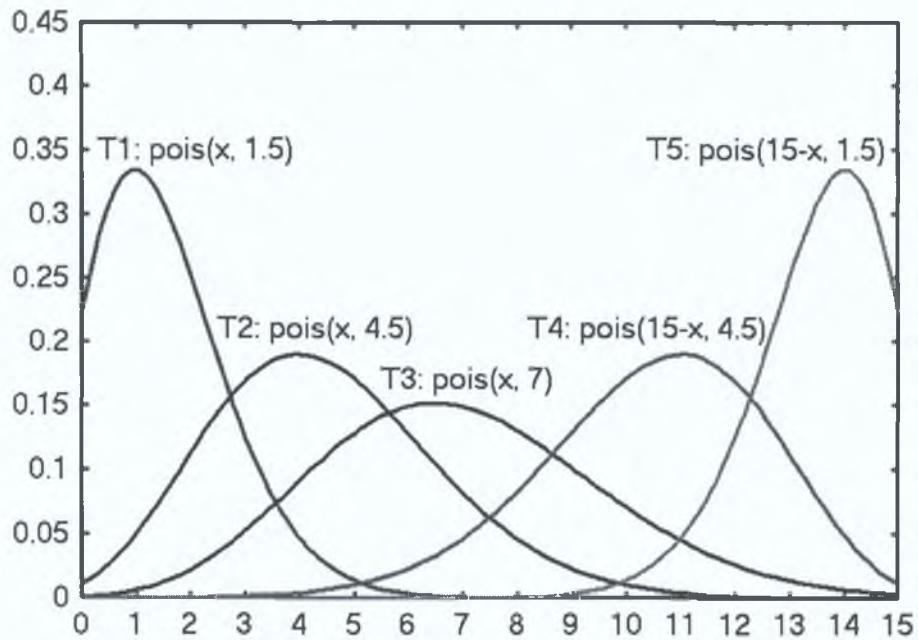


Figure 4-6 Poisson Distribution for Five Stereotypes

Next, the computation of the probabilistic values that are associated with the linguistic terms from the features F_i $i = \overline{1, n}$ of each stereotype T_h is described.

Considering the list of linguistic terms of lenght q and the Poisson distribution $\text{pois}_{T_h}(x, u_h)$ associated with the stereotype T_h , the probabilistic values (PF_{ij} where $j = \overline{1, q}$) are computed as follows:

$$\begin{aligned} PF_{ij} &= \text{avg}(\text{pois}_{T_h}(x_j, u_h)); \\ x_j &\in [step * (j - 1), step * j] \\ step &= \left\lceil \frac{15}{q} \right\rceil \end{aligned}$$

Equation 4-6

Figure 4-7 presents an exemplification for a feature F_i of the stereotype T_3 that has associated $q=5$ linguistic terms. $\text{pois}(x, 7)$ distribution is used for the computation of the probabilistic values. Table 4-3 presents the computed values associated with the linguistic terms after the normalisation process. These values are also plotted in the Figure 4-7. Thus, the process is repeated for each feature F_i of each stereotype T_h where $h = \overline{1, 5}$, in this example.

Table 4-3 Probabilistic Values Of The Linguistic Terms Associated With Feature F_i From The Stereotype T3

X INTERVAL	AVG(POIS _{T3} (X _i ,7))	NORMALISED VALUES (PROBABILISTIC VALUES)	LINGUISTIC TERMS
[0, 3]	0.017	0.053	LF _{i1}
[3,6]	0.107	0.272	LF _{i2}
[6,9]	0.136	0.419	LF _{i3}
[9,12]	0.059	0.229	LF _{i4}
[12,15]	0.011	0.034	LF _{i5}

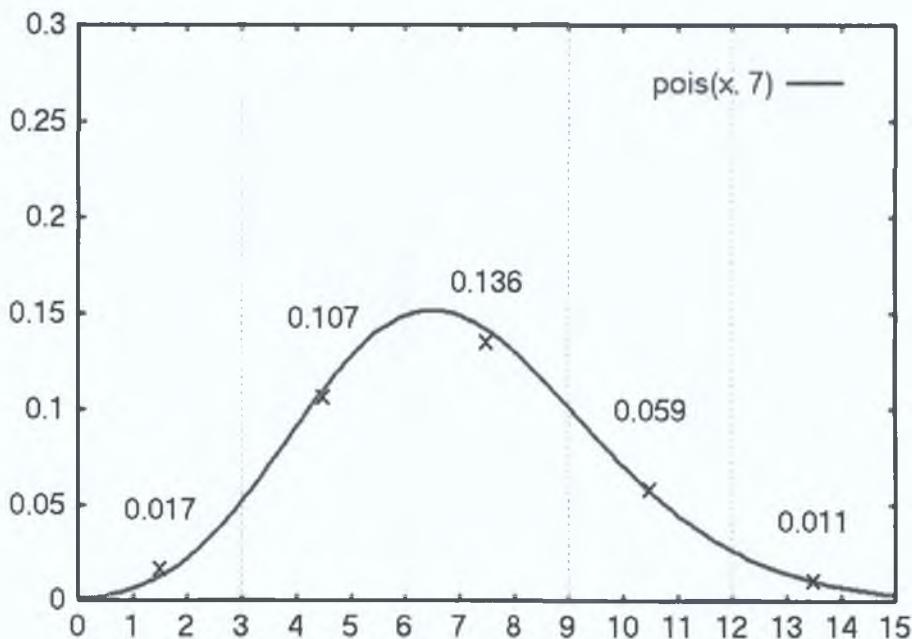


Figure 4-7 Probabilistic Values associated with q=5 Linguistic Terms of a Feature F_i from the Stereotype T3

The same computation mechanism is applied for the probabilistic values associated with the linguistic terms of a suggestion S_i , $i = 1, m$ of the same stereotype T_h .

4.3.1.3 User Classification

The goal of the classification process is to determine the stereotype classes the user belongs to and with what probability. Therefore, a degree of match between a user, characterised by a list of feature-linguistic term pairs as indicated in Equation 4-7 and each stereotype from the PPM's collection is computed as in Equation 4-8.

$$U = ((F1, LF1k_1), (F2, LF2k_2), \dots, (Fn, LFnk_n))$$

Equation 4-7

$$\begin{aligned} M(Th) &= p(Th | F1 = LF1k_1, \dots, Fn = LFnk_n) = \\ &= p(Th | F1 = LF1k_1) * \dots * p(Th | Fn = LFnk_n) \end{aligned}$$

Equation 4-8

Equation 4-8 is derived from the probability theory, assuming that all stereotypes contain the same set of features, where $p(Th | Fi = LFik_i)$ represents the a-priori probabilistic value that feature F_i from the stereotype Th to be equal with the linguistic term $LFik_i$.

Each factor from Equation 4-8 can be computed separately using the Bayes' rule as in Equation 4-9.

$$p(Th | Fi = LFik_i) = \frac{p(Fi = LFik_i | Th) * p(Th)}{p(Fi = LFik_i)} = \frac{PFik_i * p(Th)}{p(Fi = LFik_i)}$$

Equation 4-9

In Equation 4-9 $p(Th)$ represents the a-priori probability distribution of the stereotypes in the PPM and $p(Fi = LFik_i)$ represents the a-priori probability distribution of the values $LFik_i$.

Currently, it is considered that in the model all the stereotypes of the PPM have the same probability distribution, which means that the probability a client belongs to any of the stereotype classes is equal with $1/\text{NoStereotypes}$. However, different probability distributions could be associated to each stereotype according to the population of the users that have been included in each stereotype class, allowing for higher flexibility.

Since all the stereotypes contain the same set of features and the same set of linguistic terms for each feature, $p(Fi = LFik_i)$ can be regarded as a normalizing factor, which does not influence the belonging of the user to a stereotype class or another.

In conclusion, a set of degrees of match between user's characteristics and the pre-defined stereotypes from our model is obtained. All these values are normalized and the

probabilities the user belongs to each of the stereotype classes are determined. Later on, the PPM will use these probabilities in order to suggest content constraints.

4.3.1.4 Suggestion Determination

After the user was classified in one or more stereotype classes and the degree of match for each stereotype was computed, the adaptation measures that have to be performed in order to improve the user QoE are determined. For this, the groups of suggestions associated to the stereotypes the user belongs to are used. Since the stereotypes' suggestions are similar, but with different strengths, the final constraints that have to be imposed are determined after a process of weighted merge of these suggestions.

First, for each stereotype the strengths of the suggestions have to be computed taking into account the probability for the user to belong to this stereotype class. This computation is shown in Equation 4-10.

$$\begin{aligned} p'(Si = LSik_i | Th) &= p(Si = LSik_i | Th) * M(Th) \\ PS'ik_i(Th) &= PSik_i(Th) * M(Th) \end{aligned}$$

Equation 4-10

Then, the additive formula presented in Equation 4-11 from probabilistic theory is used to combine the stereotypes suggestions. In Equation 4-11 $P(E_1)$ is the probability of an event E_1 to appear, $P(E_2)$ is the probability of an event E_2 to appear and $P(E_1 \& E_2)$ represents the probability that both events E_1 and E_2 to appear. Similarly, the formula can be applied for merging the probabilities in order to determine the probability for the events E_1, E_2, \dots, E_n to appear simultaneously.

$$\begin{aligned} p(E_1 \& E_2) &= p(E_{12}) = p(E_2) + [1 - p(E_2)] * p(E_1) \\ p(E_1 \& E_2 \& E_3) &= p(E_3) + [1 - p(E_3)] * p(E_{12}) \\ p(E_1 \& E_2 \& \dots \& E_n) &= p(E_n) + [1 - p(E_n)] * p(E_{1\dots n-1}) \end{aligned}$$

Equation 4-11

For the weighted combination of the stereotypes' suggestions it was assumed that the stereotypes have independent contributions to the final group of suggestions.

Equation 4-12 presents an example of the computation of the strength of the suggestion Si to be equal to the linguistic term $LSik_i$ in case that the user belongs to two stereotypes $T1$ and respectively $T2$.

$$\begin{aligned}
 p'(Si = LSik_i | T1) &= PS^1 ik_i \\
 p'(Si = LSik_i | T2) &= PS^2 ik_i \\
 p(Si = LSik_i | T1 \& Si = LSik_i | T2) &= \\
 &= p'(Si = LSik_i | T2) + [1 - p'(Si = LSik_i | T2)] * p'(Si = LSik_i | T1) = \\
 &= PS^2 ik_i + (1 - PS^2 ik_i) * PS^1 ik_i
 \end{aligned}$$

Equation 4-12

4.3.1.5 Initialisation Phase

Initialisation of the PPM model is performed during first user access to the Web server and uses information collected during the registration process. This information is collected in two ways:

- implicitly - the Performance Monitor computes the values of some performance features, while the registration form is sent and visualized by the user;
- explicitly – questions about the client's preferred performance related issues are added to the registration form and subjective user opinion is collected. Default values for answers to the questions are considered in the case that the user does not want to perform the registration process.

The initialisation phase involves the creation of an initial model for a new user characterised by a set of pair feature-linguistic term $(Fi, LFiki) \in U$, based on stereotypical knowledge. It makes use of the pre-defined classes of users (stereotypes) that follow the structure presented in section 4.3.1.1 and provides a first set of suggestions related to the content using the classification algorithm described in sections 4.3.1.3 and 4.3.1.4.

4.3.1.6 Update Phase

The suggestions made by the stereotypes in the initialisation phase cannot be very exact due to the limited set of data used for the classification and to the limited number of stereotypes. Therefore, the goal of the second phase is to provide more accurate suggestions as each user interacts more with the system. This phase also takes into consideration any

changes in the client's satisfaction during his/her navigation on the site and the effect of the changes in the network conditions.

The update phase is based on tracking both different users' behaviours that suggest their satisfaction with the site and network events that could affect clients' perceived performance. Based on the tracked information the information stored by the PPM related to these users is dynamically revised during this update phase, which is described next.

The update phase is based on a six-step algorithm:

- 1) User's behavior events that reflect his/her satisfaction are tracked and user related perceived performance metrics are measured.
- 2) For each feature F_i of the stereotype Th a mask $Mask_{F_i}$ that reflects the user-specific changes (information gathered in step 1) is applied on the probability values PF_{ik_i} that are associated with the linguistic values LF_{ik_i} . The newly obtained probabilities are then normalized.
- 3) The process from step 2) is repeated for each stereotype of the PPM.
- 4) Re-classification of the user in one or more of the pre-defined stereotype classes is performed
- 5) The final suggestions are determined by weighting the merge of the suggestions given by the stereotype classes the user belongs to.
- 6) The current suggestions are combined with historical suggestions (generated by the PPM during the previous accessed Web pages) to determine the adaptation measures to be taken. This prevents short-term fluctuations in the network conditions from affecting the adaptation process.

The computation of the mask ($Mask_{F_i}$) for a feature F_i from stereotype Th combines some *pre-defined basic-masks* $BMask_{EM_j}$ that correspond to different user behaviour events tracked by the Behaviour Monitor and values of other performance metrics (EM_j) (different than the ones used in the stereotypes) that are measured by the Performance Monitor. The length of the mask $Mask_{F_i}$ associated with the feature (F_i) is equal with the total number of linguistic terms associated to the feature F_i (Equation 4-13).

$$\text{Length } (\text{Mask } F_i) = l_i$$

Equation 4-13

The application of the masks on a stereotype can have two types of influence: “positive” and “negative”. The first one increases the probability for a client to be classified in stereotype classes that are associated with good characteristics and decreases the probability that it will be classified in classes with bad characteristics. The immediate effect of a positive mask is that suggestions from better stereotype classes are more likely to be applied. A “negative” effect decreases the chances for a user to be classified in the stereotypes defined for good perceived performance characteristics, in consequence decreasing also the probability for the suggestions from better stereotype classes to be applied.

These effects are “imposed” by the basic-masks that are classified in two categories: “positive” B-Masks and “negative” B-Masks. The first ones correspond to the events that show an increase in the user’s satisfaction, whereas the “negative” ones are associated to events that show decreases in the user’s satisfaction. Examples of events associated with “positive” B-Masks are: the percentage of the embedded objects accessed from the cache (with the increase in number of accessed objects from the cache, the download time decreases [155]); number of sessions opened with the site (a big number means that the user was satisfied with the provided service). Examples of events that are associated with “negative” B-Masks are: number of aborted page accesses performed by the user (an aborted action reflects a bad quality download [155]); numbers of accesses during a session (user tolerance to the delay decreases with the increase in the time spent interacting with the system denoted as session length [135, 137]).

A B-Mask is represented by a one-dimensional matrix (Equation 4-14). Since all stereotypes have the same set of features and linguistic terms, the length and the values of the B-Masks computed for a feature will be the same for all stereotypes. The values of the B-Masks are different according to the type of the event they are associated to.

$$\begin{aligned} \text{BMask}_{EM\ j} &= [m_1, m_2, \dots, m_t] \\ t &= l_i / 2 \end{aligned}$$

Equation 4-14

The B-Mask $B\text{Mask}_{EM_j}$ will be added to the mask Mask_{F_i} associated to the feature F_i using the following rules:

- 1) if the B-Mask is a “negative” one then the highest probabilistic value (PF_{ik_i}) of the feature F_i will be determined and the B-Mask will be added to the Mask_{F_i} from left to right starting from the position that corresponds with PF_{ik_i} .
- 2) if the B-Mask is a “positive” one then the highest probabilistic value (PF_{ik_i}) of the feature will be determined and the B-Mask will be added to the Mask_{F_i} from right to left starting from the position that corresponds with PF_{ik_i} .

4.3.2 Illustrative Example of the Perceived Performance Model

Five stereotype classes (“Bad”, “Poor”, “Normal”, “Good” and “Excellent”) that model the perceived performance of the users are considered and illustrated in Table 4-4 to Table 4-13. Table 4-4 and Table 4-5 present the groups of features and group of suggestions for the “Bad” stereotype class. Table 4-6 and Table 4-7 present the groups of features and group of suggestions for the “Poor” stereotype class, etc.

Each stereotype class has the same structure as the one defined in Table 4-1 and Table 4-2. Four performance parameters that can influence the users’ perceived performance, are considered as features used by the model: download time (DT), client’s throughput (T), round-trip-time (RTT) and perceived performance as suggested by client (PPf). For the simplification of the model computation, these features are considered to be independent. Many researchers in the area of Web QoS consider the first three features most important from the performance point of view. The latter introduces a degree of subjective assessment specific to each user. The probabilistic value associated with each linguistic term was computed according to the specifications from section 4.3.1.2.

The DT represents the main parameter of the end-to-end performance and it is perceived directly by the users. According to its value the client satisfaction may change. Studies on the user tolerance to the DT have shown that user’s expectations are influenced by different contextual factors such as the type of task performed by the user and the duration of time the user interacts with the site [135]. Currently, there is no standard threshold for the download time. However, on average a download time higher than 10-12 sec causes disruption and users lose their attention to the task, while values higher than 30 sec cause frustration. At the same time it is significant to mention that when the user is aware of his slow connection, he/she is willing to tolerate a threshold of 15 sec.

The five stereotype classes cover a wide range of Web users that may access the adaptive system from home. They may have a variety of connection types, from very low connections such as modem-based to ISDN or even higher speed connections (over 128 kbps). These users may be located all over the world having diverse service delays. In consequence, the round-trip time (RTT) will range from less than 100 msec (e.g. the user and the server are located close to each other) to bigger than 500 msec (when the distance between the server and client has high values).

All five stereotypes have the same list of features and linguistic terms associated with each feature. The differences between the stereotypes are the probability values associated with the linguistic terms. For example, for a user that belongs to the "Bad" class (Table 4-4), the probability that he/she will perceive a download time (DT) greater than 15 sec is considered to be 86 %, the probability for the DT to be between 8 and 12 seconds is 1 % and the probability for the DT less than 5 sec is 0 %.

Content suggestions such as the number of embedded objects in the Web page (No.Ojbs), the dimension of the based Web page, (Dim.Page) and the total dimension of the embedded components (Dim.Ojbs) were used as suggestion features for the five stereotypes. These three factors represent the amount of content in a Web page, which highly affects the access to a page by a user and consequently his/her QoE [156].

Table 4-4 Group of Features for Stereotype "Bad"

FEATURE	LIST (LINGUISTIC TERM – PROBABILITY)
DT (s)	(>15, 0.86), (15-12, 0.13), (12-8, 0.01), (8-4, 0), (<4, 0)
T (Kbps)	(<28, 0.86), (28 -56, 0.13), (56-64, 0.01), (64-128, 0), (>128, 0)
RTT (ms)	(>500, 0.86), (500-300, 0.13), (300-200, 0.01), (200-100, 0), (<100, 0)
PPf	(Bad, 0.86), (Poor, 0.13), (Normal, 0.01), (Good, 0), (Excellent, 0)

Table 4-5 Group of Suggestions for Stereotype "Bad"

SUGGESTION	LIST (LINGUISTIC TERM – PROBABILITY)
No. Obj	(0-3, 0.86), (4-6, 0.13), (7-13, 0.01), (14-18, 0), (>18, 0)
Dim. Obj. (KB)	(0-10, 0.86), (10 - 20, 0.13), (20-50, 0.01), (50 -80, 0), (>80, 0)
Dim. Page (KB)	(0-5, 0.86), (5 - 10, 0.13), (10-15, 0.01), (15 -20, 0), (>20, 0)

Table 4-6 Group of Features for Stereotype "Poor"

FEATURE	LIST (LINGUISTIC TERM – PROBABILITY)
DT (s)	(>15, 0.25), (15-12, 0.52), (12-8, 0.2), (8-4, 0.03), (<4, 0)
T (Kbps)	(<28, 0.25), (28 -56, 0.52), (56-64, 0.2), (64-128, 0.03), (>128, 0)
RTT (ms)	(>500, 0.25), (500-300, 0.52), (300-200, 0.2), (200-100, 0.03), (<100, 0)
PPf	(Bad, 0.25), (Poor, 0.52), (Normal, 0.2), (Good, 0.03), (Excellent, 0)

Table 4-7 Group of Suggestions for Stereotype "Poor"

SUGGESTION	LIST (LINGUISTIC TERM – PROBABILITY)
No. Obj	(0-3, 0.25), (4-6, 0.52), (7-13, 0.2), (14-18, 0.03), (>18, 0)
Dim. Obj. (KB)	(0-10, 0.25), (10 - 20, 0.52), (20-50, 0.2), (50 -80, 0.03), (>80, 0)
Dim. Page (KB)	(0-5, 0.25), (5 - 10, 0.52), (10-15, 0.2), (15 -20, 0.03), (>20, 0)

Table 4-8 Group of Features for Stereotype "Normal"

FEATURE	LIST (LINGUISTIC TERM – PROBABILITY)
DT (s)	(>15, 0.05), (15-12, 0.27), (12-8, 0.42), (8-4, 0.23), (<4, 0.03)
T (Kbps)	(<28, 0.05), (28 -56, 0.27), (56-64, 0.42), (64-128, 0.23), (>128, 0.03)
RTT (ms)	(>500, 0.05), (500-300, 0.27), (300-200, 0.42), (200-100, 0.23), (<100, 0.03)
PPf	(Bad, 0.05), (Poor, 0.27), (Normal, 0.42), (Good, 0.23), (Excellent, 0.03)

Table 4-9 Group of Suggestions for Stereotype "Normal"

SUGGESTION	LIST (LINGUISTIC TERM – PROBABILITY)
No. Obj	(0-3, 0.05), (4-6, 0.27), (7-13, 0.42), (14-18, 0.23), (>18, 0.03)
Dim. Obj. (KB)	(0-10, 0.05), (10 - 20, 0.27), (20-50, 0.42), (50 -80, 0.23), (>80, 0.03)
Dim. Page (KB)	(0-5, 0.05), (5 - 10, 0.27), (10-15, 0.42), (15 -20, 0.23), (>20, 0.03)

Table 4-10 Group of Features for Stereotype "Good"

FEATURE	LIST (LINGUISTIC TERM – PROBABILITY)
DT (s)	(>15, 0.0), (15-12, 0.03), (12-8, 0.20), (8-4, 0.52), (<4, 0.25)
T (Kbps)	(<28, 0.0), (28 -56, 0.03), (56-64, 0.20), (64-128, 0.52), (>128, 0.25)
RTT (ms)	(>500,0.0),(500-300, 0.03),(300-200, 0.20),(200-100, 0.52),(<100, 0.25)
PPf	(Bad, 0.0), (Poor, 0.03), (Normal, 0.20), (Good, 0.52), (Excellent, 0.25)

Table 4-11 Group of Suggestions for Stereotype "Good"

SUGGESTION	LIST (LINGUISTIC TERM – PROBABILITY)
No. Obj	(0-3, 0.0), (4-6, 0.03), (7-13, 0.20), (14-18, 0.52), (>18, 0.25)
Dim. Obj. (KB)	(0-10, 0.0), (10 - 20, 0.03), (20-50, 0.20), (50 -80, 0.52), (>80, 0.25)
Dim. Page (KB)	(0-5, 0.0), (5 - 10, 0.03), (10-15, 0.20), (15 -20, 0.52), (>20, 0.25)

Table 4-12 Group of Features for Stereotype "Excellent"

FEATURE	LIST (LINGUISTIC TERM – PROBABILITY)
DT (s)	(>15, 0.0), (15-12, 0.0), (12-8, 0.01), (8-4, 0.13), (<4, 0.86)
T (Kbps)	(<28, 0.0), (28 -56, 0.0), (56-64, 0.01), (64-128, 0.13), (>128, 0.86)
RTT (ms)	(>500,0.0),(500-300, 0.0),(300-200, 0.01),(200-100, 0.13),(<100, 0.86)
PPf	(Bad, 0.0), (Poor, 0.0), (Normal, 0.01), (Good, 0.13), (Excellent, 0.86)

Table 4-13 Group of Suggestions for Stereotype "Excellent"

SUGGESTION	LIST (LINGUISTIC TERM – PROBABILITY)
No. Obj	(0-3, 0.0), (4-6, 0.0), (7-13, 0.01), (14-18, 0.13), (>18, 0.86)
Dim. Obj. (KB)	(0-10, 0.0), (10 - 20, 0.0), (20-50, 0.01), (50 -80, 0.13), (>80, 0.86)
Dim. Page (KB)	(0-5, 0.0), (5 - 10, 0.0), (10-15, 0.01), (15 -20, 0.13), (>20, 0.86)

A user that has the performance characteristics illustrated in Equation 4-15 was considered for the exemplification of the classification process and the determination of the PPM suggestions related to the web page characteristics. The user U experienced a download time of 12 sec while accessing the web page via a 56 kbps connection with the measured round-trip-time 310 msec and considered his/her QoE as “normal”.

$$U = \{(DT, 12s), (T, 56Kbps), (RTT, 310ms), (PPf, Normal)\}$$

Equation 4-15

The first step in the classification process is to determine the degree of match between these user characteristics and each stereotype of the model. This is done by using Equation 4-8. The only match values bigger than zero were obtained for the stereotypes “Poor”, “Normal” and “Good”. After the normalisation of the matching values we got the following probabilistic values for the client to belong to these classes: Stereotype “Bad” – 0.01 %, Stereotype “Poor” – 45.61 %, Stereotype “Normal” – 54.23 %, Stereotype “Good” – 0.15 %. The suggestions strengths for the stereotypes the user belongs to have to be computed using Equation 4-10 by taking into account the degree of match with each of these stereotype classes.

The next step consists of merging the suggestions associated with these four stereotypes using Equation 4-11 and a final list of suggestions for the current user is determined. Table 4-14 presents the results after normalization.

Table 4-14 Final Group of Suggestions

SUGGESTION	LIST (LINGUISTIC TERM – PROBABILITY)
No. Obj	(0-3, 0.15), (4-6, 0.37), (7-13, 0.32), (14-18, 0.14), (30-18, 0.02)
Dim. Obj. (KB)	(0-10, 0.15), (10 - 20, 0.37), (20-50, 0.32), (50 -80, 0.14), (200-80, 0.02)
Dim. Page (KB)	(0-5, 0.15), (5 - 10, 0.37), (10-15, 0.32), (15 -20, 0.14), (30-20, 0.02)

The suggestions generated by the PPM represent content properties that would offer the highest client perceived performance. For example, No.Objs gives suggestions with different probabilities related to the number of objects to be embedded in the Web page. Since the main goal is to maximize the client’s satisfaction, the model tries to offer as much information as possible while keeping the perceived performance under acceptable limits.

Therefore, important is only the maximum quantity of information (computed as in Equation 4-16) that can be delivered to the client and not the minimum. Any values for the three Web content constraints from Equation 4-16 smaller or equal than the computed ones, will maintain a good client perceived performance, which is the goal of the PPM model.

$$\text{No} \underline{\text{Objs}} \text{Max} = 0.15 * 3 + 0.37 * 6 + 0.32 * 13 + 0.14 * 18 + 0.02 * 30 = 9.96 \approx 10$$

$$\text{Dim} \underline{\text{Objs}} \text{Max(KB)} = 0.15 * 10 + 0.37 * 20 + 0.32 * 50 + 0.14 * 80 + 0.02 * 200 = 4002$$

$$\text{Dim} \underline{\text{Page}} \text{Max(KB)} = 0.15 * 5 + 0.37 * 10 + 0.32 * 15 + 0.14 * 20 + 0.02 * 30 = 1267$$

Equation 4-16

Next, **the update phase** of the PPM is exemplified step-by-step using the same stereotype based structure defined in the previous example. To simplify the example it is assumed that only one negative event has occurred: the “abort” action performed by the client, while the page is loaded on the user’s computer. An abort action is performed either when the “stop” or “reload” buttons are pressed.

In the update algorithm the first step consists of computing the mask Mask_{F_i} that reflects the change in user satisfaction for each feature $F_i \ i = \overline{1,4}$ from the stereotype structure. The mask is applied on the probability values PF_{ik} , that are associated with the linguistic terms LF_{ik} of the feature F_i . For exemplification the mask (Mask_{DT}) computed for the “Download Time” feature from the first stereotype (stereotype “Bad”) is presented next. The mask Mask_{DT} has a length equal with 5 (the total number of linguistic terms associated with this features) and its components are initialised with zero (Equation 4-17).

$$\text{Mask}_{DT} = [0.0, 0.0, 0.0, 0.0, 0.0]$$

Equation 4-17

The B-Mask associated to the “abort” event consists of a one-dimension matrix equal with 3 and all the elements equal with 0.1 (Equation 4-18). Since only one event was considered, there is only one basic-mask that will be added to the mask Mask_{DT} .

$$B\text{Mask}_{abort} = [0.1, 0.1, 0.1]$$

Equation 4-18

The highest probabilistic value of the linguistics terms associated to DT feature, from the “Bad” stereotype is 0.86 and the value corresponds to the linguistic term from position one (see Table 4-4).

Therefore the BMask will be added starting from position one resulting in a Mask_{DT} that is presented in Equation 4-19.

$$\text{Mask}_{DT} = [0.1, 0.1, 0.1, 0.0, 0.0]$$

Equation 4-19

Table 4-15 Group of Features for Stereotype "Bad"

FEATURE	LIST (LINGUISTIC TERM – PROBABILITY)
DT (s)	(>15, 0.74), (15-12, 0.18), (12-8, 0.08), (8-4, 0), (<4, 0)
T (Kbps)	(<28, 0.74), (28 -56, 0.18), (56-64, 0.08), (64-128, 0), (>128, 0)
RTT (msec)	(>500, 0.74), (500-300, 0.18), (300-200, 0.08), (200-100, 0), (<100, 0)
PPf	(Bad, 0.74), (Poor, 0.18), (Normal, 0.08), (Good, 0), (Excellent, 0)

The new probabilistic values associated with each linguistic term of the “Download Time” feature, after the mask was added, are normalized. The same procedure is applied for all other features of the stereotype Bad. The results are presented in Table 4-15.

The next step consists of applying the same update algorithm illustrated in the previous step to the other stereotypes from the PPM. The newly obtained probability values for the following four stereotypes are presented from Table 4-16 to Table 4-19.

Table 4-16 Group of Features for Stereotype "Poor"

FEATURE	LIST (LINGUISTIC TERM – PROBABILITY)
DT (s)	(>15, 0.19), (15-12, 0.47), (12-8, 0.24), (8-4, 0.10), (<4, 0)
T (Kbps)	(<28, 0.19), (28 -56, 0.47), (56-64, 0.24), (64-128, 0.10), (>128, 0)
RTT (msec)	(>500, 0.19), (500-300, 0.47), (300-200, 0.24), (200-100, 0.10), (<100, 0)
PPf	(Bad, 0.19), (Poor, 0.47), (Normal, 0.24), (Good, 0.10), (Excellent, 0)

Table 4-17 Group of Features for Stereotype "Normal"

FEATURE	LIST (LINGUISTIC TERM – PROBABILITY)
DT (s)	(>15, 0.04), (15-12, 0.21), (12-8, 0.40), (8-4, 0.25), (<4, 0.10)
T (Kbps)	(<28, 0.04), (28 -56, 0.21), (56-64, 0.40), (64-128, 0.25), (>128, 0.10)
RTT (msec)	(>500,0.04),(500-300, 0.21),(300-200, 0.40),(200-100, 0.25),(<100, 0.10)
PPf	(Bad, 0.04), (Poor, 0.21), (Normal, 0.40), (Good, 0.25), (Excellent, 0.10)

Table 4-18 Group of Features for Stereotype "Good"

FEATURE	LIST (LINGUISTIC TERM – PROBABILITY)
DT (s)	(>15, 0.0), (15-12, 0.02), (12-8, 0.17), (8-4, 0.52), (<4, 0.29)
T (Kbps)	(<28, 0.0), (28 -56, 0.02), (56-64, 0.17), (64-128, 0.52), (>128, 0.29)
RTT (msec)	(>500,0.0),(500-300, 0.02),(300-200, 0.17),(200-100, 0.52),(<100, 0.29)
PPf	(Bad, 0.0), (Poor, 0.02), (Normal, 0.17), (Good, 0.52), (Excellent, 0.29)

Table 4-19 Group of Features for Stereotype "Excellent"

FEATURE	LIST (LINGUISTIC TERM – PROBABILITY)
DT (s)	(>15, 0.0), (15-12, 0.0), (12-8, 0.0), (8-4, 0.12), (<4, 0.88)
T (Kbps)	(<28, 0.0), (28 -56, 0.0), (56-64, 0.0), (64-128, 0.12), (>128, 0.88)
RTT (msec)	(>500,0.0),(500-300, 0.0),(300-200, 0.0),(200-100, 0.12),(<100, 0.88)
PPf	(Bad, 0.0), (Poor, 0.0), (Normal, 0.0), (Good, 0.12), (Excellent, 0.88)

The following steps include a new classification of the user in stereotype classes and the computation of the final suggestions.

Considering a user with the same performance characteristics illustrated in Equation 4-15, the probabilistic values for the user to belong to PPM stereotype classes are: Stereotype "Bad" – 1.04 %, Stereotype "Poor" – 63.62 %, Stereotype "Normal" – 35.28 %, Stereotype "Good" – 0.06 %. The final list of suggestions is determined by merging the suggestions

associated with the stereotype classes the user belongs to. The results are presented in Table 4-20 and Equation 4-20.

Table 4-20 Final Group of Suggestions

SUGGESTION	LIST (LINGUISTIC TERM – PROBABILITY)
No. Obj	(0-3, 0.19), (4-6, 0.42), (7-13, 0.27), (14-18, 0.11), (30-18, 0.01)
Dim. Obj. (KB)	(0-10, 0.19), (10 - 20, 0.42), (20-50, 0.27), (50 -80, 0.11), (200-80, 0.01)
Dim. Page (KB)	(0-5, 0.19), (5 - 10, 0.42), (10-15, 0.27), (15 -20, 0.11), (30-20, 0.01)

$$NoObjs_Max = 8.86$$

$$DimObjs_Max (KB) = 34.59$$

$$DimPage_Max (KB) = 11.68$$

Equation 4-20

A comparison of the PPM classification and consequent suggested content constraints for two users with the same characteristics as in Equation 4-15 is presented next. Note that the second user has expressed dissatisfaction with the perceived performance through an abort action.

User 1	User 2
Classification: Bad – 0.01% Good – 0.15% Poor – 45.61% Excellent – 0% Normal – 54.23%	Classification: Bad – 1.04% Good – 0.06% Poor – 63.62% Excellent – 0% Normal – 35.28%
Suggestions No.Objs_Max=9.96 Dim.Objs_Max =40.02.1 KB Dim.Page_Max =12.67 KB	Suggestions No.Objs_Max=8.86 Dim.Objs_Max =34.59 KB Dim.Page_Max =11.68 KB

Figure 4-8 PPM Behaviour for Two Users

Figure 4-8 clearly shows that the unsatisfied user (User 2) was classified with higher percentages in “Bad” and “Poor” stereotype classes and with lower percentages in “Normal” and “Good” classes than the other user (User 1). Therefore, the content related constraints suggested are tougher for User 2 than those for User 1. In consequence the overall suggested quantity of information to be delivered is lower for User 2 and thus the expected performance parameters should improve determining increased end-user satisfaction.

4.3.3 Performance Monitor

Performance Monitor module is in charge with monitoring and measuring in real-time different performance metrics that may either directly affect or give an indication about the user QoE. Among them are such as response time (also referred as latency, download time and speed), round-trip time, throughput, user tolerance to delay, temporal variation of users' satisfaction and the user's behaviour, which could indicate dissatisfaction with the service (e.g. abort actions). The information gathered by the PM during the user access sessions is delivered to the PPM. The mechanism used to collect this information is based on filtering TCP packets that carry information and monitoring the signals exchanged by the HTTP protocol. Some of the most important signals tracked by PPM are: TCP SYN, TCP SYN ACK, HTTP GET REQUEST, and TCP FIN. Figure 4-9 presents the signals that are exchanged in a HTTP client-server communication process.

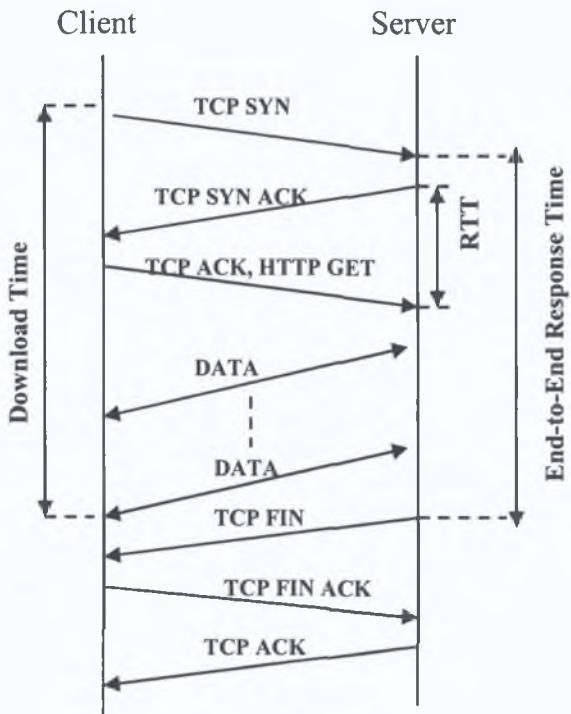


Figure 4-9 HTTP Client-Server Communication Process

The *download time perceived by the user* can be estimated on the server side using Equation 4-21, where “*EndToEndResponseTime*” is the time from the moment when the server receives the TCP SYN packet until the server sends the TCP FIN packet. This time is very close to the “*DownloadTime*” as the difference depends on the relative duration of the TCP SYN delivery from the client to the server and TCP FIN signal delivery from server to client. As a server-based solution for measuring the RTT (*round trip time*), the time between

the transmission of the TCP SYN ACK packet from the server to the client and the time when the server receives the TCP ACK back from the client is used.

$$\text{AccessTime} \approx \text{RTT} + \text{EndToEndResponseTime}$$

Equation 4-21

Since the user-perceived QoS is a highly significant issue for this research, metrics that reflect user QoE have been also taken into consideration by the PM module. The *frequency of aborted pages* is a metric defined in [157] and reflects the client satisfaction in relation to the perceived QoS. The main idea behind this metric is that if the user gets impatient due to a slow perceived access time he/she will interrupt the transfer of the web page. A user can perform an interruption by clicking either ‘stop’ or “reload” buttons while a web page is loaded slowly or by clicking a link from the page before the web page was fully loaded. However, only a subset of aborted actions reflect poor perceived QoS while many others are caused by client-specific browsing patterns. In [157] only the aborted pages with a response time higher than a given threshold (7 sec) were considered as reflecting a bad quality download. The same principle was used for PM in measuring the frequency of aborted pages.

