Intelligent User Modeling and User Adapted Interactions for eCommerce Consumers

By

Oshadi Vishanthi Rupika Alahakoon

BSc (Hons)

Thesis submitted in fulfillment of the requirements for the degree of

Doctor of Philosophy

Caulfield School of Information Technology

Monash University

November 2008

Abstract

In eCommerce vendors and consumers do not physically meet each other. As a result traditional communication channels between the consumer and the vendor have become obsolete or insufficient. To improve the understanding between consumers and vendors, user modeling and personalization has become an important and essential feature in online markets.

This thesis investigates, proposes and validates the development of a more complete, consistent and coherent model of user behavior rather than a partial view. This research separates information about the user into three main categories: general purchase behavior characteristics, domain centric purchase behavior and impulsive behavior in purchasing. It is argued that each individual user's behavior is unique, and this uniqueness could be represented by user specific combinations of these three categories. Based on these three "layers" of information categories, this thesis proposes a novel user model architecture called the Layered User Model (*LUM*).

An important factor in establishing the value of a user model is its ability to improve the interactions with the users. This should be achieved by increasing the benefits of personalization, and by reducing the obtrusiveness of the system. The thesis proposes a new online interactive product retrieval algorithm, *PIPRP* (Personalized Interactive Product Retrieval Process) to demonstrate the applicability and benefits of the new layered user model. The interactive product retrieval algorithm employ the *LUM* to successfully handle the common online product retrieval problems such as null retrievals, retrieving unmanageable number of items and retrieving unsatisfactory items.

Furthermore, the work presented in this thesis supports less obtrusive system-user interactions in both user model creation and in interactive product retrieval processes. The layered user model supports information reuse and has the ability to update and learn new user preferences without user intervention, thereby reducing the user effort required to create the model. The user model based product retrieval algorithm employs the *LUM* to provide personalization, and so reduces the user effort required in specifying user preferences in product retrievals. Therefore, we claim that the approach taken in this thesis

is capable of maintaining the balance between the benefits of personalization and the user effort.

The thesis also proposes, validates and justifies a new set of criteria for evaluating the user model in product retrieval. Several datasets are used in the evaluation and the results are discussed in the thesis.

Declaration

This thesis contains no material that has been accepted for the award of other degree or diploma in any university or other institution. To the best of my knowledge, this thesis contains no material previously published or written by any other person except where the due reference is made in the text of the thesis.

O.V.R Alahakoon

Acknowledgements

I am grateful to my supervisors Assoc. Prof. Arkady Zaslavsky and Dr. Seng Wai Loke, for their able guidance and intellectual support throughout my research work. I thank Arkady, and really appreciate his generosity and management quality demonstrated during this period. I thank Seng, and really appreciate the interest he showed in carefully reading my thesis.

I thankfully acknowledge the support for this project from the Australian Research Council (ARC) Linkage Project scheme and Microsoft Research Asia Grant LP0211384.

I take this chance to thank Prof. Liz Sonenberg for initially introducing me to research. The grounding I received gave a solid background to build up my PhD work. I would like to express my gratitude to Mr. Robert Redpath for introducing me to a teaching career.

During the PhD candidature there were many people who helped me greatly. I specially thank Resmi Nair, Laily Hashim and Wanita Sherchan who shared the office with me in different periods of time during my candidature. I greatly appreciate their friendship kindness and tolerance (especially, whenever I brought my children into the office). I also thank my colleagues Eddie Leung, Flora Salim, Glenn Jayaputara, Kutila Gunasekara, Pari Haghighi, Ruwini Kodikara and Samar Zutshi for their friendship and support. I thank Duke Fonias and See Ngieng for their technical support. I thank Akamon Kunkongkapun and Rosemary Demirtas for helping with financial issues and travel matters. I also thank Allison Mitchell for her support as the post graduate administration officer.

I thank my parents-in-law, for their help in looking after the children in difficult times of my candidature. I thank my brother Ravindra and his wife Nishadi, and my sister Tanya and her husband Mahesh for their help during the past years.

I thank my husband Damminda for his love, kindness and tolerance during my long PhD years. I am grateful to him for being an unofficial thesis supervisor, mentor, and for all the sacrifices he made for the completion of this thesis.

Last but certainly not least, I thank my three little children Navya, Laksha and Dakshina for their love and understanding during the past several years. For me, they are the epitome of true happiness.

Publications and Thesis Work Outcomes

Book Chapters

Alahakoon, O., Loke, S. and Zaslavsky, A. (2007) ‘Use of Soft Computing Techniques to Enhance the Business Value in User Models’, In Book Chapter, Soft Computing Applications in Business, (Prasad, B., Eds.), Springer-Verlag, Germany. – Accepted. To be published in 2008/09.

Journal Papers

Alahakoon, O., Loke, S. and Zaslavsky, A. (2008), ‘A Layered User Model for eCommerce Applications’, Special Issue on Web Mining for E-commerce and E-services, Online Information Review – Accepted. To be published in 2009.

Refereed Conference Publications

Zaslavsky, A., Bartolini, C., Boulmakoul, A., **Alahakoon, O.**, Loke, S. W. and Burstein, F., (2007), ‘Enhancing the IT service desk function through unobtrusive user profiling, personalization and stereotyping’, HP Software University Association, 14th Workshop, Garching/Munich, Germany.

Alahakoon, O., Loke, S. and Zaslavsky, A. (2007) ‘Capturing Buying Behavior Using a Layered User Model’, In Proceedings of 8th International Conference on E-Commerce and Web Technology, Psaila G. and Wagner R., (Eds.), LNCS, 4655, Spinger, pp. 109-112.

Alahakoon, O., (2006). ‘Use of Buying Behavioral Characteristics as General Stereotypes in User Models’. Proceedings of the 2nd International Conference on Information and Automation (ICIA 06), IEEE computer society, Colombo, Sri Lanka.

Alahakoon, O., Loke, S. and Zaslavsky, A., (2006). 'Use of Buying Behavioral Characteristics in Long Term User Models'. Proceedings of the 2nd International Workshop on Web Personalization, Recommender Systems and Intelligent User Interfaces (WPRSIUI 2006) at the 4th International Conference on Adaptive Hypermedia (AH 2006). Dublin, Ireland, June 2006.

Alahakoon, O., Loke, S. and Zaslavsky, A., (2005). 'Personalization across De-centralized Domains with a Modular User Profile Architecture'. Proceedings of the 2nd International Conference on Information and Automation (ICIA 05), IEEE computer society, Colombo, Sri Lanka.

Alahakoon, O., Loke, S. and Zaslavsky, A., (2004). 'Unobtrusive acquisition of user information for e - commerce applications'. In proceedings of the sixth International Conference on Enterprise Information Systems (ICEIS 2004), Porto, Portugal. V5, pp. 3-8.

Alahakoon, O., Loke, S. and Zaslavsky, A., (2003).'Modular and hierarchical user profile instantiation for e-commerce applications using mobile software agents'. In the proceedings of 14th Australasian Conference on Information Systems (ACIS 03), Perth, Australia. pp. 26-28.

Jayaputera, G., **Alahakoon, O**., Cruz, L., Loke, S. and Zaslavsky, A. (2003) 'Assembling Agents On-Demand for Pervasive Wireless Services', In The Second International Workshop on Wireless Information Services, (Mahmoud, Q. H., Eds.), ICEIS Press, Angers-France, pp. 21-30.

Alahakoon, O. and Alahakoon, D., (2002). 'A Framework for a Data Movement Monitoring Agent Using Dynamic Feature Maps'. Proceedings of the 1st International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2002), Singapore. 2, pp 641-645.

Presentations by the Author

Unobtrusive Personalization for E-Buyers using Layered User Profiles, Smart Internet Technology Research Group seminar Series, 19 October 2005, University of Sydney, Australia.

Personalization across De-centralized Domains with a Modular User Profile Architecture. The 2nd International Conference on Information and Automation (ICIA 05), 12 December 2005, Bandaranaike Memorial Hall, Colombo, Sri Lanka.

Capturing Buying Behaviour Using a Layered User Model, 8th International Conference on E-Commerce and Web Technology (ECWEB07), 6 September 2007, University of Regensburg, Regensburg, Germany.

Glossary

AI	Artificial Intelligence
ATR	Average Total Relevance
Attribute	Different values that an item feature can take, such as price ranges for feature 'cost'
CBR	Case Based Reasoning
DI Layer	Domain Information Layer
Domain	A domain can be a specific subject area of knowledge such as restaurants, leg-wear, real estate etc.
IM	Influence Matrix
IT	Information Technology
Feature	Items in a domain can be described using its features such as 'cost'.
LUM	Layered User Model
MAUT	Multi-Attribute Utility Theory
MR	Market Research
PBC values	Purchase Behavior Characteristics
Purchase Behavior Characteristics	Users general behavior in purchasing irrespective of the purchase domain.
Personal Information	Demographic Information such as date of birth, education, occupation,

PI Layer	Personal Information Layer
PIPRP	Personalized Interactive Product Retrieval Process
PIR-Attributes	Personal Information Related Attributes
Purchasing Behavior	User characteristics such as price sensitivity related to purchasing
TFIDF	Term Frequency Inverse Document Frequency
TI Layer	Transaction Information Layer
User	Person who interacts with the system, could be a consumer

Table of Contents

ABSTRACT ………………………………………………………………………………………….. **II**

DECLARATION ………………………………………………………………………………. **IV**

ACKNOWLEDGEMENTS ……………………………………………………………………………… **V**

PUBLICATIONS BY THE AUTHOR ……………………………………………………………. **VI**

GLOSSARY ……………………………………………………………………………………….. **IX**

LIST OF FIGURES ……………………………………………………………………………. **XVII**

LIST OF TABLES ……………………………………………………………………………….... **XX**

1 INTRODUCTION **1**

1.1 eCOMMERCE AND USER MODELING 3
1.2 MOTIVATION 4
1.3 OBJECTIVES OF THE THESIS 9
1.4 CONTRIBUTIONS OF THE THESIS 10
1.5 OUTLINE OF THE THESIS 12

2 A REVIEW OF PERSONALIZATION AND USER MODELING **14**

2.1 USER MODELING SYSTEMS 15
2.2 USER MODELING AND APPLICATION AREAS 18
2.3 DIMENSIONS OF USER MODELS 19
2.3.1 Canonical or individual user 19
2.3.2 Explicit or implicit information 20
2.3.3 Short or long-term model 20
2.3.4 Static or dynamic model 21
2.4 METHODS FOR GATHERING INFORMATION ABOUT THE USER 21
2.4.1 Explicit user inputs 22
2.4.2 Web searching 22
2.4.3 Stereotypes 23
2.4.4 Observing user behavior 26
2.5 CONTENTS OF USER MODELS 29
2.6 USER MODELING TECHNIQUES 30
2.6.1 Collaborative filtering 32

2.6.2 *Content based filtering* *38*
2.7 ALGORITHMS AND QUANTITATIVE TECHNIQUES 40
2.7.1 *Linear methods* *40*
2.7.2 *TFIDF- based models* *41*
2.7.3 *Markov chain models* *41*
2.7.4 *Neural networks* *41*
2.7.5 *Classification methods* *42*
2.7.6 *Rule induction methods* *42*
2.7.7 *Bayesian network* *42*
2.8 OBTRUSIVENESS IN PERSONALIZATION 43
2.8.1 *Number of Questions* *44*
2.8.2 *Comprehensibility of Questions* *44*
2.8.3 *Answering Cost Related to Questions* *45*
2.8.4 *Comprehensibility of Question Clusters* *45*
2.8.5 *Personal Information Content of the Questions* *45*
2.9 ELECTRONIC USER MODELS IN ECOMMERCE 47
2.9.1 *Prototype Systems* *48*
2.9.2 *Commercial Systems* *50*
2.10 ISSUES AND LIMITATIONS 53
2.10.1 *Reusability of the user model in multiple domains* *53*
2.10.2 *Necessity of knowledge engineering* *55*
2.10.3 *System adaptability* *55*
2.10.4 *Flexibility in capturing user preferences* *56*
2.10.5 *Use of personal or demographic data* *58*
2.10.6 *Sensitivity to individuality* *59*
2.11 LIMITATIONS OF USER MODELS IN ECOMMERCE 60
2.12 SUMMARY 61

3 A LAYERED USER MODEL FOR INTERPRETING CONSUMER PURCHASE BEHAVIOR 62

3.1 COMPLEXITY OF CONSUMER BEHAVIOR 63
3.2 A THREE LAYERED MODEL TO REPRESENT CONSUMER/USER BEHAVIOR 65
3.3 ANALYSIS AND CAPTURING CONSUMER PURCHASE BEHAVIOR 68
3.3.1 *Consumer Behavior Analysis and user models in MR Based Research* *68*
3.3.2 *Consumer Behavior Analysis and Models in Information Technology Based Background* *73*
3.4 BEHAVIORAL CATEGORIES AND RELATED USER INFORMATION 76
3.5 MODELING PERSONAL INFORMATION BASED PURCHASE BEHAVIOR 77
3.5.1 *Purchase Behavior Characteristics (PBC)* *80*
3.5.2 *Use of Demographics in Value Functions* *82*

3.6 MODELING DOMAIN CENTRIC PURCHASE BEHAVIOR 83
3.7 SUMMARY 86

4 CONCEPTUAL ARCHITECTURE OF THE LAYERED USER MODEL 89

4.1 THE OVERALL eHERMES ARCHITECTURE 90
4.2 THE PERSONALIZATION COMPONENT-eHERMES PERSONAL 92
4.3 THE ARCHITECTURE OF THE LUM 94
4.3.1 The user model 95
4.3.2 Domain hierarchies and n-layered user model 97
4.4 THE PERSONAL INFORMATION LAYER (PI LAYER) 100
4.4.1 Identification Information 100
4.4.2 Personal Information 101
4.4.3 Information on Buying Characteristics 104
4.5 DOMAIN INFORMATION LAYER (DI LAYER) 105
4.5.1 Identification Information 106
4.5.2 Product features and Attributes 106
4.5.3 Relevance Values 107
4.6 TRANSACTION INFORMATION LAYER (TI LAYER) 109
4.7 UPDATING THE USER MODEL 111
4.7.1 Update after each transaction 111
4.7.2 Long-term buying characteristic update 112
4.8 OBTRUSIVENESS IN PERSONALIZATION 113
4.8.1 During creation of the user model 113
4.8.2 During the PIPRP 113
4.9 SUMMARY 114

5 IMPLEMENTATION OF THE LAYERED USER MODEL 116

5.1 USER MODEL GENERATION PROCESS 117
5.1.1 The Algorithm 117
5.1.2 Flow Chart of the Algorithm 118
5.2 CALCULATION OF USER PBC VALUES USING MAUT 119
5.2.1 Multi-Attribute Utility Theory (MAUT) 119
5.2.2 Derivation of the Formula 120
5.2.3 Value Functions for PBC Values 121
5.3 USE OF PBC VALUES AS GENERAL STEREOTYPES 127
5.4 THE INFLUENCE MATRIX (IM) 131
5.5 THE DOMAIN BASED INITIAL PREFERENCES 134

5.5.1 Assigning Existing Relevance Value ... *134*
5.5.2 Assigning Calculated Relevance Values ... *135*
5.6 ALGORITHM FOR CALCULATING INITIAL RELEVANCE VALUES ... 136
5.6.1 Fuzzy Logic ... *137*
5.6.2 Use of Fuzzy Logic in Relevance Value Calculation ... *137*
5.7 AVERAGE TOTAL RELEVANCE (ATR) VALUES OF FEATURES ... 144
5.8 UPDATES IN THE USER MODEL ... 145
5.8.1 Hebbian Learning ... *145*
5.8.2 Application of Hebbian Learning in Update from TI Layer to DI Layer ... *146*
5.8.3 Application of Hebbian Learning in Update from DI Layers to PI Layer ... *148*
5.9 SUMMARY ... 152

6 FUNCTIONALITY, VALUE AND USEFULNESS OF THE LAYERED USER MODEL ... 154

6.1 IMPLEMENTATION ENVIRONMENT ... 154
6.1.1 Platform, Tools and Data Sets ... *154*
6.1.2 Experiment Preparation ... *157*
6.2 FUNCTIONALITY OF THE LAYERED USER MODEL ... 158
6.2.1 Calculation of user's PBC values - (Start-up PI layer) ... *158*
6.2.2 Initializing user's domain centric buying behavior - (Start-up DI layer) ... *163*
6.2.3 Capturing user's transaction specific buying behavior - (TI layer) ... *164*
6.2.4 Updating the user model after each transaction - (DI layer based on TI layers) ... *167*
6.2.5 Long-term update of buying characteristics - (PI layer based on DI layers) ... *169*
6.3 VALUE OF THE LAYERED MODEL ... 171
6.3.1 Ability of the model to capture individuality of the user ... *171*
6.3.2 Ability to provide personalized services from the initial interaction ... *172*
6.3.3 Ability to handle the dynamic nature of the product markets ... *173*
6.3.4 Ability to capture the individuality of the user ... *173*
6.3.5 Ability to provide personalization in multiple domains ... *175*
6.3.6 Ability to enforce control over irregular transactions ... *175*
6.3.7 Ability to support user scrutinize interfaces ... *175*
6.4 USEFULNESS OF THE USER MODEL ... 176
6.4.1 Trend analysis for the individual ... *177*
6.4.2 Trend analysis for the entire user population ... *177*
6.4.3 Usability in target advertising ... *178*
6.5 SUMMARY ... 179