A metric that reflects user behaviour relative to the perceived QoS was proposed in [137] and it expresses the *utility of a session* of requests. It was noticed that users' tolerance for latency (access delay) decreases over the duration of interaction with the site. Tests over high speed connections showed that a 10 sec latency was considered acceptable to 95 % of the participants during the first four Web page accesses, still acceptable for 80 % of the participants during the access of an extra 6 pages, but only for 60 % of accesses over the 11th page were still acceptable [137]. Similar tests were performed for latency values between 16 sec and 6 sec. The conclusion of these results was that the latency should improve over the duration of a session in order to maintain acceptable the client's experience with navigation on a web site. Following these tests a function denoted “utility of a session” which takes into account the duration of the session was defined [137] (Equation 4-22).

$$\text{Utility} = \frac{\sum_{i=1}^N \frac{\text{Threshold}(i) - \text{Latency}(i)}{\text{Threshold}(i)}}{N}$$

Equation 4-22

where *Threshold* (i) – threshold of acceptability of an access i , *Latency* (i) – delay in download completion of page i , N - length of navigation session.

The values of the *Utility* vary between -1 and 1 and are interpreted as follows:

- $Utility > 0$ indicates that the acceptable performance thresholds have been positively exceeded.
- $Utility = 0$ indicates that the performance of the service is optimal.
- $Utility < 0$ indicates unacceptable performance.

In order to maintain a positive *Utility* over a session and to keep acceptable the client's experience during a session, the *Threshold*(i) should decrease and/or *Latency*(i) must improve as i increases. Values for the *Threshold*(i) parameter have been defined in the interval 16-6 sec, with an average of 10.38 sec for high speed connections in [135]. A similar mechanism was used here for defining values in the interval 25-10 sec with an average of 15 sec as mentioned in [158] for low speed connections.

Currently there is no universal standard limit for what constitutes “acceptable” download times. However it is widely accepted that long download times cause user frustration leading to performance loss and distraction. Nielsen [159] suggested a 10 sec limit, but more recent research [135, 137, 146] showed that under certain conditions users would tolerate download times significantly longer than 10 sec. For example, Web users that frequently use high-speed connection are less tolerant with delays such as those associated with dial-up connections [146]. Users also tolerate different levels of latency depending on the nature of the task they are involved in [135]. Bouch et al. [137] found that tolerance levels vary according to both the length of time spent interacting with a particular site and the overall amount of time spent on the Web.

Based on a survey on the current research into user tolerance three zones of duration that represent how users feel were proposed in [148]: **zone of satisfaction**, **zone of tolerance** and **zone of frustration**. According to a number of studies a user is “satisfied” if the page is loaded in less than 10-12 sec. The next zone begins when the page-load time exceeds the time limit from the zone of satisfaction. The user starts to become aware of the passage of the time, slowing building up into annoyance. It is believed that it is a wide band of time between when a user is no longer satisfied and when the user becomes frustrated. Zone of frustration starts when the user reached the point when he/she is significantly frustrated.

According to a number of studies [135, 137, 149, 160] different values between 30 and 41 sec were considered as the critical point.

In conclusion the research on the effects of download time on users' subjective evaluation on the Web site performance suggests that users have some threshold (*user tolerance*) for what they consider adequate or reasonable latency. On average a download time higher than 10-12 sec causes disruption and users loose their attention, while values higher than 30 sec cause frustration. At the same time it is significant to mention that when the user is aware of the existence of a slow connection, he/she is willing to tolerate a delay that averages 15 sec but no more than 25 sec [158].

4.3.4 Extended Adaptation Model

The **Adaptation Model** is enhanced for a better adaptation of the Web content integrating knowledge of user interests with performance constraints. Thus, the enhanced AM ensures the co-operation between the personalisation (the important feature of any AHS) and performance based adaptation by taking into account information provided by the User Model and the Perceived Performance Model.

Initially, based on UM information AM proposed a page tailored to the user interests. The page consists of components (information fragments from different concepts) chosen from the Domain Model. According to the PPM suggestions the content of the proposed page is modified in order to improve the user QoE while minimising the negative impact on user's satisfaction related to the quality of the content. The adaptations performed on the proposed page consist of either the elimination (e.g. some components will not be displayed) or the modification in the properties of some information fragments from the page (e.g. changes in the resolution or size of the images). The decision, on which components are affected, is based on "**adaptation algorithm**".

The proposed adaptation algorithm uses content related constraints provided by PPM and information provided by the UM related to the strength of user interests in different concepts (defined in the DM). Hence, the enhanced AM performs changes starting with information fragments of those concepts the user is the least interested in. These changes are applied until the content constraints suggested by the PPM are matched. The PPM suggestions are used by the AM only for the content-level adaptation and not for the link-level adaptation. It is considered that the characteristics of page content (e.g. size, number

and type of the embedded components) are those that mainly affect the end-to-end performance and not the number of links.

4.3.4.1 Adaptation Algorithm

The objective of the algorithm is to determine and apply the correct transformations on a web page in order to match the PPM suggestions.

In order to match these PPM suggestions related to the embedded components, two types of transformations can be applied: modifications in the properties of the embedded components and/or eliminations of some of the components. Since images contribute with the largest quantity of information to the total size of a web page, in this work they were the only ones taken into consideration by this algorithm. In the case when some images have to be eliminated, each removed image is replaced with a link to the image. In this way, if a user does really want to see the image, the link will offer this possibility.

A block-diagram of the algorithm is presented in Figure 4-10. The algorithm uses as parameters the PPM suggestions in terms of maximum number of embedded objects ($NO_{suggest}$), maximum total size of objects ($SO_{suggest}$) and maximum size of a Web page ($SP_{suggest}$). In the algorithm, the check for the number of embedded objects is performed before the verification of the total size of the embedded components. Therefore image elimination may occur before image compression. This was preferred due to the fact that setting up multiple connections to carry a higher number of objects is time consuming and reduces the overall performance of the delivery.

Adjusting the Total Number of Embedded Objects

In the situation when the number of objects embedded in a web page ($NO_{current}$) exceeds the maximum number of objects suggested by the PPM ($NO_{suggest}$) there is a need for the elimination of some images. The image elimination mechanism makes use of information stored by the UM related to user's level of interest in the concepts defined in DM. These concepts are abstract representations of the information described through the images.

This image elimination mechanism involves the following principles:

- the image with the lowest interest for the user is removed
- the elimination process continues while $NO_{current} > NO_{suggest}$

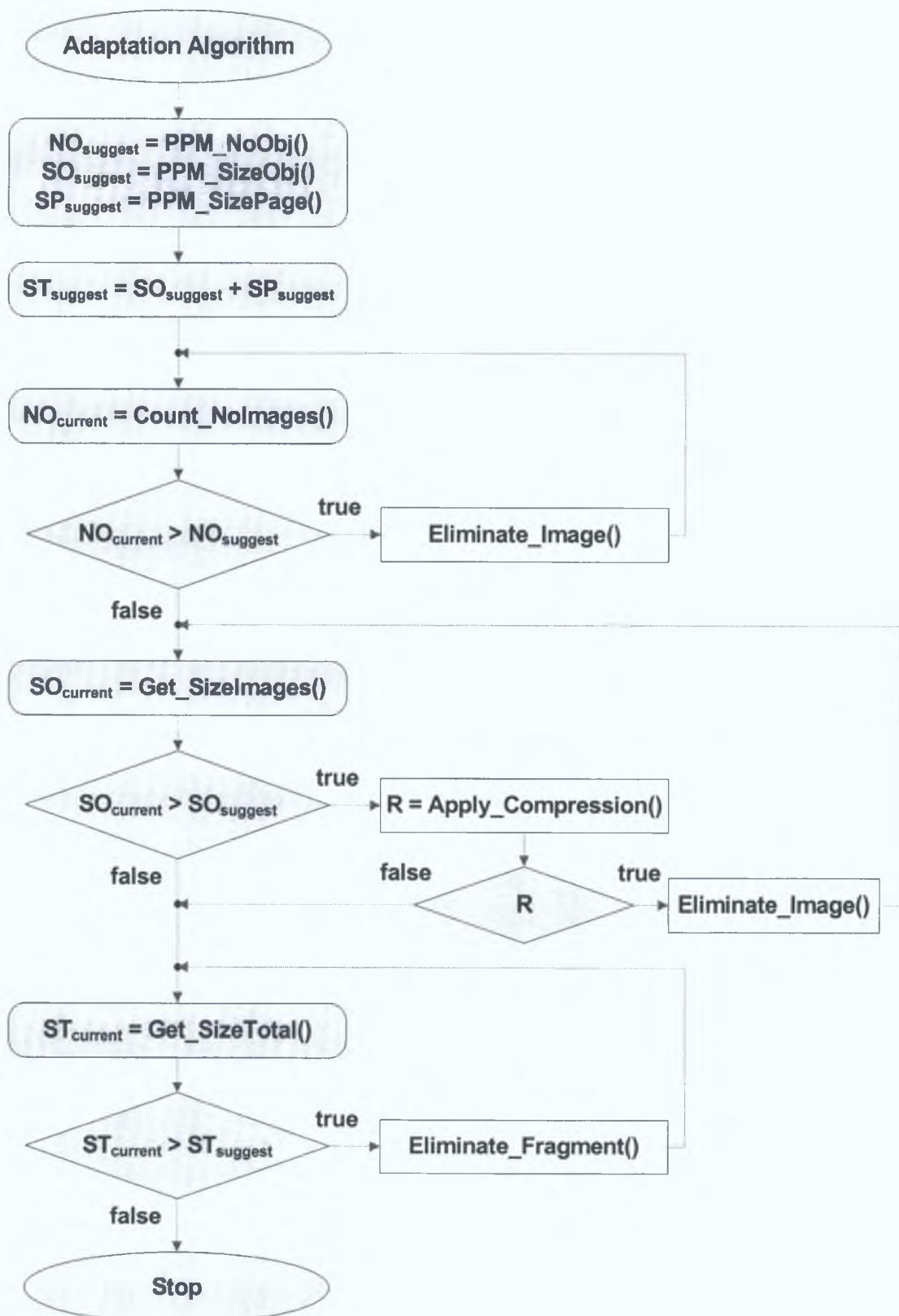


Figure 4-10 Adaptation Algorithm

A pseudo- code description of the mechanism is presented next.

```
Adaptation_Algorithm_For_TotalNo_Embedded _Objects (NOsuggest)
begin
```

```
    NOcurrent= Count_NoImages();
    while (NOcurrent > NOsuggest) do
        begin
            Eliminate_Lowest_Interest_Image ( );
            NOcurrent = NOcurrent -1;
        end;
    end;
```

Adjusting the Total Size of Embedded Objects

Matching the suggestion related to the total size of the embedded images is the next phase in the adaptation algorithm. If the total size of embedded objects (SO_{current}) is higher than the PPM suggestion (SO_{suggest}), image compression techniques and/or image elimination methods have to be applied. First, the image compression is applied and if further reduction is necessary, image elimination is applied.

Step 1: Image Compression

The first step involves trying to apply image compression. Different compression rates (expressed as percentage) are applied on each image depending on: the total reduction suggested on the total size of embedded images, on the image size and on user interest on the image. For example, if a reduction of 40 % is required to be performed on two images A and B, user being more interested in image A than image B, a smaller reduction will be applied on image A than the one performed on image B. The actual compression rates will be computed according to their relative interest. The algorithm for determination of the compression factors for each image ($R\%_i$) takes into consideration user interest on them and uses the following formulas (Equation 4-23):

$$\begin{aligned}
 S &= \sum_{i=1}^N S_i \\
 RS &= R\% * S \\
 W_i &= \frac{S_i}{S} * \overline{K}_i \quad ; \quad W_i^{normalized} = \frac{W_i}{\sum_{i=1}^N W_i} \\
 RS_i &= W_i^{normalized} * RS \\
 R\%_i &= \frac{RS_i}{S_i} * 100
 \end{aligned}$$

Equation 4-23

Notation: S_i = the size (KB) of the image “i” S = total size (KB) of the embedded images Rs = total reduction in size (KB) to be applied on the embedded images Rs_i = reduction in size (KB) to be applied on the image “i” $R\%$ = total reduction in percentage to be applied on the embedded images $R\%_i$ = reduction in percentage to be applied on the image “i” N = total number of embedded images from the Web page K_i = user’s interest in concept (image) “i” \overline{K}_i = user’s non-interest in concept (image) “i” ($100 - K_i$) normalized values

If one of the computed compression rates cannot be applied on an image (e.g. due to the fact that the quality will be lower than acceptable for the end-users. Image compression tools may have limits for the compression rate that would ensure good quality for the image), image elimination strategy (Step 2) will be applied. This assumes that the image compression algorithm has a maximum compression threshold rate in order to ensure good user perceived quality for the compressed images.

For exemplification the following case is considered. A web page consists of three embedded images and, based on the PPM suggestion related to total size of embedded images, a reduction of 20 % has to be applied. User non-interest in these images and the size of each image is presented in Table 4-21. The compression rate for each image is computed as in Equation 4-24, following the formulas from Equation 4-23. Therefore the following

compression rates should be applied on each image in order to match the PPM suggestions: 24.18 % for reduction to image A, 7.5 % to image B and 3.8 % to the third image.

Table 4-21 Size of Images and User Non-Interest in the Images

IMAGE NO.	IMAGE SIZE (KB)	USER NON-INTEREST
A	100	0.7
B	20	0.2
C	10	0.1

$$Rs = 0.2 * 130 = 26 \text{ KB}$$

$$W_1 = \frac{100}{130} * 0.7 = 0.54$$

$$W_1^{\text{Normalized}} = \frac{0.54}{0.58} = 0.93$$

$$W_2 = \frac{20}{130} * 0.2 = 0.03$$

$$W_2^{\text{Normalized}} = \frac{0.03}{0.58} = 0.0517$$

$$W_3 = \frac{10}{130} * 0.1 = 0.008$$

$$W_3^{\text{Normalized}} = \frac{0.008}{0.58} = 0.0138$$

$$Rs_1 = 0.93 * 26 = 24.18 \text{ (KB)}$$

$$R_{\%1} = 24.18\%$$

$$Rs_2 = 0.0517 * 26 = 1.5 \text{ (KB)}$$

$$R_{\%2} = 7.5\%$$

$$Rs_3 = 0.0138 * 26 = 0.38 \text{ (KB)}$$

$$R_{\%3} = 3.8\%$$

Equation 4-24

Step 2: Image Elimination

Step two is performed in the situation when image compression does not succeed to match the PPM requirements. The objective is to eliminate one or more images and to replace them with links to images. If a user does really want to see the image, the link will offer this possibility.

This strategy is applied on the original images and is based on the following principles:

- the image with the lowest interest for the user is removed

- if the recomputed total size of embedded objects from the web page ($SO_{current}$), after the elimination was performed, is still higher than the PPM suggestion ($SO_{suggest}$), perform again image compression. In this case lower compression rates will be required.

A pseudo-code based description of this strategy is presented next:

```

Adaptation_Algorithm_Step2 (SOsuggest)
begin
    SuccessCommpression = false;
    NOcurrent = Count_Nolimages();
    while (SuccessCommpression == false) do
        begin
            Compute_Img_Compression_Factors (NOcurrent, SOsuggest)
            for imgNo = 1 to NOcurrent do
                begin
                    if (Apply_Commpression(R%imgNo, imgNo) == false)
                        begin
                            SuccessCommpression = false;
                            break;
                        end
                    else
                        SuccessCommpression = true;
                end;
            if (SuccessCommpression == false)
                begin
                    Eliminate_Lowest_Interest_Image ( );
                    NOcurrent = NOcurrent -1;
                end;
        end;
    end;

Compute_Img_Compression_Factors (NOcurrent, SOsuggest)
begin
    SOcurrent = Get_SizeImages();
    SumW = 0;
    RS = SOcurrent - SOsuggest;
    for i = 1 to NOcurrent do
        begin

```

```


$$W_i = \frac{S_i}{SO_{current}} * \overline{K}_i$$

SumW = SumW + W_i
end;
for i = 1 to NOcurrent do
begin
    
$$W_i^{\text{Normalized}} = \frac{W_i}{\text{SumW}}$$

    
$$Rs_i = W_i^{\text{Normalized}} * Rs$$

    
$$R\%_i = Rs_i / S_i * 100$$

end;
end;

```

Adjusting the Size of the Web Page

The PPM suggests a maximum dimension for the based Web page ($SP_{suggest}$). The adaptation algorithm would have to enforce the size of the final web page to this limit. However often the image compression phase and the image elimination mechanism may result into objects whose total size is below the maximum size suggested by the PPM for all embedded objects ($SO_{suggest}$). In this situation the total size of the web page, including the embedded images, may be still below the sum of PPM suggested limits ($ST_{suggest} = SO_{suggest} + SP_{suggest}$) and there is no need for further adaptation related reduction.

For the case when the total size of the page including the embedded objects ($ST_{current}$) is greater than $ST_{suggest}$ those fragments from the based Web page the user is least interested in (as indicated by the User Model) are eliminated. This process continues while $ST_{current} > ST_{suggest}$.

4.4 Chapter Summary

This chapter introduced a new QoE layer to the generic AHS architecture that takes into account an extra dimension of user characterisation (end-user Quality of Experience) in the personalisation process provided by adaptive hypermedia applications. QoE is directly influenced by the operational environment through which the user interacts with AHS and by the subjective assessment of user perceived performance.

A Perceived Performance Model (PPM) that makes an interpretation of user satisfaction via different performance metrics (e.g. download time, throughput, RTT), user's behaviour metrics related to the perceived performance (user's tolerance to delay over a session, user's acceptance threshold) and user subjective assessment was proposed. These

metrics are measured in real-time by a Performance Monitor module and the computation mechanism was described in this chapter. In order to illustrate the usage of the model, the case where the users interact with the system through a low bit operational environment (up to 128 kbps) was considered. The same case study and PPM settings will be used for the simulation and qualitative tests presented in the next two chapters of the thesis.

In order to apply the results generated by the PPM and to optimise end-user QoE an adaptation algorithm was proposed and described in this chapter. The algorithm modifies or/and eliminates web page components from the structure of a web page tailored according to the information from the User Model. The decision on which components are affected is based on the information offered by the UM on the level of user's interest in the components.

The next two chapters present evaluation results and benefits provided by the proposed QoS layer. Simulation and qualitative tests were considered in the evaluation process. For testing purposes the QoS layer was embedded in an open-source adaptive hypermedia application deployed in the educational area.

Chapter V

QoE Layer Validation and Analysis

5.1 Chapter Introduction

The previous chapter introduced the proposed QoE layer, a performance-oriented extension to Adaptive Hypermedia Systems (AHS). The goal of the QoE layer is to add performance-based adaptations to the classic end-user knowledge and goal-based adaptations performed by AHS, in order to improve the end-user experience with the systems.

The role of this chapter and the following one is to investigate the performance improvements provided by the QoE layer. This chapter presents a large set of simulation tests that analyse and validate the performance improvements provided by the QoE layer stand-alone. Different constant and variable network conditions for an online learning session are simulated and data is collected and analysed. The next chapter presents the evaluation of the QoE layer when deployed on a real Adaptive Hypermedia System (AHS). This evaluation is performed by qualitative tests in the educational area and aims to analyse the learning process, students' learning performance and outcome, and system usability when performance based content adaptations are applied.

This chapter starts by highlighting the strategy used for the QoE layer analysis and validation (section 5.2). Next the setup conditions and the scenarios that were considered for the tests are presented (section 5.5.1). The simulations were performed using the Network Simulator version 2 (NS-2) and the NSWEB extension that provides support for WWW traffic generation and analysis. The QoE layer model was implemented in OTCL and C++ in order to allow for the deployment on the NS-2. Sections 5.5.2 and 5.5.3 present and discuss various results of the simulation tests. A summary of the conclusions drawn from each tested scenario is also provided in separate sub-sections.

5.2 QoE Layer Validation and Analysis Strategy

The QoE layer that was introduced in the previous chapter aims to improve the end-user QoE during the interaction with an adaptive hypermedia system, mainly in terms of system performance as perceived by the end-user.

It is well known that the development of any product or system passes through different stages. They include requirements analysis, design, implementation, integration as well as testing, deployment and maintenance. Among these phases testing is a very important one and makes sure that the system was built right. Testing is first performed on each individual component of the system before component integration and testing of the overall system.

This chapter tests the QoE layer stand alone, whereas the next chapter presents testing results that involve the full QoE-aware AHS. The goal of the QoE layer validation is to determine the correctness of its behaviour and of the provided outputs. It consists of a set of simulations, using Network Simulator version 2 (NS-2) [161], that assess the performance improvements brought by the module. The simulation scenarios involve the transmission of different web pages that are part of a simulated Web browsing session over various network conditions. Performance analysis is based on comparative measurements of two cases. The first case involves the usage of the QoE layer, while the second one does not.

QoE layer consists of the Performance Monitor (PM), Perceived Performance Model (PPM) and an extension to the Adaptation Model (AM). The core of the proposed QoE layer is the PPM that takes as input different performance metrics that reflect the current delivery conditions measured by the PM. PPM generates content constraints (suggestions) that are applied by the AM to the personalised Web content served to the user. These content constraints aim at providing the best experience to the user in the current delivery conditions. The PPM also provides a dynamical representation of both user perceived performance, during the delivery of Web content and of the user quality of experience with the system.

In consequence, to fully test the QoE layer, PPM constraints and the internal representation of the PPM for a given user are also analysed when different operational environments are simulated.

5.3 Objective of the Simulations

For qualitative testing purposes the QoE layer is deployed on an Adaptive Hypermedia System that currently runs as a courseware application and delivers adaptive educational material to the students. The educational material consists of web pages that are going to be used during the evaluation phase of the QoE-aware-AHS. Characteristics of this material are used during the simulation tests in order to build up a virtual course and to simulate a learning task performed by a student over different network environments. A learning task involves sequential accesses to a number of web pages. Time taken to complete the learning task is a sum of the access time (download time) of the visited Web pages and the time used by the user to read and accumulate knowledge. Therefore, the completion time of the task could be improved for instance by reducing the access time of the Web pages involved in the learning process. On the other hand a faster access time might have effects on the pedagogical aspects of the learning process that might lead towards better learning performance. For example long waiting time for accessing information may annoy and frustrate people making them to lose their focus and/or concentration on the learning task.

The objectives of the simulation tests are the following:

- to investigate the impact on performance when the content related constraints generated by the QoE module are applied during a simulated learning session.
- to analyse the behaviour of the PPM when a client sends web page requests over different operational environments with fixed characteristics
- to analyse the behaviour of the PPM when a client sends web page requests over different operational environments with dynamically changing characteristics.

All simulations are performed for low bit rate operational environments that involve connection speeds up to 128 kbps. Comparisons with the case when the content transmission does not involve any performance-based adaptation are performed for each studied scenario.

Performance analysis involves the assessment of the access time per page and per study session, of the quantity of transmitted date and percentage of the reduction. PPM behaviour analysis looks out to the capability of the model to track any changes in the delivery conditions that may affect the users QoE and to provide the content constraints that do not decrease the level of the user's perceived performance.

Analysis of the effects of the QoE layer on the learning process and student outcome is performed during the evaluation phase presented in Chapter 6. The evaluation tests involve real students that make use of a QoE-aware-AHS in order to perform a study task.

5.4 Network Simulator Version 2 (NS-2)

Network Simulator version 2 (NS-2) [161] is an open source, object-oriented, discrete event, network simulation environment that was built at University of California at Berkeley² in order to test models proposed in the networking research area. It was developed and written in C++ and OTcl (an object-oriented extension to Tcl/Tk proposed at Massachusetts Institute of Technology) [162], but in order to be deployed, it also requires Tcl, Tk, OTcl and TclCl to be installed. NS-2 is primarily used for simulating local and wide area IP-based networks. It provides substantial support for simulation of different network protocols such as TCP and UDP, traffic source behavior such as FTP, Telnet, WWW, etc., routing techniques over wired and wireless networks. The NS project is now a part of the Virtual InterNetwork Testbed (VINT) project³ and is supported by DARPA⁴. More information about NS-2 can be found in the NS Manual [163] or in one of the NS tutorials [164, 165].

Since NS-2 is open-source, many research groups have contributed with different extensions to the original simulator that would allow for simulation of new protocols, techniques, routing algorithms or new topologies. New workload generators and analysis tools have also been provided for different study cases. Among these extensions, the WWW Workload Generator (NSWEB) was proposed and implemented by Wallerich [166] and used for these simulation tests.

NSWEB is a WWW workload model that provides support to mimic the features of the latest standard Web protocols such as persistent and pipelined HTTP connection as specified in HTTP/1.1 (RFC 2616). It allows for a better management of the static allocation of Web pages and embedded objects and of the dynamic page popularities. A page selector mechanism that is needed to generate realistic Web request sequences was also implemented.

² University of California at Berkeley, <http://www.berkeley.edu>

³ Virtual InterNetwork Testbed (VINT) project, <http://www.isi.edu/nsnam/vint/index.html>

⁴ Defense Advanced Research Projects Agency (DARPA), <http://www.darpa.mil>

NSWEB offers a higher flexibility for the analysis of the simulation results. It generates different event logs and packet traces that can be analysed offline after the simulation. Packet traces indicate what exactly happens during a simulation run and how the packets travel through the network. Event logs present the simulation at protocol level, a more abstract level than packet tracing. Two types of events: connection related events and request transaction events are logged by NSWEB.

5.5 Simulation Tests

5.5.1 Set-up conditions

The simulations were performed using Network Simulator version 2 (NS-2) [161] and a NSWEB [166] extension. Several different connections and network properties were considered that correspond to low bit rate operational environment as perceived by people through a home Internet connection. The simulation set-up topology consists of a simple Web Server – Web Client system. Figure 5-1 exemplifies a residential client 56 kbps modem connection and round-trip-time (RTT) of 310 msec. In the absence of other background traffic, the bottleneck link is the client network connection.

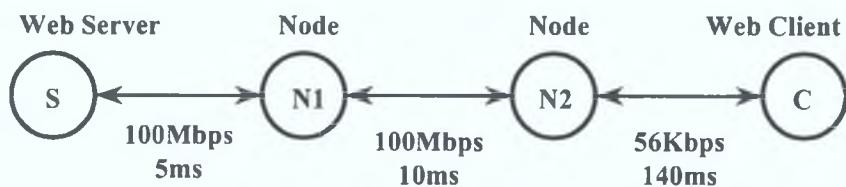


Figure 5-1 NS-2 Simulation Topology

QoE layer was deployed on the Web Server. QoE layer's PPM includes the same five stereotype classes ("Bad"-T1, "Poor"-T2, "Normal"-T3, "Good"-T4 and "Excellent"-T5) that have been presented in detail in section 4.3.2. These stereotypes cover users with performance characteristics corresponding to connection types in the range of 28 kbps-128 kbps. The same list of suggestions related to the characteristics of the delivered web content, used in the illustrative example from section 4.3.2 were considered.

Different tests were performed in NS-2 simulating Web browsing sessions performed by a Web Client in various operational environments. A simulated browsing session involves sequential requests sent by the Web Client for some web pages from the

virtual web site and the transmission of these web pages over the simulated network topology.

The tests were divided in two categories:

- The first scenario simulates a sequence of requests for ten web pages randomly selected from a virtual Web site over different types of network connections.
- The second scenario simulates a sequence of requests for web pages that are part of a learning session from the AHA! tutorial, when different types of network connections exist between the Web Client and the Web Server.

5.5.1.1 Scenario 1: Simulation of a Browsing Session to a Virtual Web Site

The first scenario involves tests that simulate a browsing session that consists of ten Web pages performed by the Web Client. The Web pages are randomly selected, based on the SURGE technique, from a virtual web site located on the Web Server. The Web site is populated with one hundred pages. These pages have different properties such as Web page size, number of the embedded objects per page and size of each embedded objects. The Web content was generated using the NS-2 Web Generator, based on a probability distribution that is indicated.

The Pareto-II Distribution was used in the NS2 tests for the generation of the Web pages. It best simulates the characteristics of the Web server resources, the distribution of a Web object size on a Web site and it is the most used function for Web traffic simulations. Table 5-1 presents Pareto-II Distribution parameters setup used during the simulations. *Shape* parameter was setup according to the Web traffic specifications presented in the NSWEB [166] documentation. *Avg* parameter was set based on the results of research [156] that analysed and characterised Web pages from most popular web sites based on the amount of content of a page, the number of bytes in the basic web page, the number of embedded objects and the total number of bytes for the embedded objects. The results presented in [156] have shown that most of the Web pages would have the size of the basic page up to 12 KB, an average of 7 up to 20 embedded objects and the total size of the embedded objects around 55 KB or higher.

Table 5-1 Parameters of Distribution Function of the Web Page Characteristics

WEB CONTENT CHARACTERISTICS	PROBABILITY DISTRIBUTION PARAMETERS
Basic Page Size	Pareto-II Distribution with $avg = 3000B$, $shape = 1.2$
Number of Embedded Objects per Page	Pareto-II Distribution with $avg = 4$, $shape = 1.5$
Embedded Objects Size	Pareto-II Distribution with $avg = 4500$, $shape = 1.2$
Pages per Web Server	100

The simulation tests assessed different types of low bit rate network connections that may be experienced by a home Web user. Therefore the bandwidth between Web Client and node N2 was varied between 28 kbps and 128 kbps while RTT between the Web server and the Web client was varied between 530 msec and 150 msec respectively.

5.5.1.2 Scenario 2: Simulation of a Learning Session on the AHA! Tutorial

The second scenario involves a simulated learning session that consists of a sequence of requests for web pages that are part of the Chapter One from AHA! tutorial. The open-source AHA! system provides an adaptive tutorial as an example of the adaptive features of the system. The properties of educational material from the AHA! tutorial such as number of web pages, size of each web page, number and size of the embedded components in a web page, are used to build up a virtual web and to populate the Web Server built using NS-2. This educational material was also used for the qualitative evaluation of the QoE-ware AHS and the results are presented in Chapter 6.

This scenario involves different tests that simulate the learning session over different types of network connections. Therefore, the throughput link connection between Web Client and node N2 was varied between 28 kbps and 128 kbps while RTT between the Web Server and the Web Client was varied between 530 msec and 150 msec respectively.

5.5.2 QoE Layer Assessment in Constant Low Bit Operational Environment

The objective of the simulation tests is to analyse QoE layer behaviour in general and that of the Perceived Performance Model (PPM) in particular when a learning session

takes place. Different network conditions that do not change in real-time during a browsing session are simulated. The analysis includes a comparison with the case when the QoE layer is not applied. The learning session involves a sequence of requests for randomly generated Web pages (Scenario 1) or for a given set of pages that are part of the AHA! tutorial (Scenario 2). Different performance parameters such as *Access Time (download time) per page*, *Aggregate Access Time per session*, *Quantity of Data Transmitted* are measured. Aggregate Access Time per session is measured as the sum of the access times per page. The chain of the *content related constraints* generated by the PPM during the navigational session is also analysed.

5.5.2.1 Simulation Results for Scenario 1

The First scenario involved Web sessions that deliver a sequence of ten web pages randomly selected from a web site populated with one hundred pages. The characteristics of the selected pages are presented in Table 5-2 and they follow the observations made in [156].

Different tests simulated the transmission of these pages over different types of network connections. The simulations involved two cases. In the first case the generated web pages were transmitted directly, while in the second case the content constraints suggested by the PPM for each web page were applied before the Web Server delivered each page. In order to match the PPM suggestions the adaptation algorithm presented in Chapter 4 was applied. This algorithm consists of embedded objects size reduction and/or their elimination. The assessment of the QoE layer benefits for each type of connection is presented in the following sections.

Table 5-2 Characteristics Of The Generated Web Pages Used For The Study Session

WEB PAGE ID	BASIC PAGE SIZE (KB)	NUMBER OF EMBEDDED OBJECTS	TOTAL SIZE OF EMBEDDED OBJECTS (KB)	TOTAL SIZE OF WEB PAGE (KB)
1	9.18	8	82.39	91.57
2	3.10	8	57.96	61.06
3	3.17	6	93.96	97.13
4	10.80	8	190.22	201.02
5	5.61	6	37.73	43.34
6	3.42	9	169.01	172.43
7	12.24	5	64.37	76.61
8	9.38	7	57.68	67.06
9	5.4	10	134.32	139.72
10	3.38	5	36.30	39.67

5.5.2.1.1 Connection Type 1: up to 28 kbps

The first simulation involved a serial transmission of the ten web pages over a connection up to 28 kpbs and with a RTT over 500 msec. A comparison of the measured access time per page for the two cases, when using the QoE content suggestions and without making use of them, is presented in Figure 5-2. This example illustrates a network environment characterised by a 28 kbps connection bandwidth and 530 msec RTT. Similar results were obtained for other values in the specified intervals.

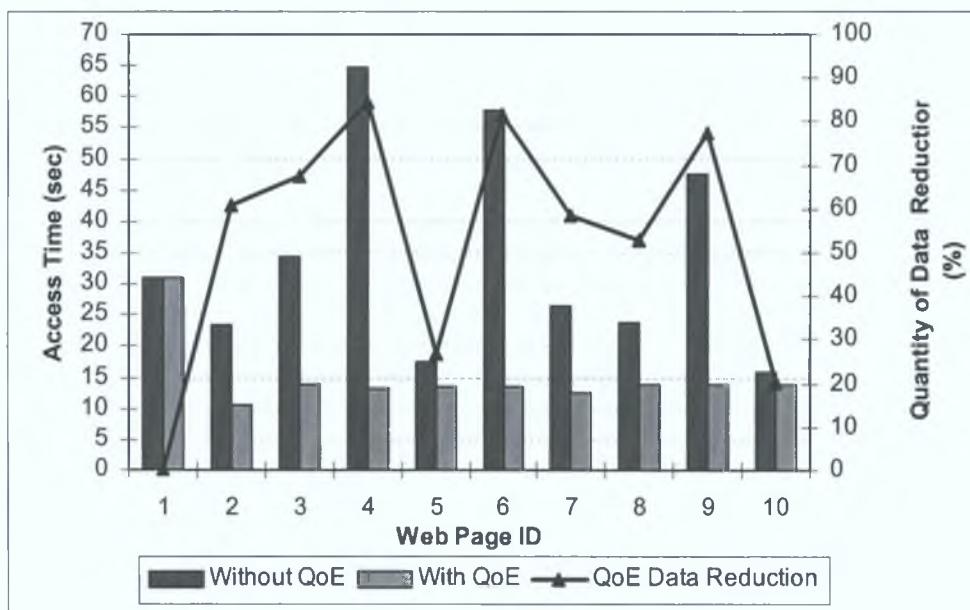


Figure 5-2 Access Time per Page and Web Page Size Reduction for a Browsing Session Over a 28 kbps Connection Speed When Content Constraints Suggested by the QoE Layer were Applied.

One can clearly notice that large Web pages such as Page 4, Page 6 and Page 9 required a high waiting time (over 45 sec) in order to retrieve all the Web page content. As result the users may be annoyed and lose their attention on the performed task. Therefore the access time has to be decreased. By applying the content constraints generated by the PPM, the quantity of data transmitted over the network was reduced. Different percentages of reduction were applied for each page. As consequence, the access time per page has significantly decreased below 15 sec, threshold that is considered acceptable by the users aware of low speed connection. Improvements of the access time have determined a lower aggregate access time per session with QoE layer (149.91 sec) than for the non-QoE case (343.44 sec).

It should also be noticed that apart from the first page, the QoE layer has applied modifications on all the other Web pages and the access time per page was decreased to fit in the acceptable range. Since the QoE layer requires some performance metrics to be measured after a Web page is delivered, the PPM suggestions can be only applied starting with the transmission of the second page. The percentage of size reduction for each page is plotted in Figure 5-2 with the thick black line. It is significant to notice that the highest reduction was performed on pages 4, 6 and 9.

Next the PPM outputs and user classification in stereotypes, after a Web page was requested is illustrated. With the increase in the number of requests for Web pages sent by a

Web Client the PPM model “learns” about the current delivery conditions. It combines the previous suggestions for a given user with the current ones, keeping the most up-to date profile while also taking into consideration historic behaviour.

Table 5-3 presents the Web page characteristics suggested by the model after a Web page was transmitted. These suggestions were applied on the following requested Web pages. Therefore the characteristics of the next Web page to be delivered will not exceed the PPM suggestions.

Table 5-3 Web Content Related Constraints Suggested by the PPM and User Classification into Stereotypes After Each Requested Web Page Over a 28 kbps Connection

ID OF REQUESTED WEB PAGE	PPM OUTPUTS			USER CLASSIFICATION				
				DEGREE OF MATCH WITH A STEREOTYPE (%)				
	SIZE PAGE (KB)	NO. OBJS	SIZE OBJS (KB)	T1 BAD	T2 POOR	T3 NORMAL	T4 GOOD	T5 EXCELLENT
1	7.26	5.0	16.61	66.69	32.76	0.55	0.00	0.00
2	8.94	6.0	22.69	2.45	82.93	14.62	0.00	0.00
3	8.22	6.0	23.42	12.42	83.92	3.66	0.00	0.00
4	7.86	7.0	23.78	12.42	83.92	3.66	0.00	0.00
5	7.64	7.0	24.00	12.42	83.92	3.66	0.00	0.00
6	7.50	7.0	24.15	12.42	83.92	3.66	0.00	0.00
7	7.40	7.0	24.26	12.42	83.92	3.66	0.00	0.00
8	7.33	7.0	24.33	12.42	83.92	3.66	0.00	0.00
9	7.26	7.0	24.39	12.42	83.92	3.66	0.00	0.00
10	7.21	7.0	24.44	12.42	83.92	3.66	0.00	0.00

The results show that the model succeeds to “learn” rapidly about the current network conditions and to suggest correct characteristics for Web pages such as their download time becomes acceptable from user QoE point of view. After the first three pages were requested and delivered the PPM state is stable, generating the same outputs as long as the network conditions do not change. As result the download time of the following seven web pages varies between 12.5 sec and 14.00 sec, depending on their respective sizes but do not exceed the 15 sec threshold.

5.5.2.1.2 Connection Type 2: between 28 kbps and 42 kbps

A second set of tests was performed over an improved delivery environment that involves a bottleneck bandwidth between 28 kbps and 42 kbps and RTT in the range of 300-500 msec.

Figure 5-3 illustrates the access time per page when the same ten Web pages presented in Table 5-2 were transmitted over such as setup network. For the case when the content related suggestions provided by the QoE layer were considered, some of the Web pages were modified as the total size of these Web page decreased, their download time was reduced and an improvement in the aggregate access time per session from 222.88 sec to 124.55 sec was recorded. As the connection speed increased, a smaller number of Web pages required size reduction.

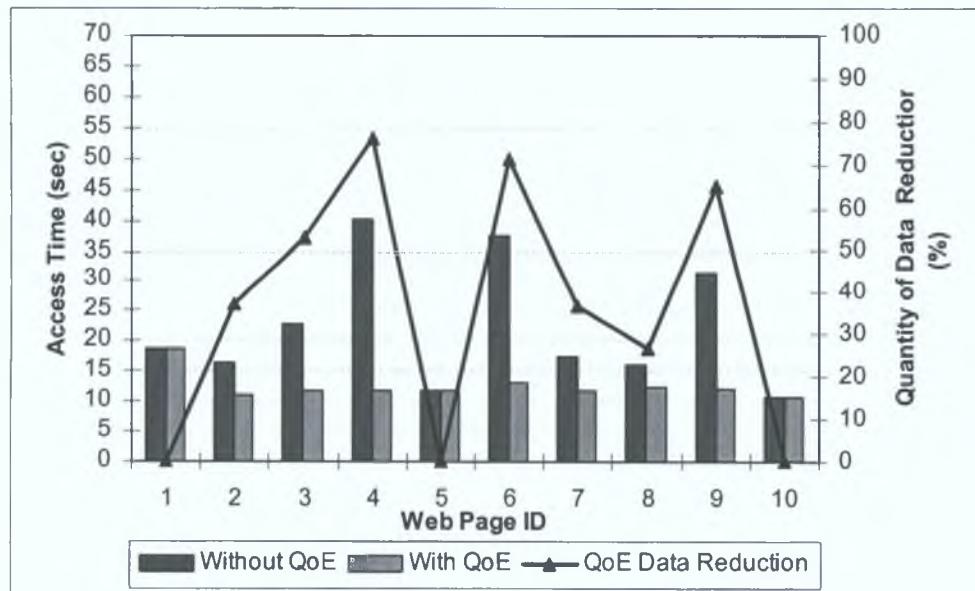


Figure 5-3 Access Time per Page and Total Web Page Size Reduction for a Browsing Session Over a 42 kbps Connection When Content Constraints Suggested by the QoE Layer were Applied.

Details about PPM outputs and user classification in stereotypes in this case are presented in Table 5-4.

Table 5-4 Web Content Related Constraints Suggested by the PPM and User Classification into Stereotypes After Each Requested Web Page over a 42 kbps Connection Throughput

ID OF REQUESTED WEB PAGE	PPM OUTPUTS			USER CLASSIFICATION				
	SIZE PAGE (KB)	NO. OBJS	SIZE OBJS (KB)	DEGREE OF MATCH WITH A STEREOTYPE (%)				
				T1 BAD	T2 POOR	T3 NORMAL	T4 GOOD	T5 EXCELLENT
1	10.46	8.0	27.74	0.96	88.97	10.07	0.00	0.00
2	11.58	9.0	33.88	0.0	45.60	54.20	0.20	0.00
3	11.95	9.0	35.93	0.0	45.60	54.20	0.20	0.00
4	12.14	9.0	36.95	0.0	45.60	54.20	0.20	0.00
5	12.25	9.0	37.56	0.0	45.60	54.20	0.20	0.00
6	12.05	9.0	36.50	0.06	77.22	22.7	0.02	0.00
7	12.12	9.0	37.00	0.0	45.60	54.20	0.20	0.00
8	12.24	9.0	36.27	0.06	77.22	22.7	0.02	0.00
9	12.09	9.0	36.69	0.0	45.60	54.20	0.20	0.00
10	12.15	9.0	37.03	0.0	45.60	54.20	0.20	0.00

The model has succeeded to determine the optimal settings for a Web page delivered over a non-changeable network environment after only two transferred pages. The PPM outputs ensure an acceptable download time as perceived by the end-user for the transfer of each Web page. By considering the PPM content - related suggestions the access time per page has varied between 10.9 sec and 13.03 sec with an average value of 11.77 sec across all

pages. This value reflects the improved delivery conditions in this 42 kbps case in comparison with the previous case.

Similar results were obtained for other settings for bottleneck link bandwidth and RTT in the specified range.

5.5.2.1.3 Connection Type 3: between 42 kbps and 56 kbps

The third set of simulations involved a delivery environment characterised by a bandwidth in the range of 42 kbps and 56 kbps and RTT in same 300-500 msec interval. This scenario brings an improvement in the connection bandwidth between the node N2 and Web Client (Figure 5-1) in comparison with the previous tests. The access time per page and the total study session was measured and analysed.

Figure 5-4 illustrates a comparison between the two cases that involve the usage or not of the QoE layer content related suggestions for delivering Web pages in the case when a 56 kbps connection with a RTT = 310 msec was considered. Taking into account the suggested performance-based adaptations different percentages of reduction in the Web page size were applied and the download time per page did not exceed 14 sec. This value is below the 15 secs limit for acceptable download time for a user aware of low bit-rate connection. With the decrease in the access time per page, aggregate access time per session was significantly reduced from 172.56 sec to 102.05 sec achieving a 40.8 % improvement.

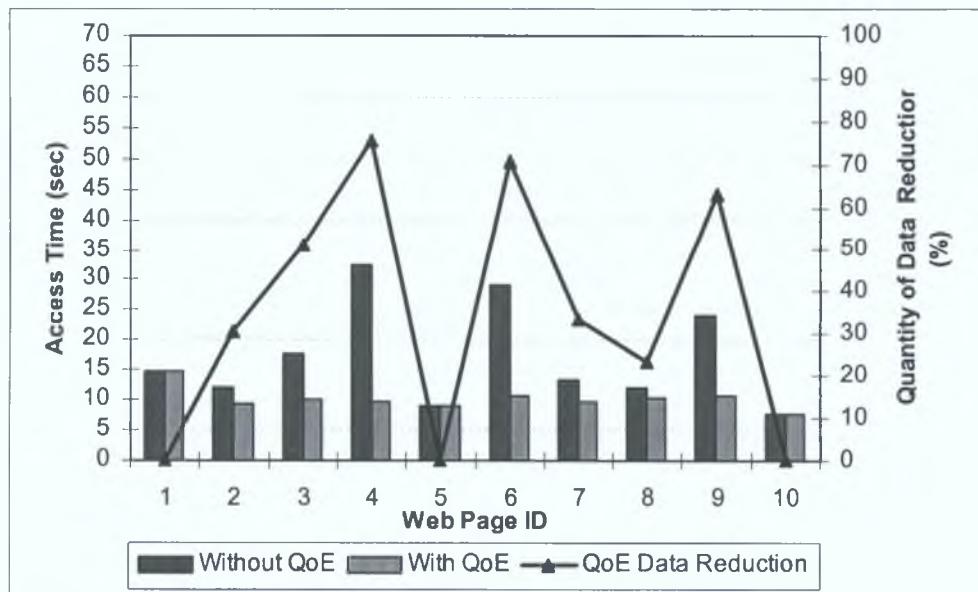


Figure 5-4 Access Time per Page and Total Web Page Size Reduction for a Browsing Session Over a 56 kbps Connection When Content Constraints Suggested by the QoE Layer were Applied.

Table 5-5 presents the Web page characteristics suggested by the model for requested Web pages during browsing session. The degrees of match between the user characteristics and the PPM stereotypes are also indicated for each page.

Table 5-5 Web Content Related Constraints Suggested by the PPM and User Classification into Stereotypes After Each Requested Web Page over a 56 kbps Connection

ID OF REQUESTED WEB PAGE	PPM OUTPUTS			USER CLASSIFICATION				
	SIZE PAGE (KB)	NO. OBJS	SIZE OBJS (KB)	T1 BAD	T2 POOR	T3 NORMAL	T4 GOOD	T5 EXCELLENT
1	11.09	8.0	31.19	0.06	77.22	22.7	0.02	0.00
2	11.89	9.0	35.60	0.0	45.60	54.20	0.20	0.00
3	12.16	9.0	37.08	0.0	45.60	54.20	0.20	0.00
4	12.30	10.0	37.81	0.0	45.60	54.20	0.20	0.00
5	12.38	10.0	38.25	0.0	45.60	54.20	0.20	0.00
6	12.43	10.0	38.55	0.0	45.60	54.20	0.20	0.00
7	12.47	10.0	38.76	0.0	45.60	54.20	0.20	0.00
8	12.70	10.0	38.92	0.0	45.60	54.20	0.20	0.00
9	12.51	10.0	39.04	0.0	45.60	54.20	0.20	0.00
10	12.67	10.0	40.89	0.0	18.52	80.40	1.06	0.00

These results confirm the model's fast learning behaviour in relation to the delivery conditions. The combined effect of the PPM suggestions provided after each requested Web page with those generated for the previously accessed pages ensure a smooth variation in the Web page properties when the performance parameters change their values. For example a decrease in the download time of a Web page may be caused by either improvements in

network conditions or due to the small size of the transferred Web page. In consequence the PPM classifies this user with higher percentages in stereotype classes that represent users with better environment conditions and with lower degree of confidence in stereotype classes that correspond to worse delivery conditions.

One can notice from Figure 5-4 that the download time for Page 10 was quite small (7.75 sec) in comparison to the one measured for the other Web pages. This was due to the fact that the original Web page was smaller and because the delivery conditions improved. However the PPM suggested better Web page characteristics in this instance. By combining these current suggestions with the previous ones, a smoother variation in the suggested optimal characteristics for a Web page was obtained. This prevents eventual noise in the PPM suggestions to influence negatively the overall behaviour of the adaptive system.

5.5.2.1.4 Connection Type 4: between 56 kbps and 64 kbps

A new set of simulations were run in order to assess the improvements brought by the QoE layer for the case when the bottleneck link bandwidth between Web Client and Web Server is between 56 kbps and 64 kbps. RTT was considered between 200 msec and 300 msec. The two cases, when Web page performance based adaptations (With QoE) were considered and no content modification was involved (Without QoE) were compared and analysed.

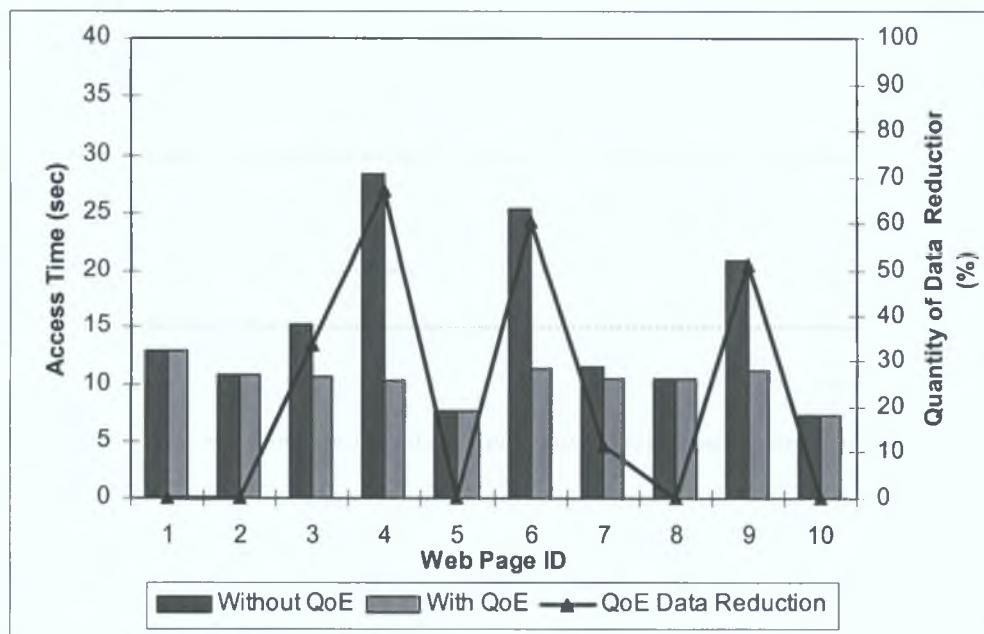


Figure 5-5 Access Time per Page and Total Web Page Size Reduction for a Browsing Session Over a 64 kbps Connection When Content Constraints Suggested by the QoE Layer were Applied.

Figure 5-5 illustrates the measured access time per page and the reduction in the total size of each page during a study session that involved the delivery of ten Web pages over a connection with 64 kbps and RTT=240 msec. The results confirm the observations made in the previous tests. By using the QoE layer the access time per page drops below 12 sec regardless of the original size of the Web page. This is compared with the download time of over 20 sec for large pages such as Page 4, Page 6 and Page 9 when no performance-based adaptations were performed. As the average access time improved a significant improvement in the aggregate access time per session was also recorded from 150.42 sec to 102.93 sec accounting for 31.5 % improvement.

Table 5-6 Web Content Related Constraints Suggested by the PPM and User Classification into Stereotypes After Each Requested Web Page over a 64 kbps Connection

ID OF REQUESTED WEB PAGE	PPM OUTPUTS			USER CLASSIFICATION				
	SIZE PAGE (KB)	NO. OBJS	SIZE OBJS (KB)	T1 BAD	T2 POOR	T3 NORMAL	T4 GOOD	T5 EXCELLENT
1	14.08	12.0	47.88	0.0	17.05	81.97	0.98	0.00
2	14.45	12.0	50.20	0.0	4.66	90.68	4.66	0.00
3	14.58	12.0	50.97	0.0	4.66	90.68	4.66	0.00
4	14.64	12.0	51.35	0.0	4.66	90.68	4.66	0.00
5	14.89	12.0	53.21	0.0	1.12	79.48	19.40	0.00
6	14.88	12.0	53.09	0.0	4.66	90.68	4.66	0.00
7	14.87	12.0	53.01	0.0	4.66	90.68	4.66	0.00
8	14.87	12.0	52.96	0.0	4.66	90.68	4.66	0.00
9	14.86	12.0	52.87	0.0	4.66	90.68	4.66	0.00
10	14.97	13.0	53.67	0.0	1.12	79.48	19.40	0.00

Table 5-6 presents the content related suggestions generated by the PPM after the delivery of each Web page. Once again the model succeeded very fast to determine the optimal characteristics of a page for a given network environment and to classify the user in the correct stereotype classes.

5.5.2.1.5 Connection Type 5: between 64 kbps and 96 kbps

The following tests analysed the same study session over a network environment characterised by bottleneck bandwidth between 64 kbps and 96 kbps and RTT between 100 msec and 200 msec.

For example, Figure 5-6 shows the access time per page and the reduction of the total size of each Web page when the study session is performed over an environment with 96 kbps connection bandwidth and 150 msec RTT.

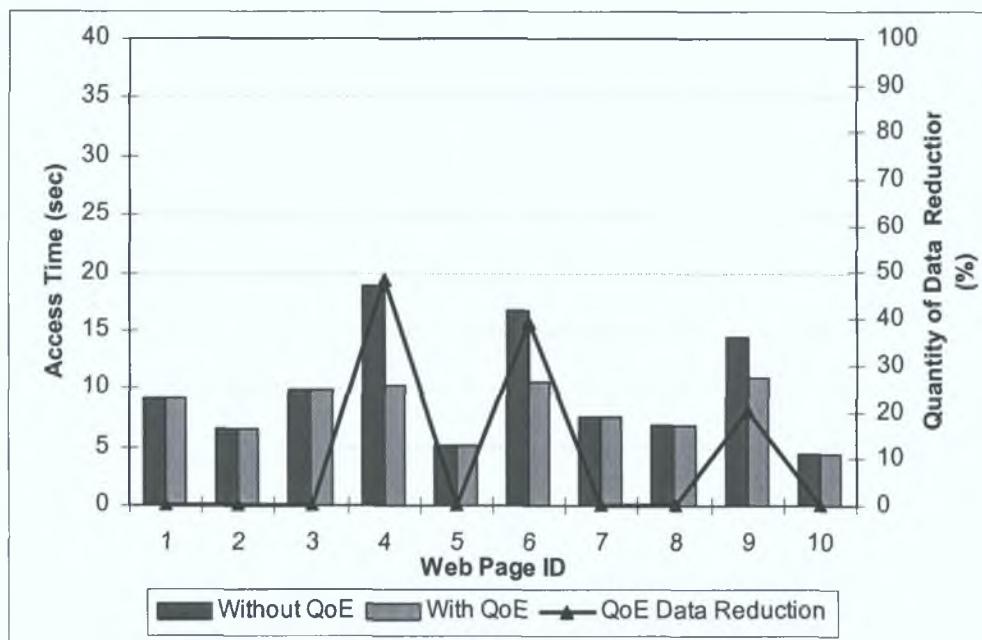


Figure 5-6 Access Time per Page and Total Web Page Size Reduction for a Browsing Session Over a 96 kbps Connection When Content Constraints Suggested by the QoE Layer were Applied.

Since the network conditions have improved in comparison with the previous case, a higher quantity of information is transferred in the same time unit and access time per page below 12 sec is provided when the QoE layer was used. Therefore the aggregate access time per session was improved from 99.91 sec when the original Web pages were directly transferred, to 81.24 sec when QoE suggestions were applied and modified Web pages were delivered.

Next the optimal Web content characteristics as suggested by the PPM after the transfer of each page is presented (Table 5-7). During these simulations, PPM suggestions were applied only to Page 4, Page 6 and Page 9, pages that had higher properties than the optimal suggested ones. The characteristics of the other pages were such that did not require any QoE layer adaptation in order to maintain the access time per page below 12 sec.

Table 5-7 Web Content Related Constraints Suggested by the PPM and User Classification Into Stereotypes After Each Requested Web Page over a 96 kbps Connection

ID OF REQUESTED WEB PAGE	PPM OUTPUTS			USER CLASSIFICATION				
	SIZE PAGE (KB)	NO. OBJS	SIZE OBJS (KB)	T1 BAD	T2 POOR	T3 NORMAL	T4 GOOD	T5 EXCELLENT
1	18.21	16.0	79.02	0.0	0.17	46.24	53.58	0.01
2	19.24	17.0	87.07	0.0	0.02	15.36	84.55	0.07
3	18.89	17.0	84.38	0.0	0.17	46.24	53.58	0.01
4	18.73	17.0	83.04	0.0	0.17	46.24	53.58	0.01
5	19.03	17.0	85.46	0.0	0.02	15.36	84.55	0.07
6	18.89	17.0	84.38	0.0	0.17	46.24	53.58	0.01
7	19.09	17.0	85.92	0.0	0.02	15.36	84.55	0.07
8	19.24	17.0	87.06	0.0	0.02	15.36	84.55	0.07
9	19.13	17.0	86.17	0.0	0.17	46.24	53.58	0.01
10	19.24	17.0	87.07	0.0	0.02	15.36	84.55	0.07

5.5.2.1.6 Connection Type 6: between 96 kbps and 128 kbps

The last set of tests involved a network connection characterised by a bandwidth between 96 kbps and 128 kbps and RTT up to 200 msec was simulated. The same type of performance analysis as for the previous tests was performed. Figure 5-7 illustrates the results of the tests that involved a 128 kbps connection with RTT= 150 msec. As it can be noticed the access time per page decreased below 10 sec that is considered by most of the users as an excellent time. Since the connection had higher bandwidth, much smaller reductions in the Web page sizes were performed in this scenario. At the same time the QoE content constraints affected only 3 pages out of 10. The aggregate access time per session dropped from 78.58 sec, when the original Web pages are transmitted, to 66.05 sec for the case when adaptive adjustments of the Web page sizes were applied.

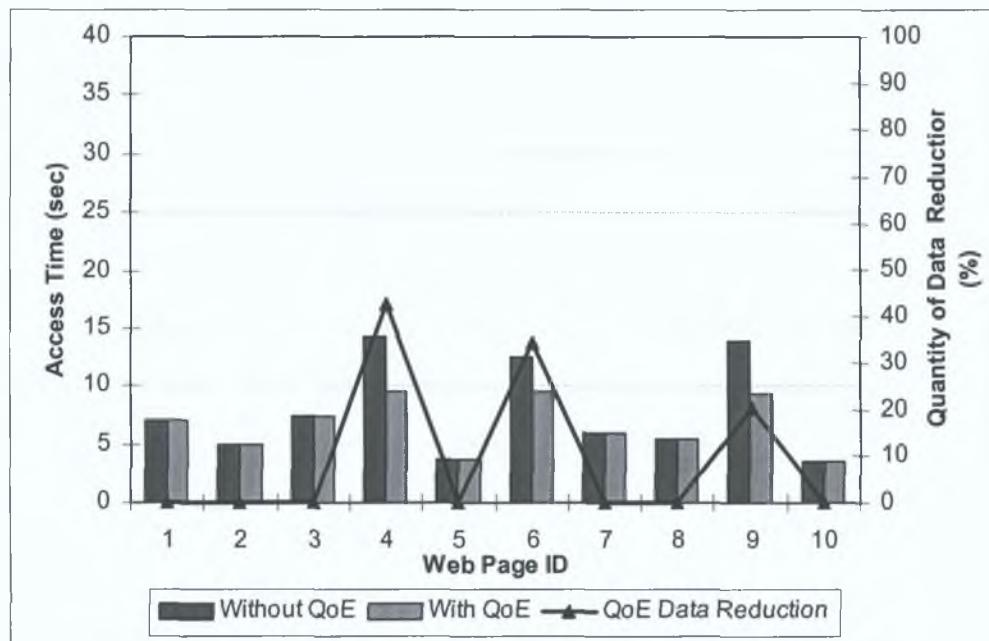


Figure 5-7 Access Time per Page and Total Web Page Size Reduction for a Browsing Session Over a 128 kbps Connection Speed for the Case When Content Constraints Suggested by the QoE Layer were Applied.

The analysis on the PPM outputs presented in Table 5-8 confirms the good behaviour of the proposed performance model and the correct classification of the user in stereotypes.

Simulation results from tests that involved lower values for the bandwidth in the 96 kbps – 128 kbps interval and other values for RTT, less than 200 msec were similar with the ones presented and analysed.

Table 5-8 Web Content Related constraints Suggested by the PPM and User Classification into Stereotypes After Each Requested Web Page over a 128 kbps Connection

ID OF REQUESTED WEB PAGE	PPM OUTPUTS			USER CLASSIFICATION				
	SIZE PAGE (KB)	NO. OBJS	SIZE OJBJS (KB)	T1 BAD	T2 POOR	T3 NORMAL	T4 GOOD	T5 EXCELLENT
1	20.27	19.0	95.11	0.0	0.02	15.36	84.55	0.07
2	20.27	19.0	95.11	0.0	0.02	15.36	84.55	0.07
3	20.27	19.0	95.11	0.0	0.02	15.36	84.55	0.07
4	19.75	18.0	91.09	0.0	0.17	46.24	53.58	0.01
5	20.00	18.0	93.02	0.0	0.17	46.24	53.58	0.01
6	19.70	18.0	90.69	0.0	0.17	46.24	53.58	0.01
7	19.78	18.0	91.32	0.0	0.02	15.36	84.55	0.07
8	19.84	18.0	91.79	0.0	0.02	15.36	84.55	0.07
9	19.66	18.0	90.37	0.0	0.17	46.24	53.58	0.01
10	19.79	18.0	90.37	0.0	0.0	4.65	94.34	1.01

5.5.2.1.7 Summary of the Results

This section summarises the previously presented simulation results that assess the behaviour of the PPM model and the benefits of the QoE layer in various constant low bandwidth delivery network conditions.

Figure 5-8 shows simulation results for the Aggregate access time for a learning session that involved 10 pages, when various connection types were used. Connection bandwidth took values in the range of 28 kbps to 128 kbps whereas the RTT in the 100 msec -1000 msec interval.

It could be noticed that the required reduction in Web page sizes transmitted decrease with the increase of the available bandwidth for the selected ten Web pages, in order to maintain the download time at an acceptable level throughout the study session. Most significantly one can notice that for these low connections QoE layer improved the Aggregate access time with up to 56.2 % for the 28 kbps case. This improvement was achieved by reducing the quantity of data sent during the learning session with up to 62.7 %.

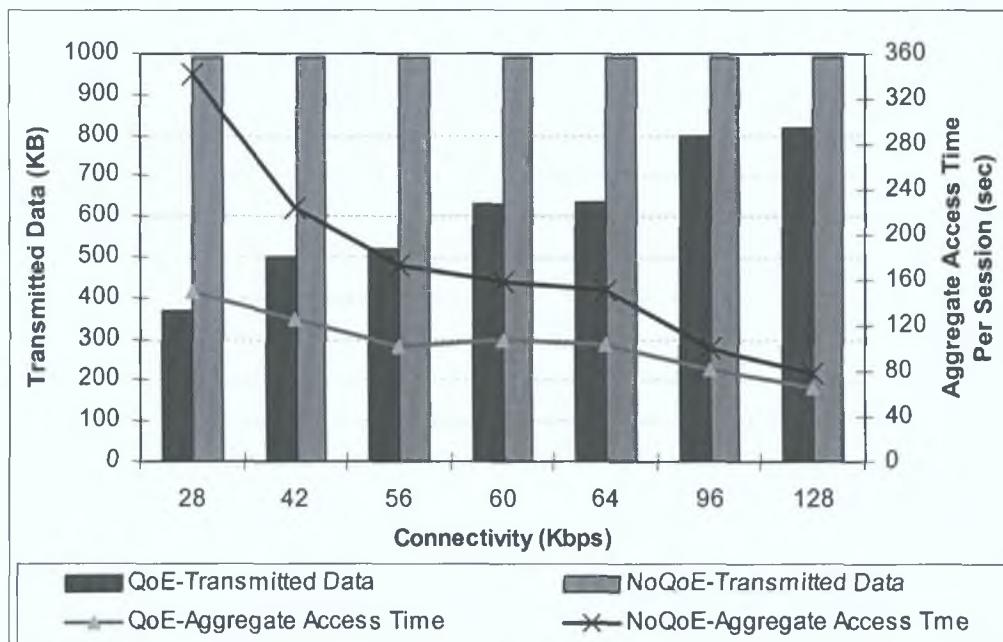


Figure 5-8 Analysis on the Execution Time of a Learning Task and the Total Quantity of Data Transferred during the Learning Session Versus Different Connection Types when the QoE Suggestions Were Taken into Consideration Before Delivering a Web Page

The PPM behaviour as the network connectivity improves, is analysed next. Figure 5-9 and Figure 5-10 show that the model succeeded to determine the optimal characteristics for the Web pages such as the number of embedded objects, total size of the embedded objects and size of the basic Web page. As the network connectivity improves the model provides higher values for the Web page characteristics in order to allow for more information to be transferred to the users while maintaining an acceptable download time.

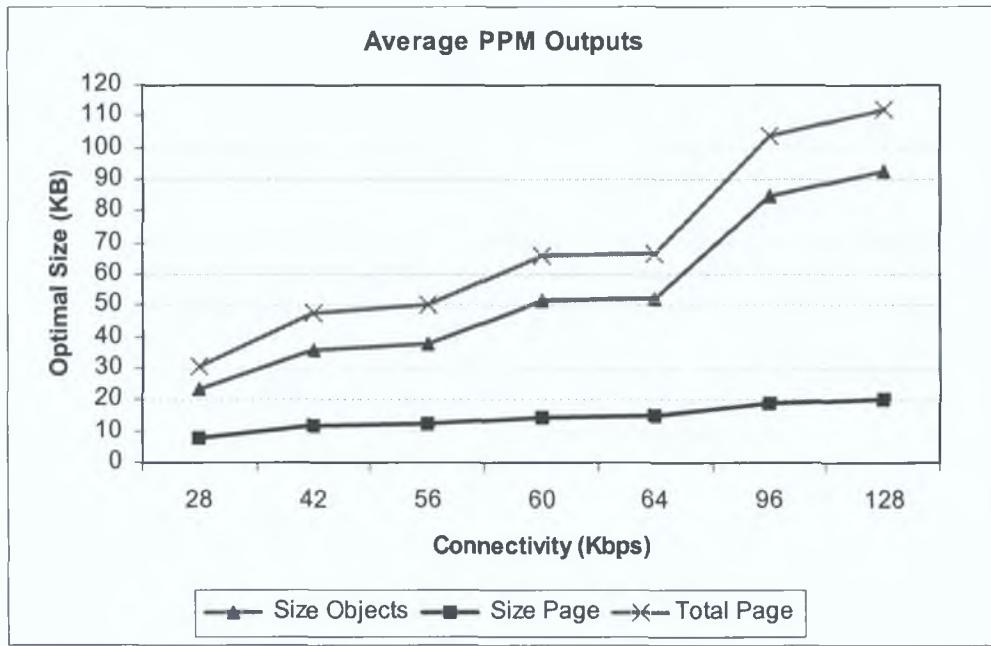


Figure 5-9 PPM Suggestions on the Average Size of the Based-Web page and of the Embedded Objects as well as Average Total Size of the Delivered Web Page for a Learning Session over Different Network Conditions

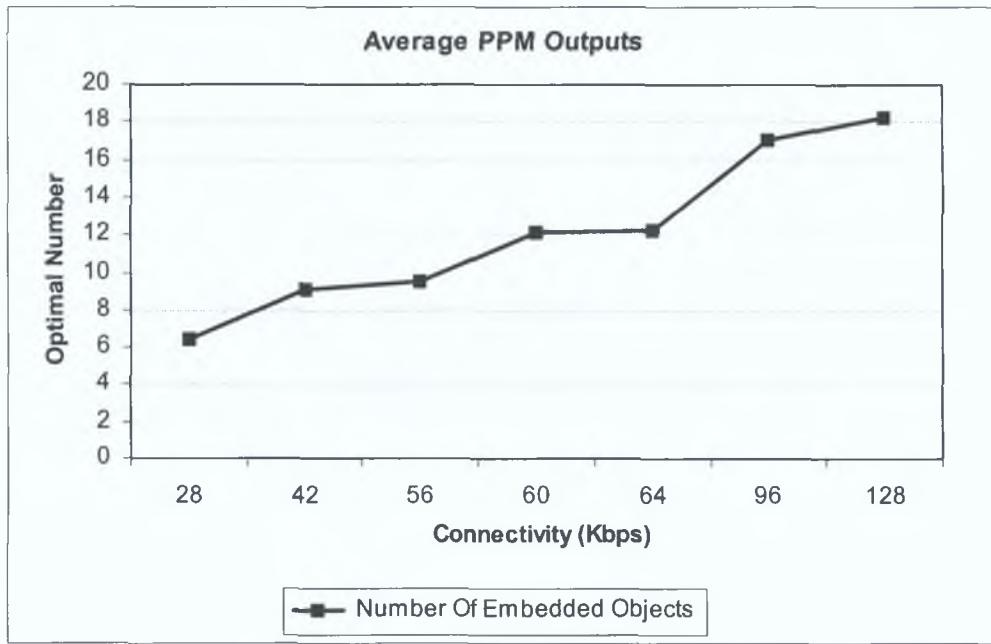


Figure 5-10 PPM Suggestions on the Optimal Number of Embedded Objects in a Web Page Transmitted for a Learning Session over Different Network Conditions

For example if for 28 kbps connection the Web page was restricted to an average of 7.6 KB, it could contain only an average of 6.4 objects of maxim 23.3 KB in total size; For 60 kbps connection based -Web pages of up to an average of 14.6 KB and 12.1 embedded

objects of total size up to an average of 51.2 KB were suggested for transmission. The quantity of information transmitted in the best of the delivery conditions considered (128 kbps connectivity) Web page up to 19.9 KB with an average of 18.2 objects totalising an average of 92.5 KB..

The Web content - related suggestions provided by the PPM ensure a download time per page not higher than 15 sec for very slow connections such as 28 kbps and not higher than 10-12 sec for better network connections (up to 128 kbps).

Table 5-9 summarises the user classification in the PPM stereotype classes during a learning session over different types of network conditions considered. For each connection type an average degree of match with each stereotype was computed as a combination of the degrees of match computed by the model after each delivered Web page.

Table 5-9 Average Degree of Match Between User's Characteristics and Each Stereotype Class Determined by the PPM Model for a Learning Session Over Different Types of Non-Changeable Network Environments

CONNECTION BANDWIDTH (KBPS)	USER CLASSIFICATION				
	AVERAGE DEGREE OF MATCH WITH EACH STEREOTYPE (%)				
	T1 BAD	T2 POOR	T3 NORMAL	T4 GOOD	T5 EXCELLENT
28	16.85	78.71	4.44	0.0	0.0
42	0.11	56.26	43.48	0.14	0.0
56	0.01	45.05	53.67	0.27	0.0
60	0.0	5.54	88.68	5.78	0.0
64	0.0	5.19	87.57	7.24	0.0
96	0.0	0.10	30.80	69.06	0.04
128	0.0	0.07	22.48	77.21	0.24

The values in the table show how the heaviest weight of the classification is placed on the “Bad” – “Poor” classes for 28 kbps connectivity and shifts to “Normal” for 60-64 kbps and to “Normal” - “Good” classes when the connection bandwidth increases up to 128 kbps.

5.5.2.2 Simulation Results for Scenario 1 with a Different Set of Web Pages

The previous tests involved the transmission of 10 pages randomly selected from a set of 100 generated Web pages. In order to validate the correct behaviour of the QoE Layer extra simulation tests that involved a different set of 10 pages randomly selected were performed. The characteristics of these new 10 pages are presented Table 5-10. The same cases of network conditions previously studied were simulated and the same performance parameters were measured.

Table 5-10 Characteristics Of The New Set of Ten Web Pages that Were Generated

WEB PAGE ID	BASIC PAGE SIZE (KB)	NUMBER OF EMBEDDED OBJECTS	TOTAL SIZE OF EMBEDDED OBJECTS (KB)	TOTAL SIZE OF WEB PAGE (KB)
1	9.18	8	82.39	91.57
2	3.10	8	57.96	61.06
3	3.17	6	93.96	97.13
4	10.80	8	190.22	201.02
5	5.61	6	37.73	43.34
6	3.42	9	169.01	172.43
7	12.24	5	64.37	76.61
8	9.38	7	57.68	67.06
9	5.4	10	134.32	139.72
10	3.38	5	36.30	39.67

Next, the results for the Study Case: connection bandwidth 28 kbps that involves the highest level of adjustments as result of the performance-based adaptation are presented.

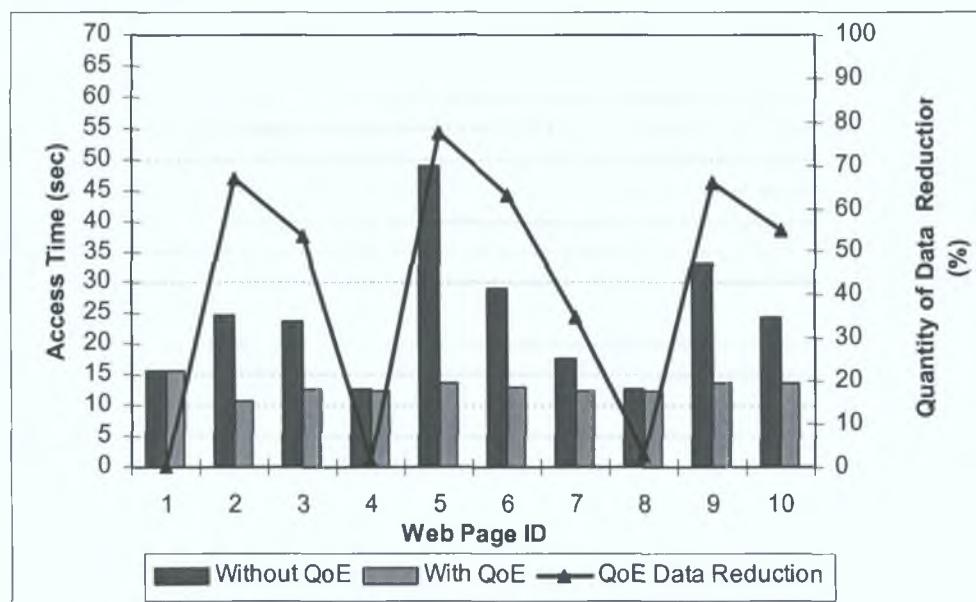


Figure 5-11 Access Time per Page and Web Page Size Reduction for a Browsing Session Over a 28 kbps Connection Speed When Content Constraints Suggested by the QoE Layer were Applied

As already observed in the tests with the first set of ten pages, the QoE Layer has applied performance-based adaptations on the delivered Web pages in order to maintain an access time no higher than 15 sec. Improvements of the access time have determined a lower aggregate access time per session (130.84 sec) than in the case when no adaptations are performed (243.17 sec).

Table 5-11 illustrates the PPM outputs and the user classification in stereotypes after each Web page was requested. A comparison with the results from Table 5-3 indicates that PPM provides the same suggestions on the optimal characteristics of the Web pages when the network conditions are the same, regardless of the set of web pages used during the tests. This is because PPM performs the same classification in stereotype classes in both cases.

These new test results confirm that the QoE Layer has the same behaviour as in the previous tests, regardless of the properties of the generated Web pages. Therefore only the characteristics of the operational environment and the changes that may appear in the network conditions influence the functionality of the QoE Layer.