7 LAYERED USER MODEL FOR PERSONALIZED INTERACTIVE PRODUCT RETRIEVAL 180

7.1 RELATED WORK ... 181
7.1.1 Online Product Retrieval in eCommerce ... 181
7.1.2 Searching Methods ... 184
7.2 THE PURCHASING PROCESS IN ECOMMERCE ... 188
7.2.1 Requirement Elicitation ... 188
7.2.2 Product Search ... 189
7.2.3 Product Presentation ... 189
7.2.4 Problems Encountered ... 190
7.3 A NEW TECHNIQUE FOR PERSONALIZED PRODUCT RETRIEVAL ... 192
7.3.1 PIPRP - Algorithm ... 192
7.3.2 Similarity Calculation ... 195
7.3.3 Example of Similarity Calculation ... 197
7.4 EXPERIMENTATION ... 198
7.4.1 Experimental Plan ... 199
7.4.2 Demonstration of Algorithm Steps ... 201
7.4.3 Step 1: Personalized Initial Query Specification ... 201
7.4.4 Step 2: Personalized Query Expansion ... 205
7.4.5 Step 3: Further Filtering Based on the User Model – Personalized Filtering ... 208
7.4.6 Step 4: Further Query the User ... 211
7.4.7 Step 5: Calculate Similarity between the Query and the Items ... 214
7.4.8 Results ... 216
7.4.9 Comparison of results ... 219
7.5 EVALUATION OF EHERMES PERSONAL ... 221
7.5.1 Background ... 222
7.5.2 Evaluation of User Model Accuracy ... 222
7.5.3 Evaluation of Online Product Selection ... 223
7.5.4 Proposed Evaluation Criterion ... 224
7.5.5 Evaluation Using the Proposed Criterion ... 225
7.6 SUMMARY ... 233

8 CONCLUSION AND FUTURE WORK ... 236

8.1 RESEARCH SUMMARY ... 236
8.2 SUMMARY OF CONTRIBUTIONS ... 238
8.2.1 Justification of the three layered architecture ... 238
8.2.2 Design and implementation of the novel layered user model ... 239
8.2.3 The new product retrieval algorithm ... 240

8.2.4 Evaluation of LUM and PIPRP *241*
8.3 FUTURE WORK 241

BIBLIOGRAPHY **244**

APPENDIX A **252**

APPENDIX B **260**

List of Figures

Figure 1.1: Changes in user behavior observed in volatile eCommerce markets 6
Figure 2.1: Domains where user models were employed to provide personalization.......... 18
Figure 2.2 : A stereotype in the family of life styles in SeAN – reproduced from (Ardissono *et. al.*, 2001a) requested permission 24
Figure 2.3 : A stereotype in SeAN– reproduced from (Ardissono *et. al.*, 2001a) requested permission 24
Figure 2.4 : "Housewife" stereotype in TV domain, – reproduced from Ardissono *et. al.*, (2004) requested permission 25
Figure 2.5 : Categorization of User Modeling Techniques 31
Figure 2.6: question types – reproduced from Spiekermann *et. al.*, 2001 with permission. 46
Figure 2.7 : Categories and examples of eCommerce systems 48
Figure 2.8: A mapping of the user model design considerations and approaches 61
Figure 3.1: Information contribution to the model from MR and IT areas of research 67
Figure 3.2: Roy Morgan Value Segments^{TM}* (reproduced from URL: http://www.roymorgan.com/products/values-segments/values-segments.cfm with permission) 71
Figure 3.3 : Consumer segments (S_1, S_2, S_3, …) formed in MR user models...................... 72
Figure 3.4 : Domain based user preferences captured in IT user models 74
Figure 3.5 : Key benefits in each Unilever category- reproduced from Patel and Schlijper (2003) with permission.. 79
Figure 4.1: eHermes Architecture 91
Figure 4.2: eHermes Personalization Component – Overview of the Architecture............. 92
Figure 4.3 : The Overall Architecture of the eHermes personalization component 93
Figure 4.4 : The main screen – eHermes PERSONAL 95
Figure 4.5 : The login form screen 95
Figure 4.6: Three layered User model architecture 96
Figure 4.7: The expanded user model to demonstrate the n- layers.................................... 99
Figure 4.8: Number of domains inter connected as a hierarchy.. 100

Figure 4.9: Registration Screen 101
Figure 4.10 : PI layer of the user model belonging to user 'John Smith'. 102
Figure 4.11 : Selecting the domain out of a collection of domains.................................. 105
Figure 4.12 : Sections of the DI layer of the same user "John Smith"............................. 106
Figure 4.13 : Sections of TI layer of "John Smith" for transaction "TxnId =1"............... 110
Figure 4.14 : Obtrusiveness of the information along layers of the user model 113
Figure 5.1 : The user model generation process illustrated as a flow chart 118
Figure 5.2 : Three step mapping of general stereotypes 129
Figure 5.3 : Steps involved in traditional stereotyping 130
Figure 5.4 : User needs are fulfilled by product attributes 131
Figure 5.5 : More than one PBC is influencing an attribute 135
Figure 5.6 : T-membership function corresponding to above function definition 138
Figure 5.7 : Four membership functions for cost attributes based on Price Sensitivity..... 140
Figure 5.8 : Six relevance fuzzy sets for Quality Consciousness..................................... 141
Figure 5.9 : Five relevance fuzzy sets for Quality Consciousness for "Service"............... 141
Figure 5.10 : Two relevant fuzzy sets for 'Adventurer' for the feature leg-wear colour... 143
Figure 5.11: Influence of a PBC over a discrete attribute 144
Figure 5.12 : Structure of a Hebbian network 145
Figure 5.13 : Feature as a single node in the output layer, attributes in the input layer..... 147
Figure 5.14 : Feature 'Cost' as the output layer and the four attributes as the input layer 148
Figure 5.15 : Process of PBC value update 149
Figure 5.16 : Allocating the initial relevance to individual attributes............................... 150
Figure 5.17 : PBC Chl influences the attributes of the feature 151
Figure 5.18 : Example calculation of new PBC 152
Figure 6.1: Hierarchically organized product attributes... 156
Figure 6.2: Characteristics of six demographically different individuals 160
Figure 6.3: Eight demographically similar individuals with varying age 161
Figure 6.4: Individuals with same family type and gender 162
Figure 6.5 : Initial relevance values calculated for the restaurants for User41 164
Figure 6.6 : Initial relevance values calculated for the leg- wear domain for User41 164
Figure 6.7: User preferences towards feature/attribute 1/1, 1/2 and 1/4 166
Figure 6.8 : User preferences towards feature/attribute 2/6, 3/31 and 4/32 166

Figure 6.9 : User preferences towards feature/attribute 5/36, 10/115 and 12/121 167
Figure 6.10 : Attribute relevance values from the DI layer for Restaurants 168
Figure 6.11 : Attribute relevance values from the DI layer of leg-wear 168
Figure 6.12 : Initial and recalculated PBC values 170
Figure 6.13 : Re-calculated domain based PBC values and the initially calculated vales . 170
Figure 6.14: Highest to lowest abstraction, when identifying user behavior 172
Figure 6.15 : Initial and after transaction relevance values - User 29160 and User 2240 . 174
Figure 6.16: Consumer types based on the PBC values 178
Figure 7.1 : Product retrieval methods in eCommerce 181
Figure 7.2 : Categorization of product retrieval methods 185
Figure 7.3 : Sets of records resulting from each step 193
Figure 7.4 : Product selection algorithm as a flow chart 194
Figure 7.5 : Comparison of four different items IA, IB, IC, and ID to a query Qu 197
Figure 7.6 : Screen – Users initial query selection 202
Figure 7.7: Screen – Interactive product selection 203
Figure 7.8 : Steps involved in personalized query expansion 205
Figure 7.9 : Records retrieved using the expanded query 207
Figure 7.10 : Records available after personalized filtering 210
Figure 7.11 : Feature selection for further querying 211
Figure 7.12 : Filtered according to ‘Popularity’, constraint 212
Figure 7.13 : Filtered according to ‘Location’ constraint 213
Figure 7.14 : Displaying a selected Restaurant 213
Figure 7.15 : Item display screen 214
Figure 7.16 : Final outcome for User41’s scenario a1 215
Figure 7.17 : Message - start of Step 4 with the record set resulted after Step 3 216
Figure 7.18 : ATR values calculated for User41’s DI layer after the scenario 41-e 227
Figure 7.19 : User preference for the restaurant category 228
Figure 7.20 : Result retrieved for the above query 230

List of Tables

Table 2.1: Well known existing user modeling systems 16

Table 2.2 : Information elicitation methods 27

Table 2.3: Information content in user models in existing well known systems 29

Table 2.4: User modeling techniques in existing well known systems 32

Table 3.1: Val Segments and their descriptions 80

Table 3.2 : Roy Morgan Value Segments and their descriptions 82

Table 4.1: Explanation of Figure 3 94

Table 4.2: A sample set of data with descriptions 107

Table 4.3 : Summary of the layers of the user model 114

Table 5.1: PBC s and contributing demographics 121

Table 5.2 : Attribute Codes for FT 123

Table 5.3: Attribute Codes for WH 123

Table 5.4: Attribute Codes for G 123

Table 5.5: Attribute Codes for Occ 123

Table 5.6: Attribute Codes for AG 123

Table 5.7: Effect of PBC on relevance of product/item attributes 127

Table 5.8: A section of the Influence Matrix for Restaurants domain 133

Table 5.9: Section of the Influence Matrix for leg-wear domain 133

Table 5.10 : A section of the $IM_{restaurant}$ - non-PIR-attributes 134

Table 5.11 : Discrete Features and their Attributes 136

Table 5.12: Attribute values for an item priced $20 136

Table 5.13: Ranges and equations of each membership function –"Cost" 139

Table 5.14: Crisp and Fuzzy representations of relevance values 139

Table 5.15: Ranges and equations for membership functions "Décor" 140

Table 5.16: Range of each membership function/attribute 142

Table 5.17: Range of each membership function/attribute 143

Table 5.18 : Relevance values corresponding to different stages in the retrieval process . 146

Table 6.1: Restaurant transactions for User 41 159

Table 6.2 : Leg-wear transactions for User 41 159

Table 6.3: Information in *PI* Layer of the user model belonging to six individuals 160

Table 6.4 : *PI* layer of individuals having the same demographics other than the age...... 161
Table 6.5: *PI* layer of individuals having the same family type and of the same gender .. 162
Table 6.6 : Relevance value change after each transaction, for nine attributes 165
Table 6.7 : The initial, updated based on the domain and current PBC values.................. 169
Table 6.8: Similar demographics resulted in identical *PBC* values 174

Chapter 1

Introduction

The tremendous growth of the Web has resulted in the accumulation of a massive amount of electronic data and information. As the volume of online information grows with an astonishing speed, users find retrieving the information they need a challenging task. Although there are sophisticated search engines available, Web searchers can still end up with loads of information which are mostly irrelevant. Search engines use either human made or automatically generated indices for Web information retrieval. But since the Web is a large collection of unorganized information, indexing everything becomes a difficult task. These issues result in not only information overload, but also a high possibility of not retrieving what is sought by the user. Therefore, it is clear that a search engine alone cannot deal with the large and complex source of available information.

Search engines look at the data overload problem only from the point of view of data. Therefore, no matter how sophisticated the search engine is, still users have to clearly specify their needs every time they search for information. Search engines treat both the expert frequent user and the novel user in the same manner. Each time a user visits a website, the same information is gathered during a similar search, disregarding previously provided information. This gives rise to the need for a method of recording and maintaining previously captured user data for reuse. Such user information containing knowledge about the user's characteristics, past and present needs and interests can be used to form a model of the user's behavior and preferences. Thinking along these lines there has been many personalization efforts described in the literature which combine user models with retrieval strategies of search engines. It has been demonstrated that more meaningful and relevant retrievals are possible by incorporating a "profile" or a "model" of the user with his/her search preferences. However, it can be seen from these efforts that handling user data is a bigger challenge than product retrieval using product descriptions alone.

With the advancement of technology, the Internet has become a common information source for people from different backgrounds. Among these users, differences such as social and educational backgrounds and personal preferences are pervasive. Catering for information needs of such a diverse user population is an additional challenge. It is observed (in the past work) that user behavior varies not only from individual to individual but even for the same individual based on the domains they interact (or purchase) within. Therefore, for effective personalized services, each individual should be separately modeled: if possible within each different domain they seek personalization in.

There is a range of definitions in the literature for a "user model" or a "user profile". According to Ross (2000),[1] more general definition for a user profile could be as follows.

> "*A user profile is an explicit representation of the properties of an individual user; it can be used to reason about the needs, preferences or future behavior of the user*"

Work on user modeling and personalization dates back to the 1970s. Elaine Rich developed one of the first electronic systems called GRUNDY (Rich, 1979;1983;1989) to recommend books out of a large collection of fiction. The initial application dependent user modeling capabilities were built into the application system itself. As mentioned in Kobsa (2001) from the mid-eighties there was a trend toward more generic user modeling systems. Then in the late 1990s several commercial user modeling systems based on client server architecture were born. These systems have the capability of serving several users at the same time.

During the past years, personalization has been provided to users in several different domains for different purposes. These domains can be categorized into several major areas such as information retrieval, eCommerce, entertainment recommendations, and adaptive hypermedia systems. In the information retrieval domain user models were employed to

---

[1] Intelligent User Interfaces: Survey and Research Directions. *Technical Report CSTR-00-004, Department of Computer Science, University of Bristol.*

avoid information overload. In the area of eCommerce, to attract users to websites it is important to provide interactions specially tailored to them. A user model also becomes useful when finding out the preferred products out of large collections of items, and for targeted advertising purposes. In the entertainment domain where massive collections of CDs, or movies are available, a user model is used to recommend the items that users may like. Finally in expert systems, student models have been used to provide users with personalized guidance in learning. This avoids situations such as providing an expert user with instructions suitable for a beginner.

To fulfill the requirements of a diverse user population and to manage the information explosion in product and services related issues, user modeling and personalization techniques look very promising. Already in a number of domains, user adaptive systems have proven to perform well compared to non adaptive systems (e.g. Amazon.com in book recommendations). The current technological trends may position personalization and user modeling as an essential part in most of the electronic systems; especially eCommerce, where a growing number of new products and the massive number of online consumers can become impossible to manage without personalization. The next section provides a glimpse of the role of personalization and user modeling particularly in eCommerce systems.

1.1 eCommerce and User Modeling

With technological growth eCommerce has become part of every day life for many users. Purchasing and selling online is familiar to many users in well known websites such as Amazon.com[2], eBay[3], or Movie finder[4]. Vendors tend to form electronic Web stores due to their low overheads and large customer base (the whole world can visit the store!). Consumers are attracted to Web stores since they get to navigate through a large selection of items, without physically visiting the store. But at the same time they can find navigating a massive product base challenging and finding the best suited products almost impossible. The main reason for this is that the consumer and the vendor do not physically meet each other. This leads to a lack of product knowledge on the consumer's side and a lack of consumer understanding from the vendor's perspective. The consumers find it difficult to obtain the items they search for, and the vendors find it difficult to spot the suitable

[2] Amazon.com
[3] eBay ebay.com
[4] EOnline! (http://www.eonline.com/movies/index.jsp)

individuals or consumer segment to market their products. In such an environment user modeling is extremely important for eCommerce applications. A vendor can use the model of the user to understand user needs, and use appropriate marketing strategies to sell the right product to the right customer at the right time. At the same time, a consumer is able to search for his/her requirements in massive product bases with less effort. In addition users are able to store information about themselves and their recurring needs, without having to provide them repeatedly every time they visit a Web store.

An eCommerce site can greatly benefit by personalizing their product offerings, sales promotions, product news, advertisements banners etc. According to Fink and Kobsa (2000) it has been reported that eCommerce sites offering personalization perform well in drawing new users to the site, turning visitors into buyers, thereby increasing revenue, increasing advertising efficiency, and improving customer retention rate and brand loyalty. There are issues such as privacy concerns and trustworthiness that are closely coupled with personalization and eCommerce. Handling and managing such diverse issues and at the same time making personalization possible is a challenging task.

1.2 Motivation

A brief background of user modeling and personalization in general as well as in eCommerce, was provided in the previous sections. The special requirements of the field that are inherent to eCommerce were also discussed. Furthermore, in the analysis, the necessary requirements for the success of user modeling and personalization in the current as well as future eCommerce environment were highlighted.