Table 5-11 Web Content Related Constraints Suggested by the PPM and User Classification into Stereotypes After Each Requested Web Page Over a 28 kbps Connection

ID OF REQUESTED WEB PAGE	PPM OUTPUTS			USER CLASSIFICATION				
	SIZE PAGE (KB)	NO. OBJS	SIZE OBJS (KB)	T1 BAD	T2 POOR	T3 NORMAL	T4 GOOD	T5 EXCELLENT
1	7.26	5.0	16.61	66.69	32.76	0.55	0.00	0.00
2	8.94	6.0	22.69	2.45	82.93	14.62	0.00	0.00
3	8.22	6.0	23.42	12.42	83.92	3.66	0.00	0.00
4	7.86	7.0	23.78	12.42	83.92	3.66	0.00	0.00
5	7.64	7.0	24.00	12.42	83.92	3.66	0.00	0.00
6	7.50	7.0	24.15	12.42	83.92	3.66	0.00	0.00
7	7.40	7.0	24.26	12.42	83.92	3.66	0.00	0.00
8	7.33	7.0	24.33	12.42	83.92	3.66	0.00	0.00
9	7.26	7.0	24.39	12.42	83.92	3.66	0.00	0.00
10	7.21	7.0	24.44	12.42	83.92	3.66	0.00	0.00

5.5.2.3 Simulation Results for Scenario 2

Scenario 2 involves the transmission of a set of Web pages from the chapter one of the AHA! tutorial over low bit rate network environments. This section presents the results of simulations in terms of download time, aggregate access time per session and Web content related suggestions proposed by the QoE layer. These tests will be complemented by the qualitative evaluation of the QoE layer when deployed on the AHA!, an adaptive hypermedia system, whose results are presented in Chapter 6.

Different simulation tests were run in order to analyse how different low bandwidth network conditions affect a learning session. Simulations similar to the ones from Scenario 1 were performed in NS-2 using the same topology illustrated in Figure 5-1, section 5.5.1. The bandwidth of the connection was varied between 28 kbps and 128 kbps and RTT took values between 100 msec and 1000 msec.

Figure 5-12 summarises the simulation results for cases studied with bandwidth of 28, 42, 56, 60, 64, 96 and 128 kbps respectively. One can observe that for the slowest connection case, Aggregate access time per learning session was improved by 37.80 % by decreasing the total quantity of data transmitted by 42 % when the QoE layer was used. Smaller reductions on the Web content were performed for improved network conditions. For the 128 kbps case no content adaptation was required since the characteristics of each Web page were below the PPM suggestions and in consequence the access time per page was below the acceptable threshold of 12 sec.

User surveys performed during the qualitative evaluation and presented in Chapter 6, confirmed that this data reduction did not significantly affect the quality of delivered information.

As Page 3 from the AHA! tutorial chapter is one of the largest in chapter one, next we analyse the effect QoE layer has on it. Page 3 involved the highest QoE performance-based adaptations during the simulation tests for all types of connections. Figure 5-13 presents the access time per page and the percentage of reduction in the size of the Web page after the QoE content related suggestions where applied. The highest improvement of the access time (64.0 %) was obtained for the slowest connection case (28 kbps). During this simulation case the highest quantity of data reduction (57.70 %) had been applied. The plot clearly shows that the access time for Page 3 did not exceed 12-15 sec threshold when the QoE layer was used.

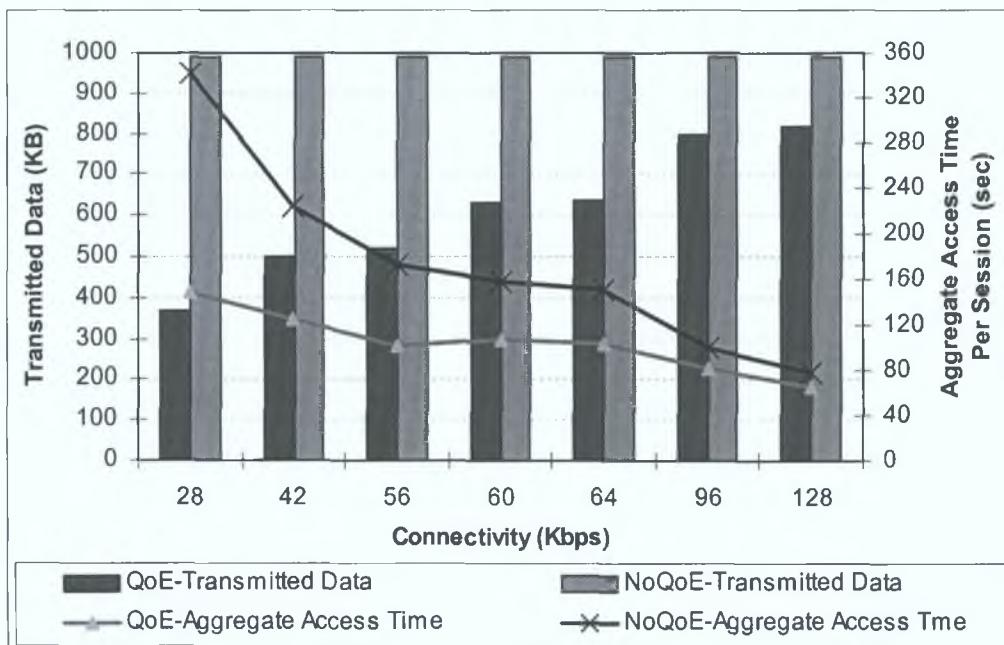


Figure 5-12 Analysis on the Execution Time of a Learning Task and the Total Quantity of Data Transferred during the Learning Session on the AHA! Tutorial Versus Different Connection Types when the QoE Suggestions Were Taken Into Consideration Before Delivering a Web Page

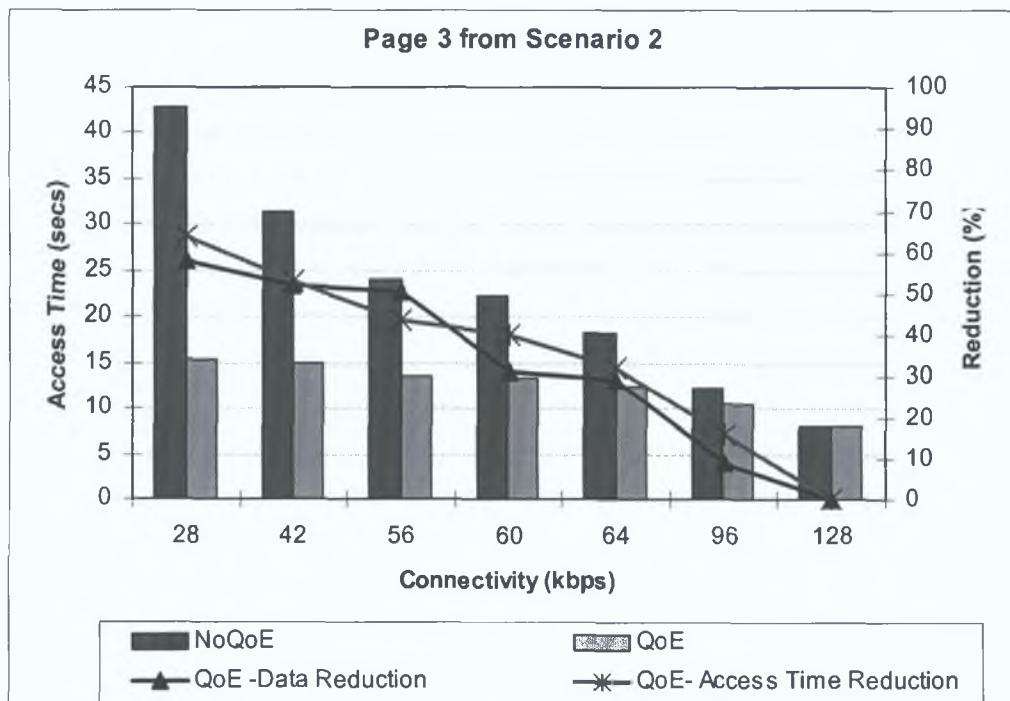


Figure 5-13 Access Time and Total Size Reduction for Page 3 from the Chapter One AHA! Tutorial When the Web Page was Delivered Over Different type of low bit operational environments

Next an analysis of the PPM behaviour and the content related suggestions proposed by the model is performed. Figure 5-14 and Figure 5-15 indicate that the model succeeded to track the current delivery conditions and to compute the optimal Web page characteristics. The better the network conditions between the Web server and the Web Client, the higher Web page characteristics are proposed.

For instance, if in the 28 kbps case the based-Web page was on average 10.45 KB with a maximum an average of 6.3 embedded objects totalising 28.3 KB in the 128 kbps case the average Web page was 20.8 KB with a maximum of an average of 16.6 objects that total 96.1 KB could be transmitted in an acceptable time limit.

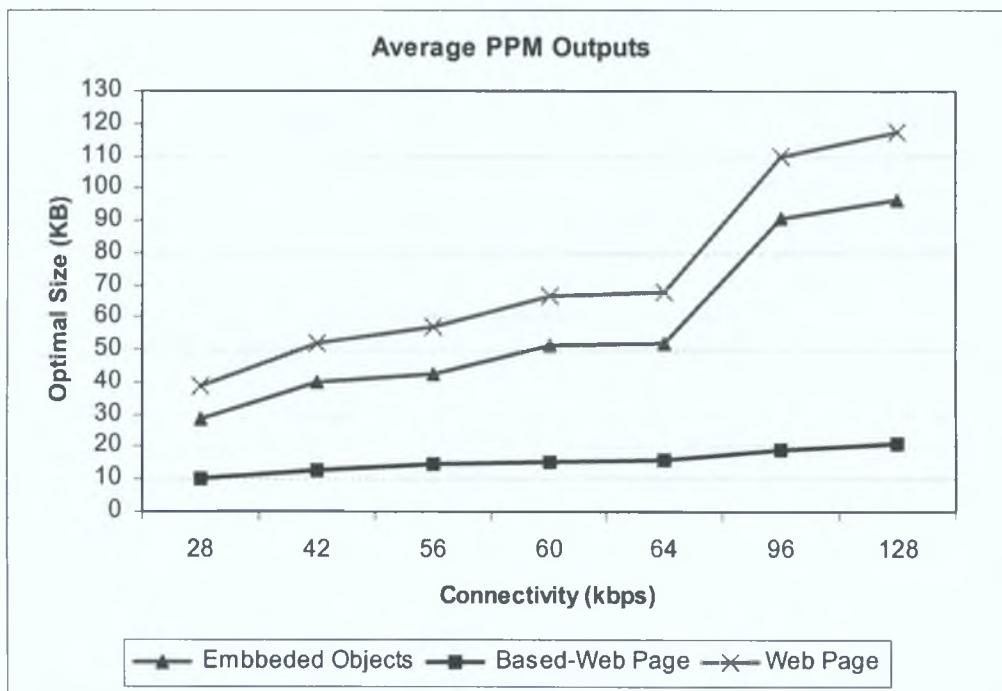


Figure 5-14 PPM Suggestions on the Average Size of the Based-Web page and of the Embedded Objects as well as Average Total Size of the Delivered Web Page for a Learning Session on the AHA! Tutorial Over Different Network Conditions

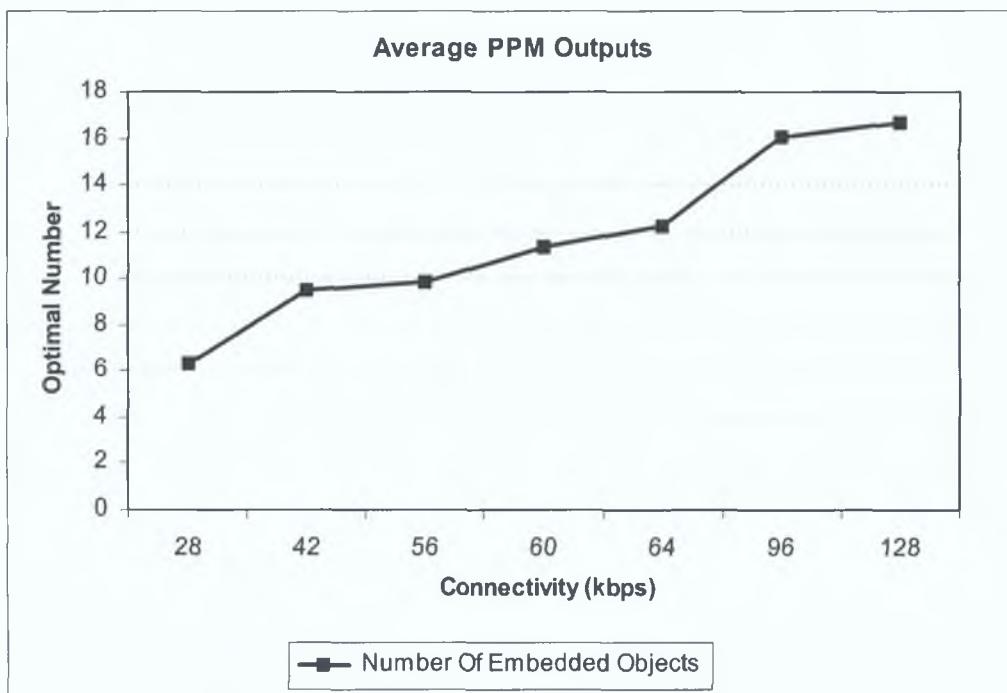


Figure 5-15 PPM Suggestions on the Optimal Number of Embedded Objects in a Web Page Transmitted for a Learning Session on the AHA! Tutorial over Different Network Conditions

5.5.2.4 Summary of the Results

Simulation tests illustrated in sections 5.5.2.1 and 5.5.2.3 have assessed the performance improvements brought by the QoE layer for a learning session over a low bit rate network environment that does not vary in time. Different types of delivery conditions were considered characterised by bandwidth in the range of 28 kbps-128 kbps and RTT in the range of 100 msec -1000 msec.

Two scenarios for a learning session were simulated. The first scenario involved a sequence of ten Web pages randomly selected from a Web site populated with one hundred pages. Details on the selected ten pages are presented in Table 5-2. The second scenario involved a learning session on the first chapter from the AHA! tutorial.

The simulation results for both scenarios have shown that when using the QoE layer the access time per page did not exceed 15 sec for very low network conditions (up to 42 kbps) and 10-12 sec for higher bit rate network delivery conditions (up to 128 kbps). This indicates that PPM successfully determined the current delivery network conditions, made right suggestions on the characteristics of the delivered Web pages providing a satisfactory end-user QoE.

As the access time per page was reduced the Aggregate access time per learning session decreased with up to 56.0 % in the first scenario and with up to 35.0 % in the second scenario. With the increase of the network bandwidth, smaller improvements were obtained such as 18.70 % for Scenario 1 over 96 kbps and 9.60 % for Scenario 2 over 96 kbps; 15.90 % for Scenario 1 over 128 kbps and 0 % for Scenario 2 over 128 kbps.

In conclusion, QoE layer brings high performance benefits for end-users with low connection speed that access a web site that was designed mainly for end-users with higher network connection properties such as 128 kbps as in these simulated scenarios.

5.5.3 QoE Layer Assessment in Variable Low Bit Operational Environment

The objective of the simulation tests is to analyse the QoE layer behaviour and PPM outputs when a learning session is performed over a network environment that changes in time. The learning session involves Web Client accesses to ten Web pages from the Web Server over the topology described in section 5.5.1 according to Scenario 1 (5.5.1.1). The following two cases were considered:

- the network conditions improve from 56 kbps to 64 kbps and then up to 96 kbps. In the same time RTT decreases from 310 to 240 msec and then to 150 msec.
- the network conditions degrade from 96 kbps to 64 kbps and then down to 56 kbps. In the same time RTT increases from 150 to 240 msec and then to 310 msec.

The analysis of the test results includes the following performance parameters: access time per page, aggregate access time per session and quantity of transmitted data. A comparison of values of these performance parameters with those measured in the case when the QoE layer was not used was also performed.

In the same manner as in the non-changeable environment, the aggregate access time per session was measured as the sum of the access times for each requested Web page during the learning session.

The simulation results for the two cases studied are presented next.

5.5.3.1 Step-Wise Changeable Environment from 56 kbps to 96 kbps

The user network connection properties illustrated in Figure 5-1 include a bottleneck link between Node N2 and the Web Client whose bandwidth was changed in three stages during the simulated learning session. In the first stage connectivity was 56 kbps and RTT = 310 msec. The first three pages generated during the learning session were transferred over these network settings. The second stage involved an improvement of the network delivery conditions to 64 kbps and 240 msec RTT while the following three web pages were accessed by the end-user. During the last stage the network conditions further improve and the last 4 pages from the leaning session were transmitted over a link characterized by 96 kbps bandwidth and 150 msec RTT.

Figure 5-16 illustrates the changes into the network delivery conditions during the learning session.

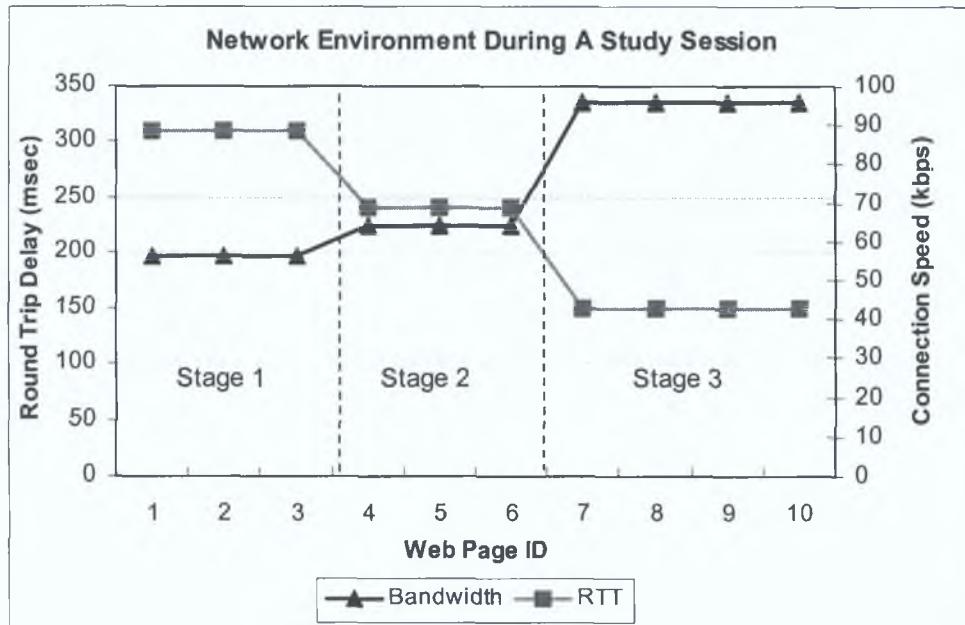


Figure 5-16 Network Properties During a Simulated Learning Session that Involves Ten Web Pages

In a similar manner as for the non-variable environment simulations (section 5.5.2) the access time per page and The QoE layer determined reduction of the size of each delivered page were analysed. Figure 5-17 shows the measured values when the QoE layer was used in comparison with the case when no performance-based content adaptations were performed. When the QoE layer was used the access time per page was below 15 sec for the

lowest connection stage (56 kbps) and further decreased to below 10 sec for the best connection stage.

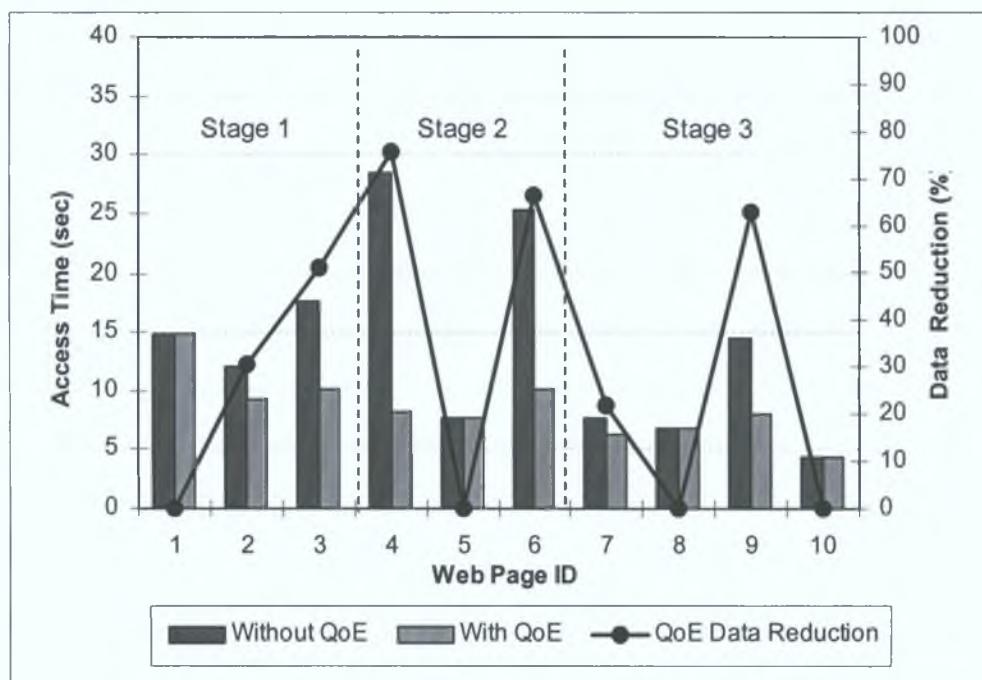


Figure 5-17 Access Time per Page and Web Page Size Reduction for a Study Session that Involves Ten Web Pages in Changeable Network Environment from 56 kbps to 96 kbps

Since the current PPM suggestions on the optimal characteristics for the Web page to be delivered are combined with the previously generated ones in worse network conditions, the final QoE layer suggested characteristics for Web pages improve slower than the network conditions. This is due to the learning behaviour of the performance model that tries to overcome sharp fluctuations of the network environment and protect against eventual noise in the recorded delivery conditions. In consequence, higher size reductions were proposed on the largest Web pages such as Page 4, Page 6 and Page 9 during stage 2 and stage 3 than those performed when the pages were delivered over a constant 64 kbps or 96 kbps connections. For example, the size of Page 6 was reduced with 66.1 % when delivered over a 64 kbps connection throughput that recently evolved from 56 kbps (Figure 5-17) in comparison with 60.5% when delivered over constant 64 kbps links (Figure 5-5).

A very significant achievement when using QoE-based adaptations in comparison with the case that no performance adaptation was performed is that aggregate access time per session has improved with 38.3% confirming the benefit of the proposed solution.

Table 5-12 Web Content Related Constraints Suggested by the PPM and User Classification Into Stereotypes After Each Requested Web Page over a Changeable Network Environment from 56 kbps to 96 kbps During a Learning Session

ID OF REQUESTED WEB PAGE	PPM OUTPUTS			USER CLASSIFICATION				
				DEGREE OF MATCH WITH A STEREOTYPE (%)				
	SIZE PAGE (KB)	NO. OBJS	SIZE OJBJS (KB)	T1 BAD	T2 POOR	T3 NORMAL	T4 GOOD	T5 EXCELLENT
1	11.09	8.0	31.19	0.06	77.22	22.7	0.02	0.00
2	11.89	9.0	35.60	0.0	45.60	54.20	0.20	0.00
3	12.16	9.0	37.08	0.0	45.60	54.20	0.20	0.00
4	12.83	10.0	40.93	0.0	4.67	90.67	4.66	0.00
5	13.44	11.0	44.87	0.0	1.12	79.48	19.4	0.00
6	13.66	11.0	46.15	0.0	4.67	90.67	4.66	0.00
7	14.61	12.0	53.14	0.0	0.02	15.36	84.55	0.07
8	15.32	13.0	58.38	0.0	0.02	15.36	84.55	0.07
9	15.64	13.0	60.67	0.0	0.17	46.24	53.58	0.01
10	16.10	14.0	64.12	0.0	0.02	15.36	84.55	0.07

Table 5-12 illustrates the PPM outputs and user classification in PPM stereotype classes, after each delivered Web page. One can notice that the model succeeded to classify the user in better stereotype classes as the network conditions improved. For example Page 4 was delivered over an improved connection to 64 kbps. Therefore the user was classified for example with 90.6 % in the “Normal” stereotype class in compression with 54.2% when the user accessed the previous three pages under poorer network conditions. In consequence the

model suggested higher values for the Web page characteristics that match the improvements in network conditions.

Similar PPM behaviour was noticed when the network conditions further improved to 96 kbps and Page 7, Page 8, Page 9 and Page 10 were transmitted. As the number of Web pages delivered over similar network condition increase, the influence of the suggestions generated for previous network stages (e.g. Stage 1 - 56 kbps and Stage 2 - 64kbps) decreases and the PPM determines the optimal content - related suggestions for the new network conditions (e.g. Stage 3 - 96 kbps).

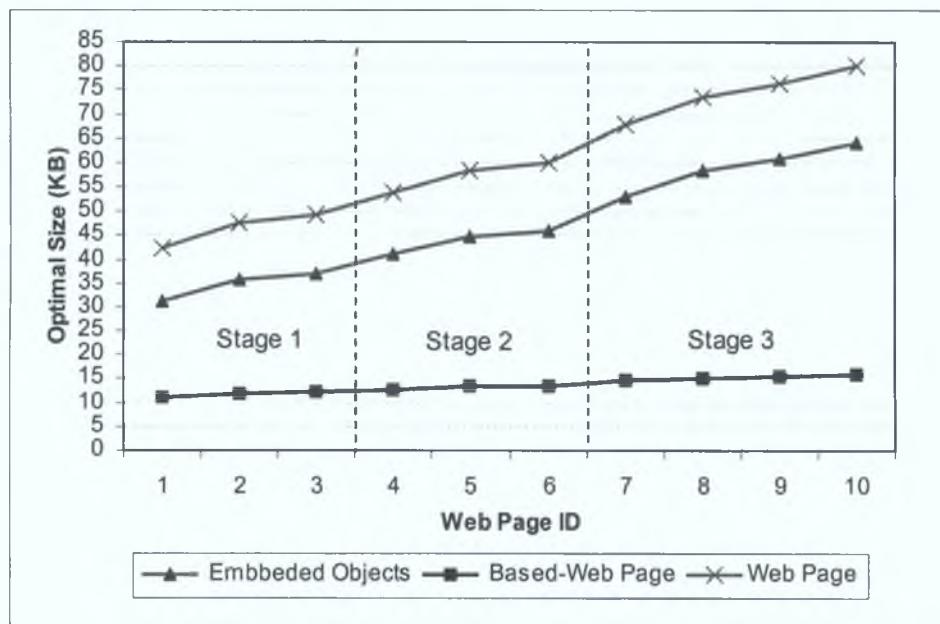


Figure 5-18 PPM Suggestions on the Optimal Size of the Based-Web Page and of the Embedded Objects as well as Total Size of the Web Page, during a Ten Page Learning Session, with the Step-Wise Improvement of the Network Conditions

Figure 5-18 and Figure 5-19 summarise the information presented in Table 5-12 and illustrate the evolution of PPM suggestions related to the optimal characteristics for the delivered Web page with the step-wise improvements in the network condition and in the number of the accessed Web page during the learning session.

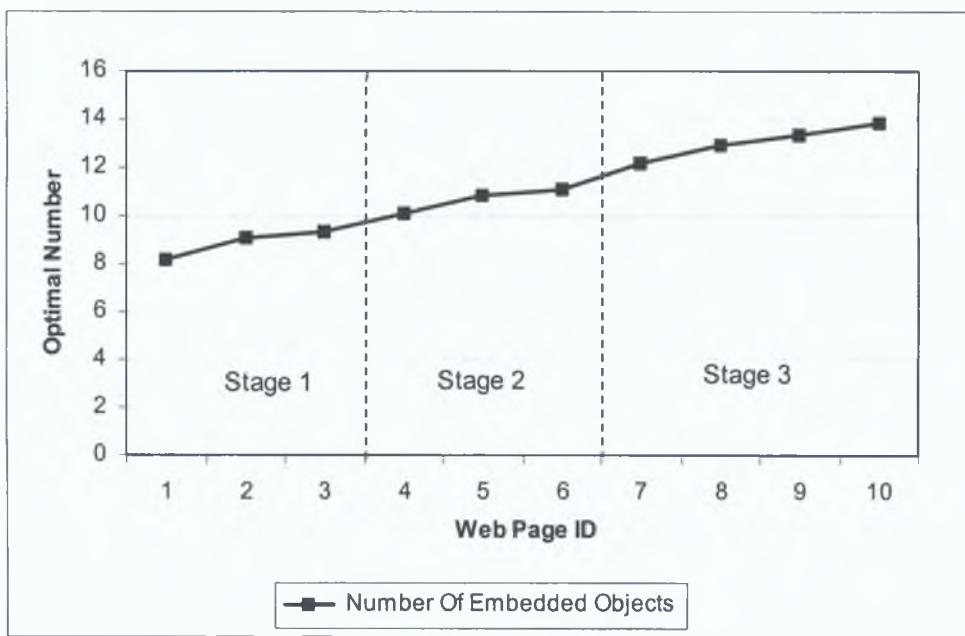


Figure 5-19 PPM Suggestions on the Optimal Number of Embedded Objects during a Ten Page Learning Session with the Step-Wise Improvement of the Network Conditions

5.5.3.2 Step-Wise Changeable Environment from 96 kbps to 56 kbps

The goal of this simulation was to analyse the same learning session that involves the access of ten Web pages when the network conditions deteriorate from 96 kbps bandwidth and 150 msec RTT to 56 kbps and 340 msec RTT via an intermediate step characterized by 64 kbps bandwidth and 240 msec RTT.

Figure 5-20 illustrates the changes into the Client- Server network conditions over the learning session.

An analysis performed on the access time per page and on the quantity of transmitted information (Figure 5-21) reveals that the usage of QoE layer reduced the access time for each page below 10 sec for high bit rate connection (Stage 1 – 96 kbps) and below 15 sec for low bit rate connection (Stage 3 – 56 kbps). The Aggregate access time per session was also reduced on average by 23.4 %.

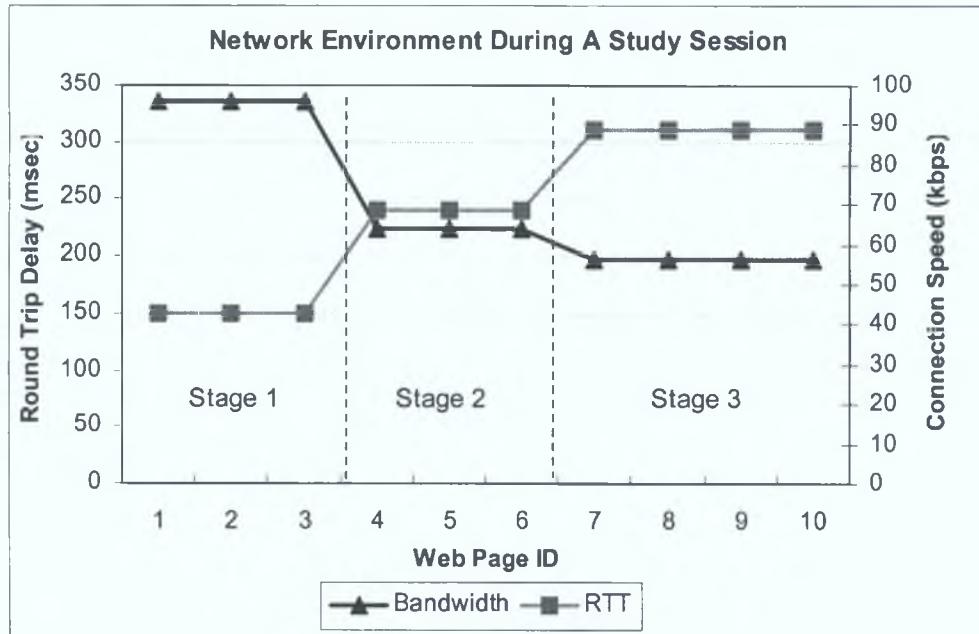


Figure 5-20 Network Properties During a Simulated Learning Session that Involves Ten Web Pages

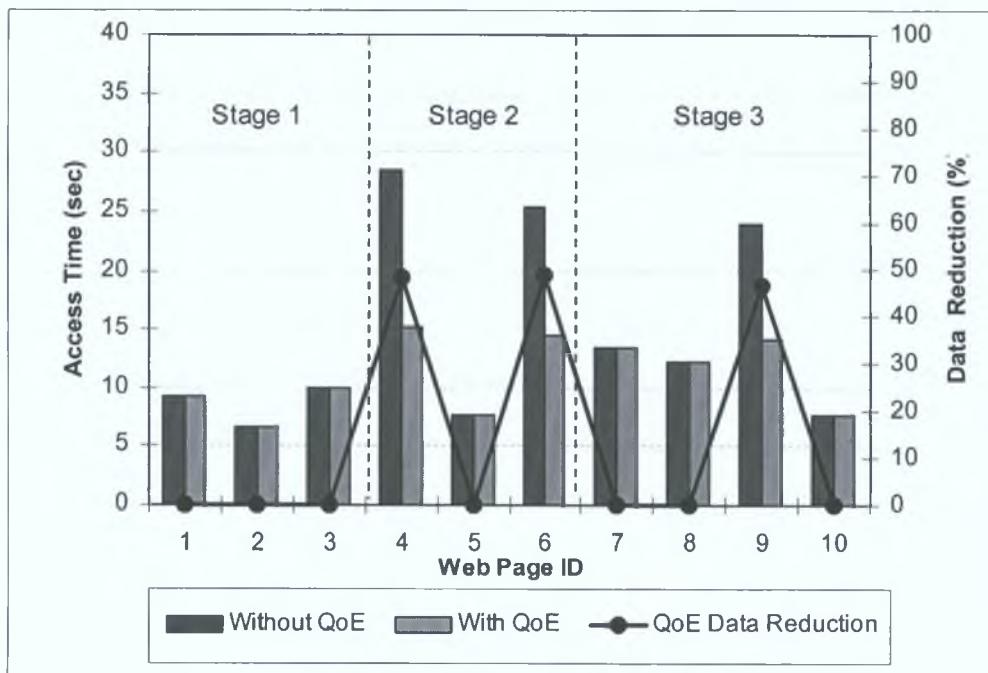


Figure 5-21 Access Time per Page and Web Page Size Reduction for a Study Session that Involves Ten Web Pages in Step - Wise Changeable Network Environment from 96 kbps to 56 kbps

The influence of the generated suggestions for Web pages delivered over the highest bit rate connection case (Stage 1 – 96 kbps) was also noticed in the final suggestions proposed for the lowest connectivity case (Stage 3 – 56 kbps) that followed. In consequence

a smaller reduction of the delivered quantity of information was performed for each page delivered during the Stage 3 (56 kbps) in comparison with the case when all the learning session was performed over a constant 56 kbps connection (Figure 5-4). Although the access time for these pages such as Page 7, Page 8, Page 9 and Page 10 was slightly higher (an average of 11.85 sec) than for the constant connection case (an average of 9.64 sec) it did not exceed the 12 sec threshold value that is considered acceptable for low home connections.

Table 5-13 Web Content Related Constraints Suggested by the PPM and User Classification Into Stereotypes After Each Requested Web Page over a Step – Wise Changeable Network Environment from 96 kbps to 56 kbps During a Learning Session

ID OF REQUESTED WEB PAGE	PPM OUTPUTS			USER'S CLASSIFICATION				
	SIZE PAGE (KB)	NO. OBJS	SIZE OJBJS (KB)	T1 BAD	T2 POOR	T3 NORMAL	T4 GOOD	T5 EXCELLENT
1	18.21	16.0	79.02	0.0	0.17	46.24	53.58	0.01
2	19.24	17.0	87.07	0.0	0.02	15.36	84.55	0.07
3	18.89	17.0	84.38	0.0	0.17	46.24	53.58	0.01
4	17.46	15.0	73.98	0.02	35.06	64.92	0.0	0.0
5	17.15	15.0	71.31	0.0	1.12	79.48	19.40	0.0
6	16.63	14.0	67.40	0.0	17.05	81.97	0.98	0.0
7	15.94	14.0	62.77	0.2	63.59	36.0	0.21	0.0
8	15.40	13.0	59.30	0.2	63.59	36.0	0.21	0.0
9	15.01	13.0	56.60	0.2	63.59	36.0	0.21	0.0
10	14.91	12.0	56.69	0.0	18.52	80.4	1.06	0.0

Table 5-13 presents the PPM content related suggestions and the user classification in stereotype classes. Once again the results confirm that the model successfully tracks the changes in the network environment. As the network delivery conditions deteriorate the user is classified with higher percentages in stereotype classes that suggest tougher constraints on the optimal characteristics of a Web page. For example during Stage 1 case (96 kbps) the user was classified with the highest percentage in the “Good” stereotype class (average 63.90 %), while during Stage 2 the user was classified in the “Normal” stereotype class with an average of 75.45 %.

Figure 5-22 and Figure 5-23 summarise the information presented in Table 5-13 and illustrate the evolution of the PPM suggestions on the optimal characteristics of Web pages with the step- wise degradation of the network conditions during the learning session.

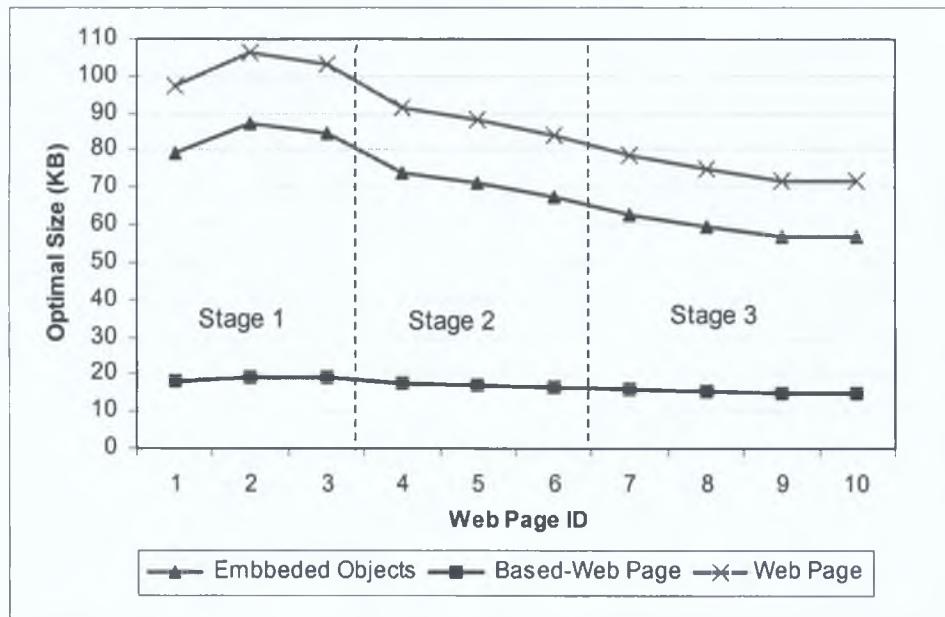


Figure 5-22 PPM Suggestions of the Optimal Size of the Based-Web Page and of the Embedded Objects as well as Total Size of the Web Page, during a Ten Pages Learning Session, with the Step – Wise Degradation in the Network Conditions

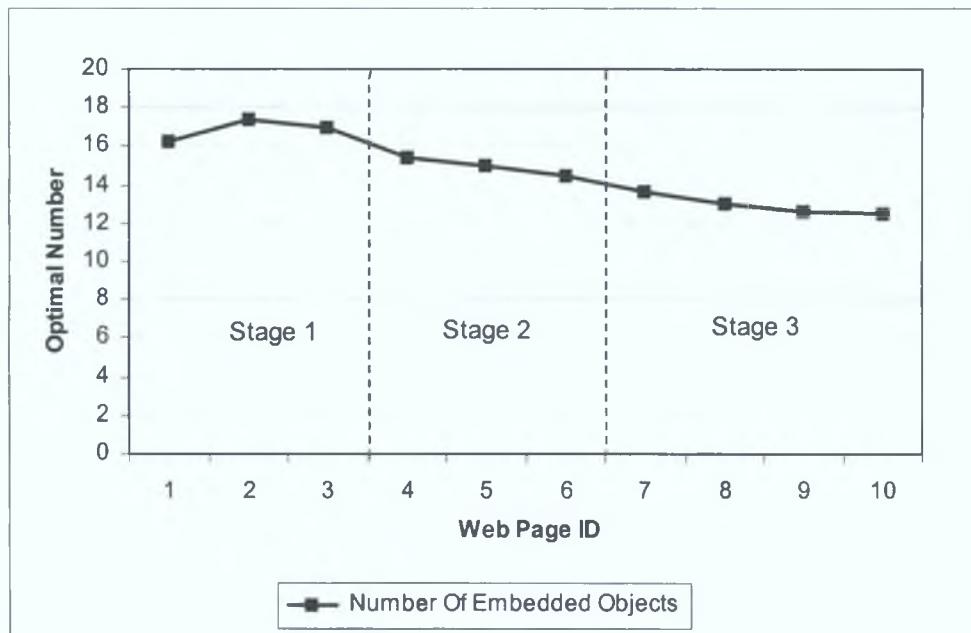


Figure 5-23 PPM Suggestions of the Optimal Number of Embedded Objects during a Ten Pages Learning Session with the Step – Wise Degradation in the Network Conditions

5.5.3.3 Summary of the Results

The simulation tests presented in section 5.5.3 analysed a learning session performed over a step-wise changeable network environment that involves three network stages. The learning session consisted of a sequence of ten pages randomly selected from a Web site populated with 100 pages. Details on the selected pages are presented in Table 5-2. Two cases were considered. In the first case (Case 1) the network bandwidth improved from 56 kbps – Stage 1 to 64 kbps – Stage 2 and up to 96 kbps – Stage 3 while RTT decreased from 310 msec to 150 msec. The second case (Case 2) involved the degradation in the network bandwidth from 96 kbps down to 56 kbps while RTT increases.

The test results have shown that the PPM model successfully tracked the changes in the network environment from one stage to another. In consequence the user was classified with a higher percentage in a stereotype class that corresponds to a better end-user network connectivity for Case 1 and to a worse network connectivity for Case 2. Therefore higher (Case 1) or lower (Case 2) Web page properties were suggested and imposed by the QoE layer during the browsing session.

Regardless of the network stage the access time per page did not exceed 15 secs when the QoE layer was used. Improvements of the aggregate access time per learning session by 39.8% for Case 1 and 23.4 % for Case 2 were also obtained.

An analysis on the PPM outputs during a learning session over a changeable network environment indicates that the “learning” behaviour of the PPM model about the delivery conditions overcomes sharp fluctuations of the network environment. PPM combines the current generated PPM suggestions for a Web page to be delivered over a given network stage with the ones generated for the previous pages delivered over a different network stage. In consequence the values of the Web page properties increase (Case 1) or decrease (Case 2) slower than the changes in the network environment.

5.6 Chapter Summary

The aim of the tests presented in this chapter was to investigate the performance improvements provided by the QoE layer when performance-based Web page content adaptations are performed during a simulated learning session. Performance was analysed in term of access time per page, aggregate access time per learning session and quantity of transmitted information. The behaviour of the Perceived Performance Model (PPM) and the optimal web page characteristics generated by the PPM after a Web page was delivered were also analysed.

Using Network Simulator version 2 (NS-2), learning sessions over different types of low bit rate network environments characterised by end-user connection bandwidth in the range of 28 kbps - 128 kbps were simulated. Constant and step-wise changeable network conditions, during the learning session, were set up for the simulation tests.

The test results showed that the access time per page did not exceed 12-15 sec threshold considered acceptable for low bit rate environments. More significantly the aggregate access time per learning session improved by up to 56 % in constant network conditions and by up to 39.8 % in step-wise changeable network conditions when the QoE layer was used. These improvements were due to controlled reductions of the quantity of transmitted information.

Chapter VI

QoE Layer Evaluation In Educational Area

6.1 Chapter Introduction

The evaluation of Adaptive Hypermedia Systems (AHS) is a active research topic regardless of the applicability area of the adaptive systems. As Hook [167] noted, evaluating a system is an intricate task and it becomes even more difficult if the system is adaptive. Although AH research has rapidly evolved after 1996 and promising results were produced especially in educational context, as reported by Brusilovsky in [5], this area still lacks standards and evaluation frameworks for measuring the effectiveness of adaptations performed by AHS.

In order to highlight current trends in the evaluation of educational AHS, the first section (6.2) of this chapter briefly outlines a number of strategies and testing techniques used by different researchers. Based on these observations the strategy used for the evaluation of the proposed QoE layer is defined in section 6.3.2. This involves the open-source AHA system (which has been intensive tested in the educational area) and QoEAHA, an QoE-based enhanced version of the AHA!

Next, the objectives of the experimental tests are highlighted and the set-up conditions and testing scenarios presented. The QoEAHA evaluation (section 6.3.3) investigates the feasibility and utility of applying the QoE layer in order to support extra adaptation based on the end-user perceived performance. The learning capabilities offered by the system and the students learning outcomes are also studied. The benefits provided by the QoE extension were then analysed by comparing the test results of QoEAHA with the original AHA! system.

6.2 Evaluation Issues in Adaptive E-Learning Systems

6.2.1 Current Evaluation Approaches

Currently, the most used method in the evaluation of adaptive educational systems adopts a “*with or without adaptation approach*” [168], considering that the evaluated system can have adaptive and non-adaptive versions. The experiments are conducted between two groups of learners, one working with an adaptive version of the system and the other with its non-adaptive version. This conventional method involving the comparison of an adaptive and a non-adaptive version of an application is debatable [169] and highly depends on how the non-adaptive version was obtained.

A possibility is to “disable” all adaptive features of the adaptive version [169]. Since most adaptive systems are developed with particular adaptive techniques in mind, removing those techniques affects the system’s basic functionality and the comparisons are always in favour of the adaptive systems. Although highly used, this approach does not offer fair results.

Another possibility is for this comparison to be performed with the original non-adaptive system prior to adding adaptive functionality. This lacks the advantage of a well-structured domain model as in the adaptive version and may lead again to unfair results according to some opinions [169]. This is especially true since both the information content of the pages and the link structure and/or presentation layout may be different in the two versions. Short explanations, additional comparisons can be added, changes in the presentation style can be performed and/or presentation length can be modified. Links or link destinations can be added, removed, sorted and/or annotated [169].

A third possibility is to disable some adaptive features from the adaptive version of the system. This allows for the comparison to be made between two adaptive versions of the same system, having different degrees of adaptiveness. This type of comparison is used to show the benefits of some adaptive techniques against others.

From the evaluation strategy point of view, two main directions were proposed for the assessment of AHS:

- Evaluation of the system as a whole
- Layered-based system evaluation

6.2.1.1 System Evaluation “As a Whole”

This first approach aims at **adaptive system evaluation “as a whole”** [170] and is very often used in the educational area. The evaluation process focuses mainly on the overall learners’ performance and their satisfaction related to the use of the adaptive system. This user satisfaction can be quantified by selected and measurable criteria. In this context the most used criteria in the evaluation process of educational systems are: task completion time, learning performance assessed by comparing the results of a pre-test and post-test, number of navigation steps, number of times the subjects revisited “concepts” they were attempting to learn, users satisfaction reflected through questionnaires [171, 172, 173, 174].

6.2.1.2 Layer-Based System Evaluation

Very recently, a new approach was recommended for the evaluation of the adaptive applications and advocated by a number of researchers [170, 175, 176, 177, 178]. This approach involves a **layered evaluation of adaptive applications**. Unlike the previous approach that focuses on assessing users’ performance and satisfaction in relation with the system as a whole, layered evaluation assesses the success of the adaptation by decomposing the system into different layers and evaluating them one by one [179]. The different layers reflect various aspects and stages of the adaptation, starting from low-level input data acquisition and user monitoring to high-level assessment of the behavioural complexity of the users. Although the current proposed frameworks are described at different levels of granularity [176, 180], the evaluation process is mainly divided in two phases or layers: evaluation of the interaction assessment and evaluation of the adaptation decision-making [175].

Karagiannidis et. al [175] has proposed a framework for layered evaluation that consists of two layers:

- *Layer 1: Interaction Assessment Evaluation* tests if the system detected the learner’s goals, knowledge, preferences, interests, user’s experience with the respect of the hyperspace. It also assesses whether the assumption drawn by the system concerning characteristics of the user-computer interaction is valid. This evaluation is based on comparison between experts’ opinions and information stored in learner (user) model.
- *Layer 2: Adaptation Decision-Making* tests if the selected adaptive technique is appropriate, valid and meaningful for learner’s goal or improves interaction for specific learner’s interests, knowledge, etc. This evaluation consists of tests

based on scenarios that involve a particular goal for the learner and assess the success of quality improvement. Learners and/or experts can evaluate the tests.

The division of the evaluation process into the two layers that also reflect the main phases of the adaptation may help to determine where the fault (if any) of the adaptive system may be and to target the solutions accordingly [170]. For example it can be the case that adaptation decisions are reasonable but they are based on incorrect system assumptions, or that the system assumptions are correct but the adaptation decision is not meaningful. Both cases can happen at the same time, too.

A more detailed approach was proposed by Weibelzahl et. al [176] and consists of a framework for layered evaluation based on four layers:

- *Layer 1: Evaluation of the Reliability and Input Data.* This evaluation prevents unreliable input data to result in miss-adaptation.
- *Layer 2: Evaluation of Inference.* This layer evaluation tests the inference mechanism in different environments under real world conditions
- *Layer 3: Evaluation of Adaptation Decision.* The idea of this evaluation is that if some user properties have been inferred, several adaptation possibilities exist. (e.g. with/without adaptive guiding, with/without link annotations).
- *Layer 4: Evaluation of Interaction.* In this case human system interaction has to be evaluated to prevent confusion and dissatisfaction of the users. Different objective and subjective measures are taken into account such as: system usability, solution quality, frequency of tasks success, number of required hints, etc.

One can notice that both evaluation strategies (evaluation “as a whole” and layered evaluation) aim at assessing three important features of the educational applications:

- usability of the application,
- learner achievement
- learning performance.

6.2.2 Evaluation Tests of Adaptive Educational Systems

6.2.2.1 Usability Evaluation Tests

One of the most important features of any software application is its usability. According to the ISO 9241 [181] standard, usability represents the effectiveness, efficiency and satisfaction that a software application offers to its users in a given context of use and task.

In an educational environment, the usability of a software application is related to its pedagogical value [182]. Although there is a large amount of knowledge relating to educational software usability evaluation strategies [183], currently there are no well-defined techniques for usability evaluation of e-learning (distance learning) environments [184]. This is due to the fact that e-learning is an area of relatively short history, users of e-learning tools can access them through various computer, network and social contexts and the characteristics of a typical users of e-learning services can not be easily predicted [185].

Some of the most used methods proposed in the literature to be applied during the usability evaluation are: *query techniques* (interviews and questionnaires), *logging of user performance* in laboratory conditions, *timing and keystroke level measurements*, *subjects' observation* through adequate equipment, *heuristic evaluation*, etc. These methods are applied after the subjects have interacted with the system by performing one or multiple tasks. Usually the usability is analysed through five major characteristics: efficient to use, easy to remember, pleasant to use, easy to learn, few errors.

Questionnaires and interviews are the most widely used technique since they provide a quantitative and qualitative measure of usability and they serve as an objective comparison of two systems. This technique offers a concise test of usability, it gets directly the users' viewpoint and attitude and it is suitable for a wide range of end-users, especially students. A big advantage is that it does not require the presence of an evaluator. In this context, Preece [186] has suggested a list of guidelines for creating questions for the questionnaires, a list that currently is widely used for the usability evaluation of the web-based systems.

Heuristic evaluation is also a widely accepted method for diagnosing the system's usability due to the fact that it can be completed in a relatively short period of time. This methodology involves an expert that evaluates the system using a set of recognized usability principles. These principles called "heuristics" were discussed and presented by Nielsen in [187].

6.2.2.2 Learner Achievement Evaluation

In the evaluation of a learning process both quality and quantity of learning (learning outcome) should be assessed. Therefore, learner achievement (defined as the degree of knowledge accumulation by a person after studying certain material) continues to be a widely used barometer for determining the utility and value of learning technologies. It is analysed in the form of *course grades, pre/post-test scores, or standardized test scores*.

A course grade is a certification of competence that should reflect, as accurately as possible, a student's performance in a course. There are multiple methods for assigning grades, such as weighting, distribution gap method, curve, percent grading, relative grading, and absolute standard grading.

Pre/Post test scores are also a viable methodology to assess the extent to which an educational intervention has had an impact on student "learning". Pre-test is used to determine subjects' prior knowledge on the studied domain, while post-test is used to examine learning outcomes after the learning related intervention.

Standardized tests scores give a "standard" of measure of students' performance when a large numbers of students (often geographically distributed) take the same test.

Tests, quizzes or exams are methods used to evaluate students and assess whether they learned what it is expected to be learned. Jacobs and Chase [188] made a distinction between the three terms: *tests, quizzes and exams*, based on the scope of content covered. *An examination* is the most comprehensive form of testing. *A test* is more limited in scope, focusing on particular aspects of the course material. *A quiz* is even more limited and usually is administered in fifteen minutes or less.

Among these evaluation methods, tests are the most important ones for the evaluation of adaptive Web-based learning systems, mainly due to two main reasons [71]:

- 1) Testing offers a feedback on the correctness of the answers, helping to optimise the learning process.
- 2) Testing results are the most reliable source of evidence that a user has learned a concept.

In general tests, quizzes or exam -based evaluation may consist of five different types of test items:

- *Yes-No (True-False) test items*: users have to answer to questions by selecting either "Yes" or "No" answer only.

- *Forced-Choice test items*: users have to answer a question by selecting only one of the alternative answers that are provided.
- *Multi-Choice test items*: users have to answer a question by selecting all correct answers from the list of possible provided answers.
- *Essay (Free-Form) /Short Answer test items*: users can type an answer to the question freely into the form. Short answers are usually only one to three paragraphs long.
- *Gap-Filling (Completion) test items*: users have to type in characters or numbers to complete a word or a sentence.

Each type of test items has its relative strengths and weaknesses and they are discussed next.

Yes-No (True-False) tests: They measure the ability of the subject to identify the correctness of statements of facts, definitions of terms, statements of principles, etc. These tests can sample many more bits of information in a given time period than any other type of test format. Research does indicate that Yes-No testing is sufficiently reliable and valid for periodic in-classroom testing. However, because of random guessing with 50-50 chances, these tests can be less reliable than other tests, unless the number of questions asked is high.

Forced-Choice tests and Multi-Choice tests: These types of tests consist of a sentence that describes a problem and a series of possible answers or alternatives (usually 3 to 5). They can address many learning targets, can be used to assess both simple knowledge and complex concepts and can be answered quickly. They are easy to score, can be considered objective because all potential item responses are identified, but lack the ability to address learner produced responses. Multi-choice tests have a higher degree of difficulty than the forced-choice tests and both are more difficult than Yes/No tests.

Essay/Short Answers tests: These tests are most advantageous when assessing complex learning outcomes and higher-level thinking skills. They are relatively easy to construct, do not permit guessing and cannot be answered by simply recognizing the correct response. Among the limitations of Essay Tests are that they are difficult to score, their scores are less reliable than the ones from the previous types of tests, the score is often influenced by the readers overall impression of the student and these tests provide a very limited sample of the content in the typical unit of study.

Gap-Filling (Completion) tests: They make scoring faster and less subjective. They are used to measure the recall of memorized information. Completion test items preclude the kind of guessing that is possible on limited-choice items since they require recall and a definite response rather than simple recognition of the correct answer. They are more difficult to score than forced-choice items and scoring often must be done by the test writer since more than one answer may be considered correct. On the whole, completion items have little advantage over other item types unless the need for specific recall is essential.

Every type of test has a general value for difficulty and relevance for the tested concept. These test items can be used to wrap up a course, lesson, section, or subsection and very significantly, they help in the evaluation of the learning outcome

6.2.2.3 Learning Performance Evaluation

Learning performance refers to how fast a study task (e.g. learning task, searching for a piece of information or memorising information displayed on the computer screen for example) takes place. The most used metric used for measuring learning performance provided by an Adaptive Hypermedia System for Education is *study session time* [173, 71, 189, 190]. The completion time for a study session is measured from the start of the session, when the subject logins into the system and starts to study, until the subject starts answering the questions from the evaluation test. Other metrics worth mentioning are: *number of navigation steps* performed during a study session [172, 173, 174, 71], *number of pages revisited* [172], *average time spent per page* for studying the information, *average access time*, etc.

6.2.2.4 Assessment of Evaluation Results

The assessment of the usability evaluation is performed in terms of overall usability of the web-based course system and usability of each category of questions that reflects different characteristics of the system such as efficient to use, easy to remember, pleasant to use, easy to learn, few errors, etc. Mean values and standard deviations of the results are computed.

The assessment of the learner achievement is performed in terms of final scores from the quizzes, tests or exams, achieved by the subjects when one or another version of the adaptive educational system is used. The results are analysed by computing mean values and standard deviations of the final scores.

Learning performance is analysed through some measured performance metrics such as study session time, number of accesses to a page, etc.

For scientific credibility, different statistical methods for data analysis are used for the comparison of different versions of adaptive systems. The most used statistical analysis methods in the evaluation of educational systems are: average, standard deviation, T-Test, ANOVA, F-Test, Q-Test, etc.

Correlation analysis (e.g. Pearson correlation and Spearman correlation) could also be performed between the usability evaluation results and student marks from the learner achievement assessment. The goal is to investigate the existence of any relationship between the two sets of data.

More details on these statistical methods for data analysis are presented in Appendix A

6.3 Evaluation of the QoE Layer: QoEAHA System

For illustration and testing purposes the QoE layer proposed and described in Chapter 4 has been deployed on the open-source AHA! system, forming QoEAHA.

The QoE layer was implemented in Java, the programming language used to implement the AHA! system. The Performance Monitor was implemented in C+ in order to allow for faster measurement of the network related parameters in real time.

The AHA! system [191] was developed at the Eindhoven University of Technology, in the Database and Hypermedia group. The system was first deployed and used in educational area as an adaptive hypermedia courseware application that supports the "Hypermedia Structures and Systems" course (see TU/e course [189, 190]). AHA! has a number of advantages that allowed us to use it for the demonstration of the benefits brought by the QoE layer. Different tests were performed using the AHA! tutorial.

Among the AHA! system advantages are the following:

- The AHA! system has been extensively tested and is accepted as a good Adaptive Hypermedia System by the research community.
- AHA! is a simple general-purpose hypermedia adaptive system,
- The AHA! system architecture respects the AHAM architecture [25].
- AHA! is open source and therefore allows for extensibility.

- The distributed open-source AHA! version 2.0 includes an adaptive tutorial as example for the adaptive features of the AHA! system. This tutorial is a course about the AHA! system and how it can be used to develop one's own adaptive courses.

6.3.1 AHA! System

The AHA! system [25] (Adaptive Hypermedia Architecture) was designed and implemented at the Eindhoven University of Technology, as part of the AHA! project sponsored by the NLnet Foundation. The main goal of the AHA project was to create an environment for making websites that adapt themselves to the user's characteristics [192]. The development of the system has started in 1996 with an online text-based courseware application on the subject of hypermedia that has provided both adaptive content and linking. Since then, the system was improved and extended leading to the current AHA! version 3.0. AHA! was studied and tested by several research groups from different countries (e.g. Eindhoven TU/e University - Netherlands, University of Southampton- UK, University of Pittsburgh - USA, University of Cordoba - Spain, Universidade Federal do Rio Grande do Sul - Brasil).

Currently, AHA! 3.0 is an open source Java-servlet-based software environment, general-purpose adaptive hypermedia system through which different adaptive applications can be created. The system runs on both Linux (or Unix) and Microsoft Windows operating systems.

6.3.1.1 AHA! Architecture

The AHA! system's architecture is based on the AHAM reference model and consists of four components that work closely together: Domain Model (DM), User Model (UM), Adaptation Model (AM) and Adaptation Engine (AE). AHA! deviates from this general model in only one detail by not separating the Domain Model and Adaptation Model that are stored together [193]. In consequence, the author of an AHA!-based application has to define the concepts and the attributes associated to each concept (DM) along with a list of requirements and generate rules (AM).

Requirements specify under which conditions the user is ready to access a concept while generate rules (condition action rules) indicate how the user model is updated [194]. All the information is stored by the AHA! system in a XML file or in a mySQL database.

Concepts are used by the author of an AHA application to represent topics of the application domain in the DM. Each concept has associated a number of attributes. The AHA! system provides a list of pre-defined attributes such as “access”, “visited”, etc. that might be associated with a concept. The author can also define new attributes extending the existing set.

Some of the concepts are linked to pages or fragments of information represented through XHTML files augmented with AHA!-specific tags (e.g. “if” and “a” tags). These tags are used for the generation of optional headers and footers and for the conditional inclusion of fragments. The AHA! engine looks at all AHA tags in order to perform the adaptation [195]. It then translates the AHA/XHTML content into plain HTML before sending the page to the client browser. Currently only the XHTML format is supported but the AHA! system can be easily extended to serve other document formats combined with AHA! tags. SMIL is one of the formats that are currently considered.

For the description of the UM, AHA! uses the overlay-based technique [25]. This technique implies that for every concept and attribute from the Domain Model of the application an associated concept and attribute is stored in the User Model. UM also stores user-related aspects that are not part of the subject domain of the application and reflect user-related characteristics. This information is represented through the attributes of a concept called “personal”. Just like for the other concepts, the author can add new attributes to the “personal” concept in order to provide adaptation based on domain-independent user characteristics [196].

The Adaptation Engine consists of a set of Java servlets that are activated when the Web server receives requests for a Web page from a browser. For each user request an AHA! servlet processes the requested page through the interpretation the embedded AHA tags and sends an adapted page to the user. The Web page consists of modified links and fragments conditionally included or excluded based on the interpretation of the AHA tags. AHA can include fragments that are part of a page as well as stored as separate objects [196].

6.3.1.2 AHA! Functionality

AHA! is a Java server-side application that allows the Website developers to add automatic personalization features to their web content. The system intercepts HTTP requests from the user’s browser, retrieves the requested web content from the external

resource, applies adaptation on the content by including or excluding fragments and by modifying links, and forwards the response to the end-user's browser.

There are three categories of users with different access rights that can interact with the AHA! system [195]:

- *End-Users* that access the web site pages. At every interaction of the user with the system, the User Model is updated based on page access.
- *Authors* that are in charge with the design of AHA! applications. They create the structure of the Domain Model, define concepts and associate them with fragments/pages, and specify the Adaptation Model rules. The authors can design the application making use of the authoring tools provided by the AHA! system.
- *Manager* (system administrator) is in charge with the initial configuration of the AHA system and with the registration of the authors that have the rights to develop AHA applications.

6.3.1.3 Authoring Interface

In order to develop an adaptive Web-based application that makes use of an AHS, an overall conceptual structure of the domain application has to be designed and a set of adaptation rules has to be indicated. Currently most of the developed AH systems do not have associated any authoring tool that would allow easier representation of the DM or a simple specification of the AM rules.

In consequence, the AHA! research team put a lot of effort into the development of authoring support in order to ensure the usability of the AHA! system for adaptive Web based applications. The AHA! system provides two Java Applet-based authoring tools that allows for an easier definition of the Domain/Adaptation model at low level (**Concept Editor Tool**) and high level (**Graph Author Tool**). These authoring tools allow an author to define the Domain Model for an adaptive application along with the Adaptation Model associated with it. The tools are server side Java-based applications that manipulate the XML files that describe the conceptual structure of applications.

However, currently AHA! does not provide any authoring tool of the web content (e.g. pages or fragments) and the material has to be developed using an external

XML/HTML editor. On the other hand the author of an AHA! application does not necessarily need to be also the author of the web content. He/she can re-use existing fragments or pages created by other people.

A brief description of the functionality of the AHA! authoring tools is presented next. More details about the tools and examples on how the tools can be used are provided in [195, 196] and on the AHA! project web site [191].

6.3.1.3.1 Concept Editor Tool

The Concept Editor is a low-level Java Applet based graphical authoring tool used to define both the concepts and adaptation rules. It is most suitable for applications that require different and many adaptation rules. If the application uses only a few types of adaptation rules commonly used by most of AHS then the Graph Author Tool may be used [195]. By using the Concept Editor Tool the author has control on writing all the individual adaptation rules.

For each newly created concept the Concept Editor Tool offers a list of predefined attributes and adaptation rules. It also offers the possibility to add new attributes and rules to each concept and also to remove the old ones.

Unfortunately the usage of Concept Editor Tool may require a lot of repetitive work from the author when defining rules associated with the related concepts. For example, adaptation rules that show the knowledge propagation from sections to chapter are only repetitions of the same basic adaptation rule applied for each concept representing a section.

6.3.1.3.2 Graph Author Tool

Graph Author is a high-level Java applet-based graphical tool used to define the concepts and concept relationships. Based on a drag-and-drop action from a hierarchical list of defined concepts concept relationships are created. The tool offers a list of pre-defined relations that can be associated between two concepts such as: “inhibitor”, “prerequisite”, “knowledge_propagation”, “knowledge_update”, etc. Other types of relationships may also be incorporated in the tool and used by the author [195, 196].

With each relationship defined between two concepts the system automatically performs a translation of the concept relationship to AHA adaptation rules (high-level to low-level representation). Therefore, when a new relationship is incorporated in the Graph Author tool the translation into adaptation rule has also to be provided. The tool knows how

to combine different relationships between the same concepts into correct AHA! adaptation rules that express the meaning of the combined given rules.

6.3.2 QoEAHA Evaluation Strategy

Although many AEHS have been proposed and developed, there is a significant lack of evaluation strategies and comprehensive empirical studies to measure the usefulness and effectiveness of adaptation within the systems and between the systems. There is also much debate on how adaptive hypermedia applications should be evaluated since there is no standard or agreed evaluation framework for measuring the value and the effectiveness of adaptation yielded by adaptive systems. In order to determine the evaluation strategies for QoEAHA system a survey of the research in the adaptive education area with emphasis on Web-base AEHS has been undertaken and the results were presented in subsection 6.2.

From the two main evaluation directions that take into account the system as a whole and the division the evaluation process into layers respectively, the second approach has been selected in this research. Layer-based evaluation strategy evaluates both the adaptation phases and the design stages. Thus during the evaluation process one can determine with greater accuracy the location of possible malfunctions of the adaptive system components that may not perform the best.

This choice was determined by the following facts:

- Since the AHA! was used, layer-based evaluation technique helps to narrow the QoS AHS evaluation target from the whole system that includes both the original AHA system and QoE extension to the evaluation of the QoE layer. The original AHA system was already extensively tested and accepted as a good adaptive hypermedia system by the research community [42, 191, 192, 193, 194, 195, 195, 196, 197, 198, 199, 200]. Thus, the evaluation of the QoEAHA will assess only the benefits brought by the QoE Layer extension and not the User Model based adaptation mechanism performed by the AHA!.
- Since AHA! is a robust and well-tested system, QoS AHS evaluation is reduced to Adaptation Making Evaluation phase, which corresponds to the second evaluation layer as defined in [175]. This phase assumes that the information about the users is correctly acquired making the adaptation possible and aims at testing the performance of the chosen adaptation techniques that use this information only. The correct acquisition of the values of the measured

performance metrics performed by the PM module was also verified during the simulation tests. Therefore, this phase tests if the enhanced adaptive system provides appropriate adaptations to the users' knowledge and their current operational environment.

- Since the QoS AHS adaptation includes both the adaptation performed by the original AHA! system - User Model-based content and link adaptation and the end-user perceived performance-based adaptation suggested by the QoE layer ,the Adaptation Making Evaluation phase analyses their combined effect.

The role of the QoE layer is to enhance AHS with the end-user perceived performance-based adaptation that improves the overall user experience with the system. Therefore, the evaluation strategy involves a comparison between the original AHA! system and the QoS AHS and assesses the performance improvements, learning capabilities and end-user perception.

6.3.2.1 Objectives of the Evaluation Experiment

Although the AHA! system is a general-purpose Adaptive Hypermedia System, it was deployed and intensively used in the educational area. Therefore the evaluation of the QoEAHA system was performed in the same domain area. This decision was supported by the fact that AHA! system provides an adaptive tutorial as testing material. The content of this tutorial was used as educational material for the students in the experimental tests performed during our research. As the material was already designed a prior to the proposal of the QoE layer, it provides independent testing material for both objective and subjective evaluations performed by us.

The objectives of the experiment were the following:

- to investigate the impact of QoE extension on student performance
- to measure the usability and effectiveness of the QoE-aware AHA system in comparison to the original AHA! system
- to assess any improvement brought by the QoEAHA in terms of users' QoE

Usability evaluation was performed through an on-line usability questionnaire filled-out by the students after they completed a required task. The task that the students had to

perform during the experiment involved the study of one chapter (“AHA! installation”) from the AHA! tutorial and to take an on-line evaluation test for assessing their learning outcome.

Next, the impact of the QoE layer on student performance was investigated. This was measured by comparing the performance of the students that used the two alternative systems. Students' performance was assessed in term of the two most important metrics: *learner achievement* and *learning performance*. Other assessment criteria such as number of revisited web pages and study time on the modified pages have also been investigated. Details on the experimental results are presented in the next sections.

The analysis of students Quality of Experience when the two version of the AHA system were used was also performed. On-line questionnaires that assess user opinion in relation to performance issues and user's satisfaction on the perceived QoS were used at the end of the study session. An analysis of the student answers when using both versions of the AHA! system was performed in order to assess the benefits brought by the proposed QoEAHA.

The results of the evaluation tests have been assessed and analysed through different statistical analysis methods (presented in section 6.2.2.4) in order to check the significance of the results.

Another aspect worth examining was to establish whether there is any correlation between:

- the performance of individual students and their judgment over system usability and delivery performance.
- the length of the study period and the students' final marks

The correlation analysis was performed by calculating correlation coefficients, as presented in Appendix A. Conclusions were drawn and presented in the following sections.

6.3.2.2 Experimental Setup

The experiments took place in the Performance Engineering Laboratory, School of Electronic Engineering, Dublin City University. Two sets of task-based scenarios were developed and carried out in laboratory settings. The scenarios were created in order to provide a real usage context for participants, as they would interact with the system in real study conditions.

The subjects involved in the tests were randomly divided into two groups. One group used the original AHA! system, whereas the second one used QoEAHA. The subjects were not aware of what system version they were using during the experiment. No time limitation was imposed on the execution of the required tasks. None of the students had previously used any of the two versions of the AHA! system and none of them has accessed the test material prior to take the tests. Therefore no previous practice with the environments was assumed for any of them. The material on which the students learned consisted of the original adaptive tutorial delivered with the AHA! system version 2.0. The students were required to perform a learning session on the Chapter One from the AHA! tutorial.

For both group of subjects same network conditions were emulated between the subjects' computers and the Adaptive Hypermedia System. These conditions determined performance-related adaptations when QoEAHA version was used. The laboratory-network setup used for testing involved four PC Fujitsu Siemens desktops with single Pentium III (800MHz) processors and 128 MB memory each, a Web server IBM Netfinity 6600 with dual Pentium III (800 MHz) processors and 1 GB memory and one Fujitsu Siemens desktop computer with Pentium III (800 MHz) processor and 512 MB RAM that acts as a router and has a NISTNET network emulator installed on it. NISTNET that allows for the emulation of various network conditions characterized by certain bandwidth, delay, loss rate and loss pattern was used to create low bit rate modem-like operational environments with 56 kbps and 28 kbps respectively. Figure 6-1 graphically presents the experimental setup. These setup conditions offer similar connectivity to that experienced by residential users and are the same as the ones used in the simulation tests presented in Chapter 5.

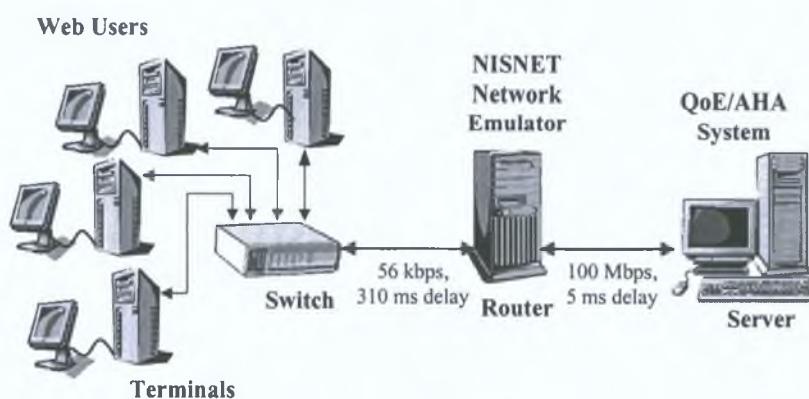


Figure 6-1 Laboratory Network Configuration for the Perceptual Testing

6.3.2.2.1 Scenario 1: Interactive Study Session

The testing scenario covered an interactive study session of one chapter from the adaptive AHA! tutorial over a 56 kbps connection speed. It also involved asking the subjects to take an on-line evaluation test in order to assess their knowledge level immediately after they completed the study task. This first experimental test involved forty-two postgraduate students from the Faculty of Engineering and Computing, Dublin City University, Ireland as subjects.

At the start of the study session the subjects were given a short explanation concerning the usage of the system and their required duties. They were asked the following:

- *complete an on-line Pre-Test evaluation* that consisted of a questionnaire with six questions related to the learning topic. The test was used to determine subject's prior knowledge about the studied domain.
- *log onto the system* and proceed to browse and *study the material*. Back and forward actions through the studied material were permitted.
- *complete a Post-Test* at the end of the study period. The Post-Test consisted of a questionnaire with fifteen questions that tested recollection of facts, terms and concepts from the supplied material. During the evaluation phase the students were not allowed to return to the studied material.
- *answer a usability questionnaire* that consisted of ten questions categorized into navigation, accessibility, presentation, perceived performance and subjective feedback.

In order to fully assess the subjects learning outcome, both Pre-Test and Post-Test were such devised that consisted of a combination of four different types of test-items most commonly used in the educational area: "Yes-No", "Forced-Choice", "Multi-Choice" and "Gap-Filling" test items. Details about these types of test-items were presented in section 6.2.2.2.

For practical reasons it is recommended that the total number of test-items in any evaluation test should not exceed 25 test items with a maximum of 5 test items for each concept involved [186]. Therefore, the Post-Test evaluation used during this experiment consisted of 15 questions as follows: 5 – "Yes/No" items, 6 – "Forced Choice" items, 3 –

“Multi Choice” items and 1 – “Gap Filling” item (see Appendix B). For time-related reasons a smaller number of questions were used for the Pre-Test evaluation representing a total of 6 as follows: 3-“Yes/No”, 2-“Forced Choice” and 1-“Multi-Choice” (see Appendix C).

Each type of test-item has different degree of difficulty and therefore a different corresponding weight in the final score has been assigned for a correct answer as follows:

- for each correct answer to a “Yes/No” question one point was given
- for each correct answer to “Forced-Choice” question two points were given
- for each correct answer to “Multi-Choice” question three points were given
- for each correct answer to “Gap-Filling” question four points were given
- for the incorrect answers no points were given

In consequence, the maximum score for the Pre-Test was 10 points, while the maximum score for the Post-Test was 30 points. The final scores of both the tests were normalized in the 0-10 range.

6.3.2.2 Scenario 2: Visual Quality Assessment

The second scenario covered a visual quality assessment for the worse network connection case - 28kbps. Since the performance improvement brought by the QoE layer involves reduction in the quantity of data sent to the users by decreasing the quality of the embedded images, user surveys were used to ascertain the impact of these modifications on user experience with the system. The visual user subjective-based quality assessment was performed on a five-point quality scale: 1 – “bad”, 2 – “poor”, 3 – “fair”, 4 – “good”, 5 – “excellent”.

This scenario also involved an objective visual quality assessment that makes use of an evaluation criteria very often used in the educational area: time taken to search for a term described in a Web page.

The subjects involved in these tests were comprised of twenty postgraduate students from the Faculty of Engineering and Computing, Dublin City University, Ireland. They were randomly divided into two groups. Each group made use of one of the two versions of the AHA!: the original and the QoE-aware AHA system. In each case the same network delivery conditions were emulated. These conditions were such set that would test the subjects’ visual

quality in the worst-case scenario that corresponds to the lowest connection case (bandwidth 28 kbps).