As mentioned above, personalized user interactions are invaluable in the growing electronic market. However, to provide personalized interactions, well organized product and user data is required. In other words, for effective personalization, it is required to model both users and products. Out of the two, modeling consumers can be argued as the greater challenge due to the complexity of human buying behavior. In addition, the heterogeneous user populations demand the modeling of each individual instead of a “one size fit all” strategy. Each consumer demonstrates individual behavior in different product domains, where each domain has its own characteristics. Domain characteristics indirectly contribute to differentiations in user behavior. To capture such complex human buying behavior there is a need of a user model which has the ability to capture the different dimensions of consumer behaviors.

Each user visiting an eCommerce website leaves important information such as product navigational data and purchase history. If this data is effectively utilized to provide the users with personalization it can be beneficial throughout the customer lifecycle. However, to provide such effective personalization, current user models urgently require overcoming certain challenges. Sierra and Dignum (2001) describes four such challenges faced by current user modeling research.

(i) capturing dynamics of preferences,
(ii) handling different ontologies,
(iii) representing fuzzy preference, and
(iv) need for techniques to learn or capture preferences.

Capturing dynamics of preferences

User preferences change rapidly within the eCommerce market due to the number of new products, media and effective advertising. The user models should have the ability to capture such changing needs of the consumer. According to the previous discussions, changes in user behavior observed in volatile eCommerce markets can be shown as in Figure 1.1. As such, user behavior is expected to change with the change of demographics, or even for the same user as the domain changes or for the same user in same domain for different transactions.

Handling different ontologies

Different websites use their own ontologies to characterize the consumer interests and products. One way of handling this situation is by introducing a common ontology for all websites. The semantic Web is a concept which attempts to find such a solution. Wikipedia[5] defines the semantic Web as follows.

> *The semantic Web is an evolving extension of the World Wide Web in which Web content can be expressed not only in natural language, but also in a format that*

[5] http://en.wikipedia.org/wiki/Semantic_Web

can be read and used by software agents, thus permitting them to find, share and integrate information more easily.

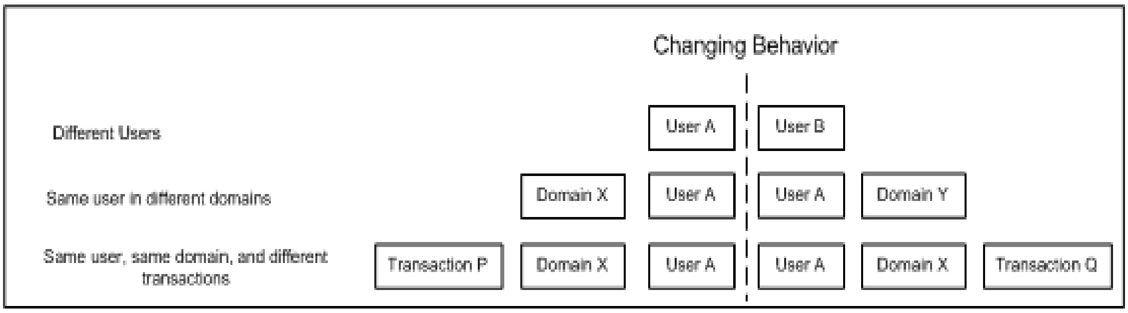

Figure 1.1 : Changes in user behavior observed in volatile eCommerce markets

It is the vision of the World Wide Web Consortium that the Web as a universal medium for data, information, and knowledge exchange. (The World Wide Web Consortium (W3C) is the main international standards organization for the World Wide Web). An alternative would be guiding the user to select preferences out of a set of options which will reduce the chance of confusion between user's own vocabulary and that of the system. But such a system will have to pay the price of knowledge engineering.

Representing fuzzy preference

User preferences towards products or their characteristics are by no means clear cut but fuzzy (e.g. uninteresting, interesting, very interesting). When modeling the users, this is another important issue that requires attention. If user preferences are measured quantitatively rather than qualitatively even the slightest difference of preferences among two users are identifiable. In Sierra and Dignum (2001) use of fuzzy logic and statistics is proposed as a solution when capturing user preferences accurately. Fuzzy logic, (Zadeh, 1965) a modified set theory in which an individual could have a degree of membership which ranged over continuum of values rather than being either '1' (true) or '0' (false). Instead of crisp boundaries imposed by conventional binary logic it uses fuzzy membership functions to assign a value of vagueness. Such an approach will successfully and more accurately capture the preferences of users.

Need for techniques to learn or capture preferences

Finally, learning user preferences without direct user inquiry needs to be supported. In other words it is necessary to develop appropriate mechanisms to unobtrusively obtain user preferences. Unobtrusiveness in personalization is an essential feature and a well discussed topic in user modeling (Rashid *et. al.*, 2002; Schwab and Kobsa, 2002; Alahakoon *et. al.*, 2004; Zaslavsky *et. al.*, 2007). When personalization is provided systems may have to interact with the user in two stages; in user model building and in product retrieval. In the process of user model generation, unobtrusiveness is achieved through either by employing implicit methods for acquiring user information or by information reuse. However, implicit methods do not acquire reliable information about the users and results in poor personalization. Therefore, we identify and investigate the possibilities of information reuse in achieving unobtrusiveness in personalization.

Unobtrusive user information acquisition by information reuse

In the literature, both academic and commercial user modeling systems formed user models containing different user information obtained using a variety of methods. Some of these are now commercially available (CDNow (Schafer *et. al.*, 2001), Movie finder[4]) while the others were prototype systems (Entrée (Burke, 2002a), Seta (Ardissono *et. al.*, 2001b)). A characteristic of these systems is that each of these have been demonstrated or used in specific domains, such as entertainment recommendations, restaurant recommendations, electronic program guides etc. Although the techniques used in these systems are suitable to be used in other domains, the information used within the profiles for predictions are not reusable in multiple domains. If each application is personalized using a separate user model, it becomes necessary for users to register in multiple websites that offer different products and services. As such, users have to provide the same information to different websites, leading to private/personal information unnecessarily being managed by multiple sites as well as users having to familiarize themselves with multiple applications. In addition, multiple websites gives rise to data repetition where users are required to reveal the previously provided data repeatedly to several websites.

Data repetition issues in user modeling and personalization has been partially solved by applications such as Microsoft passport[6], Liberty alliance[7], Lumeria[8], and Digital Me[9].

[6] Available at http://www.passport.com

[7] Liberty_Alliance, Available at www.projectliberty.org/

[8] Lumeria, Available at http://lumeria.com

These applications offer services such as storing personal information, automatically filling out forms on behalf of the owner, provision of anonymous surfing, and keeping track of usernames/passwords, thus avoiding repeated entry of the same user information. These systems are simple data stores without inferencing ability and are unable to provide personalized interactions. During the early 80's 'user modeling shell' systems made an appearance to fulfill the need of a single user model. When (the empty shell is) filled with application specific user modeling knowledge, these systems would serve as separate user modeling components in the respective applications. Providing the system with domain specific user knowledge required considerable work in such systems. As such, the application system requires considerable programming to communicate with the user modeling shell system as to use the advantages of personalization.

Few of the later developed commercial user modeling servers use a single profile but only in related domains such as tutoring different subjects within the teaching domain (Paiva and Self, 1994) and GroupLens/NetPerceptions (Resnick et. al., 1994) for different news story classification. Among more recent work, the concept of a single user profile is discussed within the ubiquitous computing environment and scrutable user profiles (Kay *et. al.*, 2002; Kay *et. al.*, 2003). In later work it has been reported, that a user model should consist of several different components. When the current eCommerce needs are closely analyzed, it is apparent that these requirements demand a personalization infrastructure that comprise centralized components, decentralized components as well as user modeling components that are embedded into the application (Fink and Kobsa, 2000).

Instead of maintaining distributed data regarding individuals, many researchers (Orwant, 1995; Shearin, 1999; Heckmann, 2003) suggest using single electronic user models that can serve multiple applications. Based on the user modeling approaches and ideas in the personalization literature, it can be argued that instead of domain based user models, maintenance of a single user model which has a suitable architecture (for instance, consisting of components) can be used over multiple domains. Both user and the vendor can benefit from such a user model. From the user's point of view it is the only source that

[9] Digital Me, Available at http://digitalme.com

is required for all personalization needs. From the vendors point of view entire consumer base can be evaluated according to the same modeling techniques.

Therefore, it can be concluded if there is a single user model which is designed to be used across the multiple domains and is able to successfully handle requirements for effective personalization that will be a valuable contribution to eCommerce as well as to personalization research. The timely requirement of such a user model which has arisen from the needs and the challenges faced by the current user modeling systems have provided the motivation for this thesis.

1.3 Objectives of the Thesis

Based on the discussion in the previous section, there are certain requirements that have to be met in order to ensure the future and current success of personalization in eCommerce. To address such challenges this thesis proposes the following:

1. Techniques for integrating user general purchase behavior characteristics with user's actual purchase behavior captured during transactions.

2. Ability to learn and update the model of the user with ongoing system-user interactions.

3. Develop techniques for efficient, practical and less obtrusive interactions with the users.

Based on these requirements, the following two objectives have been identified. The main objectives are further expanded into sub-objectives.

Objective 1: Design of a new user model architecture which can capture complex purchase behavior in the current eCommerce environment.

Sub-Objectives:

(i) To analyze the existing theories about consumer modeling and consumer segmentation to discover the information categories that are required to capture complex buyer behavior in the eCommerce interactions.

(ii) To develop a framework for incorporating the information categories identified in (i).

(iii) Design and implement a user model architecture that consists of components for capturing different information categories of user behavior based on the framework developed in (ii).

(iv) Inclusion of the following functionality in the user model as to handle the challenges in current eCommerce environment. These are related to the issues identified in section 1.2.

   i. Issue 1 - Capture changes in user preferences over time, in dynamic product markets

   ii. Issue 2 - The designed model should be usable in multiple product domains eliminating the need to use different ontologies (as the same user model is being used for all personalization needs in multiple domains).

   iii. Issue 3 - Capture the fuzziness in user preferences

   iv. Issue 4 - Able to use existing information to generate/infer knowledge about the user rather than requesting all required information from the user and hence less obtrusively create the user model.

Objective 2: Develop a new technique to successfully employ the user model, in online interactive product selection process.

Sub-Objectives:

(i) Develop a new technique for an interactive product selection process personalized with the novel user model.

(ii) Minimize obtrusiveness in user preference elicitation by employing the user model.

(iii) Evaluation of the interactive product selection process with regard to its abilities in product search and controlling obtrusiveness in system-user interactions.

1.4 Contributions of the Thesis

Corresponding to the above objectives, this thesis makes the following contributions to the area of user modeling research:

1. A novel user model architecture consisting of information layers (Layered User Model - *LUM*) is proposed, designed, implemented and validated. The main contribution are presented more descriptively as follows.

(i) A detailed study and analysis of the consumer behavior based on the past work of two research areas: information technology and market research.

(ii) Introduction of a conceptual framework for categorizing eCommerce consumer information into three main layers.

(iii) Propose, design and implement a new layered user model architecture that consists of three main information layers: personal information (*PI* layer), domain behavior information (*DI* layer) and transaction information (*TI* layer), which capture user behavior from different perspectives.

(iv) The novel user model architecture is capable of handling the following issues, and hence contributes to solving the challenges in the current eCommerce environment.

   i. Captures changes in user preferences over time using Hebbian learning technique. The user model also considers the "forgetting factor" as to maintain more accurate preferences.

   ii. Rather than using traditional stereotypes, this work introduces much flexible domain independent "*General Stereotypes*" (we refer to them as Purchase Behavior Characteristics – *PBC* values): a set of quantitative values which describe the user behavior generally in purchasing. *General stereotypes* allow the model to re-use existing information by generating/inferring knowledge about the user in multiple domains. Such ability partially eliminates the need of handling different ontologies by allowing the user model to operate in the same environment.

   iii. Calculate the user preferences based on fuzzy logic and hence achieving greater flexibility in capturing differences among users.

   iv. The user model is capable of providing personalized services even during the initial interactions. Use of *General Stereotypes* (*PBC* values) facilitates such ability hence handling the "new user" problem. The layered approach allows content-based information in the user model. Therefore, the model is also capable of capturing user preferences towards product features allowing it to include all products in the product-base in the search space. Hence the model is capable of locating new products that arrive in dynamic product markets (handle the "new item" problem).

2. The second main contribution is a Personalized Interactive Product Retrieval Process (*PIPRP*) combined with the novel layered user model (*LUM*).
   (i) Design and implementation of the *PIPRP*.
   (ii) The new user model is tested using an interactive product selection process. Several new algorithms are introduced to minimize the obtrusiveness of the system. With the support of the user model the interactive product selection process provides personalized services in all three phases of eCommerce activity: requirement elicitation, product search and product presentation.
   (iii) Definition of a set of evaluation criteria based on existing methods and evaluation of the performance of the product search process.
3. The important need of unobtrusiveness has been addressed in both user model building and product elicitation process.
   (i) With the user model creation, unobtrusiveness is achieved by information reuse to minimize explicit user inputs.
   (ii) Within the product retrieval stage, unobtrusiveness is achieved by exploiting the information in the layered user model, instead of requesting such information from the user.

1.5 Outline of the Thesis

This Thesis is organized as follows. Chapter 2 provides a discussion and a comparison of background work carried out in the user modeling area. The important issue of obtrusiveness during user interactions is formally defined. Existing user models and their contents and personalization techniques employed are discussed at length. Furthermore, this chapter discusses the existing user modeling systems in the area of eCommerce and possible ways of addressing their limitations. Finally, the advantages of user models in multi-domain distributed eCommerce are investigated.

Chapter 3 presents the theoretical foundations of the thesis. At the beginning of this chapter the design of the novel *Layered User Model* (*LUM*) architecture is presented. Then the rest of the chapter argues for the presented architecture; work carried out in market research and theoretical consumer behavioral theories are incorporated to justify the types of information captured in the user model.

Chapter 4 discusses the framework within which the *LUM* architecture is implemented. Mechanisms and algorithms for the initial creation of the layers and transfer of information between layers are discussed.

In Chapter 5, the implementation details of the *LUM* architecture and the algorithms required for the design of the layered user model are discussed. Theories, techniques, and inferencing mechanisms, their implementation and algorithms are discussed in detail.

Chapter 6 provides a comprehensive set of experiments to demonstrate the functionality, value and usefulness of the new model.

Chapter 7 further justifies the importance of the novel *LUM* applying it in eCommerce environment. The background work of online product selection strategies is discussed at length. This chapter presents the application of the *LUM* in a Personalized Interactive Product Retrieval Process (*PIPRP*). Furthermore, the chapter demonstrates the *PIPRP* step-by-step using an example and experimental results are presented. To strengthen the evaluation of the model, this chapter introduces a new evaluation criterion and evaluates the combined outcomes of the *LUM* and the *PIPRP*.

Chapter 8 provides the concluding remarks of the thesis and discusses future work.

Chapter 2

A Review of Personalization and User Modeling

In the previous chapter the importance of personalization and the motivation for the thesis were discussed. The objectives of this work and the contribution of the thesis were also presented. In this chapter the background of the work is described which consists of what has been carried out so far in the area of research and the limitations of the existing approaches.

The rest of the chapter is organized as follows. In the section 2.1, the history and categories of user modeling systems are discussed. Section 2.2 discusses the major application areas of user modeling and personalization describing the variations of requirements in each area. Section 2.3 describes the important dimensions to be decided when building a user model. The next three consecutive sections (2.4, 2.5, 2.6) describe current electronic user models with respect to their methods of information gathering, contents, and techniques used for recommendations and categorize sample models under each of these topics. Section 2.7 discusses the algorithms and quantitative techniques employed in existing user models. Section 2.8 present the electronic user models specific to eCommerce environments, whereas section 2.9 provides an analysis of obtrusiveness in personalization discussing what is obtrusiveness in system-user interactions and how it was handled in the literature. Then section 2.10 discusses the limitations of user models in eCommerce environments. Section 2.11 presents possible methods to address the above limitations. Finally, section 2.12 summarizes the chapter.

2.1 User Modeling Systems

During the past several decades (dating back to 1970's) several electronic user models have been employed to provide personalization in different application domains. These electronic user models can be categorized into five major types according to their overall performance: user models in dialog systems, shell systems, servers, recommender systems and systems that perform tasks on behalf of the user in a personalized manner. Table 2.1 presents examples of systems under each category. Next, each category is described using the examples in the table.

In the 1970's, the idea of user modeling emerged as a need for natural language interfaces to support discourse systems. The user models were employed in narrow domains, to conduct personalized dialogs between the user and the system. Due to the reason that system focus was on narrow domains, the conversations were somewhat unnatural compared to human dialogs. GRUNDY (Rich, 1989), was such a book recommender system. For example, Grundy needs the user of the system to provide a "few words that provides a good self description" (Rich, 1983). Therefore, the user had to find an appropriate strategy to deliver the correct information that the system needs in order to work out its recommendations. These systems had no capability of providing any support guiding the user in this respect (Wahlster and Kobsa, 1989).