The subjects were requested to look up for two different terms and to answer two questions related to them. The terms were introduced in two of web pages of the AHA! tutorial, more precisely, in the embedded images. As these web pages included the largest number of embedded images, the highest degradation in the quality of images in the tested delivery conditions, is caused when the QoEAHA is used. The goal of these tests was to assess whether the resulted quality of images is good enough for the subjects to be able to perform the required task. An overall assessment on the quality of all images that were viewed was performed.

6.3.3 QoEAHA Evaluation Tests

6.3.3.1 Learner Achievement

Learner achievement is defined as the degree of knowledge accumulation by a person after studying a certain material. It continues to be a widely used barometer for determining the utility and value of distance learning technologies. Learner achievement is analysed in the form of *course grades, pre/post-test scores, or standardized test scores*.

During the first scenario learner achievement was assessed by comparing Pre-Test and Post-Test scores achieved by the subjects using the QoEAHA and AHA! systems respectively. The results of the Pre-Test are shown in Table 6-1 whereas the results of the Post-Test are presented in Table 6-2.

Table 6-1 Pre-Test Results

	MEAN SCORE	MIN SCORE	MAX SCORE	SDEV
AHA!	0.35	0.0	2.0	0.55
QoEAHA	0.30	0.0	2.0	0.53

Table 6-2 Post-Test Results

	MEAN SCORE	MIN SCORE	MAX SCORE	SDEV
AHA!	6.70	4.30	9.30	1.401
QoEAHA	7.05	4.60	9.00	1.395

A test for homogeneity (T-Test) was performed first in order to determine if both groups of students had the same prior knowledge on studied domain. If the T-Test analysis on the Pre-Test scores shows that there is no significant difference between the two groups then only the Post-Test scores are used in the analysis of the learner achievement. Otherwise both Pre - Test and Post-Test sets of results have to be taken into account.

The two-sample T-Test analysis, with equal variance assumed, performed on the Pre-Test scores has proved with a 99 % confidence level that statistically both groups of subjects had the same prior knowledge about the studied subject (significance level $\alpha = 0.01$, $t = 0.21$, $t\text{-critical} = 2.42$, $p(t) = 0.41$). This result means that the learner achievement can be assessed by processing only the Post-Test scores.

Following the Post-Test results evaluation, the mean score of the subjects that used QoEAHA was 7.05 and the mean grade of those that used AHA! was 6.70. A two-sample T-Test analysis on these mean values does not indicate a significant difference in the final marks of the two groups of users. ($\alpha = 0.05$, $t = -0.79$, $t\text{-critical} = 1.68$, $p(t) = 0.21$). Therefore the results indicate no significant difference in the learning outcome between the users of the QoEAHA and AHA systems.

Since answers for three of the questions from the Post-Test questionnaire (Q9, Q12, Q15 -see Appendix A) have required the subjects to study the images embedded in the Web pages, an analysis of the students' outcome on these questions was performed. After the scores related to these three questions were normalized in the 0 to 10 range, the mean value of the students' scores was 6.3 for the QoEAHA group and 6.4 for AHA! group (Table 6-3). A two-sample T-Test analysis, with equal variance assumed, performed on the two sets of results indicates with a 99 % confident level that there is no significant difference in the students' learning achievement ($t = -0.08$, $t\text{-critical} = 2.71$, $p(t) = 0.93$, confidence level $\alpha = 0.01$). This result is very important as an adaptive degradation in the image quality (up to 34 % in size) was applied by the QoEAHA.

Table 6-3 Post-Test Results for Questions with Answers Presented in the Embedded Images

	MEAN SCORE	MIN SCORE	MAX SCORE	SDEV
AHA!	6.40	2.0	10.0	3.25
QoEAHA	6.30	2.0	10.0	3.15

In summary, these test results indicate that the QoEAHA system did not affect the learning outcome, offering similar learning capabilities as the classic AHA! system.

6.3.3.2 Learning Performance

Learning performance refers to how fast a study task takes place. A study task may involve a learning process, searching for a piece of information described in the educational material or memorising information displayed on the computer screen. The first two types of tasks often executed by a student during a study session were considered for experimental testing.

The performed tests looked to compare the Learning performance of the students when the two versions of the AHA system were used. Different metrics presented in section 6.2.2.3 were measured and assessed. *Study Session Time* is the most used metric for assessing learning performance in the educational area.

6.3.3.2.1 Learning Task

The first scenario described in section 6.3.2.2 that requires the users to perform a learning task was used for assessing the students' performance. The completion time for a learning session is measured from the start of the session, when the subject logins into the system and starts to study, until the student starts answering the post-test questionnaire.

The distribution of the study time taken by the students in order to accumulate the information provided during the first scenario using the AHA! and QoEAHA systems is presented in Figure 6-2. One can notice that on average, students that made use of the QoEAHA system (Average Study Time = 17.77 min) have performed better than the ones that used the AHA! (Avg. Study Time = 21.23 min) (Table 6-4). The very large majority of the students that used QoEAHA (71.43 %) performed the task in up to 20 min. with a large number of students (42.87 %) requiring between 15 min and 20 min of study time. In comparison, when the AHA! system was used, only 42.85 % of the students succeeded to

finish the learning task in 20 minutes. The majority of them (71.42 %) required up to 25 min. with the largest number of students (28.57 %) in the interval 20-25 min. (Table 6-5).

Table 6-4 Study Time for the Learning Taks when AHA! and QoEAHA were Used

	MEAN STUDY TIME (MINS)	MIN STUDY TIME (MINS)	MAX STUDY TIME (MINS)	SDEV
AHA!	21.23	12.95	31.84	5.90
QoEAHA	17.77	9.37	30.38	5.44

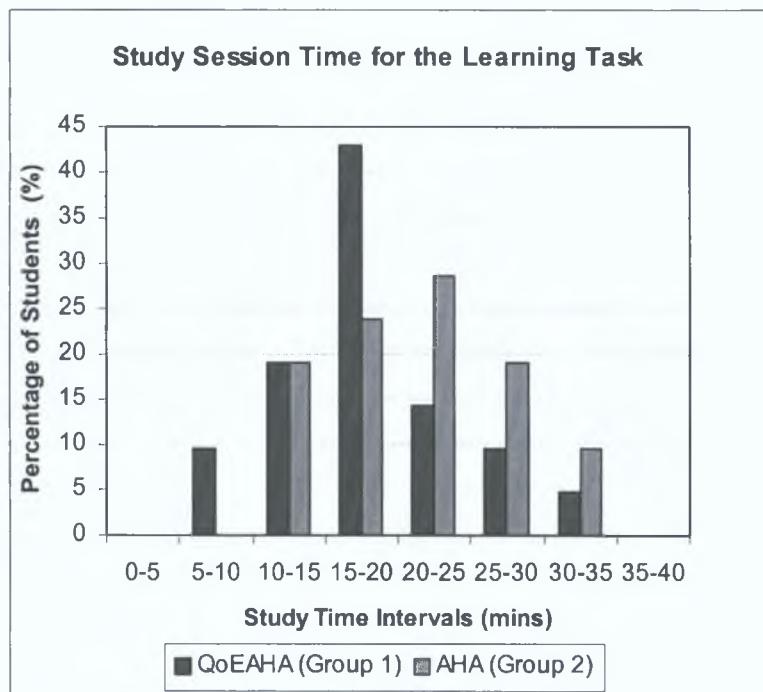


Figure 6-2 Distribution of the Study Time Performed by the Students Involved in a Learning Task

In Figure 6-2 one can also notice that 9.5 % of the students from Group 1 (using QoEAHA) succeeded to learn in less the 10 minutes while none of the students from Group 2 (using AHA!) had this performance.

Table 6-5 Percentage of Students that Have Succeeded to Learn for Different Study Period Times.

STUDY TIME INTERVAL (MINS)	QOECHA NUMBER OF STUDENTS (%)	AHA NUMBER OF STUDENTS (%)
0-10	9.5	0
0-20	71.43	42.85
0-25	85.71	71.42
20-25	14.28	28.57
25-35	14.29	28.57

Apart of the *Study Session Time* other metrics were used in order to investigate the students' learning performance such as *Average Time spent per page* and *Number of Accesses to a page performed by a person*. The last one can provide an indication on the quality of learning. Any re-visit to a page indicates that the student was not able to recall the information provided in the page and thus the learning process was of poor quality.

Among the web pages studied by the students during the learning task, two of them involved a higher number of embedded images and a larger quantity of data being transmitted. In consequence, the students perceived a longer waiting period when the AHA! system delivered the pages. QOECHA decreased the access time perceived by the students but also performed some degradation in the quality of the content. Therefore the study time on those pages was analysed when the two systems were used with the first scenario. *Study Time for a page* was measured from the moment when the system has received a request for the page until a request for a new page was sent.

Table 6-6 Data Analysis for the Study Time Spent on Page 1 and Page 2

ADAPTIVE SYSTEM	STUDY TIME ON PAGE 1 (MINS)					STUDY TIME ON PAGE 2 (MINS)				
	Avg	Min	Max	SDev	σ^2	Avg	Min	Max	SDev	σ^2
QoEAHA (Group 1)	4.28	2.09	7.91	1.48	2.2	4.23	2.15	7.7	1.44	2.09
AHA (Group 2)	5.33	3.69	8.47	1.28	1.65	5.40	3.76	8.23	1.28	1.64

The results show that on average the students from Group 1 (using QoEAHA) spent less time on both Page 1 and Page 2 for studying the information in these pages than the ones from Group 2 (using AHA!). This observation was confirmed by a data analysis procedure. The Null hypothesis (H_0) stated that there is no difference between the means values (avg.) of the two groups and thus the results of the two groups do not differ significantly. Alternate hypothesis (H_A) stated that there is a significant difference between the two groups' means.

By performing a two-sample T-Test, assuming unequal variances (σ^2), for each page, as described in Appendix A, the Null hypothesis was rejected and the alternative one accepted (Table 6-7).

Table 6-7 T-Test Analysis on Study Time results for Page 1 and Page 2

PERFORMED ON STUDY TIMES RESULTS FOR	T-TEST ANALYSIS			
	CONFIDENCE LEVEL (α)	T-CRITICAL	T-STAT	P (T)
Page 1	0.05	2.02	2.45	0.01
Page 2	0.05	2.02	2.77	0.008

This improvement in study time could be due to an eventual students' incapacity of visualising and processing the information that would make them decide to go forward

through the educational material faster. The fact that this is not the case is demonstrated by the students' marks received on three questions from the Post-Test evaluation with answers provided in these two pages. Both groups of students that have accessed the original pages through AHA! and modified ones via QoEAHA, have received similar marks. More details on these test results were presented in section 6.3.3.1.

Table 6-8 Number of Accesses to a Page Performed by a Person

ADAPTIVE SYSTEM	NO. OF ACCESSES PER PERSON PER PAGE 1					NO. OF ACCESSES PER PERSON PER PAGE 2				
	Avg	Min	Max	SDev	σ^2	Avg	Min	Max	SDev	σ^2
	QoEAHA (Group 1)	1.43	1.0	3.0	0.60	0.36	1.38	1.0	3.0	0.59
AHA (Group 2)	1.76	1.0	4.0	0.83	0.69	1.71	1.0	4.0	0.90	0.81

Table 6-9 T-Test Analysis on Number of Accesses Per Person to Page 1 and Page 2

PERFORMED ON NUMBER OF ACCESSES PER PAGE PER PERSON	T-TEST ANALYSIS			
	CONFIDENCE LEVEL (α)	T-CRITICAL	T-STAT	P (T)
Page 1	0.08	1.43	1.49	0.07
Page 2	0.08	1.43	1.47	0.07

Number of accesses to a page performed by a person was also measured and analysed for the same two pages. The average value of this parameter for Page 1 was 1.43 when the QoEAHA system was used and 1.76 for the AHA! system. Similar values were obtained for Page 2: 1.38 and respectively 1.70 (Table 6-8). An unpaired two-tailed T-Test analysis, with unequal variance assumed, has statistically confirmed with at least 92 %

confidence that there is a significant difference in the number of visits performed by a student to Page 1 and Page 2 when the two versions of the AHA! were used (Table 6-9).

Next, how the effect of the version of the AHA! system used by the students has affected the Number of Accesses to a page was investigated by analysing the variability of the test samples. The results from Table 6-8 show that the standard deviation and variance of Group 1 results are lower than the ones that correspond to Group 2 results for both pages. A F-Test analysis was performed to determine if the variance between the two groups is statistically significant. The results presented in Table 6-10 confirm that Group 1 and Group 2 results do not have the same variance and the difference between the two groups' variances is statistically significant.

Table 6-10 F-Test Analysis on the variance of Group 1 and Group 2 Number of Accesses Samples.

PERFORMED ON NUMBER OF ACCESES PER PAGE PER PERSON	F-TEST ANALYSIS			
	CONFIDENCE LEVEL (α)	F-CRITICAL	F-STAT	P (F)
Page 1	0.08	1.89	1.93	0.07
Page 2	0.05	2.12	2.34	0.03

Therefore, looking at these results one can conclude that Group 2 results have a higher dispersion than the ones obtained by Group 1. This indicates that a larger number of students (an average of 55 %) that used the AHA! system (Group 2) required more than one access to Page 1 and Page 2 for learning. At the same time, a large majority of students (an average of 65 %) that used the QoEAHA (Group 1) performed only one access to the same pages (Figure 6-3 and Figure 6-4) showing that in general these students have succeeded to focus better on the studied material.

Although both groups of students achieved the same learning outcome at the end of the experiment (6.3.3.1), Group 1 that recorded a lower number of re-visits to the web pages has obtained its results in a shorter study period (see section 6.3.3.2).

In conclusion the students that have used the QoEAHA system for studying the educational material have succeeded to learn in a shorter period of time than those that have used the AHA! system. Results show that an improvement of 16.27 % in the study time for the whole learning session was obtained when the QoE aware version was used. This is due to the fact that the material was delivered faster to the students and the students were constantly focused on their task. Long periods of waiting time for getting access to the material annoys the people and disturbs their concentration on the learning task.

This observation is confirmed when assessing the time spent for studying two of the most complex pages. On average, a shorter time was allocated for studying the material by the students that made use of the QoEAHA system than those that have used the AHA! system.

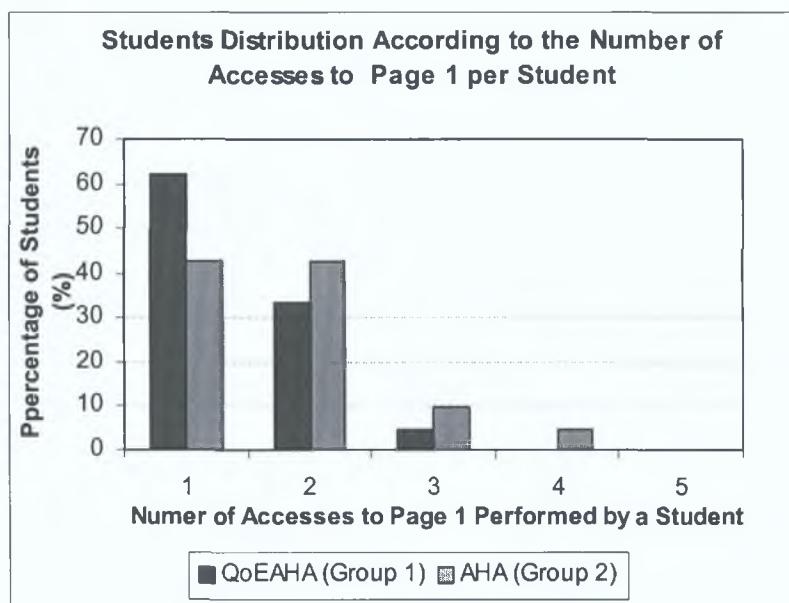


Figure 6-3 Students Distribution Based on Number of Visits to Page 1

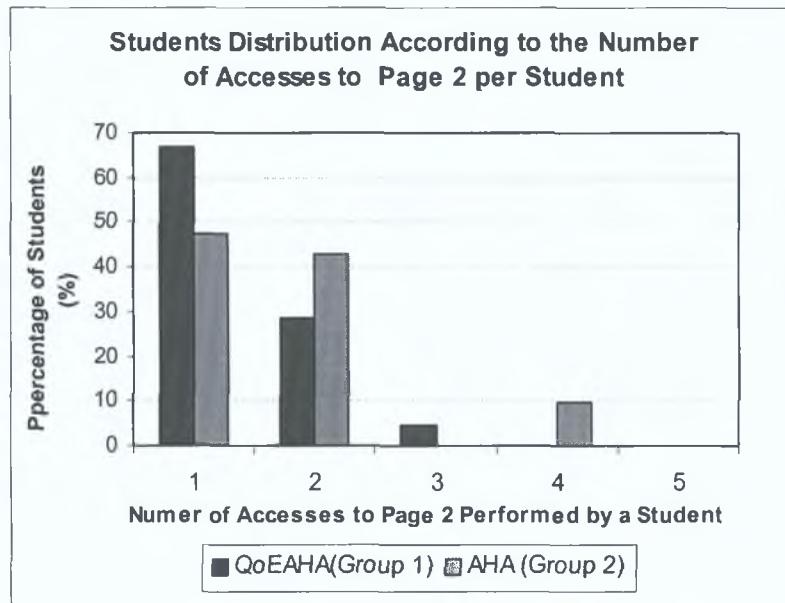


Figure 6-4 Students Distribution Based on Number of Visits to Page 2

6.3.3.2.2 Search for a Term Task

The second scenario described in section 6.3.2.2 was used for assessing the students' learning performance when executing a "search for a term" task. The task requires the users to look up two terms and answer two questions related to them by browsing through two complex web pages selected from the AHA! tutorial. The two terms are introduced in the embedded images of the web pages. Completion time for the searching task is measured from the moment when the subject sends a request for the page until the student starts answering the question.

The goal of this test was to show that although QoEAHA has reduced the quality of the images the learning performance was not significantly affected.

The test results, presented in Figure 6-5 show that the time taken to search for a term is similar for the two versions of the AHA! system. It is also worth mentioning that all the subjects have successfully completed the task and answered correctly the questions.

Statistical analysis was conducted in order to check the student performance variation of the two groups. An unpaired two-tailed T-test was applied on the test results. This analysis suggested that there is no significant difference in learning performance between the users of the two systems ($t\text{-statistic}=-0.12$, $t\text{-critical}=2.08$, confidence level =0.05, $p(t)=0.89$).

In consequence the controlled degradation in the quality of the images that has improved the delivery performance has not affected subjects learning performance.

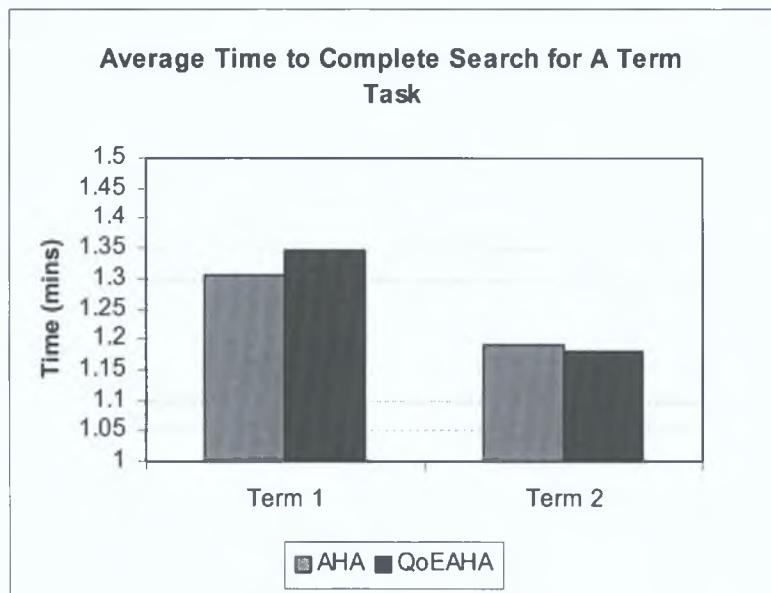


Figure 6-5 Search for a Term Average Time

6.3.3.3 Usability Assessment

The main goal of the usability assessment evaluation strategy is to measure the usability and effectiveness of the QoE-aware AHA system in comparison to the original AHA! system. The methodology of study involved the usage of the online questionnaire technique. This is one of the most widely used techniques in the education area. More details about this and other evaluation techniques were presented in section 6.2.2.1.

This evaluation technique was selected mainly due to the followings issues:

- it provides a quantitative measure of usability and allows for the comparison of two system versions
- it is widely accepted and used in the educational area
- it offers a concise test of usability and it is suitable for wide range of end-users, especially students
- this techniques gets directly the users' viewpoint
- it does not require the presence of an expert evaluator

The first scenario described in section 6.3.2.2 that involved two groups of students performing a learning task by making use of QoEAHA and AHA! respectively was used. At the end of the study session both groups of subjects were asked to complete an online usability evaluation questionnaire consisting of ten questions related to key usability aspects and performance issues. The answers were expected to be given on the Likert five-point scale: 1-poor, 2-fair, 3-average, 4-good, 5-excellent. The questions were created using the widely used guidelines suggested by Preece [186] for evaluating Web sites. They were categorized into: navigation, presentation, subjective feedback, accessibility and user perceived performance.

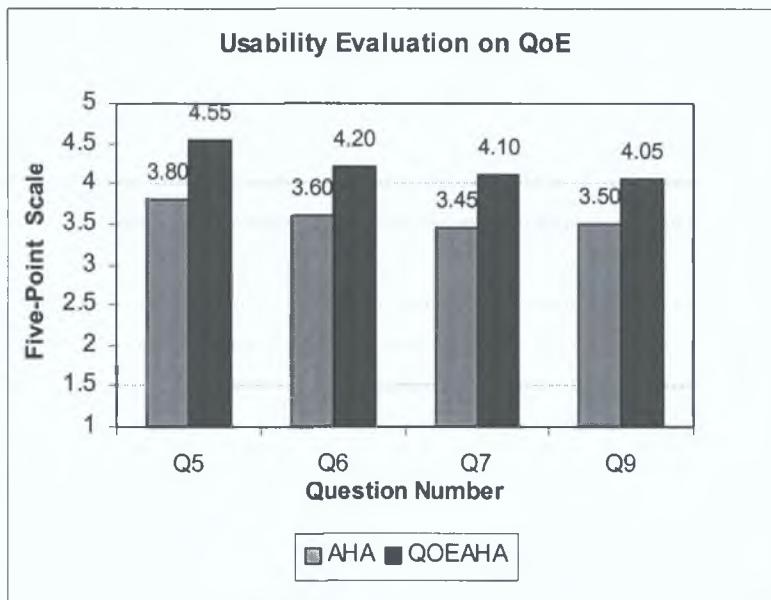


Figure 6-6 Usability Evaluation Results on Questions that Assessed the End-User QoE

The last two question categories seek to assess the end-user QoE. Four questions of the survey relate to these two categories. These four questions assess user opinion in relation to the overall delivery speed of the system (Q6), the download time of the accessed information in the context of Web browsing experience (Q7), the user satisfaction in relation to the perceived QoS (Q9) and whether the slow access to the content has inhibited them or not (Q5). The results of the QoE related questions for both AHA! and QoEAHA systems are graphically presented in Figure 6-6.

One can notice from the chart that the QoEAHA system has provided a better QoE for the end users, improving the users' satisfaction, which was above the "good" level for all questions. The AHA! system scored just above the "average" level, significantly lower than QoEAHA! This good performance was obtained in spite of the subjects using slow

connection (56 kbps) during the study session and not being explicitly informed about this. A two-sample T-Test analysis on the results of these four questions confirmed that users' opinion about their QoE is significantly better for QoEAHA than for AHA!, fact stated with a confidence level above 99 %, ($p<0.01$).

On overall, the mean value of QoE usability assessment, assuming that the questions were of equal importance was 4.22 for QoEAHA and 3.58 for AHA. This leads to an improvement of 17.8 % with the QoEAHA system.

The usability assessment on the other questions related to the navigation and presentation features achieved an average score of 3.83 for AHA! and 3.89 for QoEAHA, demonstrating that these features were not affected by the addition of the QoE enhancements.

Finally, an overall assessment of all questions from the usability questionnaire when all ten questions were considered of equal importance shows that the students considered QoEAHA system (mean value=4.01) significantly more usable then the AHA! one (mean value=3.73). These results were also confirmed by the unpaired two-tailed T-Test ($t=2.44$, $p<0.03$) with a 97 % degree of confidence. This increase of 7.5 % in the overall QoEAHA usability was mainly achieved due to the higher scores obtained in the questions related to end-user QoE.

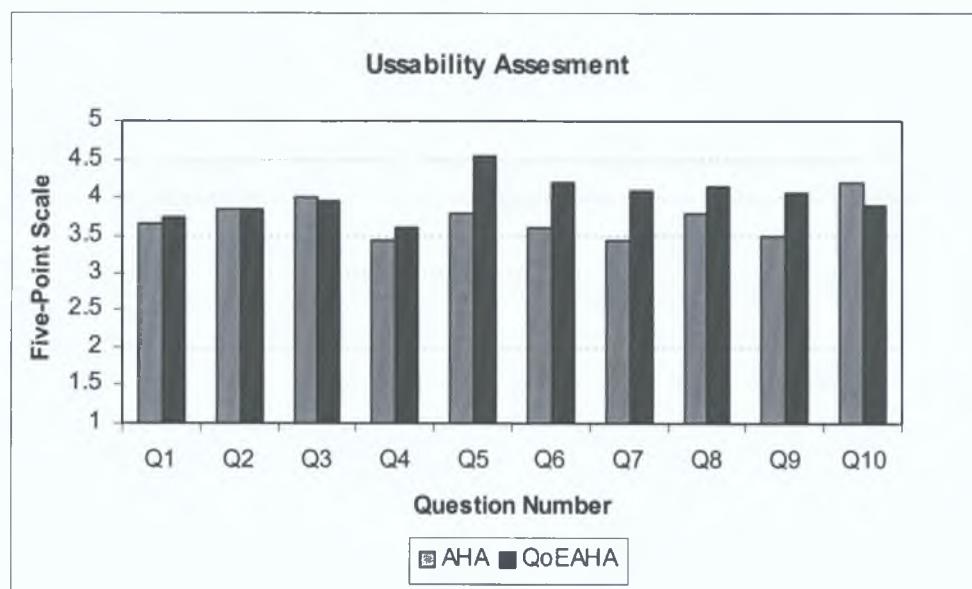


Figure 6-7 Comparative Presentation of the Answers on Usability Questionare

By examining in detail the provided answers (Figure 6-7), one can notice that only for the last question (Q10) AHA! has a slight advantage over QoEAHA while in most of the other questions QoEAHA received a higher score. Q10 is related to the user satisfaction with the quality of the provided images. The advantage of the AHA! system is justified by the fact that, as mentioned in the previous chapters, QoEAHA system performs controlled image degradation in order to improve the end-user perceived performance. Yet, these image degradations did not disturb the users since they scored this question with an average of 3.9, very close to the “good” level.

6.3.3.4 Visual Quality Assessment

The second scenario described in section 6.3.2.2 aims at visual quality assessment. The goal of this test was to investigate the effect the quality degradation performed in a controlled manner by the QoE enhancement layer has on end-users.

When the test students have completed the execution of the search for a term task (test results were presented in section 6.3.3.2.2) they were required to grade the overall quality of the two web pages seen, which included the images involved in this experiment. The visual quality was graded on a five-point quality scale (1-bad, 2-poor, 3-fair, 4-good, 5-excellent). Since the network environment between student computers and the adaptive system was set-up for the lowest connection case (28 kbps), a quality reduction of 57 % for the first page and 18 % for the second one respectively was performed. This was set because the aim of this test is to assess the visual quality in the worst-case situation.

Table 6-11 Perceptual Quality Grading for the AHA and QoEAHA Systems

	QUALITY REDUCTION (%)	AVERAGE QUALITY GRADE (1-5)	STDEV
Page 1 with AHA	0.0	4.5	0.53
Page 1 with QoEAHA	57.0	4.3	0.48
Page 2 with AHA	0.0	4.1	0.56
Page 2 with QoEAHA	18.0	3.7	0.48

The test results (Table 6-11) show that the size reduction applied on the first page did not significantly affect the visual quality of images (4.4 % reduction in the graded average perceived quality). Regardless of the AHA! version tested, lower quality grades were given to the images from Page 2 than those of the Page 1. This may be due to the fact that 50 % of the subjects found the colour combination used in the original images from Page 2 that describe the AHA! configuration interface (grey colour text on purple background) inappropriate. In the second page where the images were richer in colour compared to those from the first page, higher perceived quality degradation (9.7 %) was reported.

However, even in the worst-case situation (28 kbps) that requires the highest content reduction, the viewers' perceived quality was close to the "good" perceptual level. This suggests that the cost of image quality reduction is not significant as far as user-perceived quality is concerned while at the same time yielding significant improvements in download time.

Visual quality was also assessed in the experiment that involved the first scenario (study task). One question (Q10) of the usability questionnaire asked user opinion in relation to the quality of the delivered images. Details on these results were described in section 6.3.3.3. The average mark on this question, when QoEAHA system was used was 3.9, once again very close to the "good" level.

6.3.3.5 Correlation Between Individual Student Performance and Judgment on System Usability

One aspect worth examining is to determine whether there is or not any correlation between the performance of individual students and their judgment over system usability. Therefore, the goal of this analysis is to examine if students that performed well in the post-test evaluation, thought that the system was more usable, while students with much lower scores expressed bad opinions. A strong correlation between the two set of results (Post-Test and Usability) would have discredited to a certain extend the results of the usability evaluation experiment.

The correlation is studied by calculating the Spearman coefficient (r_s) as presented in Appendix A. The r_s coefficient takes values in the range of -1 and 1 and indicates the degree of correlation between two sets of data.

The correlation analysis has been performed for both QoEAHA and AHA! systems. The two sets of data consisted of the post-test evaluation marks and the usability questionnaire scores. For the QoEAHA the value of the Spearman coefficient was $r_s=0.23$ while for the AHA! this value was $r_s=0.03$. As both coefficients have low values there is no strong correlation between the two data sets.

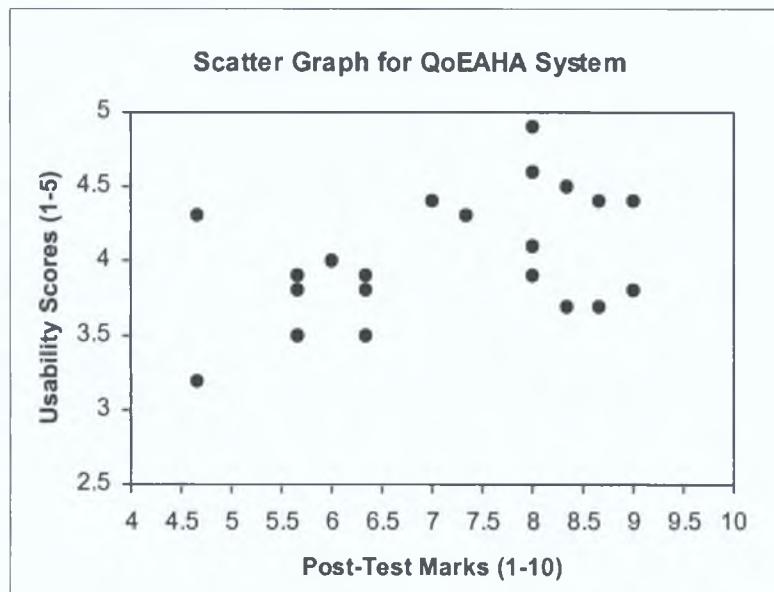


Figure 6-8 Correlation Graph Between Students Performance and Usability When Using QoEAHA

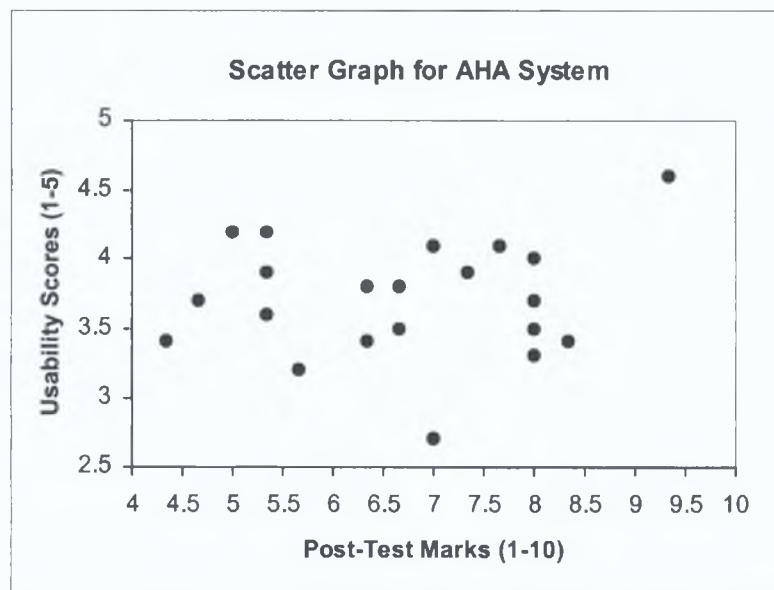


Figure 6-9 Correlation Graph Between Students Performance and Usability When Using AHA!

An alternative solution for examining this correlation is by representing graphically the values in a scatter plot. By inspecting Figure 6-8 and Figure 6-9 one can establish that there is no significant correlation between the two sets of data.

Summarizing, no correlation has been found between the students learning outcome and their judgment on the system usability. The opinion expressed by the students in the usability questionnaires was not influenced by his/her final score in the Post-Test evaluation.

6.3.3.6 Correlation Between Study Session Time and Individual Student Performance

The goal of this study is to investigate whether a longer period of study on the provided educational material has determined higher marks in the Post-Test evaluation. The analysis is performed for both studied cases, AHA! and QoEAHA respectively, and involved the computation of the Spearman coefficient.

The input set of data for Spearman correlation test consists of values for study periods taken by each student and marks obtained by each student in the evaluation test. For the QoEAHA system the value of the Spearman coefficient was $r_s=0.27$ while for AHA! this value was $r_s=0.004$. As both coefficients are low values, there is no strong correlation between the two data sets. A scatter plot for each of the systems is presented in Figure 6-10 and Figure 6-11 respectively.

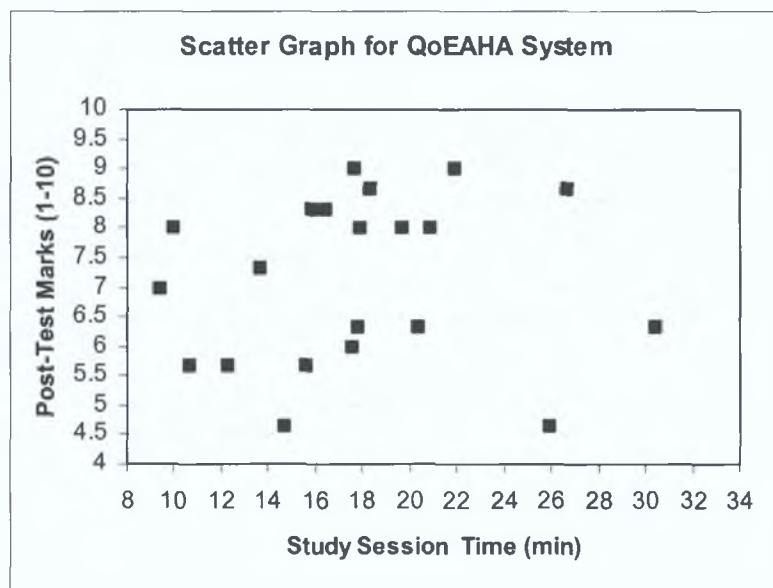


Figure 6-10 Correlation Graph Between Students Study Session Time and Students Evaluation Marks Using QoEAHA

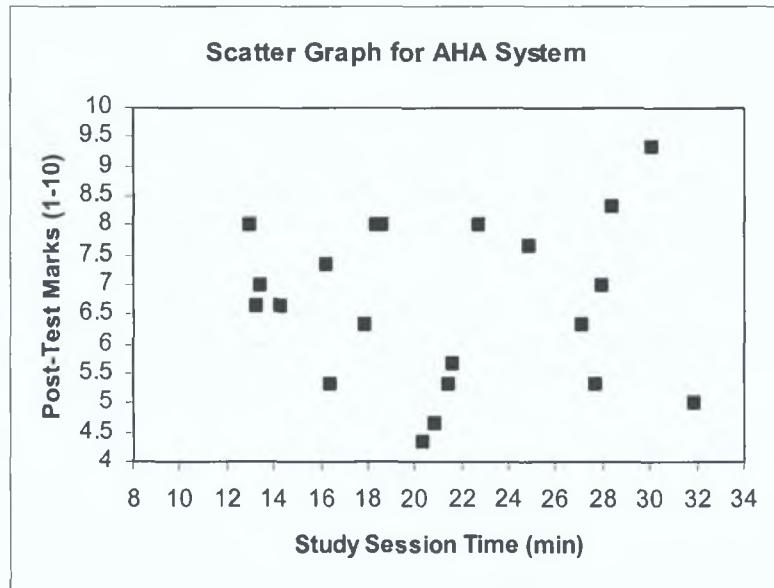


Figure 6-11 Correlation Graph Between Students Study Session Time and Students Evaluation Marks When Using AHA!

Analysing the results one can conclude that there is no strong correlation between the study period and the evaluation marks. In consequence it can be said that a low mark received by a student was not due to a short period of time allocated by him/her for the study.

6.3.4 Reflections on the Evaluation Results

The QoE layer proposed in Chapter 4 has been evaluated in the educational area by deploying it on the open source AHA! system. As result, the QoEAHA system was built. Over 60 students from Dublin City University took part in the evaluation process of the QoEAHA that involved two scenarios. The first scenario covered an interactive learning session whereas the second scenario involved visual quality assessment through a search task.

The QoEAHA evaluation strategy involved a comparison of the system with the original AHA! system. Therefore the subjects were randomly divided into two groups making use of one of the two versions of the AHA! system. The objective of the evaluation was to investigate the benefits brought by the QoE layer to the students learning performance and perceived experience with the system when a low bit operational environment (e.g. home connection) was considered.