These early user modeling systems were built into the application and impossible to be reuse in a different application. On the system developer's side, building the user modeling component was a costly process. Therefore, the urge for *re-usable* user modeling systems emerged. Making use of developments in the field of Expert systems, the next generation of user modeling systems, called "shell systems" was born (Kobsa, 2007). Shell systems are capable of modeling users irrespective of an application domain, by separating the user modeling functionality from the user-adaptive application. The application developer has to provide the empty "shell' with application dependent information and develop the interface and interactions between the application and the shell system. Examples for shell systems among the above are GUMS (Finin, 1989), *UMT* (Brajnik and Tasso, 1992), TAGUS (Paiva and Self, 1994), and *um* (Kay, 1995). Shell systems maintain application independent user data that can be used by any application, in addition to application specific user data. But all the inference rules and stereotypes are predefined.

Table 2.1: Well known existing user modeling systems

System/Tool and Year	Description
Dialog Systems	
PersonaLogic[10] (1998)	Search tool for online catalogues
GRUNDY (Rich, 1979)	A book recommender
Shell Systems	
GUMS (Finin, 1989)	Allows applications to define stereotypes, answer queries about the user based on currently held information, at run time accept and store new facts about the user, informs the application about inconsistencies.
UMT (Brajnik and Tasso, 1992)	Allows applications to define stereotypes and rules for contradiction detection
User Modeling Servers	
(a) Commercial User Modeling servers	
GroupLens (Resnick *et. al.*, 1994)	The personalization engine is utilized for array of systems in eCommerce, knowledge management, online advertising, e-mail marketing companies, and for supporting call centers to provide clients with personalized suggestions.
(b) Research Prototype systems	
BGP-MS (Kobsa and Pohl, 1995)	Can be used as a network server with multi-user multi-application capability.
Doppelganger (Orwant, 1995)	Accept user information from h/w and s/w devices. Machine learning techniques are put at the disposal of user modeling developers.
CUMULATE (Brusilovsky, 2004)	A student adaptive educational system. Collect student information from multiple servers.
Personis (Kay *et. al.*, 2002)	Maintain a main user model and application dependent 'personas' for each user.
Recommender systems	
Lifestylefinder (Krulwich, 1997)	eCommerce web site recommendations
Entrée (Burke, 2002a)	Restaurants recommendations and critique based navigation
Quickstep (Middleton *et. al.*, 2002)	Web based research paper recommender system
Latizia (Lieberman, 1995)	Retrieve interesting web pages
Syskill & Webert (Pazzani *et. al.*, 1996)	A s/w agent learns to rate web pages, that interest the user.
SETA (Ardissono *et. al.*, 1999)	Online catalogue generation
EPG TV (Ardissono *et. al.*, 2004)	Personalized TV program guides.
News Dude (Billsus and Pazzani, 1999)	News story classification
Tapestary (Goldberg *et. al.*, 1992)	E-mail and news filtering- recommend interesting news articles to users
ELFI (Schwab and Kobsa, 2002)	Provide users with personalized research grant information
Identity Management Systems	
Liberty Alliance [7] (2000)	Assist user's to visit or take part in services from multiple websites under the same authcate.

[10] See the slide at http://web.media.mit.edu./~pattie/ECOM/sld018.htm

Digitalme[9]	An open source information management system maintaining user information as InfoCards. User's are able to maintain multiple InfoCards for different sites and able to control amount of information revealed.
Lumeria [8] (1998)	Maintain a consumer Super Profile where information is allowed to share between sellers and marketers. In return consumers get personalized services.
MS Passport [6]	There are two usages of MS Passport; One 'sign in' and the other 'wallet'. The sign in allows single authcate for several websites and MS wallet maintain credit card and transaction information and allows secure monetary transactions

Commercial user modeling servers developed later, such as GroupLens (Konstan, 1997) are also generic user modeling systems. User modeling servers have the additional advantage of functioning separately from the application system. Unlike shell systems, server systems are not functionally integrated into the application but communicate with the application through inter-process communication. This architecture allows several applications to use the same server for different purposes. Kobsa (2007) further categorizes user modeling servers as "academic" and "commercial".

Recommender systems are employed to guide the users in a personalized manner to interesting or useful objects in a large space of possible options. Most of the time, these applications are tailored to perform well in a given application domain. Most of them maintain long term user models that are usable in the same domain. Notable examples are (Rich, 1979; Pazzani *et. al.*, 1996; Ardissono *et. al.*, 2001b; Middleton *et. al.*, 2001; Ardissono *et. al.*, 2004).

Finally there are systems known as "Identity Management Systems" or "User Provisioning Systems". These systems generally maintain user data in a centralized repository and use the data to provide users with personalized services. They are capable of account linking; and therefore able to automatically fill out forms on behalf of the user. But they lack inferencing capabilities and therefore their services are limited.

User modeling systems in Table 2.1 were implemented in different application areas or domains. Such application areas/domains have their own characteristics which have a strong influence over the user models required by the user modeling systems. The main application areas where user models were used are discussed below.

2.2 User Modeling and Application Areas

As previously stated, electronic user modeling is used in several different domains and these domains can be categorized into several major areas such as information retrieval, eCommerce, entertainment recommendations, and adaptive hypermedia systems. Each of these domains has characteristics that are specific to them and therefore the objective of personalization varies, influenced by their varying interests.

In the information retrieval domain user models have been employed to avoid information overload. Information filtering according to a user model results in retrieving more relevant web pages or news articles. In the area of eCommerce, to attract users to websites it is important to provide personalized interactions. A user model also becomes useful when finding out preferred products out of large collections of items, and also for targeted advertising purposes. In the entertainment domain a user model is employed to recommend CDs or movies that the users may like out of massive collections. Finally in expert systems, student models have been used to provide users with personalized guidance and tutoring. This avoids situations such as providing an expert user with instructions suitable for a beginner. Categorization according to application areas is illustrated in Figure 2.1.

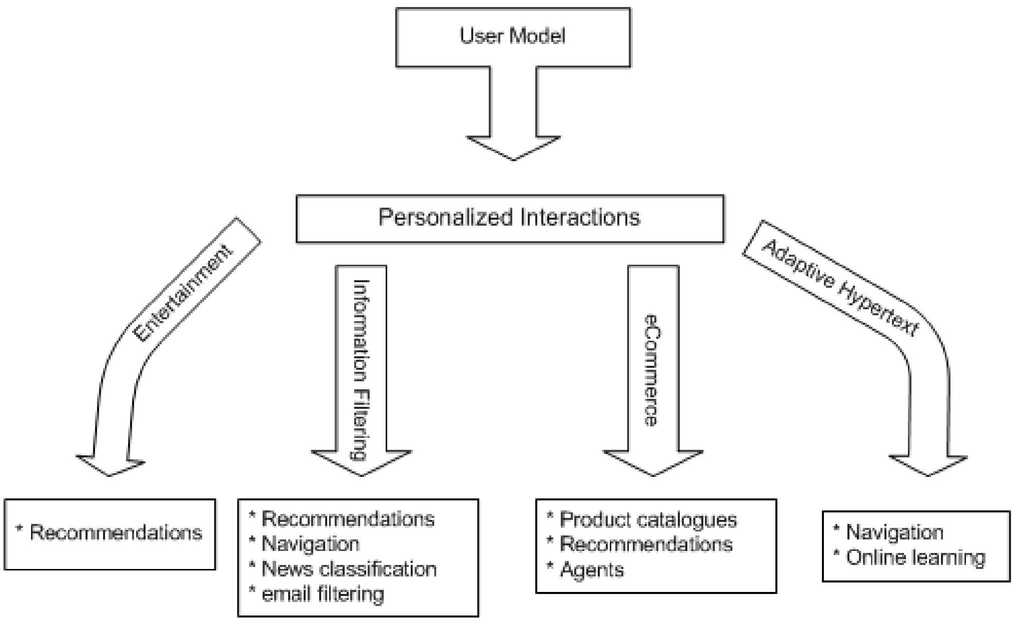

Figure 2.1 : Domains where user models were employed to provide personalization

Each of these domains (or application areas) has their own characteristics. For example, in the information filtering domain (news article filtering) recommending more news articles under a once recommended topic is trivial. But in the entertainment domain an adult fond of a certain genre of movies will continue to like movies of the same kind. Therefore, it is observed that different user modeling methods and techniques perform well depending on the nature of the application area (Towle and Quinn, 2000).

When designing a user model, once the domain is known there are other aspects that contribute to the efficiency of the user model. In Ross (2000), the electronic user models were classified along the four major dimensions as: what is modeled (canonical user or individual user), source of modeling information (explicit user model constructed using direct user involvement or an explicit user model based on system extracted user behavior), time sensitivity of the model (short term or long term), and update methods (static model or a dynamic model). Each of the dimensions are discussed next.

2.3 Dimensions of User Models

When deciding on the methods of developing a user model for a given application, the above described dimensions are decided depending on the application domain characteristics. Impact of these categories on the application domain is described below under each category.

2.3.1 Canonical or individual user

If the items are targeted at a particular audience, a canonical user model has the ability to capture such group preferences. As an example, modeling user groups are effective in the entertainment domain. It is possible to identify user groups where certain demographic groups may be interested in similar items. For example, people in a certain age group may have a trend towards a certain type of entertainment such as music or movies. Again in the information filtering and adaptive hypermedia, people working in the same areas of studies may form groups. Therefore, stereotyping users in such domains is effective. However, in eCommerce where a heterogeneous user population carries out their purchasing in frequently changing product markets, such stable groups are difficult to identify.

2.3.2 Explicit or implicit information

In some domains, people will tolerate a certain amount of inaccuracies in recommendations due to the low investments required. For example, following recommendations for borrowing items does not incur high cost, and therefore, customers tolerate a certain level of inaccuracy. In contrast, if purchasing an expensive commodity the accuracy of the recommendation will be considered as quite important. In such situations explicit user details will help to provide better suited predictions.

Since eCommerce applications are involved in monitory transactions, predictions for such domains are crucial. An eCommerce application will prefer explicit information from the user about his/her current need rather than implicit information collected by the system. Implicit user information such as previous purchases, purchases by similar users and time taken to look at an item, may be combined with the current explicitly specified preferences. But an application such as recommending movies or an information link in information retrieval will be less crucial and implicit user information will be considered as sufficient. For example, there is no monetary commitment in CDs or books recommendation or electronic program guides, since generally the CDs or books are borrowed from a library.

2.3.3 Short or long-term model

In certain domains, individuals exhibit recurring behavior patterns. For example, people tend to prefer movies of similar or same genre (e.g. crime, comedy, etc). Although maintaining a long term model is expensive, such valuable information can be put to ongoing use. If supported with dynamic updates, long term models are also capable of handling temporal decay of the item relevance.

A short term user model is suitable for capturing one time interests such as real estate property purchases. Certain application domains such as news article filtering or eCommerce, demonstrate portfolio effects. In other words, products that user has seen or purchased affect the interest in other products of similar description. In eCommerce, if the user purchases an electrical item he/she may not purchase the identical item for the next few years. When recommending news articles, a user is less interested in reading the same news for the second time. But user may need to read more about interesting recent events.

Again, there can be user's long-term general preferences that could not be captured by a short-term model. Therefore, selection of a short or long term user model is strongly linked to the product domain.

2.3.4 Static or dynamic model

Static user models do not update its contents, and hence, cannot capture changing user needs. Such a user model is only capable of retrieving current needs. A static user model can be effective over a long period of time only if the changes are negligible and so, suitable for capturing information such as user demographics.

With time, user needs, goals and preferences change. Therefore, interest in certain products varies with time. A dynamic user model can capture the changing user needs over time. With this kind of user model, it is possible to consider user's past preferences for predicting future interests. Therefore, when there are long term behavior patterns to capture, a dynamic long term user model is useful.

For a user model to perform effectively and efficiently, the above discussed application specific characteristics have to be taken into consideration. However, research carried out in certain domains ignore the possibility of such effects (Towle and Quinn, 2000). When designing a user model within a certain application domain, aspect such as (i) the methods of gathering information about the user, (ii) information recorded in the user model and (iii) the user modeling techniques used to combine the acquired information for future predictions have to be determined appropriately. Under each of those topics, varieties of methods have been exploited by existing user models which are discussed in the next three sections.

2.4 Methods for Gathering Information about the User

The more information the system possesses about the user, the better its prediction ability to provide services tailored for the individuals. However, techniques for acquiring this information in an unobtrusive manner that does not overly burden the user are still the subject of ongoing research. The accuracy of information is highest when explicitly obtained from the user, but this might burden the user. Therefore, in addition to obtaining the information via explicit user inputs, there are implicit user information acquisition methods such as searching the web, monitoring user behavior, mapping the user to existing user segments and employing agents based information retrieval methods.

2.4.1 Explicit user inputs

Explicit user information is gathered during system user interactions. This is carried out by (a) the user filling in a form, (b) during a dialog between the user and the system or (c) the user rating a list of items or features presented by the system.

Filling out forms to express user needs is an impersonalized static questioning process. Every user is supposed to fill out the fields in a questionnaire. In dynamic dialogs, user information is acquired during the dialog where each user is presented with a different set of questions. In the ratings based methods, the user inputs are ratings towards a list of items indicating user preference. This can be either binary such as *like* or *dislike* or the users provide a rating towards items according to a scale (say 1-5).

Although explicit information is high in accuracy, when obtaining such information, obtrusiveness has to be balanced carefully. This can be achieved by minimizing the number of questions directed to the users by maximizing the information gain of each question. Other aspects that are related with obtrusiveness of interactions were discussed in detail in section 2.4.

2.4.2 Web searching

Since the Internet contains a large volume of data about users in the form of heterogeneous data sources, it is possible to retrieve certain reliable information from these information sources. Although Web searching approach could reduce the information requests directed to users, privacy concerns and P3P regulations, limit the scope of this method. In the literature there are personalization systems which have used existing online information. For example, Pazzani, 1999 obtained user demographics from their personal home pages to be used in recommendation of restaurants. There were many problems and drawbacks related to this approach. For example, in the above work, some people did not have a home page which will be the majority case in eCommerce applications. Also, with this method, information obtained by employing techniques such as text mining would be less reliable.

2.4.3 Stereotypes

Rather than collecting behavior observations of an individual, it is possible to map a new user to already existing behavioral data of a population of users. A stereotype represents a collection of attributes that often co-occur in a group of people. Such existing behavior could be treated as defaults, which can be overridden by specific observations.

User stereotyping first appeared in the Grundy system (Rich, 1979). At the time of system design, user groups and subgroups were identified, and typical needs and expectations of these groups were determined. Then during system runtime, new users were assigned to one or more groups or subgroups based on available initial information. Then the new users were assumed to inherit all the typical characteristics of these groups. Grundy, used a hand coded set of stereotypes which were formed based on general knowledge and intuition of the author. Grundy's attempts to automate the acquisition of stereotypes have been limited to the adaptive refinement of numeric parameters, rather than the construction of the stereotype. Since Rich's introduction of the notion of stereotypes, they appeared in a number of user modeling shell systems such as BGPMS (Kobsa and Pohl, 1995), *um* (Kay, 1995), Dopplegager (Orwant, 1991) and UMT (Brajnik and Tasso, 1992) confirming their importance and value.

Double stereotypes were introduced in the KNOME system (Chin, 1989), to reason from user action to a classification of their expertise and then to derive user's possible actions starting from user expertise. SeAN (Ardissono *et. al.*, 2001a), is a multiagent system, which generates adaptive hypermedia for accessing on-line electronic news servers. It aims to personalize both the selection of topics and the level of detail of the presentation of each news item. SeAN utilized *families of stereotypes* along four different dimensions (Interests, Expertise, Cognitive characteristics, and Lifestyles) to capture different conceptual characteristics of the user. In Figure 2.2 and 2.3 are two stereotypes belonging to two different *families* (lifestyles and interests) obtained from Ardissono *et. al.* (2001a). These stereotypes contain both the classification and predictive information and therefore domain dependent and cannot be used in a different domain.

In later work, SETA (Ardissono *et. al.*, 2001b) and electronic TV program guides (Ardissono *et. al.*, 2004) used lifestyle survey based stereotypes. In all instances, a new user belongs to one or more stereotypes to a certain percentage, and hence, inherited the characteristics according to the ratio. These stereotypes are predefined. They did not have a method to update the changing needs of the user population. Since the stereotypes were built based on only a sample of population, the initially identified groups and characteristics can become non-representative when a larger portion of the data is used or substantial growth of the user population occurs. Stereotype of a 'Housewife' used in EPR in TV domain is shown in Figure 2.4. Similar to the stereotypes used in SeAN, this too contains both the classification and predictive information and strictly belongs to the given domain.