Different educational-based evaluation techniques such as learner achievement analysis, learning performance assessment, usability survey, visual quality assessment correlation analysis between individual student performance and judgment on system usability and between study time and student marks were applied in order to fully assess QoEAHA.

Learner Achievement was analysed in the form of final scores from pre/post tests achieved by the students after a learning session. The test results show that the QoE aware AHA! system did not affect the learning outcome of the students. Both groups of students received similar marks on the final evaluation test. Therefore QoEAHA has offered similar learning capabilities to the classic AHA! system.

Learning Performance assessment involved a more complex evaluation and assessed how fast a required task is performed by a student. Two types of tasks, often performed by the students during a normal study session, were considered: learning task and search for a term task. The first task required a deeper look into the educational material, while the second one involved only a brief look up for a piece of information.

The results have shown an improvement of 16.27 % in the execution of the learning task when the QoEAHA system was used. Most of the students (71.43 %) that made use of the QoEAHA have succeeded to finish the task in up to 20 minutes while only 42.85 % of students from AHA! group have finished in the same period. Improvements to the learning performance brought by the QoEAHA were also confirmed through two other metrics: *Time spent per page* and *Number of accesses to a page performed by a user*. On average, the students from QoEAHA group spent less time per page and they performed a smaller number of re-visits to a page than those from AHA! group. These improvements brought by the QoEAHA were due to the fact that the educational material was delivered faster and the students were constantly focused on their task. Long period of waiting time for a required page annoys people and disturbs their concentration on the task.

Results on the execution of the search for a term task have confirmed that controlled degradation into the quality of the content did not affect the functionality of the system. Both groups of students have succeeded to complete the task in similar period of time.

System Usability was investigated through an online questionnaire, with answers on a five-point scale, filled out by both groups of students at the end of the study task. The questionnaire assessed different aspects of the system such as navigation, presentation,

accessibility, user QoE and subjective feedback. The results show that the students appreciated better the QoEAHA system for providing high QoE to the end-users. All the questions related to the end-user QoE were marked on average over the “good” level (between 4.10 and 4.55 points) while the AHA! system was scored between 3.45 and 3.8. Questions related to the other aspects of the system (e.g. navigation, presentation) achieved similar marks with both systems demonstrating that the embedded QoE layer did not affect them. Finally an overall assessment of the system usability has shown that students considered QoEAHA significantly more usable than AHA!. QoEAHA has achieved 7.5 % increase in usability due to the high marks awarded on QoE related questions.

Another evaluation technique used during the tests involved a *Visual Quality Assessment* in the lowest connection case (28 kbps). It involved a survey related to content quality performed after the execution of the search for a term task when the two versions of the AHA! system were used. Although the students from the QoEAHA group experienced significant controlled quality degradations (up to 57 %), the results have reported only a small degradation in the users opinion (up to 9.7 %). However the results still remained close to the “good” perceptual level, when compared to the one expressed by the AHA! group of students. Therefore, the cost of quality reduction is not significant as far as user-perceived quality is concerned whereas QoEAHA provides significant improvements in the learning time.

The last type of technique used for the evaluation process of the QoEAHA system involved *Correlation Analysis* based on Spearman coefficient. The studies have shown that for both QoEAHA and AHA! systems there is no strong correlation between individual student Post-Test scores and their judgement on system usability. Therefore the feelings expressed by the students in the usability survey were not influenced by the marks received in the evaluation test. No correlation has also been found between the length of the students study period and their final marks. Students that have spent more time on the material have not necessarily received higher marks. The same conclusion was drawn when very short periods of study were considered, as the results show there were students that have received high marks after a short period of study. In conclusion the subjects did really put all their effort when they took part in the experiment and they performed as they would have performed the required task in a real environment.

6.4 Chapter Summary

The aim of the research presented in this chapter was to deploy the proposed QoE layer on an Adaptive Hypermedia System (AHS) and to investigate the usability and effectiveness of the proposed extension. The choice to use AHA! as the foundation for building a QoE-aware-AHS, lately named QoEAHA, was based on issues such as availability (open-source), trust (tested), etc. However the proposed QoE module could be easily embedded in any AHS that respects the principles of the AHAM model.

Since AHA! was intensively tested in the area of education and the educational material was provided together with the system, we have decided to make use of the same courseware and application area for testing the QoEAHA.

The first section of this chapter examined current techniques used in the evaluation of the adaptive educational applications. Based on this analysis, the QoEAHA evaluation strategy was underlined. It involved experimental testing through objective and subjective methods. The goal of these experimental tests was to analyze the students learning process, learning outcome and their opinion on the system when the QoEAHA was used. The results were compared with the ones gathered in the case when the AHA! system was used. The conclusions drawn from the tests were summarized in a separate section.

The following chapter, the final one, concludes the work proposed in this thesis and indicates possible future directions for extending both the QoE framework and this research in the area of AHS.

Chapter VII

Conclusions

7.1 Chapter Introduction

This chapter summarises the research reported in this thesis and highlights the significant achievements of the proposed Quality of Experience Layer (QoE). The most important results of both objective and qualitative tests involving QoE layer are briefly mentioned. This chapter also summarises the contributions of this research and underscores the benefits of the proposed solution when it is deployed on an AHS for Education (AEHS). Some future work directions are suggested at the end.

7.2 Main Achievements of the Research

This research *explored a new dimension of individual differences between Web users* that focuses on end-user QoE and *proposed a novel solution for increasing end-user QoE* when personalised Web content is provided to end-users that avail of Web services over various and changeable operational environments and network delivery conditions.

The proposed solution bridges for the first time two research areas: Adaptive Hypermedia that focuses on serving adapted Web content based on user characteristics, and Web QoS which modifies the volume of data transmitted in response to observed performance problems. The solution consists of a QoE layer enhancement for the classic AHS that:

- Analyses multiple factors such as network connectivity properties, Web page characteristics and user behaviour that affect Web service performance as perceived by the end-user.
- Estimates both users' satisfaction related to their Web experience and current user operational environments by making use of different performance-based

metrics such as round-trip time, throughput, download time, tolerance for delay and utility of a session.

- Performs correlation between these metric values and optimal Web page characteristics that provide the best QoE, by making use of a novel stereotype-based Perceived Performance Model (PPM).
- Applies performance-based content adaptation to Web pages based on PPM content characteristics suggestions and strength of user interest in the fragments of information presented in the personalised Web page. A proposed adaptation algorithm determines the correct performance-based adaptation strategy in order to maximise user's satisfaction with the quality of delivered information.

The proposed QoE layer was extensively tested using both simulation and qualitative-based measurements. Low bit-rate operational environments were considered.

The simulation tests assessed the performance improvements due to the QoE layer when performance-based adaptive Web content is delivered over various low-bit rate network environments. The results show that regardless of the network state the download time per page did not exceed the 12-15 sec threshold providing a satisfactory end-user QoE. More significantly the aggregate access time per learning session decreased by up to 56% in constant network conditions and by up to 39.8% in step-wise changeable network conditions. The analysis of the PPM behaviour in constant and step-wise changeable environments indicated that QoE layer successfully determined the current delivery conditions, tracked quickly changes in the network environment that occurred during a learning session and provided the optimal Web page characteristics that maximise the end-user QoE.

The benefits due to the proposed QoE layer in adaptive personalised e-learning were highlighted by deploying the proposed layer on AHA!, an open-source AHS applied in the educational area, and by performing qualitative evaluation. This evaluation involved a comparison of the QoE-aware AHA (QoEAHA) system with the original AHA! system by using different educational-based analysis techniques. While the learner achievement analysis indicates that QoEAHA offers similar learning capabilities to the AHA! system, learning performance and usability analysis highlight the advantages provided by the QoEAHA system. The usage of the QoE layer provided an improvement of 16.27 % in the execution time of a learning task. The students that made used of the QoEAHA system were constantly focused on the execution of the task and therefore they spent less time per page and they performed a smaller number of re-visits to a page. The system usability analysis

shows that the students considered the QoEAHA system significantly more usable than AHA due to a 17.8 % improvement in the end-user QoE-related usability.

7.3 Conclusions Drawn from the Research

Next the most significant conclusions drawn from both objective and qualitative testing are presented.

7.3.1 Simulation Testing Conclusions

- *When a learning session in constant low bandwidth delivery network conditions is performed, QoE layer succeeded to determine very quickly the optimal characteristics of the Web pages* for a given non-changeable network environment and to perform the correct performance based adaptations. These adaptations have provided to the end-user an average access time per page no higher than 15 sec for very low speed connections (up to 42 kbps) and between 10 sec and 12 sec for better network connectivity (up to 128 kbps).
- *Improvements in the Access Time per Page have determined a reduction of the Aggregate Access Time per navigation session.* The lower the network connectivity, the higher the reduction obtained. For example for a 28 kbps constant connection speed when using QoE layer the aggregate access time decreased by 56.2 % while for the 128 kbps case by 15.9 % when the same set of ten Web pages were selected and delivered. These values were computed in comparison with the case when no performance-based adaptations are performed on the delivered Web pages.
- An analysis of the PPM behaviour for different types of non-changeable delivery conditions has shown that *as network connectivity improves the model provides higher values for the Web page characteristics in order to allow for more information to be transmitted while maintaining a satisfactory download time.*
- *When a learning session is performed over a three-stage step- wise changeable network environment* (between 56 kbps and 96 kbps), the *QoE layer tracked successfully and quickly the changes* into the network environment from one stage to another and PPM suggested *optimal characteristics of the Web pages for the current network conditions*. Regardless of the network state the access time per page did not exceed 15 sec when QoE layer was used.

- An analysis of the PPM behavior in a step-wise changeable network environment shows that *as network conditions improved the PPM model succeeded immediately to classify the user with higher percentages in stereotype classes that correspond to better network connectivity.* In consequence, the model proposed higher values for the Web page characteristics that match the improvements in the network conditions and allow higher quantity of information to be delivered thus improving the end-user QoE. Analogous fast learning and sensitive behaviour of PPM was noticed when the network delivery conditions deteriorated. The end-user was classified with higher percentages in stereotype classes that suggest tougher constraints on the optimal characteristics of Web pages.
- *The “learning” behaviour of the PPM model on the delivery conditions overcomes sharp fluctuations of the network environment* as it combines the current generated PPM content related suggestions for a Web page to be delivered with the ones generated for previously accessed pages. This behaviour of the PPM is shown by the simulation test results for the step-wise changeable network environment case.
- Simulations have shown important improvements of the access time per page and of the aggregate access time per session, which were due to the *controlled reduction of the quantity of transmitted information performed by QoE.*

7.3.2 Qualitative Testing Conclusions

- *The QoE layer does not affect the learning outcome of the students.* Both groups of students that made use of AHA! or QoEAHA systems respectively during the learning process, received similar marks on the final evaluation test. Therefore, the *QoEAHA offers similar learning capabilities as the classic system.*
- Student learning performance, assessed in terms of execution time of a task, time spent per page and number of accesses to a page performed by a user, improved when the QoEAHA system was used by the students to perform a learning task. 16.27 % improvement in the learning task execution time was obtained and most of the QoEAHA group students (71.43 %) finished the task in up to 20 min. while only 42.85 % of the AHA! group students finished in the same period of time. Also, on average, the students from QoEAHA group spent less time per

page and they performed a smaller number of re-visits to a page than those from the AHA! group.

- *The usage of QoE layer brought significant learning performance improvements that are due to the fact that the educational material is delivered faster and the students are constantly focused on performing their tasks.* Long periods of waiting for a required page annoys people and disturbs their concentration on the task.
- Execution of a search for a term task performed by the students confirms that *controlled degradation of the quality of the Web content*, as result of the performance based adaptation performed by the QoE layer, *does not affect the functionality of the system*. Both groups of students that made use of AHA! or QoEAHA have succeeded to complete the same task in similar periods of time.
- *Visual quality assessment surveys on the delivered educational material* when the QoEAHA was used over very low connection speeds (28 kbps and 56 kbps) *have indicated a small degradation (up to 9.7 %) in the users' opinion on the perceived quality* of the content in comparison with the case when AHA! system was used in the same conditions. *However the average quality grade was 3.9 which is very close to the “good” perceptual level on the 1-5 perceptual quality scale.*
- *An overall assessment of system usability has shown that students considered QoEAHA system significantly more usable than AHA!* QoEAHA has achieved 7.5 % increase in the usability survey due to the higher marks awarded on the QoE-related questions. All the questions related to the end-user QoE were marked on average over the “good” quality level (between 4.10 and 4.55) while the AHA! system was scored between 3.45 and 3.8. This led to an improvement of 17.8 % of the end-user QoE-related usability. Questions related to the other aspects of the system such as navigation and presentation have achieved similar marks in both situations (3.83 for AHA! and 3.89 for QoEAHA) demonstrating that the additions of the QoE layer did not affect them.
- *Correlation analysis indicates no strong correlation between students' final evaluation scores and their marks on system usability survey.* Therefore the feelings expressed by the students in the usability survey were not influenced by the marks received in the evaluation test and the survey results are consistent and trustworthy. *No correlation has also been found between the length of the*

students study period and their final marks. Students that have spent more time on the material have not necessarily received higher marks. Therefore, one can conclude that the subjects did really put all their effort when they took part in the experiment and they performed as they would have performed the required task in a real environment.

7.4 Future Work

Although the performance of the proposed QoE layer is already very good there are some aspects in relation to the applicability of the layer or to its potential extension that could be further explored.

7.4.1 Wider Range of End-User Operational Environments

PPM was designed as a general purpose performance modelling solution that takes as input an integer that defines the number of stereotypes, a list of performance parameters (group of features) and possible values associated with each parameter (list of linguistic terms), and a list of properties of the delivered information (list of suggestions) and possible values associated with each property (list of linguistic terms). It provides as output the optimal values for the properties of the delivered information for a user that operates in a given network environment.

The proposed PPM was instantiated and tested for the situation where end-users operate in various types of home-like low bit rate environments. The model can be extended to also cover higher bit rate connections. In this case, an instance of the PPM will consist of a different list of linguistic terms associated with the features and suggestions and eventually a higher number of stereotypes. Simulation tests may assess the performance benefits brought by the QoE layer when a user changes connectivity from various wired connections and especially when the background traffic of different types and with various variation patterns affects the load of the delivery network.

7.4.2 Optimisation of the QoE Layer

Currently the proposed PPM makes use of the stereotype-based technique to construct and infer end-user QoE related information. Since PPM is similar to the User Model (UM) in gathering and storing information about the user from the perceived performance point of view, other techniques such as Bayesian networks applied in User Modelling could be used in the PPM. Therefore a possible research direction would be to

investigate other techniques already applied in user modelling, for manipulating performance related data and for inferring information on user's experience with the system. These techniques would also allow other strategies to be used for combining the historical information accumulated by the system. An analysis in terms of content-related suggestions generated by the PPM when the new modelling techniques are used and end-user QoE when these suggestions are applied could be performed.

Another optimisation may consist in looking for other solutions for combining the PPM content related suggestions for the current accessed page with the historical ones. Currently an average-based strategy is used which gives equal weights to all suggestions generated by the model during the previous accesses performed by the user. A new solution may consist of providing higher weights to suggestions generated more recently when some network performance parameters account for certain variation patterns. Therefore the model could track the changes in the network faster and thus provide better final content related suggestions, optimising the end-user QoE.

7.4.3 QoE E-Learning System that Provides Personalised Multimedia Content

The delivery of multimedia content to end-users over heterogeneous network delivery conditions and to various devices presents significant challenges. Therefore, another direction is to broaden the use of the QoE layer to applications that deliver personalised multimedia content to e-learners. The goal would be to maximise their experience with the learning process that could be performed on different devices via different types of networks. This novel approach would bring together research in the areas of QoS multimedia networking and Adaptive Hypermedia.

The current QoE-aware AHS performs adaptive delivery of Web pages that consist of text and images only. The proposed extension would allow for the delivery of high quality multimedia-based educational material to mobile users. The material would be personalised according to user characteristics, device and network connectivity properties, and would be adjusted in real-time during the streaming process based on variations in delivery conditions by switching between different quality versions of the same piece of information.

This extended QoE layer would monitor and analyse in real-time the values of multimedia-streaming-related parameters such as delay, loss, jitter and bit rate; end-user perceived multimedia quality estimation metrics; device-based properties such as display

size, power, etc. and would make suggestions about the optimal type and characteristics of multimedia stream (bit rate, frame rate, resolution) delivered to the user in order to provide a good level of QoS.

7.4.4 Performance Oriented Open Adaptive Hypermedia Systems

In the early 1990s different research directions were formed within the hypermedia research community such as Adaptive Hypermedia (AH) and Open Hypermedia (OH).

AH researchers combine and extend ideas from Artificial Intelligence and User Modeling fields to create complete systems that help the user in knowledge acquisition and enable personalized access information. The hypermedia material is located together with the adaptive system and the authors of the adaptive hypermedia system know the Web content at design time of the system. More details about AH were provided in Chapter 2 of this thesis.

OH research addresses the issues of integrating hypermedia functionality into existing applications and focuses on cross-application access and distributed hypermedia. This required the separation of links from documents, allowing the hyperstructure to be processed separately from the media it relates to. Thereby they proposed Open Hypermedia Systems (OHS) as middleware components that provide centralised locations for links from where they could be maintained, processed and propagated to other systems easier than if they had been embedded into existing documents. The OH community is currently addressing issues related to interoperability between different OHS by proposing communication protocols (e.g. Open Hypermedia Protocol) and a set of guidelines and standards for the representation of resources that allow for interoperation (e.g. Web semantic technologies) and generation of the hypertext structure.

Although a crossover between the OH and AH researchers would help to find solutions to shared problems in some application areas (e.g. education) and even to yield new research directions, these two communities still continue to exist independently of each other. In the early 2000 some research groups [201, 202] attempted to join the two areas by bringing AH techniques in the OH field and proposing Open corpus Adaptive Hypermedia System (OAHS). OAHS is a system that adapts hypermedia documents to individual needs of the user regardless of the origin or location of the materials. The materials may be part of a tutorial, may refer to the content from a personal Web page or could be learning objects that belong to an open repository of learning material.

OAHS may be useful in the educational area where a large collection of learning objects exists. Since many authors can easily add learning objects to an open learning repository, oversupply of information may occur. Therefore an OAHS will decide which is the best learning object for a user, based on user's interest and current knowledge state, best suitable learning style, etc. An issue that may rise is the network connection performance between the source of the learning object and the OAHS, and user-OAHS. The selection of the learning object should be based not only on user's personal characteristics but also on the connectivity properties in order to allow for a fast transfer from the source to user terminal. In this context a possible further research direction is to explore a combination of the proposed QoE layer with OAHS and to analyse the improvements this layer may provide for the educational area. The trade-off between performance and content suitability could be investigated.

7.5 Chapter Summary

This last chapter highlights the significant achievements of the research work described in this thesis and presents conclusions drawn after the QoE layer was designed and tested. The results of tests, both objective and qualitative, are briefly summarised. Then several possible directions to extend this work are presented.

Appendix A

Statistical Analysis

A.1 Average

The most common thing one can do with a data set once collected is to compute its average value. The average is a statistic information that characterizes the typical value of the data and eliminates the random scattering of values. There are several common methods of selecting a "typical" value for data. The most common method is the *arithmetic mean*. The mean value is computed by adding up all the data values and dividing by the number of data items. The mean term is also most commonly referred to as the average. Thus, the term average is considered as a synonym for the mean.

Other methods for the measurement of the typical values include *median*, *geometric mean* and *mode*. *Median* is the value of the point, which has half the data smaller than that point and half the data larger than that point. *Mode* represents the value of the random sample that occurs with the greatest frequency. It is not necessarily unique. *Geometric mean* of "n" numbers is expressed as the n-th root of their product.

A major disadvantage of the mean value is that it is sensitive to outlying points, whereas the median will be unchanged.

A.2 Variance

The variance (σ^2) is one of several indices of variability that statisticians use to characterize the dispersion among the measures in a given set of data (population). It indicates how spread out a distribution is. The variance is computed as the average squared deviation of each number from its mean (Equation 0-1).

$$\sigma^2 = \frac{\sum_{i=1}^N (value_i - mean)^2}{N}$$

Equation 0-1

A.3 Standard Deviation

The standard deviation is another indice of variability to characterize dispersion among measures in a given set of data. The standard deviation (σ) is a statistic value that tells how tightly all the various examples are clustered around the mean in a set of data. When the examples are pretty tightly bunched together the standard deviation is small. When the examples are spread apart the standard deviation is relatively large.

Standard deviation is usually displayed using a bell curve representation of a normal distribution. A normal distribution of data means that most of the examples in a set of data are close to the "average," while relatively few examples tend to one extreme or the other. When the examples are tightly packed together the bell-shaped curve is steep and standard deviation is small. When the examples are spread apart the bell curve is relatively flat.

Standard deviation is the measure of spread most commonly used in statistical practice when the mean is used to calculate central tendency. It is calculated as the square root of variance (Equation 0-2).

$$\sigma = \sqrt{variance}$$

Equation 0-2

Because of its close links with the mean, standard deviation can be greatly affected if the mean gives a poor measure of central tendency. Standard deviation is also influenced by outliers. One value could contribute largely to the results of the standard deviation. In that sense, the standard deviation is a good indicator of the presence of outliers.

Standard deviation is also useful when comparing the spread of two separate data sets that have approximately the same mean. The data set with the smaller standard deviation has a narrower spread of measurements around the mean and therefore usually has comparatively fewer high or low values. An item selected at random from a data set whose standard deviation is low has a better chance of being close to the mean than an item from a data set whose standard deviation is higher.

A.4 T-Test

T-Test is the most widely used statistical test because it is simple, straightforward, easy to use, and adaptable to a broad range of situations. It is used to compare how groups of subjects perform in two different test conditions. The T-Test analysis allows for the determination if there is a statistically significant difference between the means of two groups, at a certain confidence level.

There are two types of T-Tests: paired and unpaired. The *paired (two-sample) T-Test* implies that the measures were taken on the same individuals on two different occasions, or that there is some other inherent dependency among the groups. For example, a paired T-Test can be used to examine two sets of scores across time as long as they come from the same students. The unpaired T-Test implies that the two groups of subjects are unrelated and therefore independent samples are used.

Certain assumptions must be met in order to conduct a T-Test analysis such as data points of each sample/group are normally distributed, all individuals are selected at random from the population and have equal chance of being selected, sizes of the compared groups are as equal as possible (but some differences are allowed) and smaller than 30.

In order to determine if there is any statistical difference between the mean values of two sets of data (\bar{X}_1 and \bar{X}_2) the following steps are performed:

Step 1: State the statistical hypotheses. For example, the *null hypothesis* states that there is no difference between the population means and thus the results of the two groups do not differ significantly. *The alternate hypothesis* states that there is a difference between the population means (Equation 0-3). In this case, the T-Test is used to prove or to discard the mentioned null hypothesis.

$$\begin{aligned}H_0 &: \bar{X}_1 = \bar{X}_2 \\H_A &: \bar{X}_1 \neq \bar{X}_2\end{aligned}$$

Equation 0-3

Instead of determining whether the means of the two groups were different, T-Test can also be used to determine whether the mean value of the first population is higher than the other one. In this case Equation 0-4 states the statistical hypotheses

$$H_0 : \bar{X}_1 \leq \bar{X}_2$$

$$H_A : \bar{X}_1 > \bar{X}_2$$

Equation 0-4

Step 2: Decide whether to use an unpaired T-Test or paired T-Test based on the type of samples used in the test.

Step 3: Set the significance level (α) of the test. The most common level used by researchers is typically stated at the 0.05 – 0.01 level. For example a value of $\alpha=0.05$ indicates a 95 % level of confidence of the result, whereas $\alpha =0.01$ - a confidence level of 99%.

Step 4: Decide whether a one-tailed T-Test or a two-tailed T-Test is used. Generally, researchers select two-tailed tests because they are not sure whether the difference between the means being compared will be positive or negative. In cases where the researcher has done some previous research and is sure the difference between the means will be either positive or negative, a one-tailed test can be used.

Step 5: Perform the T-Test computing degree of freedom (df), t-statistic (t), t-critical and the probability (p). The t-statistic is a number that represents the actual size of the difference between the two test means and is computed according to well-defined formulas. The t-critical is a number obtained from a statistical table related to either for one-tail or two-tail T-Test. It represents the size of the mean difference required for the alpha level selected. The probability p is the "calculated alpha." It represents the probability of the difference in the test means coming from the same population.

Step 6: Interpret the results and decide whether to

- reject the Null Hypothesis and accept the Alternative Hypothesis . This is the case when the absolute value of the t-statistic $>$ t-critical and $p < \alpha$

or

- retain the Null Hypothesis and reject the Alternate Hypothesis. This is the case when the absolute value of the t-statistic $<$ t-critical and $p > \alpha$

Example

Let's assume $\alpha = 0.05$. If the p value is less than 5% ($p < 0.05$), there is statistical significance between the two groups and the null hypothesis can be rejected. If the p value is

greater than 5%, there is no statistical significance between the two groups. A p value of less than 5% ensures that the outcome is not just as a result of coincidence, but is indeed as a result of the variables in the experiment.

A.5 ANOVA Test

The previous paragraph has shown how T-Test is used to compare means from two groups. ANOVA Test is used to compare means values from k independent groups ($\overline{X_1}, \overline{X_2}, \dots, \overline{X_k}$), where k is 2 or greater. In fact, T-Test is considered to be a special two-group version of ANOVA. If the means of only two groups are being analysed by ANOVA, the same results are achieved as would have been achieved conducting the analysis with a t-test

A similar mechanism as the one performed for the T-Test, based on a five steps algorithm is used by the ANOVA Test.

Step 1: State the statistical hypotheses.

$$H_0 : \overline{X_1} = \overline{X_2} = \dots = \overline{X_k}$$

H_A : At least two means are significantly different

Step 2: Set the significance level (α) of the test.

Step 3: Compute degree of freedom (df), mean square, f-ratio and significance value (sig.) from the data.

Step 4: Look up the table value of f-critical for the computed degrees of freedom (df) for the given confidence level.

Step 5: Make decision to reject or retain H_0 . If H_0 rejected, conduct post hoc comparisons as needed.

ANOVA uses various statistics (sum of squares, degrees of freedom and mean squares) to produce the f-ratio and the sig. value. The f-ratio is a measure of how different the means are relative to the variability between each sample. The larger this value, the greater the chance that the differences between the means are due to something other than chance alone, namely real effects. For the f-ratio to be considered statistically significant it has to be greater than f-critical and thus the null hypothesis is rejected. Like the T-Test, if the

significance value (sig) is less than an alpha level (e.g. $\alpha=0.05$) the result is statistically significant and the null hypothesis can be rejected. Therefore an ANOVA test determines whether there is a difference among the means of the groups.

The limitation of the ANOVA test is that it does not make pair-wise comparisons. The ANOVA analysis only indicates if there is a significant difference between at least one pair of the group means. It does not indicate what pair or pairs are significantly different. This leads to the need to conduct a post hoc multiple comparison test called Tukey. The Tukey test compares each group mean to each other group mean. For each pairing (e.g. (\bar{X}_r, \bar{X}_p) , where $r, p \in \overline{1, k}, r \neq p$), the difference of the group means is computed and compared with Tukey Honestly Significant Difference (HSD). HSD is determined based on the “q table”, df, α and k. If the mean difference of a pair does not exceed HSD than the two group means do not differ from each other.

A.6 F-Test

It offers a statistical analysis of the equality of the two population variances or standard deviations. It allows for deciding if the two variances are comparable with a certain confidence level. This test can be a two-tailed test or a one-tailed test. The two-tailed version tests against the alternative that the standard deviations are not equal. The one-tailed version only tests in one direction, that is the standard deviation from the first population is either greater than or less than (but not both) the second population standard deviation. The choice is determined by the problem. For example, if a new process is tested, one may only be interested in knowing if the new process is less variable than the old process. Therefore, two-tailed f-test is used.

Like the T-Test, where a t value is computed, one calculates an f value and compares this one to a table value (f-critical). Thus, F-Test analysis involves the following five steps:

Step 1: State the statistical hypotheses. The null hypothesis states that two population variances corresponding to the two samples are equal. The alternate hypothesis for the two-tailed F-test states that there is a difference between the standard deviations of the two populations. For the one-tailed version the hypothesis states that the standard deviation of one population is higher (or lower) than the other one.

Step 2: Set the confidence level (α) of the test. The most common level is stated at the 0.05-0.01 level. For example, $\alpha=0.05$ indicates a 95% level of confidence of the result.

Step 3: Calculate the *f-statistic* value and *p* value.

Step 4: Look up the table for the value of *F* for the degrees of freedom used to calculate both variances and for a given confidence level.

Step 5: Interpret the results. If the calculated *F* is greater than the value retrieved from the table, then the null hypothesis is rejected. Otherwise, the two samples could have come from the same population of measurements.

A.7 Q-Test

A problem, which often arises when making multiple measurements is that one of the measurements, may yield a result that differs excessively from all of the others. We are then faced with the problem of deciding whether to retain this questionable result and include it in the calculation of the average, or to reject it as being unreliable. This decision can be made with the help of the Q-Test.

Q-Test is a statistical test to determine if a data point that is very different from the other data points in a set can be rejected. Only one data point may be discarded for one usage of the Q-test. The Q-test involves applying statistics to examine the overall scatter of the data set. This is accomplished by comparing the gap between the suspect point (outlier) and its nearest neighbour within the range. A *Q* value is calculated by taking the ratio of the difference between the suspect data point (outlier) and its nearest neighbour to the total range of the data (Equation 0-5). The computed *Q* value is compared to a tabulated value *Q* critical (Q_c) from the statistical tables at a given confidence alpha level (α).

$$Q = \frac{|Outlier - ClosestValue|}{|HighestValue - LowestValue|}$$

Equation 0-5

If the measured *Q* is greater than Q_c , then that suspected data point can be excluded on the basis of the Q-test. For large data sets ($N > 10$) a data point that lies more than 2.6 times from the average may be excluded.

This test should be applied sparingly and never more than once to a single data set. If there are more than one suspected value then we have a group of data with a lot of scatter, and all the values have to be kept.

A.8 Correlation Test

Correlation is a technique for investigating the relationship between two quantitative, variables X and Y. Several different correlation coefficients can be calculated, but the two most commonly used are Pearson's correlation coefficient and nonparametric Spearman's rank correlation coefficient.

Pearson correlation calculations are based on the assumption that both X and Y values are sampled from populations that follow a Gaussian distribution (at least approximately) and the calculation is performed on the actual values. With large samples, this assumption is not too important. If a Gaussian assumption is not made Spearman correlation is selected instead.

Spearman correlation is based on ranking the two variables, and so makes no assumption about the distribution of the values. The calculation, which is similar with that for the Pearson correlation, is carried out on the ranks of the data.

Correlation analysis is performed in the same as any other statistical test of significance and involves the following steps:

Step 1: Formulate the null hypothesis that states there is no correlation between the datasets.

Step 2: Set the significance level (α). E.g. $\alpha=0.05$.

Step 3: Calculate the *correlation coefficient (r)* for the test data. Either Pearson or Spearman formula is used. The degree of freedom (df) is also determined.

Step 4: Look up in the correlation table for critical value of r ($r_{critical}$), for a given df and α value.

Step 5: Interpret the results. If the calculated value of r is less than or equal to $r_{critical}$, accept the null hypothesis. This means that there is no proof of significant correlation between the variables. If the calculated value of r is greater than $r_{critical}$ from the table, reject the null hypothesis and accept the alternative hypothesis.

Once a corelation has been found between the two sets of data, the strength of correlation is determined as follows:

- $|r| \geq 0.67 \Rightarrow$ strong correlation
- $0.34 \leq |r| \leq 0.66 \Rightarrow$ moderate correlation
- $|r| \leq 0.33 \Rightarrow$ weak correlation

A.8.1 Pearson Correlation

The Pearson correlation involves the computation of the r coefficient that measures the strength of the association and the direction of an eventual linear relationship between the X and Y variables.

The first step in studying the relationship between two continuous variables is to draw a scatter plot of the variables to check for linearity. The correlation coefficient should not be calculated if the relationship is not closed to linear. For correlation only purposes, it does not really matter on which axis the variables are plotted.

The following are properties of Pearson's correlation:

- The value of r does not depend upon the units of measurement
- The value of r does not depend upon which variable is labeled X and which variable is labeled Y
- $-1 \leq r \leq 1$. A positive value of r means a positive linear relationship, a negative value of r means a negative linear relationship.
- $r = \pm 1$ happens only when all the points of the scatter plot lie exactly on a straight line.
- r measures only the *linear* relationship between X and Y .

A.8.2 Spearman Correlation

While desirable, it is not always possible to use a parametric test such as the Pearson method. Fortunately, there are also non-parametric correlation tests. One of the most frequently used is the Spearman test. Calculation of the Spearman rank order correlation coefficient (r_s) is used when the data consists of ordinal variables (i.e. variables with an ordered series where numbers indicate rank order only). Although this is a non-parametric

statistic, it may be a better indicator than the Pearson coefficient of a non-linear relationship between two variables.

In order to perform the Spearman test, the data must first be converted into rank order. When converting to rank order, the smallest value on X becomes a rank of 1, etc. An example of conversion is presented in Table 0-1. Next, the five-steps algorithm is applied and Spearman correlation coefficient (r_s) is computed.

Values of r_s range from +1 (perfect correlation), through 0 (no correlation), to -1 (perfect negative correlation).