ADULT SUPERIOR COMMITTED:
profile:
age: <26: 0; 26-35: 0.1; 36-45: 0.4; 46-65: 0.4; >65: 0.1
gender: Male: 0.5; Female: 0.5
education: primary school: 0; secondary school: 0.2; university: 0.8
education field: economy: 0.1; {politics, law, sociology}: 0.5; {medicine, biology}: 0.05, scientific: 0.05, human sciences: 0.3...
hobbies - theatre: a lot: 0.3; some: 0.5; a little: 0.2; not at all: 0;
hobbies - traveling: a lot: 0.5; some: 0.4; a little: 0.1; not at all: 0;
hobbies - body care: a lot: 0; some: 0.3; a little: 0.5; not at all: 0.2;
hobbies - shopping: a lot: 0; some: 0.2; a little: 0.8; not at all: 0;
hobbies - doing sport: a lot: 0; some: 0.1; a little: 0.3; not at all: 0.6;

Figure 2.2 : A stereotype in the family of life styles in SeAN – reproduced from (Ardissono *et. al.*, 2001a) requested permission

FINANCIAL PROFESSIONAL:
profile:
age: <20: 0; 20-25: 0.1; 26-35: 0.2; 36-45: 0.3; 46-65: 0.3; >65: 0.1
gender: Male: 0.8; Female: 0.2
job: manager: 0.57; self-trader: 0.3; self-employed: 0.05; ...; student: 0.01
job field: {financial, banking insurance}: 0.8; {politics, law, civil services}: 0.14; ...
reason of connection: work: 0.9; personal: 0.1
hobbies - theatre: a lot: 0.1; some: 0.3; a little: 0.4; not at all: 0.2;
hobbies - watching sports: a lot: 0.4; some: 0.3; a little: 0.2; not at all: 0.1;
...
predictions on interests:
economy: high: 1; medium: 0; low: 0; null: 0
politics: high: 0.7; medium: 0.3; low: 0; null: 0
sport: high: 0.2; medium: 0.4; low: 0.3; null: 0.1
culture: high: 0; medium: 0.2, low: 0.5; null: 0.3
technology: high: 0; medium: 0.3; low: 0.6; null: 0.1

Figure 2.3 : A stereotype in SeAN– reproduced from (Ardissono *et. al.*, 2001a) requested permission

Housewife
<u>Classification data</u>
Age *[personal data]*: *Importance*: 1, *Values*: (less_than_15, 0) (15/24, 0) (25/34, 0) (35/44, 0.5) (45/54, 0.5) (55/64, 0) (more_than_64, 0)
Gender *[personal data]* *Importance*: 1, *Values*: (male, 0) (female, 1)
Books*[interest]* : *Importance*: 0.6, *Values*: (low, 0.8) (medium, 0.2) (high, 0)
<u>Prediction part</u>
movies-sentimental, *Interest*: 1; serial-soap, *Interest*: 1; TV news, *Interest*: 0,2; fashion programs, *Interest*: 0,5; cooking programs, *Interest*: 1; …

Figure 2.4 : "Housewife" stereotype in TV domain, – reproduced from Ardissono *et. al*., (2004) requested permission

Work carried out in the Doppelganger user modeling system (Orwant, 1995) used clustering mechanisms to identify user communities within the entire user population. They maintained two types of user communities. There were twenty-two permanent user communities (such as *student, artist, children* and *Media Labbers*) which were chosen to be created at the system design time. Although the choice to create them was made by a human, they change overtime as new information becomes available. In addition there were automatically created stereotypes using an unsupervised clustering algorithm. This way, emerging salient traits were represented in communities without explicit human intervention.

Another community approach is introduced in Paliouras *et. al.* (1999). They identify user communities (class description) based on users' system usage data by employing induction of decision trees. Users belonging to the same community are expected to have similar preferences towards items. When the stereotypical behavior is identified within a cluster such information are used to derive more complete models of the users.

Work in Schwab and Kobsa (2002) follow a similar approach to identify user groups. They cluster the existing user models based on explicit information on preferred features, and descriptions of the clusters are used as predefined stereotypes. This way dynamic evolution of the stereotypes is made possible.

With automatic stereotype generation the following issues persist: users can be grouped along different dimensions, in some domains it is difficult to identify well-distinguished user groups, one user can fit into contrasting user groups, and additionally in a dynamic market users may change their memberships, too frequently demanding costly re-clustering. Schwab and Kobsa (2002) employed a less costly nearest neighbor method to overcome these limitations. But the cluster formation is possible only after the system has obtained a reasonable number of users. Additionally, the system needs preference information of the user to compute the nearest neighbors.

2.4.4 Observing user behavior

User behavior during system-user interactions can provide valuable information. Eliciting such information is important to be utilized as evidence to predict future user requirements. This can be carried out by observing the individual user and keeping track of the actions performed. Often possible actions depend on the application domain. In domains such as information retrieval book marking a web page indicate a positive action where the user is interested in the particular web page. In the literature, actions such as browsing a web page, skipping a hyperlink, or putting an item in the shopping cart were used as evidence of an individual's interest when predicting or guessing about the user requirements.

This method can result in misleading the system beliefs regarding the user. For example, say the user is interested only on a subtopic in the above web page example. The system may come to the conclusion that, he is interested in the main topic and forward the user with more similar information, which can be annoying and tiresome to the user. Again the user may be looking at that web page on behalf of another person. If the user profile is changed according the new behavior, future system findings will not fulfill user needs. Although this method is less intrusive to the user, the information obtained may not be reliable. This situation arises since the systems' interpretation of the user actions may not be the actual reason behind the observed action.

Table 2.2 lists a sample of existing user modeling applications describing them with respect to the user information elicitation methods they employed. The darkened cells are not applicable.

It can be seen that ELFI (Schwab and Kobsa, 2002) and Letizia (Lieberman, 1995) do not use explicit user information. But on contrary, they require users to use the system for an initial period of time to construct a useful profile. Rest of the systems collects both implicit and explicit user information in order to base their predictions. Every system elicit explicit user information uses them as start up information to initialize the user model. As mentioned before, though explicit information is intrusive to collect, such information is important to reveal the actual needs of the user. Systems such as ELFI and Letizia are in the information retrieval domain which has less monetary commitment, where users will tolerate mismatches in recommendations.

Table 2.2 : Information elicitation methods

System/Tool	Explicit Inputs	Implicit Inputs		
		Observing User Behavior	**Stereotypes**	**Other**
BGP-MS (Kobsa and Pohl, 1995)	Application specific initial user interview	Predefined assumptions based on observed user actions during interactions	A hierarchy of predefined stereotypes. General and application dependent.	
Personis (Kay *et. al.*, 2002)	Questions to answers	h/w & s/w sensors based and during interactions with applications		Pre-existing user information (Components) are reusable in different personas.
Doppelganger (Orwant, 1991)	Questions to answers	h/w & s/w sensors based and during interactions with applications	Users belongs to different Communities by fractions	
GroupLens (Resnick *et. al.*, 1994)	Ratings are obtained from the users, towards the articles they have read.	Navigational data, past ratings towards items they knew and purchased.		Ratings of the similar users
Lifestyle Finder (Krulwich, 1997)	Answers for questions			Map to one or more user information clusters
Entrée (Burke, 2002a)	Attribute preferences or example restaurant User feedback			
Quickstep (Middleton *et. al.*, 2001)	Explicit feedback, registration information	Browsed URLs		publication list from the organization database, correlation to similar users
Syskill & Webert (Pazzani *et. al.*, 1996)	User ratings for web pages on a three point scale- both (+)ve and (-)ve preferences.			

Letizia (Lieberman, 1995)		Terms entered in search engines, Visited web pages,		
SETA (Ardissono *et. al.*, 1999)	Answers to questions.	User viewing an item indicate the interest	Map users to one or more pre-defined s/types. Inherit fractions of behavior depending on the degree of match.	
EPG TV (Ardissono *et. al.*, 2004)	Answers to questions Explicit ratings.	Info in the set top box - recorded interesting programs	Map users to one or more s/types. Inherit fractions of behavior depending on the degree of match	
Grundy (Rich, 1979)	Initially a self description followed by questioning to map user to stereotypes		User model get populated with only stereotypes	
News Dude (Billsus and Pazzani, 1999)	Selecting the interesting news channel, Input comments using available options/ratings	User options are used to identify long term general news preferences.		
ELFI (Schwab and Kobsa, 2002)		User model get initiated based on user behavior, then inherit from similar users		
Tapestary (Goldberg *et. al.*, 1992)	Descriptive ratings as annotations and needs to query for own need			

According to the table, applications catering for domains such as eCommerce, where there is a monitory commitment is involved, tends to collect more reliable explicit information to strengthen their predictions. This concludes that the domain in question, contribute to the systems' intrusiveness towards the user due to the nature of its information requirement. As shown in the table, regardless of the method, the type of user information acquired by each system is different.

For example, in (Pazzani, 1999) demographics were collected implicitly while SETA acquired the same explicitly at registration. Types of information contained in user models are discussed next.

2.5 Contents of User Models

Information contained in a user model can be either explicitly acquired or implicitly derived information. Some contain user interests towards items in the product collections while the others maintain user preferences towards product descriptions. Table 2.3 summarizes the contents of user models in existing well known systems.

Table 2.3 : Information content in user models in existing well known systems

System/Tool	Item List	Feature list	Personal Information
BGP-MS (Kobsa and Pohl, 1995)		Derived user capabilities dependent on the application	Information such as user's abilities obtained by asking the user
SETA	Items put in the shopping cart	Item features and their preferred attributes	Demographics required to mapping user to a stereotypes
Ringo/Firefly (Shardanand and Maes, 1995)	Item list and ratings		
EPG TV	List of rated programs	TV Program features and their preferred attributes	Demographics required to mapping user to a stereotypes
GroupLens/ NetPerseption	Item list and ratings		
Grundy (Rich, 1979)		Categories of book genres and user preference towards them	Personal information from the stereotypes
Personis Kay *et. al.*, 2002		Feature/ abilities along with ratings	Few demographics
News Dude (Billsus and Pazzani, 1999)	Each news story as a TF-IDF vector	Hand selected set of domain specific features/words – each news story as a Boolean feature vector	
Quickstep (Middleton *et. al.*, 2002)		Research paper topics and calculated interest	Use personal information from a department database to link with people with similar interests
Latizia (Lieberman, 1995)		Set of weighted keywords	
Syskill&Webert (Pazzani *et. al.*, 1996)		Each topic has a user model consisting of two sets of weighted key words (likes, dislikes)	

Often preferences towards items are represented as a list of ratings. These ratings can be either binary or of higher scale. Binary ratings allow user only to categorize their preference as 'like' or 'dislike' while higher scale ratings allow to specify a more granular preference. In Adaptive Place Advisor (Thompson *et. al.*, 2002) preference towards an item is captured as a ratio of the time user accepted an item out of the time it was recommended. In SETA, items put in the shopping cart are recorded to be used as evidence for later recommendations. Table 2.3, confirms that applications in information filtering domain maintain a set of weighted key words rather than preferred news items. The key words with a larger weight indicate its frequent presence in the user preferred articles. Similarly systems such as BGP-MS (Kobsa and Pohl, 1995), SETA (Ardissono and Goy, 2000), EPG TV (Ardissono *et. al.*, 2004), Grundy (Rich, 1989) and Personis (Kay *et. al.*, 2002) maintain domain descriptions and user preference toward each such feature.

Most of the user models in literature are difficult to read and understand. There are some exceptions such as the user models used in *um* (Kay, 1995) and Personis (Kay *et. al.*, 2002) user modeling servers. These models allow the user to read and edit its contents. According to the Table 2.3, personal information can be either demographics or user's personality related information. Personality related information is obtained explicitly by interviewing the user as in BGP-MS or derived using the stereotypes as in EPG TV, or acquired using both methods as in GRUNDY. According to the Table 2.3, demographics are generally used as initial information to predict user preferences and once user starts interactions with the system are combined with user actions to update preferences. Predictions based on initial information or the updated information is carried out exploiting recommendation techniques. Such techniques present in current user modeling applications are discussed next.

2.6 User Modeling Techniques

As mentioned previously in Chapter 1, a user model contains the information regarding the user's needs, requirements and preferences. This information needs to be explicitly captured as user inputs or as implicit user information such as past preferences, which are gathered by the user modeling system during the previous interactions. Such information combined with the information already available with the system is used to predict the possible user requirements. Product information or systems knowledge about its user population or similar information which is available prior to system start-up falls into the latter category. Burke (2002a), refer to them as *background information.* A user modeling system employs an algorithm to combine the above input data and system data to arrive at its suggestions about the user needs. The two types of information and the algorithms employed to combine them, vary from system to system. Figure 2.5 shows this categorization of user modeling techniques.

All such different techniques basically belong to two major techniques: collaborative filtering and content based filtering. Collaborative techniques use the assumption that people with similar taste will rate things similarly. There are alternative ways of performing collaborative filtering, such as user-to-user correlation, item-to-item correlation or demographic correlation. Content based techniques analyze item descriptions to identify items that are of particular interest to the user. The most popular and known method is to obtain user ratings towards item features. Knowledge based filtering and utility based filtering can be categorized under content based filtering, since the filtering is carried out based on item descriptions (contents).

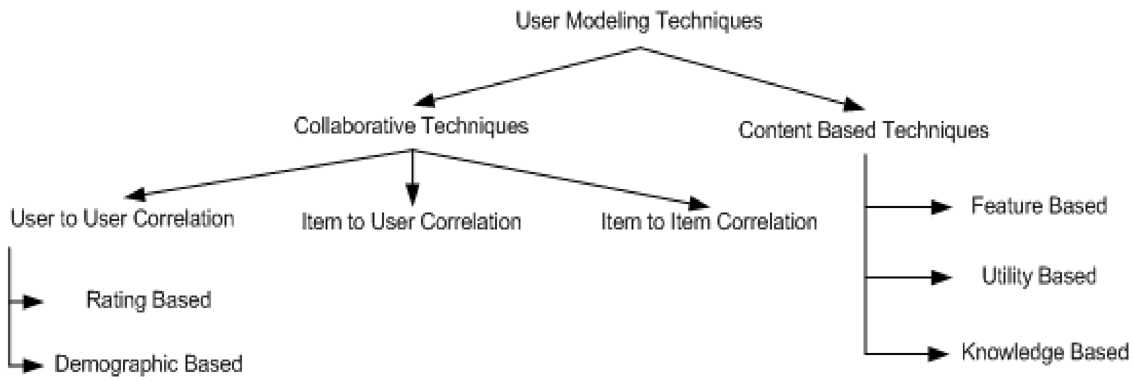

Figure 2.5 : Categorization of User Modeling Techniques

Table 2.4 present several known user modeling systems in the literature, along with the main user modeling technique/s they employed. The darkened cells are not applicable. The techniques are discussed below with respect to several issues such as the types of information required to provide personalization, time sensitivity of the model (long-term or short term), typical contents of the user model, and obtrusiveness of the system-user interaction. Strengths and drawbacks of such techniques are also presented.

Table 2.4 : User modeling techniques in existing well known systems

System/Tool	Collaborative	Content based
GroupLens/NetPerseption (Resnick *et. al.*, 1994)	User-to-user and User-to-item correlation	
Lifestylefinder (Krulwich, 1997)	Clustering similar individuals /User-to-user	Knowledge based
Entrée (Burke, 2002a)		Knowledge based
Quickstep (Middleton *et. al.*, 2002)	Preferred research areas of people belonging to a similar community – closer to demographic correlation	Papers in the interesting research areas are retrieved with the support of topic ontology - knowledge based
Latizia (Lieberman, 1995)		Feature based
PersonaLogic[10]		Utility based approach
SETA (Ardissono and Goy, 2000)		Knowledge based
EPG TV (Ardissono *et. al.*, 2004)		Feature based/ Knowledge based
Grundy (Rich, 1989)		Knowledge based
News Dude (Billsus and Pazzani, 1999)	Item-to-user collaboration	Long term interest in news stories - using the occurrence of a set of hand selected words in each story
Syskill&Webert		Feature based
ELFI (Schwab and Kobsa, 2002)		Personalized Research Grant Information
Tapestary (Goldberg *et. al.*, 1992)	Indirect user-to-user collaboration, since many people can post evaluations	Filtering of commented items includes content based/knowledge based criteria
Firefly (Shardanand and Maes, 1995)	User-to-user collaboration	

2.6.1 Collaborative filtering

Collaborative filtering is believed to be the most familiar, widely used and most mature of the techniques (Burke, 2002a). From results described so far, it is the best performing technique, especially in the eCommerce environments (Ardissono *et. al.*, 2004).

The main idea in collaborative filtering is that the users, who exhibit some form of similarity in their behavior, can serve as recommenders to each other on previously unencountered data items. A typical collaborative filtering based user model consists of an array of items and the ratings towards the items which are provided by the user.

The new user is provided with an array of items, which is a representative sample of the available item space. Then the user provides his/her recommendation or ratings to each item which are then used to recognize user communities with similar interests, and generate new recommendations or predictions for the target user based on the preferences of community members. These similar users are called *neighbors*. The most common method to find out similar users is calculation of Pearson's correlation coefficient. Pearson's correlation reflects the degree of linear relationship between two variables. It ranges from +1 to -1. In Schafer *et. al.* (2007) calculation of similarity between a user *u* and a neighbor *n* is calculated using the Pearson correlation coefficient as follows. Term $CR_{u,n}$ denotes the set of correlated items between *u* and *n*. Here r_{u_i} denote the rating given by *u* for the item *i*, and $\overline{r_u}$ is the mean rating of *u*. Subtraction of the mean value rating, reduces the dissimilarities among users (Schafer *et. al.*, 2007). Such dissimilarities in user ratings are known as the *shift of average ratings* problem in collaborative filtering technique. (More information about the *shift of average ratings* is discussed later in this section).

$$userSim(u,n) = \frac{\sum_{i \in CR_{u,n}} \left(r_{u_i} - \overline{r_u}\right)\left(r_{n_i} - \overline{r_n}\right)}{\sqrt{\sum_{i \in CR_{u,n}} \left(r_{u_i} - \overline{r_u}\right)^2} \sqrt{\sum_{i \in CR_{u,n}} \left(r_{n_i} - \overline{r_n}\right)^2}}$$

A correlation of +1 indicate a that there is a perfect positive linear relationship between the target user and the neighbor. Considering the ratings of neighbors, for an item *i* rating of a user *u* is calculated as follows.

$$pred(u,i) = \frac{\sum_{j \in ratedItems(u)} itemSim(i,j).r_{u_i}}{\sum_{j \in ratedItems(u)} itemSim(i,j)}$$

In item-to-user correlation, slightly different calculations are carried out to measure the similarity between two items. Depending on the similarity, the rating for target item is calculated. Basically there are two types of collaborative filtering methods (Breese *et. al.*, 1998):

- Memory based, and
- Model based

In memory based methods, to predict user ratings for a given item, the ratings of the entire user population or a sample of the population is used. The 'like-minded' users are found using clustering algorithms such as the k-nearest neighbor algorithm. In model based methods, a new user model for a new user is formed with the information from the existing users. The model is then used to predict future preferences of the user. In the literature, model based methods employed a variety of learning techniques such as neural networks (Jennings and Higuchi, 1993) and Bayesian nets (Condliff and Lewis, 1999) to create the user model. In either method, there are issues connected with obtaining the initial user ratings.

The greatest strength of the collaborative approach is said to be its capability of recommending items in the absence of machine readable product descriptions (Burke, 2002a). Therefore, it is capable of recommending complex objects such as movies or music which needs to be described with respect to qualitative measures. But when the user ratings are considered still there is quality related issues such as *variance in ratings*, connected with collaborative filtering. There are two such problems identified among the user ratings; *shift of average ratings* and *different rating scales* (Jin and Si, 2004). The *shift of average ratings* problem is related to the fact that more tolerant users tend to provide higher ratings than more rigorous users. In the literature this problem is addressed by subtracting the mean rating of each user from his/her ratings given for items (shown in the equation). The problem of *different rating scales* occurs due to "conservative" and "liberal" users. They either tend to rate within a close range or a wider range respectively. This is accounted for by dividing the rating of each user by the variance of the ratings.

Although explicit user inputs provide grounds for accurate predictions, it is directly related to the intrusiveness of the system. In collaborative methods, new users are required to rate several number of items to build their initial user model, which is called the 'start-up' problem. Since the information is requested solely for model building, users consider this as an unnecessary effort. First, How many items should a user rate, before receiving recommendations, and secondly, what items to present for user ratings? These two issues are inter-related: the number of items presented can be reduced if their information gain is high. In other words, depending on how much the system can learn by looking at a rating, the number of items needed to be rated can be reduced. Therefore, when acquiring user ratings, selecting the initial set of items presented for user ratings is a crucial task. For example, if a movie recommender asks a new user if he/she likes 'Titanic', the system learns very little about the user. Being a very popular movie, there is a high chance of the user liking it. Therefore, knowing the new user likes it tells very little about the user. On the other hand presenting the user with an item he/she can't give an opinion (never heard of it) is too is trivial. In Rashid *et. al.* (2002) several strategies have been evaluated to recognize an effective item presentation. They are: randomly presenting lists of items, first presenting the most popular items or pure entropy based item presentation. Since these strategies alone do not perform well, they were combined to acquire more balanced strategies with higher information gain as well as adequate user ratings for user identification. Alternatively there are other plausible strategies such as combining content based methods, and use of sophisticated entropy methods.

In addition to the above discussed *new user* problem, collaborative filtering also suffers from the *new item* problem. Whenever a new item is added to the item collection that item has no ratings from any of the users. Since nobody has rated this before, the collaborative technique has no way of presenting it by correlating the user preferences. In the literature such situations are handled by combining content based methods (Balabanovic and Shoham, 1995). Although, over time, this mature user model performs well, the initial start-up problem still exists.

Apart from the initial start up problems, collaborative user models are prone to security problems such as profile injection attacks (Sandvig *et. al.*, 2007). The open nature of collaborative filtering allows attackers to inject biased ratings to force the system to provide recommendations that are advantageous to them. Although there are investigations and research being carried out to control such situations, the original problem still persists. And such remedies add up more computation overhead to the existing recommendations methodology.

In collaborative filtering, different algorithms were employed to find the possible rating towards an item. Such algorithms discussed in literature correlated users, items or demographics to generate new ratings for a user. These methods are known as (a) user-to-user correlation, (b) item-to-user correlation, (c) item-to-item correlation and (d) demographic based correlation. Each of these methods are discussed next.

User-to-user correlation

This is the traditional collaborative filtering method. In this approach, the user population is searched to identify the users who preferred items that are similar to the items preferred by the active user (for whom the recommendations are sought) (Konstan, 1997). Like-minded uses are identified in this way and the other items preferred by those users are recommended to the active user.

This traditional method of user based correlation suffers from severe sparsity and scalability problems (Sarwar *et. al.*, 2001). In large databases where the number of item records exceeds millions, a sparsity of ratings can occur. To obtain accurate predictions, each user needs to provide ratings for a considerably large percentage of the items. But if the number of items increases, this becomes an unachievable task for any user. For example, a user who has rated 1% of the items (1% of 2 million is 20,000) may not be able to get any accurate recommendations. This situation may result in poor accuracy of system recommendations. In the user based approach computations grow with both the number of users and the number of items and can lead to a scalability problem. To overcome these two limitations the collaborative techniques have followed variations such as item based collaboration.

Item-to-user correlation

Unlike the user-to-user approach, the item-to-user approach looks into the set of items rated by the current user. The idea is to identify if the target item is important to the target user. This kind of collaboration is used in News Dude (Billsus and Pazzani, 1999) to identify user's short term news story preferences. First, the similarities between the target item and the items the user has already rated are computed. This is carried out by isolating the users who have rated both the target item i and the comparing item j. Then the similarity of the two items is calculated. There are a number of different ways of computing similarity between two items (Sarwar *et. al.*, 2001). For example, in Sarwar *et. al.* (2001) correlation based similarity is computed as follows. Let the users who rated both items *i* and *j* be denoted by *u*. Then, the correlation similarity is given by,

$$sim(i, j) = \frac{\sum_{u \in U} \left(R_{u,i} - \overline{R}_i\right)\left(R_{u,j} - \overline{R}_j\right)}{\sqrt{\sum_{u \in U} \left(R_{u,i} - \overline{R}_i\right)^2} \sqrt{\sum_{u \in U} \left(R_{u,j} - \overline{R}_j\right)^2}}$$

Here, $R_{u,i}$ denotes the rating of user *u* on item *i*, $\overline{R_i}$ is the average rating of the i^{th} item.

Item-to-item correlation

In this approach, when the target user rates an item, the item population is searched to identify other items that are usually sold together with the target item (Sarwar et.al., 2001). The item descriptions are not considered, but the individual items are compared. For example, the recommendation would be “People who bought item X, also bought item Y”.

Demographic based correlation

This method requires user’s personal attributes such as age and gender to form demographic classes. First the target user (the user seeking recommendations) is mapped to a demographic class. Then, based on their demographic class, items preferred by the members of the class are recommended to the target user. The demographic based methods do not need a history of ratings as in other correlation methods. Lifestyle finder (Krulwich, 1997) uses a similar method called demographic generalization where a commercially available database of demographics has been used to classify users. Initially a dialog is used to assign the user to a demographic cluster, and then the websites and items that they may be interested in are suggested.

Pazzani (1999) attempts to find regularities among users based on their demographics. Initially user‘s personal information is extracted from their homepages, using the Winnow algorithm (Littelestone and Warmuth, 1994; Blum *et. al.*, 1995). Similar recommendations are then offered to demographically similar users. The user models utilized for similarity calculations had the form of user demographic vectors. Pazzani's experiments claim that demographic based recommendations showed an increase in accuracy over random guessing. They suggest that demographic based correlations may be combined with other information and exploited to increase the precision of predictions. This method also can be employed to solve the new user problem encountered in pure collaborative filtering methods.

When the domain objects (products) are targeted to particular user groups, (e.g. Books/CD/music/news trends among young generations etc) demographics based methods perform well. But since demographic information is most predictive of user preferences, people are reluctant to provide such information. Therefore, irrespective of the acquisition method, demographics information itself has to be considered as of high obtrusive nature.

2.6.2 Content based filtering

Content based filtering obtains user preferences towards product descriptions and attempt to retrieve the items that consist of user preferred features. This approach relies on the assumption that user's previous preferences and interests as reliable indicators for his/her future requirements. One powerful feature of content based filtering is its ability to predict relevance of items that are new to the item collection. For example, high turnover items like news articles and items in large item spaces are suitable candidates for this kind of predictions. There are three different content filtering approaches using product descriptions: (a) feature based approach, (b) utility based methods and (c) knowledge based methods.

Feature based approach

In feature based approach, the user rates the features of the items. Then the system searches the item collection for the items with highly rated features. Since content based filtering has its roots in information retrieval, most of them allow only positive ratings indicating user's preference. Typically, a feature based user model consists of a set of features and the ratings provided for each of these features. But the type of user model derived depends on the learning methods employed. In the literature, learning methods such as decision trees, k-nearest neighbor, relevance feedback based methods, linear classifiers and probabilistic methods (naïve Bayes) have been used (Billsus and Pazzani, 1999).

Calculating the similarity of items is another major issue in content based filtering. Methods such as cosine-based similarity, correlation-based similarities and adjusted cosine based similarity have been used (Sarwar *et. al.*, 2001). Content based methods demand descriptive item listings under many features, and so, are only suitable for certain domains, such as personalized book selection, where the items could be described using title, author etc. Although this allows describing the features of an item the user needs, if a certain description is missing, then the user doesn't have the chance of selecting a value for such a feature, even though it is of extreme importance to the user. Similar to collaborative user models, content based user models capture long-term preferences and become mature over time.

Utility based filtering

This method attempts to capture current user needs in the form of a utility function. Item utility is calculated using the product descriptions and corresponding weights. The items that satisfy the utility function are retrieved from a large collection of items. The user has to define his/her utility function and hence this method will help when the user is familiar with the domain and know what exactly he/she needs. Utility based methods do not build long-term user models. Each time a user searches for a new item, the system has to derive a new utility function and apply it to the objects under consideration. PersonaLogic[10] employ utility based recommendations after interviewing their customers. This method allows user to constrain even non-product attributes where users get to present his/her requirements in a more flexible way. Since deriving the utility function needs several user inputs and there is no reuse of such provided information, the information usage in utility based methods is weak (Burke, 2002a).

Knowledge based filtering

Similar to utility based methods, knowledge based methods too, do not maintain a long-term user model. The main drawback of knowledge based methods is their requirement of knowledge engineering. Such systems need knowledge about the products, users and additional expert knowledge to combine the user with appropriate products. Burke (2002a) refers to such knowledge as functional knowledge. For example, the system should know that a person requiring a family car is referring to a large size vehicle. Once all this types of knowledge is available, they attempt to map the user need with available options using their functional knowledge about the user and the products. Knowledge engineering is not only a time consuming task, it may not always capture user expectation. In the above example, although according to system knowledge a family car is equal to a large car, the user may expect a safer car. Since similarity is more complicated than it appears to be, knowledge engineering is a crucial task in knowledge based methods. On the other hand, since they do not maintain a long term user model, users have to submit their preferences each time they need personalized services, which make the this method inefficient with respect to information reuse. The success of personalization with such systems depends on the effort put into knowledge engineering and its accuracy. The above discussed facts can be categorized as positive and negative outcomes of each of the techniques.

2.7 Algorithms and Quantitative Techniques

Irrespective of the technique used to build the model, the underling statistical methods or algorithms are extremely important for the performance of a user model. In Zukerman and Albrecht (2001) number of such commonly used methodologies such as linear models, TFIDF- based models, Markov models, neural networks, classification methods, rule induction methods and Bayesian networks were described. These methods are briefly introduced below.

2.7.1 Linear methods

Linear methods use weighted sums of linearly related values to compute an unknown value. Calculation of Pearson coefficient to identify the similarity of two users is an example of using linear methods (Schafer *et. al.*, 2007). In addition, linear models are used in hybrid user modeling system when combining the recommendation scores of two different techniques (Claypool *et. al.*, 1999).

2.7.2 TFIDF- based models

Term Frequency Inverse Document Frequency (TFIDF) method is often used in the information filtering domain to rate or retrieve documents that match the user query (Salton and McGill, 1983; Salton, 1989). In TFIDF based methods, each item on the document is represented as a vector of weights. When representing a document, each weight corresponds to a term in the document. Once each document and the user query are represented as a vector, depending on the approach taken by the system, either similarity of two documents or the similarity between the query and a document is calculated using vector cosines. In the literature, short-term interest of a user was represented in the user model as a TFIDF, and used to compute a score to new stories (Billsus and Pazzani, 1999). Quickstep (Middleton *et. al.*, 2002) used TFIDF for a different purpose: to assist classification of research papers in the database, according to an ontology.

2.7.3 Markov chain models

A Markov Chain model (Russel and Norvig, 1995) is a mathematical model for describing a certain type of stochastic process that move in a sequence of phases through a set of states. When there are a number of observed events, the occurrence of the next event is predicted from the probability distribution of the observed events in the past. In user modeling systems, Markov models have been widely used to represent and analyze web navigation data. In (Borges and Levene, 2007) different lengths of web navigation sessions have been analyzed and combined to predict the next link choice of unseen user navigation sessions.

2.7.4 Neural networks

Neural networks (Haykin, 1998) have been used to represent user preferences. In content based methods the nodes of the neural net store the item descriptions and the edges have been used to represent the strength of association between features. The more representative nodes are expected to predominate in the network leading to a more accurate representation of user interests. The network is modified on the basis of the items accepted and rejected by the user. The accepted items contribute positive evidence or energy to the network whereas rejected ones contribute negative energy. For example, Jennings and Higuchi, (1992) used NNs to represent user preferences for news articles under content based approach. Boone, (1998) employed an agent, that filters e-mails through a neural network which is previously trained with feature vectors of past messages. Graef and Schaefer, 2001 used different types of neural networks to model user preferences in collaborative filtering as a solution in handling large number of users and their ratings.

2.7.5 Classification methods

In the user modeling literature, usage of both supervised and unsupervised classification methods are observed. Unsupervised classification methods were employed to automatically forming user communities among information users, to construct stereotypes (Paliouras *et. al.*, 1999). In another instance, research papers to be recommended were classified using a k-Nearest Neighbor type classifier that uses a set of example documents as a training set (Middleton *et. al.*, 2002).

2.7.6 Rule induction methods

Rule induction methods (Russel and Norvig, 1995) are used to find out item-to-item correlation in collaborative methods. These methods are similar to classification methods except that they need to be aware of the feature of the items and the class label. This information is used in forming rules for the given situation. In Lawrence *et. al.*, (2001) personalization for supermarket product recommendations were carried out using association rule mining. Item based correlation is applied based on both the expected appeal for a product and the user's past purchased items.

2.7.7 Bayesian network

Often Bayesian Networks (BN's) are formed based on a training set of data. An important property of BN's in user modelling is its ability to support both content based and collaborative methods at the same time. It is possible to form the user model by training the BN based on the existing user data and then further fine tune using the users actual preferences obtained during interactions. The model can be built within a short time period of a few hours or couple of days off line. Since it needs training, this kind of model is suitable for slow changing domains. In Ardissono *et. al.* (2004), a BN was employed to observe user's TV viewing preferences. The BN was initialized with a uniform distribution of probabilities on its nodes. Then, each time the user expresses interest in a TV program, the BN is updated by feeding it with such evidence. News Dude (Billsus and Pazzani, 1999) employed a Bayesian classifier to learn user's long-term general interests in news stories. Each news story was represented as a Boolean feature vector, where the features were a set of hand selected words. In another instance, Conati *et. al.*, (2002) employed BNs in long-term knowledge assessments in student modeling.

2.8 Obtrusiveness in personalization

As mentioned in Chapter 1, in our work we pay attention to obtrusiveness of the system in both user model building and product selection processes. Since there is no proper definition to measure obtrusiveness, in this section we describe and define the issues to control in order to minimize obtrusiveness.