Table 0-1 Spearman Rank Order Conversion

X	Y	CONVERT TO RANKS:	X	Y
7.3	4.2		2	1
5.15	7.6		1	2
8.99	9.12		3	4
9.01	8.7		4	3

Appendix B

Post Test Evaluation Questions

User ID:

1. Is the following statement true? AHA system is an application installed at the client.

- Yes
- No

2. The software environment required by the AHA to run includes: (multiple answers)

- Tomcat Web Server
- Windows/Linux Operating System
- Sun Java SDK
- Oracle

3. The concept structure of an AHA application can be modified by:

- Author
- Any user
- Technician
- Administrator

4. In order to access AHA application the client needs to have the following software installed:

- Acrobat Reader
- Windows Media Player
- Web browser
- Text Editor

5. Is the following statement true? AHA system is implemented using Java Servlets.

- Yes
- No

6. Is the following statement true? The use of a Web server to run AHA system is optionally.

- Yes
- No

7. AHA system was tested on the following operating system:

- Solaris
- Windows 2000
- BSD Unix
- DOS

8. Currently AHA! works only with [redacted] database system.

9. When installing Tomcat Web Server you need to provide:

- User name
- User name and password
- Password
- Nothing

10. The steps required for creating AHA context include: (multiple answers)

- Login on TomCat Web Server
- Registration with ODBC
- Create configuration file
- Set-up context properties
- Save the settings

11. AHA System for Tomcat Web Server is:

- Web browser
- A service
- An applet
- A database

12. Is the following statement true? AHA system requires cookies to be enabled.

Yes

No

13. Is the following statement true? An author is allowed to create AHA applications

Yes

No

14. The following operations are provided in relation to an AHA user: (multiple answers)

Add a new user

Modify user settings

Change user status

Delete all users

Remove a user

15. Each author of a AHA application has associated

A list of courses

A list of users

A database

A picture

Appendix C

Pre Test Evaluation Questions

User ID:

1. The concept structure of an AHA application can be modified by:

- Author
- Any user
- Technician
- Administrator

2. Is the following statement true? The use of a Web server to run AHA system is optionally.

- Yes
- No

3. Is the following statement true? AHA system is an application installed on the client side.

- Yes
- No

4. The steps required for creating AHA context include: (multiple answers)

- Login on TomCat Web Server
- Registration with ODBC
- Create configuration file
- Set-up context properties
- Save the settings

5. Is the following statement true? AHA system requires cookies to be enabled

- Yes
- No

6. Each author of a AHA application has associated

- A list of courses
- A list of users
- A database
- A picture

Appendix D

Usability Evaluation

Questionnaire

User ID:

1. Assess ease of access to the required information

- poor
- fair
- average
- good
- excellent

2. Assess the web-based learning system as a whole

- poor
- fair
- average
- good
- excellent

3. Assess the web-based learning system navigational structure

- poor
- fair
- average
- good
- excellent

4. Assess the aspect of the presentation of the web site.

- poor
- fair
- average
- good
- excellent

5. Did slow access inhibit your use of the site?

- very much
- much
- average
- little
- very little

6. Assess the overall delivery speed.

very slow

slow

average

good

very high

7. In the context of your experience with the web browsing how quick was the download of the information?

very slow

slow

average

fast

very fast

8. Express your satisfaction when working with the site

poor

fair

average

good

excellent

9. Express your satisfaction with the performance of the delivery of the information

- poor
- fair
- average
- good
- excellent

10. Express your satisfaction with the quality of the images

- poor
- fair
- average
- good
- excellent

References

- [1] NetCraft Survey 2005 <http://news.netcraft.com/archives/2005/03/01/>
- [2] Hobbies' Internet Timeline <http://www.isoc.org/guest/zakon/Internet/History/HIT.html> v.5.0.
- [3] <http://www.etforecasts.com>
- [4] "A primer on e-learning...The framework, the market, the players" KMWorld Magazine, Vol. 12, No. 2, February, 2003
- [5] P. Brusilovsky, "Adaptive Hypermedia", User Modeling and User-Adapted Interaction Journal, Vol. 11 No. 1-2, pp. 87-110, 2001
- [6] Vannevar Bush "As We May Think", 1945, www.ps.unisb.de/duchier/pub/vbush.shtml
- [7] J. Nyce, P.Kahn, "A Machine for the Mind: Vannevar Bush's Memex", In From Memex to Hypertext: Vannevar Bush and the Mind's Machine, J. Nyce and P. Kahn, Eds. Academic Press, Inc., 1991
- [8] Xanadu Project <http://xanadu.com>
- [9] N. Yankelovich, M. Norman S. Drucker, "Intermedia: The Concept and the Construction of a Seamless Information Environment." IEEE Computer Journal No.21, pp.81-96, 1988
- [10] J. H. Walker, "Document Examiner: Delivery Interface for Hypertext Documents", 1st ACM Hypertext Conference, Chapel Hill, North Carolina, USA, pp. 307-323, 1987
- [11] F. Halasz, "Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems", CACM Vol. 31, No.7, pp. 836-852, 1988
- [12] D. Lowe, W. Hall, "Hypermedia and the Web: An Engineering Approach", Wiley, ISBN: 0471983128, 1999
- [13] T. H. Nelson, "Literary Machines", Self-published, Swarthmore, PA, 1981
- [14] T. Nelson, "All for One for All", In Proceedings of ACM Hypertext Conference, Chapel Hill, North Carolina, USA, 1987

- [15] <http://www.hyperdictionary.com/>
- [16] B. Shneiderman, G. Kearsley, "Hypertext Hands-On: An introduction to a New Way of Organizing and Accessing Information", Addison, Wesley, 1989
- [17] J. Multisilta, S. Pohjolainen, "Hypermedia and Animation in CACSD Education", Int. Journal of Mathematical Education in Science and Technology, Vol. 26, No. 4, pp. 599-618, 1995
- [18] M. Bieber, F. Vitali, H. Ashman, V. Balasubramanian and H. Onias-Kukkonen, "Fourth Generation Hypermedia: Some Missing Links for the World Wide Web", International Journal of Human-Computer Studies, Vol.47, pp. 31-65, 1997
- [19] A. Kobsa, J. Koenemann, W. Pohl, "Personalised Hypermedia Presentation Techniques for Improving Online Customer Relationships", The Knowledge Engineering Review, Vol. 16, No.2, pp. 111–155, Cambridge University Press, 2001
- [20] P. Brusilovsky, "Methods and Techniques of Adaptive Hypermedia", User Modeling and User-Adapted Interaction Journal, Special Issue on Adaptive Hypertext and Hypermedia Vol. 6, No. 2-3, pp. 87-129, 1996
- [21] C. Boyle, A. O. Encarnacion, "MetaDoc: An Adaptive Hypertext Reading System", User Models and User Adapted Interaction Journal, Vol. 4, No. 1, pp. 1-19, 1994
- [22] A. Kobsa, D. Müller, A. Nill, "KN-AHS: An Adaptive Hypertext Client of the User Modeling System BGP-M", 4th International Conference on User Modeling, Hyannis, MA, USA, pp. 31-36, 1994
- [23] I. Beaumont, "User Modeling in the Interactive Anatomy Tutoring System ANATOM-TUTOR", User Models and User Adapted Interaction Journal, Vol. 4, No. 1, pp. 21-45, 1994
- [24] H. Hohl, H. D. Böcker, R. Gunzenhäuser, "Hypadapter: An Adaptive Hypertext System for Exploratory Learning and Programming", User Models and User Adapted Interaction Journal, Vol. 6, No. 2-3, 1996
- [25] P. De Bra, L. Calvi, "AHA! An Open Adaptive Hypermedia Architecture" The New Review of Hypermedia and Multimedia, Vol. 4, pp. 115-139, 1998
- [26] K. Höök, J. Karlgren, A. Wærn, N. Dahlbäck, C. G. Jansson, K. Karlgren, B. Lemaire, "A Glass Box Approach to Adaptive Hypermedia", User Models and User Adapted Interaction Journal Vol. 6, No. 2-3, 1996
- [27] J. Hothi, "Using An Open Hypermedia System to Develop New Techniques in Adaptive Hypermedia", PhD Thesis, University of Southampton, 2001
- [28] T. Joachims, D. Freitag, T. Mitchell, "WebWatcher: A Tour Guide for the World Wide Web", Proceedings of IJCAI97, Nagoya, Aichi, Japan, 1997

- [29] N. Mathé, J. Chen, "User-Centered Indexing for Adaptive Information Access", User Models and User Adapted Interaction Journal, Vol. 6, No. 2-3, pp. 225-261, 1996
- [30] C. Kaplan, J. Fenwick, J. Chen, "Adaptive Hypertext Navigation Based on User Goals and Context", User Models and User Adapted Interaction Journal, Vol. 3, No. 3, pp. 193-220, 1993
- [31] P. Brusilovsky, L. Pesin, "ISIS-Tutor: An Adaptive Hypertext Learning Environment", JCKBSE'94, Japanese-CIS Symposium on Knowledge-based Software Engineering, Pereslavl- Zalesski, Russia, pp. 83-87, 1994
- [32] C. G. Thomas, "BASAR: A Framework for Integrating Agents in the World Wide Web", IEEE Computer Journal, Vol. 28, No. 5, pp. 84-86, 1995
- [33] C. Bailey, S. El-Beltagy, W. Hall, "Link Augmentation: A Context-Based Approach to Support Adaptive Hypermedia", S. Reich, Tzagarakis, M. & De Bra, P. (Eds), Hypermedia: Openness, Structural Awareness, and Adaptivity, Springer, LNCS 2266, pp. 239-251, 2001
- [34] P. Maglio, S. Farrell, "LiveInfo: Adapting Web Experience by Customization and Annotation", First International Conference on Adaptive Hypermedia and Adaptive Web-based Systems (AH2000), Springer, LNCS 1892, pp. 144-154, 2000
- [35] P. Brusilovsky, J. Eklund, E. Schwarz, "Web-based Education for All: A Tool for Developing Adaptive Courseware", The 7th International World Wide Web Conference, pp. 291-300, 1998
- [36] G. Weber, M. Specht, "User Modelling and Adaptive Navigation Supporting WWW-based Tutoring Systems", The 6th International Conference on User Modeling, pp. 289-300, 1997
- [37] P. De Bra, L. Calvi, "AHA: A Generic Adaptive Hypermedia System", ACM Conference on Hypertext and Hypermedia (HYPERTEXT'98), 2nd Workshop on Adaptive Hypertext and Hypermedia, Pittsburgh, USA, pp. 5-12, 1998.
- [38] R. Wieringa, E. Dubois, "Integrating Semi-formal and Formal Software Specification Techniques", Information Systems Journal, Vol. 19, No. 4, pp. 33-54, 1994
- [39] B. Campbell, J.M. Goodman, "HAM: A General Purpose Hypertext Abstract Machine", Communications of the ACM Journal, Vol. 31, No. 7, pp. 856-861, 1988
- [40] J. M. Spivey, "The Z Notation: A Reference Manual", Prentice Hall International Series in Computer Science, Prentice-Hall, Inc., 1989

- [41] F. Halasz, M. Schwartz, "The Dexter Hypertext Reference Model", Communications of the ACM Journal Vol. 37, No. 2, Grønbæk K. and Trigg R. (Eds.), pp. 30-39, 1994
- [42] P. De Bra, G. J. Houben, H. Wu, " AHAM: A Dexter-based Reference Model for Adaptive Hypermedia", ACM Hypertext '99 Conference, Darmstadt, Germany, pp. 147-156, 1999
- [43] N. Koch, M. Wirsing, "Software Engineering for Adaptive Hypermedia Systems", The 3rd Workshop on Adaptive Hypertext and Hypermedia, pp. 205-210, 2001
- [44] N. Koch, M. Wirsing, " The Munich Reference Model for Adaptive Hypermedia Applications" The Second International Conference on Adaptive Hypermedia and Adaptive Web Based Systems (AH '02), Springer, LNCS 2347, pp. 213-222, 2002
- [45] F. Albrecht, N. Koch, T. Tiller, "SmexWeb: An Adaptive Web-Based Hypermedia Teaching System", Journal of Interactive Learning Research, Vol. 11 No. 3-4, pp. 367-388, 2000
- [46] A. I. Cristea, A. de Mooij, "LAOS: Layered WWW AHS Authoring Model and Their Corresponding Algebraic Operations", The Twelfth International World Wide Web Conference (WWW'03), Alternate Track on Education, Budapest,Hungary, 2003
- [47] A. I. Cristea, A. De Mooij, "Adaptive Course Authoring: MOT, My Online Teacher", In Proceedings of ICT-2003, IEEE LTTF International Conference on Telecommunications, Telecommunications and Education Workshop Tahiti Island, Papetee - French Polynesia, 2003
- [48] J. Vassileva, "A Task-Centered Approach for User Modelling in a Hypermedia Office Documentation System", User Modeling and User Adapted Interaction, No.6, pp. 185-223, 1996
- [49] D. Küpper, A. Kobsa, "User-Tailored Plan Generation" in J Kay (ed.), The 7th International Conference on User Modeling (UM99), Springer-Verlag, pp. 45–54, 1999
- [50] J. Gilbert, C. Han, "Arthur: Adapting Instruction to Accommodate Learning Style", World Conference on the WWW and Internet (WebNet'99), Honolulu, USA, pp. 433-438, 1999
- [51] M. Specht, R. Oppermann, "ACE-Adaptive Courseware Environment", The New Review of Hypermedia and Multimedia Journal, No. 4, pp. 141-161, 1998
- [52] K. Cheverst, K. Mitchell, N. Davies, "The Role of Adaptive Hypermedia in a Context-Aware Tourist Guide", Communications of the ACM, Vol. 45, No. 5, pp. 47-51, 2002

- [53] L. Ardissono, A. Goy, G. Petrone, M. Segnan, P. Torasso, "Ubiquitous User Assistance in a Tourist Information", 2nd International Conference on Adaptive Hypermedia and Adaptive Web Based Systems (AH2002), Malaga, Spain, pp. 14-23, 2002
- [54] L. Ardissono, A. Kobsa, M. T. Maybury, "Personalized Digital Television: Targeting Programs To Individual Viewers", Dordrecht, Kluwer Academic Publishers, 2004
- [55] B. Smyth, P. Cotter, "A Personalised TV Listings Service for the Digital TV Age", Knowledge-Based Systems Journal, Vol. 13, pp. 53-59, 2000
- [56] J. Masthoff, "Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewers", User Modelling and User Adapted Interaction Journal, No. 14, pp37-85, 2004
- [57] G. Lekakos, G. M.. Giaglis, "Delivering Personalized Advertisements in Digital Television: A Methodology and Empirical Evaluation", 2nd International Conference on Adaptive Hypermedia and Adaptive Web Based Systems, Workshop on Personalization in Future TV Malaga, Spain, 2002
- [58] D. Chin, "Acquiring User Models", Artificial Intelligence Review, Kluwer Academic Publishers, pp. 185-197, 1993
- [59] D. Pilar da Silva, R. Van Durm, E. Duval, H. Olivie, "Concepts and Documents for Adaptive Educational Hypermedia: A Model and a Prototype", ACM Conference on Hypertext and Hypermedia (HYPertext'98), 2nd Workshop on Adaptive Hypertext and Hypermedia, Pittsburgh, USA, pp. 35-43, 1998
- [60] M. Murphy, M. McTear, "Learner Modeling for Intelligent CALL", The 6th International Conference on User Modeling (UM97), Jameson A., Paris C. and Tasso C. (Eds.), Springer Verlag Wien, pp. 301-312, 1997
- [61] M. Encarnacão, A. Stork, "An Integrated Approach to User-centered Interface Adaptation", Technical Report WSI-96-10, University of Tübingen, 1996
- [62] A. Jameson, "Numerical Uncertainty Management in User and Student Modeling: An Overview of Systems and Issues", User Modeling and User-Adapted Interaction Journal, Vol. 5, pp. 193–251, 1996
- [63] I. Zukerman, D. W. Albrecht, "Predictive Statistical Models for User Modelling", User Modelling and User-Adapted Interaction Journal, Vol. 11, pp. 5-18, 2001
- [64] W. Nejdl, M. Wolpers, "KBS Hyperbook - A Data-Driven Information System on the Web", The 8th World Wide Web (WWW) Conference, Toronto, Canada, 1999

- [65] C. Conati, A. Gertner, K. VanLehn, M. Druzdzel, "On-Line Student Modeling for Coached Problem Solving Using Bayesian Networks", 6th International Conference on User Modeling (UM97), Chia Laguna, Sardinia, Italy, pp. 231-242, 1997
- [66] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, K. Romelse, "The Lumière Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users.", The 14th Conference on Uncertainty in Artificial Intelligence, Madison, WI, Morgan Kaufmann: San Francisco, pp. 256-265, 1998
- [67] J. Kay, R. J. Kummerfeld, "An Individualised Course for the C Programming Language", The 2nd International World Wide Web (WWW) Conference, Chicago, IL, USA, 1994
- [68] G. Weber, A. Möllenbergs, "ELM-Programming-Environment: A Tutoring System for LISP Beginners", Cognition and Computer Programming, K. F. Wender, F. Schmalhofer and H. D. Böcker (eds.), Norwood, NJ, pp. 373-408, 1995
- [69] P. Brusilovsky, E. Schwarz, G. Weber, "ELM-ART: An Intelligent Tutoring System on World Wide Web", Intelligent Tutoring Systems, Lecture Notes in Computer Science, Vol. 1086, Berlin, C. Frasson, G. Gauthier, and A. Lesgold, Eds., Springer, pp. 261-269, 1996
- [70] P. Brusilovsky, E. Schwarz, G. Weber, "A Tool for Developing Adaptive Electronic Textbooks on WWW", WebNet'96 Conference of the Web Society, Boston, MA, USA, 1996
- [71] G. Weber, P. Brusilovsky, "ELM-ART: An Adaptive Versatile System for Web based Instruction", Artificial Intelligence in Education, Special Issue on Adaptive and Intelligent Web-based Educational Systems, Vol. 12, No. 4, pp. 351-384, 2001
- [72] G. Weber, "Adaptive Learning Systems in the World Wide Web", 7th International Conference on User Modeling (UM99), Banff, Canada J. Kay (ed.), Springer Wien New York, pp. 371-378, 1999
- [73] G. Weber, H. C. Kuhl, S. Weibelzahl, "Developing Adaptive Internet Based Courses with the Authoring System NetCoach", The 3rd Workshop on Adaptive Hypertext and Hypermedia, Sonthofen, Germany, J P. D. Bra, P. Brusilovsky and A. Kobsa (eds.), pp. 35-48, 2001
- [74] H. Wu, "A Reference Architecture for Adaptive Hypermedia Applications", PhD Thesis, Technische Universiteit Eindhoven, 2002
- [75] P. De Bra, J. P. Ruiter, "AHA! Adaptive Hypermedia for All", World Conference of the WWW and Internet (WebNet'2001), Orlando, FL, USA, pp. 262-268, 2001
- [76] M. Grigoriadou, K. Papanikolaou, H. Kornilakis, G. Magoulas, "INSPIRE: An Intelligent System for Personalized Instruction in a Remote Environment", The 3rd

- Workshop on Adaptive Hypertext and Hypermedia, Sonthofen, Germany, pp. 13-24, 2001
- [77] K. A. Papanikolaou, M. Grigoriadou, H. Kornilakis, G. D. Magoulas, "Personalizing the Interaction in a Web-based Educational Hypermedia System: the Case of INSPIRE", User Modeling and User-Adapted Interaction Journal Vol. 13, No. 3, pp. 213-267, 2003
- [78] M. H. Ng, R. Armstrong, W. Hall P. Maier, "JointZone: An Adaptive Web-based Learning Application", Technical Report Number: ECSTR-IAM02-002, ISBN:0854327711, University of Southampton, 2002
- [79] M. H. Ng, W. Hall, P. Maier, R. Armstrong, "The Application and Evaluation of Adaptive Hypermedia Techniques in Web-based Medical Education", Association for Learning Technology Journal, Vol.10, No. 3, pp. 19-40, 2002
- [80] S. Garlatti, S. Iksal, P. Kervella, "Adaptive On-Line Information System by Means of a Task Model and Spatial Views", Workshop on Adaptive Systems and User Modeling on World Wide Web at WWW'99 Conference, Toronto, Canada, pp. 59-66, 1999
- [81] M. Milosavljevic, J. Oberlander, "Dynamic Hypertext Catalogues: Helping Users to Help Themselves", 9th ACM Conference on Hypertext and Hypermedia (HYPERTEXT'98), Pittsburgh, PA, USA, pp. 123-131, 1998
- [82] M. Sarini, C. Strapparava, "Building a User Model for a Museum Exploration and Information-Providing Adaptive System", 2nd Workshop on Adaptive Hypertext and Hypermedia at 9th ACM Conference on Hypertext and Hypermedia (HYPERTEXT'98), Pittsburgh, PA. 1998
- [83] J. Fink, A. Kobsa, "User Modeling in Personalized City Tours", Artificial Intelligence Review Journal, Vol. 18, No. 1, pp. 33-74, 2002
- [84] L. Ardissono, A. Goy, G. Petrone, M. Segnan, P. Torasso, "INTRIGUE: Personalized Recommendation of Tourist Attractions for Desktop and Handset Devices", Applied Artificial Intelligence Journal, Special Issue on Artificial Intelligence for Cultural Heritage and Digital Libraries, Vol. 17, No. 8-9, Taylor & Francis, pp. 687-714, 2003
- [85] M. Encarnaçao, "Concept and Realization of Intelligent Support in Interactive Graphics Applications", Ph.D. Thesis, University of Tübingen, Wilhelm-Schickard-Institute for Computer Science, Germany, 1997
- [86] J. Vassileva, J. Greer, G. McCalla, R. Deters, D. Zapata, C. Mudgal, S. Grant, "A Multi-Agent Design of a Peer-Help Environment, 10th International Conference on Artificial Intelligence in Education (AIED'2001), San Antonio, Texas, USA, pp. 38-45, 2001

- [87] J. Vassileva, R. Deters, "Lessons from Deploying I-Help", Workshop on Agents and Internet Learning (AIL'2001) at the Autonomous Agents'2001 Conference, Montreal, J. Whatley & M. Beers (eds.), 2001
- [88] K. F. Gates, P. B Lawhead, D. E. Wilkins, "Toward and Adaptive WWW: A Case Study in Customized Hypermedia", New Review of Multimedia and Hypermedia Journal, Vol. 4, pp. 89-113, 1998
- [89] M. Pazzani,J. Muramatsu, D. Billsus, "Syskill and Webert: Identifying Interesting Web Sites", The 13th National Conference on Artificial Intelligence (AAAI'96), Portland, OR, USA, pp. 54-61, 1996
- [90] T. Joachims, D. Freitag, T. Mitchell, "WebWatcher: A Tour Guide for the World Wide Web. 15th International Joint Conference on Artificial Intelligence (IJCAI'97), pp. 770-775, 1997
- [91] D. Mladenic, "Machine Learning Used by Personal WebWatcher", Workshop on Machine Learning and Intelligent Agents (ACAI 99), Chania, Crete, 1999
- [92] F. A. Asnicar, C. Tasso, C. "ifWeb: A Prototype of User Model-Based Intelligent Agent for Document Filtering and Navigation in the World Wide Web", Workshop "Adaptive Systems and User Modeling on the World Wide Web" at 6th International Conference on User Modelling (UM97), Chia Laguna, Sardinia, Italy, pp. 3-11, 1997
- [93] X. Fu, J. Budzik, K. Hammond, "Mining Navigation History for Recommendation", International Conference on Intelligent User Interfaces, New Orleans, LA, pp. 106-112, 2000
- [94] J. Vassileva, "A Task-Centered Approach for User Modeling in a Hypermedia Office Documentation System", User Models and User Adapted Interaction Journal, Vol. 6, No. 2-3, pp.185-223, 1996
- [95] W. S. Li, Q. Vu, E. Chang, D. Agrawal, K. Hirata, S. Mukherjea, Y. L. Wu, C. Bufl, C. C. K. Chang, Y. Hara, R. Ito, Y. Kimura, K. Shimazu, Y. Saito, "PowerBookmarks: a System for Personalizable Web Information Organization, Sharing, and Management", ACM SIGMOD, Vol. 28, No. 2, pp. 565-567, 1999
- [96] R. M. Keller, S. R. Wolfe, J.R. Chen, J. L. Rabinowitz, N. Mathe, "A booking service for organizing and sharing URLs", The 6th International Conference on World Wide Web, Santa Clara, California, US, pp. 1103-1114, 1997
- [97] ITU-T Recommendation E.800, "Terms and Definitions Related to Quality of Service and Network Performance Including Dependability", August 1994
- [98] ISO/IEC 10746-2, "Information Technology - Open Distributed Processing - Reference Model: Foundations", International Standards Organisation, 1996

- [99] E. S. Crawley, R. Nair, B. Rajagopalan, H. Sandick, "A Framework for QoS-based Routing in the Internet", RFC 2386, <http://www.ietf.org/rfc/rfc2386.txt>
- [100] Cisco Systems, "Internetworking Technology Handbook", 2003, http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/qos.pdf
- [101] Microsoft, "Quality of Service", Technical White Paper, September 1999, <http://www.microsoft.com/windows2000/docs/QoSOver.doc>
- [102] M. Conti, M. Kumar, S. K. Das, B. A. Shirazi, "Quality of Service Issues in Internet Web Services, IEEE Transactions on Computers, Vol. 51, No. 6, pp. 593-594, 2002
- [103] 103 M. Kihl, N. Widell, "Admission Control Schemes Guaranteeing Customer QoS in Commercial Web Sites", IFIP IEEE Conference on Network Control and Engineering for QoS, Security and Mobility (NETCON), Paris, France, isbn 1-4020-7268-6, 2002
- [104] T. F. Abdelzaher, N. Bhatti, " Web Content Adaptation to Improve Server Overload Behavior", Computer Networks, Vol. 31, No. 11-16, pp. 1563 - 1577, 1999
- [105] P. Bhoj, S. Ramanathan, S. Singhal, "Web2K: Bringing QoS to Web Servers", HPL-2000-61 Technical Report, 2000
- [106] L. Cherkasova, P. Phaal, "Predictive Admission Control Strategy for Overloaded Commercial Web Server", Proc. of 8th International Symposium on Modeling Analysis and Simulation of Computer and Telecommunication Systems, pp. 500-507, 2000
- [107] H. Chen, P. Mohapatra, "Session-based Overload Control in QoS-aware Web Servers", Proceedings of IEEE Infocom 2002, New York, US, 2002
- [108] M. Kihl, N. Widell, "Admission Control Schemes Guaranteeing Customer QoS in Commercial Web Sites", IEEE Conference on Network Control and Engineering (NetCon), Paris, France, pp. 305-316, 2002
- [109] <http://www.cisilion.com/cisco/cisco-localdirector-.htm>
- [110] <http://www.alteonwebsystems.com/collateral/acedirector.pdf>
- [111] IBM Web Accelerator, www.ibm.com
- [112] M. Conti, E. Gregori, F. Panzieri, "QoS-based Architecture for Geographically Replicated Web Servers", Cluster Computing Journal, No. 4, pp. 105-116, 2001
- [113] P. Rodriguez, A. Kirpal, "Parallel-Access for Mirror Sites in the Internet", Proceedings of IEEE Infocom 2000, Tel-Aviv, Israel, 2000
- [114] Cisco Distributed Director
<http://www.cisco.com/warp/public/cc/pd/cxsr/dd/index.shtml>

- [115] Arrow Point - <http://www.arrowpoint.com>
- [116] Alteon WebSystems - www.alteonwebsystems.com
- [117] Cisco Systems Inc. www.cisco.com
- [118] Bell Labs - <http://www.bell-labs.com>
- [119] A. Tannenbaum, "Computer Networks, 3rd Edition, Prentice Hall, 1996
- [120] RouteScience® Adaptive Networking Software <http://www.routescience.com/>
- [121] Microsoft Corporation www.microsoft.com
- [122] Novell Inc. www.novell.com
- [123] Netscape Communications Corp www.netscape.com
- [124] IBM Corporation www.ibm.com
- [125] CacheFlow www.cacheflow.com
- [126] Network Appliance Inc. www.netapp.com
- [127] S. D. Gribble, E. A. Brewer, "System Design Issues for Internet Middleware Services: Deduction From a Large Client Trace", Proceedings of USENIX Symposium on Internet Technologies and Systems, Anaheim, California, USA, pp. 207- 218, 1997
- [128] T. M. Kroger, D. E. Long, J. C. Mogul, "Exploring the Bounds of Web Latency Reduction from Caching and Prefetching, Proceedings of USENIX Symposium on Internet Technologies and Systems, Anaheim, California, USA, pages 13-22, 1997
- [129] X. Chen and X. Zhang, A Popularity-based Prediction Model for Web Prefetching", IEEE Computer, Vol. 36, No. 3, pp.63-70, 2003
- [130] E. P. Markatos, C. E. Chironaki, "A Top 10 Approach for Prefetching the Web", In Proceedings of INET'98: Internet Global Summit, Geneva, 1998
- [131] A. Venkataramani, P. Yalagandula, R. Kokku, S. Sharif, M. Dahlin, "The Potential Costs and Benefits of Long-Term Prefetching for Content Distribution" The 6th International Workshop on Web Caching and Content Distribution, Boston, Massachusetts, USA, 2001
- [132] Y. Jiang, M.Y. Wu, W. Shu, "Web Prefetching: Cost, Benefits and Performance", 11th World Wide Web Conference (WWW), Honolulu, Hawaii, USA 2002
- [133] Empirix, "Assuring QoE on Next Generation Networks", White paper, 2001

- [134] ITU-T Recommendation G.1010, "End-CUser Multimedia QoS ategories", Nov. 2003
- [135] N. Bhatti, A. Bouch, A. Kuchinsky, "Integrating User–Perceived Quality into Web Server Design. Computer Networks Journal Vol. 33, No. 1-6, pp. 1-16, 2000
- [136] B. Krishnamurthy, C. Wills, "Analysing Factors that Influence End-to-End Web Performance", Computer Networks Journal, Vol. 33, No.1-6, pp. 17-32, 2000
- [137] A. Bouch, A. Kuchinsky, N. Bhatti, "Quality is in the Eye of the Beholder: Meeting Users' Requirements for Internet Quality of Service", In Proceedings of the ACM CHI 2000 Conference on Human Factors in Computing Systems, Hague, Netherlands, 2000
- [138] S. Khirmam, P. Henriksen, "Relationship between Quality of Service and Quality of Experience for Public Internet Service", The 3rd Workshop on Passive and Active Measurement, Fort Collens, Colorado, USA, 2002
- [139] J.B. Chen, M. Perkowitz, "Using End-User Latency to Manage Internet Infrastructure", The 2nd Workshop on Industrial Experiences with Systems Software (WIESS), Boston, MA, 2002
- [140] H. Shubin, M. Meehan, M, "Navigation in Web Applications", ACM Interactions, Vol. 4, No. 6, pp. 13-17, 1997
- [141] J. Ramsay, A. Barbesi, J. Preece, "A Psychological Investigation of Long Retrieval Times on the World Wide Web", Interacting with Computers, Vol. 10, No. 1, pp. 77-86, 1998
- [142] J. Jacko, A. Sears, M. Borella, "The Effect of Network Delay and Media on User Perceptions of Web Resources", Behaviour and Information Technology, No. 19, Vol. 6, pp. 427-439, 2000
- [143] J. Nielsen, "Designing Web Usability", New Riders Publishing, Indianapolis, 2000, ISBN 1-56205-810-X
- [144] A. Bouch, M. A. Sasse, H. DeMeer, "Off Packets and People: A User-Centered Approach to Quality of Service," Proceedings of IWQoS2000, pp. 189-197, 2000
- [145] I. Ceaparu, J. Lazar, K. Bessiere, J. Robinson, B. Schneiderman, "Determining Causes and Severity of End-User Frustration", International Journal of Human-Computer Interaction, No.17, Vol. 3, pp. 333-356, 2004
- [146] P. Selvidge, "Examining Tolerance for Online Delays", Usability News, Vol. 5, 2003
- [147] B. Schneiderman, "Designing the User Interface: Strategies for Effective Human Computer Interaction", MA: Addison-Wesley, 1998

- [148] P. J. Sevcik, "Understanding How Users View Application Performance", *Business Communications Review*, Vol. 32, No. 7, pp. 8-9, 2002
- [149] P. Servidge, "How Long is Too Long to Wait for a Web Site to Load? ", *Usability News*, 1999
- [150] J. Ramsay, A. Barbasi, J. Preece " A Psychological Investigation of Long Retrieval Times on the World Wide Web", *Interacting with Computers*, Elsevier, 1998
- [151] L. Cherkosa, Y. Fu, W. Tang, A. Vahdat, " Measuring End-to End Internet Service Performance: Response Time, Caching Efficiency and QoS", HP Labs Technical Report, 2002, HPL – 2002- 148, 20020605 External, <http://www.hpl.hp.com/techreports/2002/hpl-2002-148.html>
- [152] H. Wu, "A Reference Architecture for Adaptive Hypermedia Applications", Ph.D thesis, Eindhoven University, The Netherlands, 2002
- [153] H. Wu, E. de Kort, P. de Bra, "Design Issues for General-Purpose Adaptive Hypermedia Systems", 12th ACM Conference on Hypertext and Hypermedia (HT01), Arhus, Denmark, 2001
- [154] E. Rich, "User Modeling via Stereotypes", *Cognitive Science Journal*, Vol. 3, No. 4, 1979, pp.329-354
- [155] L. Cherkasova, Y. Fu, W. Tang, A. Vahdat, "Measuring End-to-End Internet Service Performance: Response Time, Caching Efficiency and QoS. HP Labs Technical Report HLP-2002-148, <http://www.hpl.hp.com/techreports/2002/HPL-2002-148.html>
- [156] B. Krishnamurthy, C. Wills, Y. Zhang, "Preliminary Measurements on the Effect of Server Adaptation for Web Content Delivery", In Proceedings of ACM SIGCOMM, The 2nd Internet Measurement Workshop, Marseille, France, Nov. 2002
- [157] L. Cherkosa, Y. Fu, W. Tang, A. Vahdat, " Measuring End-to End Internet Service Performance: Response Time, Caching Efficiency and QoS", HP Labs Technical Report, 2002, HPL – 2002- 148, 20020605 External, <http://www.hpl.hp.com/techreports/2002/hpl-2002-148.html>
- [158] W. Chiu, "Best Practices for High Volume Web Sites", IBM RedBooks, 2001
- [159] J. Nielsen, "Designing Web Usability", New Riders Publishing, Indianapolis, 2000, ISBN 1-56205-810-X
- [160] J. Ramsay, A. Barbasi, J. Preece " A Psychological Investigation of Long Retrieval Times on the World Wide Web", *Interacting with Computers Jurnal*, Elsevier Ed., March, 1998

- [161] Network Simulator - NS-2, <http://www.isi.edu/nsnam/ns/>
- [162] “OTcl - MIT Object Tcl”, Massachusetts Institute of Technology, September 1995, <ftp://ftp.tns.lcs.mit.edu/pub/otcl/README.html>
- [163] K. Fall, K. Varadhan, “The NS Manual (Formerly NS Notes and Documentation)”, VINT Project, <http://www.isi.edu/nsnam/ns/ns-documentation.html>
- [164] J. Chung, M. Claypool, “NS by Example”, Worcester Polytechnic Institute, <http://nile.wpi.edu/NS>
- [165] M. Greis, “Tutorial for the Network Simulator - NS” VINT Project, <http://www.isi.edu/nsnam/ns/tutorial/index.html>
- [166] NSWEB - <http://www.net.informatik.tu-muenchen.de/~jw/nsweb/>
- [167] K. Hook, “Evaluating the Utility and Usability of an Adaptive Hypermedia System”, Proceedings of IUI’97 Conference, Orlando, Florida USA, 1997, pp 179-186
- [168] C. Karagiannidis, D. Sampson, P. Brusilovsky “Layered Evaluation of Adaptive and Personalized Educational Applications and Services”, The 10th International Conference on Artificial Intelligence in Education, Workshop on Assessment Methods in Web-based Learning Environments and Adaptive Hypermedia, San Antonio, Texas, US, 2001.
- [169] P. De Bra, “Pros and Cons of Adaptive Hypermedia in Web-Based Education”, Journal on Cyber Psychology and Behaviour, Vol. 3, No. 1, pp. 71-77, 2000
- [170] P. Brusilovsky, C. Karagiannidis, D. Sampson, “The Benefits of Layered Evaluation of Adaptive Applications and Services”, The 18th International Conference on User Modeling, Workshop on Empirical Evaluations of Adaptive Systems, Sonthofen, Germany, 2001
- [171] P. Brusilovsky, J. Eklund, E. Schwarz, “Web-Based Education for All: A Tool for Developing Adaptive Courseware”, Journal of Computer Networks and ISDN Systems, Vol. 30, No. 1-7, pp. 291-300, 1998
- [172] P. Brusilovsky, “Adaptive Navigation Support in Educational Hypermedia: An Evaluation of the ISIS-Tutor”, Journal of Computing and Information Technology, Vol. 6, No. 1, pp. 27-38, 1998
- [173] M. H. Ng, W. Hall, P. Maier, R. Armstrong, “The Application and Evaluation of Adaptive Hypermedia Techniques in Web-Based Medical Education”, Association for Learning Technology Journal Vol. 10, No. 3, pp. 19-40, 2002
- [174] C. Boyle, A. Encarnacion, “Metadoc: An Adaptive Hypertext Reading System”, User Modeling and User-Adapted Interaction Vol. 4, Kluwer Academic Publishers, pp. 1-19, 1994

- [175] C. Karagiannidis, D. Sampson, "Layered Evaluation of Adaptive Applications and Services", International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2000), Trento, Italy, Springer LNCS 1892, pp. 343-346, 2000
- [176] S. Weibelzahl, G. Weber, "Advantages, Opportunities, and Limits of Empirical Evaluations: Evaluating Adaptive Systems", Künstliche Intelligenz Journal, Vol. 3, pp. 17-20, 2002
- [177] S. Weibelzahl, "Evaluation of Adaptive Systems", In Ma. Bauer, J. P. Gmytrasiewicz, and J. Vassileva, Eds, In Proceedings of the 8th International Conference on User Modeling (UM2001), Berlin, pp. 292–294, 2001
- [178] G. Elissavet, A. A. Economides, "An Evaluation Instrument for Hypermedia Courseware", Journal of International Forum of Educational Technology & Society and IEEE Learning Technology Task Force, Vol. 6, No. 2, pp. 31-44, 2003
- [179] G. D. Magoulas, S. Y. Chen, K. A. Papanikolaou, "Integrating Layered and Heuristic Evaluation for Adaptive Learning Environments", The 9th International Conference on User Modeling, Second Workshop on Empirical Evaluation of Adaptive Systems, Pittsburg, US, pp. 5-14, 2003
- [180] E. Herder, "Utility-Based Evaluation of Adaptive Systems", 9th International Conference on User Modeling (UM2003), Second Workshop on Empirical Evaluation of Adaptive Systems, Pittsburg, US, pp. 25- 30, 2003
- [181] ISO 9241 www.iso.com
- [182] D. Kirkpatrick, "Evaluating Training Programs", San Francisco, CA: Berrett-Koehler Publishers, Inc. 1994
- [183] N. M. Avouris, N. Tselios, E.C. Tatakis, "Development and Evaluation of a Computer-Based Laboratory Teaching Tool", Journal of Computer Applications in Engineering Education, Vol. 9, 2001
- [184] J. M. Heines, "Evaluating the Effect of a Course Web Site on Student Performance", Journal of Computing in Higher Education, Vol.12, No. 1, pp. 57-83, 2000
- [185] N. Tselios, N. Avouris A. Dimitracopoulou, S. Daskalaki, "Evaluation of Distance-Learning Environments: Impact of Usability on Student Performance", AACE International Journal of Educational Telecommunications, Vol. 7, No. 4, pp. 355-378, 2001
- [186] J. Preece, "Online Communities: Designing Usability, Supporting Sociability", published by John Willey & Sons, Chichester UK, 2000
- [187] J. Nielsen, "Heuristic Evaluation. Usability Inspection Methods", Wiley, New York, 1994

- [188] L. C. Jacobs, C. I. Chase, "Developing and Using Tests Effectively: A Guide for Faculty", San Francisco: Jossey-Bass Ed., 1992
- [189] TU/e Course 2L690 Hypermedia Structures and Systems, <http://wwwis.win.tue.nl/2L690/>
- [190] P. De Bra, "Teaching Through Adaptive Hypertext on the WWW", Journal of Educational Telecommunications, Vol. 3, No. 2-3, pp. 163-180, 1997
- [191] AHA! System <http://aha.win.tue.nl/>
- [192] P. De Bra, N. Stash, "AHA! Adaptive Hypermedia for All", SANE 2002 Conference, Maastricht, the Netherlands, pp. 411-412, 2002
- [193] P. De Bra, A. Aerts, D. Smits, N. Stash "AHA! The Next Generation", ACM Conference on Hypertext and Hypermedia, University of Maryland, USA, 2002
- [194] P. De Bra, A. Aerts, D. Smits, N. Stash, "AHA! Version 2.0, More Adaptation Flexibility for Authors", AACE ELearn'2002 Conference, Montreal, Canada, pp. 240-246, 2002
- [195] N. Stash, P. De Bra, "Building Adaptive Presentations with AHA! 2.0", PEG Conference, St. Petersburg, Russia, 2003
- [196] P. De Bra, N. Stash, B. De Lange, "AHA! Adding Adaptive Behavior to Websites", NLUUG Conference, The Netherlands, 2003
- [197] P. De Bra, "Link-Independent Navigation Support in Web-Based Adaptive Hypermedia", Journal of Web Engineering, Vol. 2, No. 1&2, pp. 74-89, Rinton Press, 2003
- [198] P. De Bra, A. Aerts, B. Berden, B. De Lange, B. Rousseau, T. Santic, D. Smits, N. Stash, "AHA! The Adaptive Hypermedia Architecture", ACM Hypertext Conference, Nottingham, UK, pp. 81-84, 2003
- [199] P. De Bra, "Adaptive Educational Hypermedia on the Web", Communications of the ACM, Vol. 45, No. 5, pp. 60-61, 2002
- [200] P. De Bra, A. Aerts, D. Smits, N. Stash, "AHA! Meets AHAM", Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH2002, Springer LNCS 2347, Malaga, Spain, pp. 381-384, 2002
- [201] C. Bailey, W. Hall, D. E. Millard, M. J. Weal, "Towards Open Adaptive Hypermedia", Proceedings of 2nd International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, Malaga, Spain, pp.36-46, 2002
- [202] N. Henze, W. Nejdl, "Knowledge Modelling For Open Adaptive Hypermedia", Proceedings of Proceedings of 2nd International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, Malaga, Spain, pp. 174-183, 2002