An eCommerce system needs to request information from the users to ensure successful delivery of personalized contents. Therefore, it is important to capture the current user need while at the same time making it a pleasant interaction for the user. If the system provides personalized interactions, often creation of a user model is required. To build the user model system may need to request user's personal information or product related information. Alternatively, a system which provides personalization without a user model may needs to know the product preferences in the requirements elicitation process. In addition, during the search process there will be information required either to narrow down or expand the search outcomes. Finally, the system needs to obtain the user's opinion about the system choices. Although high personalization could be achieved by obtaining more information from the user, it may lead to obtrusive system-user interactions. Such situations will have a negative impact on the services provided thus negating the benefits from higher personalization. Research has been carried out in order to attempt to maximize personalization with minimal obtrusiveness by either manipulating product information or profiles of the users to reduce the number of questions (Rashid *et. al.*, 2002). As highlighted in Bergmann and Cunningham (2002), in addition to the number of questions other factors such as comprehensibility of questions, answering cost related to the question, and question clustering also contribute to the obtrusiveness of the system interactions. In the literature, work carried out on search costs in online environments claim that people are concerned about the type of facts that are requested from them (Annacker *et. al.*, 2001). They calculated a value PIC (Personal Information Content) to measure the intrusiveness of a question in an electronic sales interaction (Spiekermann *et. al.*, 2001). Based on all such ideas we argue that obtrusiveness of a system depends on the following facts.

(i) Number of questions
(ii) Comprehensibility of questions
(iii) Answering cost related to questions
(iv) Comprehensibility of question clusters
(v) Personal information content of the questions

2.8.1 Number of Questions

The number of questions directed to a user should be kept to a minimum. This can be achieved by avoiding requesting the same information more than once and by avoiding trivial questions where answers do not contribute to narrow down the search space. In user model creation process, either implicit information use or reuse of explicit information is a remedy. The number of questions in a static dialog or in a standard form filling application remains static. But this can be manipulated in dynamic dialogs, where the number of questions depends on the user's initial request and product attribute distributions.

2.8.2 Comprehensibility of Questions

Current eCommerce systems have to cater for a heterogeneous user population, where each user has a different background and experience. The vocabulary used in questions can be difficult to understand or may need technical expertise. Such difficulty can be lessened by providing users with possible answers for the system questions as options. Rather than open ended questions, provision of such facility will provide users with domain knowledge required to clearly specify their preferences.

2.8.3 Answering Cost Related to Questions

Some questions directed to the user can associate a cost; especially in the product retrieval process. For example, before answering a question related to a technical requirement of an appliance, the user needs to pay a fee to a certain body to obtain the necessary information. In such a situation, such high cost questions should be prioritized by low cost questions.

2.8.4 Comprehensibility of Question Clusters

This issue is more connected with the obtrusiveness in product retrieval. In surveys or in pre-designed electronic forms, related question sequences appear close to each other. But if providing personalized interactions, the question sequence for each user changes. Then, there should be necessary measures taken not to confuse the user by asking related questions far apart from one another. In other words, the questions should be organized into clusters. For example, when selecting a restaurant, asking the preference for each cuisine type independently will be confusing. If the questions "Do you prefer Chinese cuisine?" and "Do you prefer Malaysian cuisine?" were asked at two different points in the dialog, it may sound strange to the user. It would be more meaningful if the questions are clustered according to a logical order. For example, the question could be "What is your preferred cuisine?" and then all the different cuisine types become possible answers.

2.8.5 Personal Information Content of the Questions

When asking questions, the amount of personal information users have to reveal have an impact on the obtrusiveness of the question. As mentioned before the work by (Spiekermann *et. al.*, 2001;2003;2004) has demonstrated that the eCommerce users, answer product related questions willingly, whereas they show a concern when personal information related questions are asked. They also claim that in addition to the personal information, other factors such as perceived *legitimacy* (How necessary is it to reveal the information for successful product selection?), *importance* (How important is the outcome of the question?) and *difficulty* (How comprehensive is the question?). They argue that depending on the above three aspects, users chose either to answer or reject a question. Their work, group the questions directed to users into four categories: personal questions not necessarily required in the product selection process (abbreviated as Pd), personal questions which are required in the product selection process (Pepr), product usage related questions (U) and product attribute related questions (Peip).

The work demonstrates the online user's preference towards answering each of these four question types. Figure 2.6, shown below (obtained from Annacker *et. al.*, 2001 with permission) illustrates the percentages of question *types* presented in answered questions as the *legitimacy* of the questions varies from 'low' to 'high'.

This confirms that if the legitimacy is low, then users answer them only if the questions are less personal (either Peip or of U type questions). The users tolerate personal questions as the *legitimacy* of a question increases. It shows that if any personal questions are asked from the users, they should be convinced that those questions are necessary to select the products they prefer. If the questions are more about the products, then user concern seems less when answering a question. It is apparent in the first bar, where a high percentage of the answered questions belonging to the type Peip (product attribute related questions). This work can be considered as a guide, when preparing questions for online dialog systems. Most importantly, it demonstrates that there is an effect of personal information content over the customer's willingness in question answering during online shopping interactions.

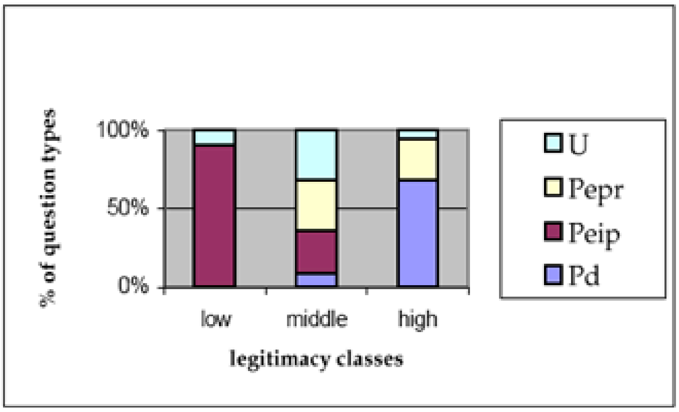

Figure 2.6 : question types – reproduced from Spiekermann *et. al.*, 2001 with permission

Furthermore, if the personal questions are to build a user model, then the resulting model should be able to provide benefits for the user effort. This issue is thoroughly analyzed in Rashid *et. al*. (2002). However, we argue if the personal information is reused within the created user model over number of domains where the user is able to enjoy the benefits of personalization in every interaction, then the users will be convinced about their efforts.

2.9 Electronic user models in eCommerce

eCommerce sites provide their customers with personalized services in number of instances such as during electronic catalogue navigation, product recommendations and target advertising, which are provided either based on a long term user model of the customer or preference information obtained for the current transaction. There are number of systems that provide personalization in the eCommerce domain. These systems can be categorized according to the services they deliver Jameson, 2001.

(i) Recommender systems that help users find the items they may like using a long-term adaptive use model (SETA, Firefly, Amazon.com)

(ii) Recommender systems that help navigate product spaces (PersonaLogic[10], Tête-à-tête (Guttman and Maes, 1999), Entrée (Burke, 2002a))

(iii)Information management systems – which provide security and manage user data reusing ones entered personal information, (Ms Passport,[6] Liberty Alliance,[7] Digital me[9]).

Interactive dialogs (e.g. Gateway Virtual Notebook Expert), life like characters/MIAW agents (e.g. Artificial Life) or 3D VRML (Virtual Reality Modeling Language) product spaces, which provide a user friendly atmosphere to guide users in information spaces

Since the focus of the thesis is on 'User Models' the first two types of systems are discussed in detail. Since the aim of the thesis is to design and develop a user model which can be efficiently and effectively used in product searches, we also consider the positive contributions from the other types of systems. Some of these systems described are academic prototypes while others are commercially available. Figure 2.7 presents a representative sample of the main eCommerce systems found in the literature that are either commercial systems or academic prototypes.

2.9.1 Prototype Systems

Academic prototype systems shown in Figure 2.7 are discussed next. Few of these systems are currently available online.

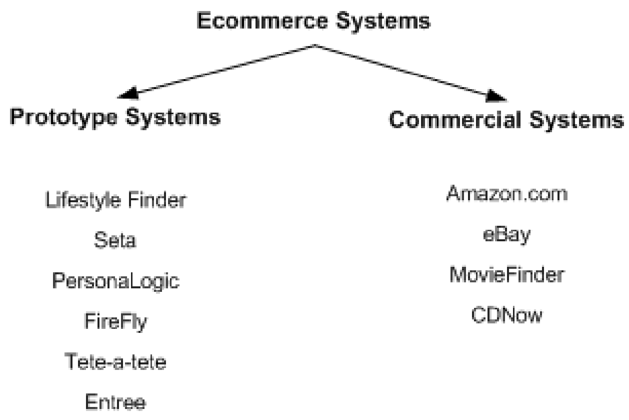

Figure 2.7 : Categories and examples of eCommerce systems

Lifestyle Finder

Lifestyle Finder (Krulwich, 1997) recommends interesting websites to its users that selling a range of products and services. It collects explicit user inputs to map the user to a single demographic cluster out of an existing collection of demographic clusters. These clusters have been obtained using a commercially available database (PRIZM database by Claritas Corporation) of demographic data that encompasses the interests of people nationwide in the USA. The PRIZM system divides the population of the United States into 62 demographic clusters according to their purchasing history, lifestyle characteristics and survey responses. Therefore, the interests of people living in a geographical suburb get clustered. The resulting user model consists of all the preferences specific to the mapped cluster of people. This method of user model building is called "demographic generalization". Although not obvious, this model uses both content based and collaborative techniques. Collaboration is achieved via the common clusters. Individual user models are formed partially mapping users to segments. Then, the system takes a content based approach, searching for the interesting websites according to the user model.

SETA

SETA (Ardissono *et. al.*, 1999; Ardissono and Goy, 2000) is a web based multiagent system that dynamically generates personalized electronic product catalogues. Apart from recommending, SETA also helps the user to navigate in the large and unfamiliar product space while providing explanations. SETA explicitly collects demographic information at the beginning of the interaction to map the user to a stereotype. The stereotypes are fixed and based on a prior survey data. The stereotypes are characterized by a classification part and a predictive part. The classification part concerns socio-demographic characteristics while the predictive part is made of corresponding personality traits and user preferences towards product features. The user model consists of user preferences towards product features which are initialized by the stereotypes.

Once the initial user model is formed based on stereotypes, the system interactively generates catalogue pages demonstrating details of the items that the user may be interested in. User interest is obtained based on the user's behavior during the product navigation. For example, requesting more information on a particular item feature (photo copying facility in fax machines) will indicate the interest in that feature. Therefore, this will result in presenting the user with more items with the similar capabilities (photocopiers). The user model will be updated indicating user interest in the new item category.

PersonaLogic[10]

PersonaLogic[10] is a tool that helps users to find out the products of their interest by guiding them through a large product feature space. It takes a content based-utility based approach by allowing the users to specify constraints on product features. Initially the users are supposed to answer a detailed questionnaire regarding product features and provide their preferences on a scale of five, from "no option" to "extreme" (Maes *et. al.*, 1989). PersonaLogic[10] filters products that do not meet the hard constraints and prioritize the remaining products using soft constraints which need not be completely satisfied. Finally users are presented with an ordered list of products that satisfy all of the user defined constraints. Since PersonaLogic[10] does not maintain a user model, each new product search is freshly carried out guided by the user inputs.

Firefly/Ringo

Firefly (Shardanand and Maes, 1995) is a recommender using the user-to-user collaborative method. It compares the user's product rating with those of the other users and recommends the items preferred by 'nearest neighbors'. This system was used to recommend products such as music and books.

Tête-à-tête

Tête-à-tête (Guttman and Maes, 1999) is an agent mediated eCommerce system that helps users to find both the suitable product and the most suitable vendor to purchase that product. It is capable of obtaining user preferences towards items features as well as non-item features such as delivery schedule and vendor reliability. Therefore, it does more than a recommender system. In the literature Tête-à-tête is referred to as a "product and merchant brokering" system (Guttman *et. al.*, 1998).

Entrée

Entrée (Burke, 2002a) is a restaurant recommender system. It belongs to a family of recommender systems called "find me" systems which offers an easy-to-use interface. Entrée was designed to recommend restaurants to conference attendees from different cities (in the US). Therefore, the users are supposed to provide an example restaurant that they are familiar within their home city to find a similar restaurant in another city. Alternatively preferred values for a set of features are accepted as the initial input. Then the navigation is conducted according to user's critiques. User can criticize a system selection along predefined set of features such as the cost, niceness, quietness, décor and liveliness by pressing a button. Entrée can be categorized as a content based-knowledge based recommender system. Entrée depends on a considerable amount of knowledge engineering.

2.9.2 Commercial Systems

Commercial user modeling systems shown in Figure 2.7 are discussed next. Amazon.com[2] uses a long-term user model and provides users with recommendations, while the others offer personalized services without using a model of the user.

Amazon.com[2]

Amazon.com is a pioneer in the area of commercial product recommendation. It sells a variety of items and provides their customers with recommendations as an additional service to increase their sales. They employ two different types of recommendations which are based on collaborative filtering/user-to-user correlation and item-to-item correlation.

The recommendations start after system registration by providing an email address and password. The customers 'Your recent history" page contain all the recent searches while the "Customers who bought items in your recent search also bought" list contains the recommendations for other items. The *books* section of Amazon.com provides detailed information about each book with respect to its contents and its purchase history. For each book, two lists of recommendations are provided: one list for the other books by the same author and the other list recommend books frequently purchased by customers who purchased the selected book (user-to-user correlation). For better recommendations customers are encouraged to provide ratings for the books they have already purchased. Customers can perform a search to locate the books that are related to the search topic and then allowed to rate them on a 1-5 scale.

CDNow is a store connected with Amazon.com under the category of *music*. CDs searched and selected for browsing offer several personalized features such as, "Better together", "Customers who bought this item also bought", "Customers viewing this page may also interested in these sponsored links", "Looking for product X?", and "What do customers ultimately buy after viewing items like this". It also encourage the customer to rate the item under "Rate this Item to improve your recommendations" in a scale of 1-5.

Although Amazon.com provides recommendations both across domains and within a single domain, only the recommendations provided within a single domain (such as books or CDs) looks promising while cross domain recommendations do not make sense most of the time.

eBay[3]

eBay.com™ (www.ebay.com) operates in 33 countries and owns a large customer base of 1.35 million. eBay facilitate an online market place for people to buy and sell their products. eBay does not provide recommendations for commodities but provide recommendations on buyers and sellers. Both buyers and sellers are allowed to contribute feedback comments about the members they have done business with. A satisfaction rating is a combination about their performance along the lines of correctness in item description, reasonability of postal cost, communication etc. The resulting rating is positive, negative or neutral followed by a detailed comment. There is a marking scheme and an award scheme connected with the customer reliability percentage. This performance profile is used to obtain an understanding of the seller or buyer before bidding for an item. Any seller can block buyers with a bad reputation from bidding for their commodities.

eBay also offers personalized facilities such as My eBay and Bid Assistant. My eBay is a central place where the users can store the items they are currently watching, past purchases, items bid, items won, ratings received etc. Since the information in My eBay is not used for inferencing, it cannot be called a “model of the user”. The Bid Assistant is a feature that performs automatic bidding on behalf of the user. Once the user set up maximum bids for a group of items, the bid assistant follows item after item bidding up to the maximum bid.

MovieFinder.com[4]

MovieFinder.com is an online system that helps users to find currently showing movies. It is maintained by E! Online.

This site allows the user to register with the site and then rate any movies they have watched. Then each movie gets two letter ratings (A-F) indicated by a letter such as “A+”, “A-“, one from the editors of the site and the other averaged over all customer ratings.

This site also provides the top ten movies of a selected category of movies. When the user selected the preferred category of movies (based on site’s categorization), the top ten movies of that category defined by the editors can be seen. MovieFinder.com recommends movies in general, but no user models are maintained for its users.

2.10 Issues and Limitations

The systems discussed in sections 2.9 follow different approaches in handling the massive product collections in electronic markets. In this section such past and current models of personalization are summarized and analyzed with a view to provide the foundations and justification for the work presented in this thesis.

We list the positive and negative effects observed in the existing systems using the following broad topics. These six topics were selected based on the existing work. Directly related notable examples include (Towle and Quinn, 2000; Burke, 2002a; Kay *et. al.*, 2002; Pu and Kumar, 2004; Kobsa, 2007).

(i) Reusability of the user model in multiple domains

(ii) Necessity of knowledge engineering

(iii) System Adaptability

(iv) Flexibility in capturing user preferences

(v) Use of personal or demographic data

(vi) Sensitivity to individuality

These are discussed below in detail; showing how these issues are inter-related and how the same problem impose both negative and positive effects on the system performance.

2.10.1 Reusability of the user model in multiple domains

Acquiring user information, especially explicit information to create the user model is costly. For example, requesting demographics, feature preferences or ratings to initiate personalization may be considered as a burden by the users. In the literature most of the user models created is suitable for predicting user behavior in a single domain. If reusability of the user models in multiple domains is possible, that will result in not requiring user information for each and every domain.

Theoretically, any system using collaborative methods can use the available ratings to provide recommendations in multiple domains, since they consider only the similarity of the users. For example, the recommendation would be of the type "people who bought X, also bought Y". Burke (2002a), refers to this characteristic in collaborative techniques as the "ability to identify cross genre niches". But similar behaving individuals in a given domain cannot be expected to behave the same in all the other domains. For example, two users who browse the same restaurant may prefer another restaurant in common, but the chances of them both purchasing a digital camera may be very narrow. Therefore, if the ratings are not acquired for a particular domain (e.g. for digital cameras) then such recommendations generated with no other solid evidence (such as ratings in that domain) other than similarity of the users or items may result in error.

Among the content based approaches reusability is achieved by developing the system as a shell. For example, SETA's framework was developed in a more generic way as a reusable shell (Ardissono *et. al.*, 2001b). Still the survey based stereotypes which were used to initially populate the user model are not reusable in any other domain than electronic equipment. Therefore, the user model or sections or components of the user model are not reusable in any other domain.

As previously mentioned information management systems such as MS passport[6] and Liberty Alliance[7] facilitate reuse of user personal information in a number of domains. But these cannot be labeled as user modeling systems, since they lack inferencing capability. In the literature one instance of user model reuse is exhibited in the Personis (Kay *et. al.*, 2002) user modeling server. Personis maintains application dependent 'Personas' that are connected with the main user model. With user's authorization, information in the main user model is passed on to applications to be included in their own Persona. Similarly, user information collected in an application dependent persona can be passed back to the main user model to be kept for disposal of future applications. The server operates a number of 'resolvers' which have the ability to interpret user data. Different applications use their choice of resolvers depending on the type of interpretations they require. Although these resolvers require knowledge engineering, Personis can be highlighted as a system that efficiently utilized user information.

2.10.2 Necessity of knowledge engineering

Depending on the recommendation technique and the nature of the information elicited from the user, some of the above systems require pre-processed information or availability of knowledge at the initiation. Content based methods needs descriptive data about the products and in some occasions about the users. PersonaLogic[10], Entrée (Burke, 2002a) and tête-à-tête (Guttman and Maes, 1999) require product descriptions while SETA utilizes both descriptive information about the user and the products. For example, SETA uses a set of predefined stereotypes that match user information to product features. Being a knowledge-based system, Entrée requires additional knowledge to match the user requirements with the suitable items. Burke (2002a) described this as *Entree knows that a need for a romantic dinner spot could be met by a restaurant that is "quiet with an ocean view"*.

Systems based on collaborative filtering such as Amazon.com or firefly (Shardanand and Maes, 1995) don't require knowledge about the products or descriptive information about the users. However, all collaborative systems need a large historic database to start recommending, since the initial user ratings needs to be compared. If not, there is no way of evaluating the initial ratings of the early users.

2.10.3 System adaptability

Adaptability of user modeling systems has both benefits and drawbacks. Systems such as Amazon.com, SETA (Ardissono *et. al.*, 1999), and FireFly (Shardanand and Maes, 1995) maintain adaptive user models that become more stable over time. Systems following collaborative approach such as Amazon.com and FireFly collect more and more ratings towards items during each transaction with the system. As the number of ratings provided by the user grows, the system is able to accurately map users to their nearest neighbors. In content based approaches such as SETA, the importance value of parameters in the user model grows according to demonstrated user preferences. In both instances when a user indicates a certain preference toward an item or item features, it is introduced to the user model and becomes established with further positive ratings. Although adaptability is a positive quality, this results in the difficulty in differentiating user's current requirements from his/her past preferences. Generally, in a long term user model the initial user preferences are subject to temporal decay when the user requirements changes according to the market changes (Towle and Quinn, 2000). In all adaptive user models, user's new habits and preferences are gradually introduced into the user model while there are mechanisms to make the influence of old habits less prominent in the user model. But the older preferences are not wiped out completely from the user model and can interfere with the recommendations. Burke (2002a)[11] refers to this quality as the "stability vs plasticity" problem. Burk explains the problem with an example.

"A steak-eater who becomes a vegetarian will continue to get steakhouse recommendations from a content-based or collaborative recommender for some time, until newer ratings have the chance to tip the scales."

Among the above described systems, Lifestyle Finder, Entrée, PersonaLogic[10] and Tête-à-tête do not maintain long term user models and do not suffer from the described problem. As a drawback they do not have the learning ability and require user effort in every interaction for specifying the current request. Even when the user interacts with the system several times, he/she does not receive any guidance or suggestions from the system as there is no learning ability present.

2.10.4 Flexibility in capturing user preferences

A user modeling system should be able to provide users with personalized services from the point of user registration. In some approaches, start up needs considerable user effort. For example, in collaborative methods a new user needs to provide the system with ratings towards a list of system selected items as to clarify his/her preferences, since the system needs sufficient evidence to map the user to another user with similar ratings. A user with only a few ratings cannot be mapped effectively. Therefore, the new user should rate a number of items to express his/her preferences. In addition the set of items user initially rate has to represent the entire product population. The most difficult of all, the items presented should be familiar to the user or else user will not be able to rate these items.

---

[11] Burke, R. (2002) Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction,* VOL 12pg 331-370.

Therefore, in collaborative methods, the set of items the user rates at the beginning needs careful selection.

In content based methods this problem does not occur, since the user query specifying preferred item features is used to do the retrieval. For example, in SETA (Ardissono *et. al.*, 1999), new users have their user model populated by stereotypic information for personalized services immediately after registration.

Every time a user visits a website, he/she has a requirement defined either completely or partially in their mind. There should be a way of expressing it to the system rather than letting the system guess. In this regard, systems following utility based method such as PersonaLogic[10] and Tête-à-tête (Guttman and Maes, 1999), has the greatest strength in providing the user to specify his/her needs explicitly as a utility function. Utility functions formed in these systems are MAUT (Multi-Attribute Utility Theory) based and are able to capture preference attributes along with weights. The only drawback associated with this method is that both systems do not maintain user models. Therefore, each time the user browses for items, he/she needs to perform the tedious task of redefining the utility function.

Critique based systems such as Entrée (Burke, 2002a) SmartClient (Viappiani *et. al.*, 2006) allows a much flexible interface for capturing the initial requests. Entrée allows the user to provide either an example item known or preferred values for a set of system selected attributes. Compared to Entrée, SmartClient provide the user with a more flexible interface by leaving the user to specify preference for any attribute. However, neither of these systems employs a user model and the user is supposed to provide preference information during each interaction. Adaptive Place Advisor (Thompson *et. al.*, 2002) is a critique based dialog system which uses a user model. However, the underlying user model is primitive and may not be suitable for handling online massive product catalogues.

In SETA (Ardissono *et. al.*, 1999), the system guesses the user's current need by observing the browsing behavior. A new user who is unfamiliar with the system and trying to understand system operation may not properly express his/her requirements via browsing behavior. Therefore, direct specification of the user request is more effective than indirectly presenting user requests to the system.

In a collaborative approach, when the user request is presented as ratings towards items or item features, if the current need is dissimilar to previous expected behavior, then the system may get confused with contradicting preferences. For example, a user who always prefers restaurants which do not offer alcohol will get grouped into a set of neighbors who are similar. If he/she decides to take a friend to a restaurant which offers alcohol and then rated a restaurant known for fabulous selection of alcohol, this will map him/her away from his/her usual group of people. Although he/she indicated preferences toward restaurants known for good liquor, his/her past performance will result in keeping him/her away from such preference groups. As such the user will receive recommendations which do not belong to either groups resulting in dissatisfaction. Although not in the eCommerce domain, in Billsus and Pazzani (1999), a hybrid of short term and long term user models were used for news story classification. The idea was to capture the frequently changing preferences in the short term model and the long term preferences in the other.

Another issue connected with lack of flexibility in capturing user preference is portfolio effects. In other words, when items are presented to users only according to the user model and without a recent request it can be something that the user already possesses (Prasad, 2005). This situation does not occur in a preference based product search where the user always declares the need for a certain item.

2.10.5 Use of personal or demographic data

Due to the privacy concerns, online users are reluctant to reveal their personal information. Therefore, systems that use only anonymous user information are much appreciated by the users. Systems such as Lifestyle Finder (Krulwich, 1997) and SETA (Ardissono *et. al.*, 1999) use demographic information to form the user model. But systems with a collaborative approach base their user models only on the similarity of users and therefore, require only the anonymous ratings.

Demographics are powerful information that can reveal the reality about the user. Although privacy concerns exist, use of demographic data solves several other issues connected with user models. For instance if demographic information was present, the ‘new user’ problem discussed in section 2.9.2, do not arise. SETA (Ardissono *et. al.*, 1999) use demographic information to map users to initial stereotypes. In addition, use of demographic information in user models incorporate user's personality traits into the user model. Although it is not specifically known what demographics contribute the most, a number of researchers have investigated this issue (Brusilovsky, 2001).

Furthermore, most of the systems that provide personalization need user registration in their sites. For this purpose, users are requested for their personal information such as the birth date, profession etc. For example, eBay do not use demographics in providing personalization, but still collect detailed demographics to grant membership. Therefore, requesting demographics seems like an unavoidable step in providing personalization.

2.10.6 Sensitivity to individuality

This issue is somewhat related to the above discussed "Flexibility in capturing user preference" issue. Sensitivity of the user model to individual's interests grows with the explicit user inputs. In collaborative approaches when similarity of the user ratings are considered to locate preferences, such implicit information do not retrieve the reasons behind a rating (Towle and Quinn, 2000). For example, two users with similar ratings for a restaurant may do so one being contented with the cuisine it offers and the other contented with the atmosphere. If the other restaurants preferred by the first user are recommended to the second user there could be a greater chance for dissatisfaction. All content based approaches are free of this problem.

Sensitivity to individual's preferences is lost when users are offered group preferences. A problem related to this is known as the "Gray sheep" (Claypool *et. al.*, 1999) problem. Lifestyle finder allocates users to user groups. Sometimes individuals are difficult to map to a single cluster or they only belong to the 'edge' of a group. In collaborative approach there are individuals who do not match any neighbors or groups consistently. People who exhibit such unique behavior will rarely receive accurate predictions (Claypool *et. al.*, 1999). The same results can be expected in stereotype usage and any demographic or other group based recommendations.

Electronic shops do not need shelf space. Therefore, the variety and volume of products can grow without a limit. In the collaborative approach, when new items arrive, there should be methods to retrieve ratings for such items, or they cannot be recommended to people. To overcome this problem, FireFly/Ringo (Shardanand and Maes, 1995) chose the lists of items for initial user ratings following a different approach. Half of the item list was selected from the most popular items while the rest of the items were randomly selected from the entire item database. But still new items will take time to obtain a sufficient number of ratings. Systems following any of the content based approaches do not face this problem. Since the items are retrieved based on the feature values, any item satisfying the query are presented to the user.

2.11 Limitations of user models in eCommerce

In the previous section, six broad issues that are associated with current eCommerce personalization efforts were discussed. As mentioned previously these six issues were selected mainly based on the work by (Towle and Quinn, 2000; Burke, 2002a; Kay *et. al.*, 2002; Pu, 2004; Kobsa, 2007). We believe that the six aspects discussed in the previous section need to be carefully analyzed when forming a new user model architecture to provide personalization in current and future eCommerce environments. Figure 2.8 lists the above aspects on the left, and highlights how they can be manipulated to come up with better performing user model architecture.

On the left are the facts to decide when designing a user model. On the right the suitable approaches are given. If these approaches are combined the resulting user model can be described as follows.

"It should be a component based user model, which has the ability to capture complex user needs in a number of layers. This facilitates the components to function separate from one another and thereby reuse the existing information when required. The user model should initiate with pre-engineered knowledge about the user to be able to provide personalization from the start of user interactions. It should be able to capture the individuality of the user over time and hence be able to capture and adapt to the user's changing needs. The user model should follow a content based approach allowing the user to declare his/her needs relating them to product properties. This will also result in a more meaningful user model and will facilitate information reuse across domains."

In the rest of the thesis we design and investigate a user model fulfilling the above requirements.

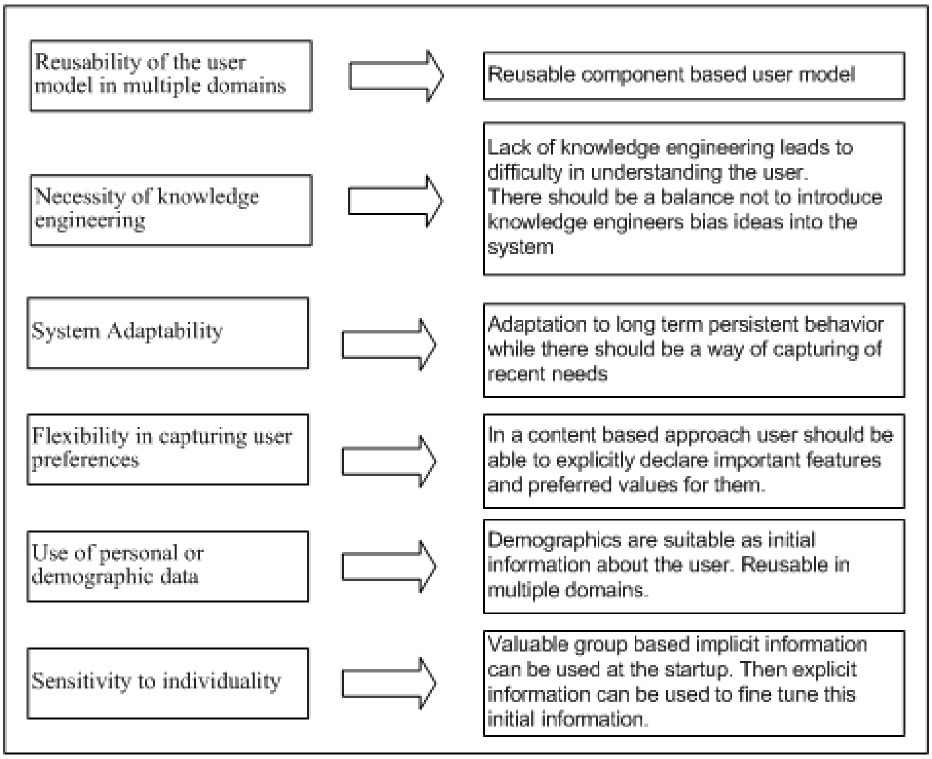

Figure 2.8: A mapping of the user model design considerations and approaches

2.12 Summary

This chapter presented a detailed survey of the user modeling systems that appear in the literature. Initially the application areas that employ user models were presented. Then, the dimensions of user models were discussed. Next, the existing user models were analyzed with respect to, how unobtrusiveness in user model creation is handled, methods of information gathering to build user models, contents of user models and user modeling techniques. The existing prototype and commercial systems employing user models in the eCommerce were discussed. Finally, the limitations that exist in the current eCommerce user models and how such limitations can be overcome to meet the current and future eCommerce environment were presented. In the next chapter, a conceptual model which can address the highlighted issues is presented.

Chapter 3

A Layered User Model for Interpreting Consumer Purchase Behavior

In the previous chapter, existing electronic user models have been discussed at length. Such analysis revealed that different perspectives of user information have been exploited in current user models. As a result, these user models exhibit positive as well as negative aspects in capturing user behavior. If varying perceptions about users are to be combined in a single user model, then the complexity of the user behavior should be captured. To accomplish such a single user model, further study and investigation is required to identify the user characteristics (that describe user expectations), and then the types of information that can capture such characteristics. It has been emphasized that a descriptive content based user model is a promising technique to address the difficulties with personalization in eCommerce.

In this chapter, novel concept of a Layered User Model (*LUM*) is presented. The preceding sections justify the novel user model by argument, basing the model on varying theories. The introduction of the novel concept of a layered user model presented in this chapter provides the foundation for the rest of the thesis.

In the next section of this chapter, the complexity of consumer behavior in current commercial environments is discussed. The novel concept of a layered user model is presented and described in section 3.2. Section 3.3 analyzes the existing consumer buying behavior models to identify the user characteristics they capture. This is carried out in order to justify the information layers of the novel user model. In section 3.4, identification of behavioral categories and related user information that are important to understand the complex user behavior are further clarified. Section 3.5 discusses and justifies identification of consumer stereotypes in eCommerce purchasing. This section further confirms user information categories that are important for a user model providing supporting theories. In section 3.6, modeling domain based purchase behavior is discussed. Finally, the chapter is summarized in section 3.7.

3.1 Complexity of Consumer Behavior

Consumer buying behavior has been analyzed at length in the field of economics under market research and behavioral theories. According to (Fink and Kobsa, 2000), these early user models have focused on traditional brick and mortar markets with their inherent simplicity (e.g. consumer behavior can be predicted from a few key characteristics), linearity (i.e., future consumer behavior can be predicted from the past behavior), and time invariance (i.e., market rules always apply). In such an environment, knowing and remembering a consumer and serving him according to individuality was possible. The consumers were thus expected to behave in what would be considered as 'rational' buying behavior. Such rationality is partially enforced by constraints such as limited number of items available, difficulty (distance etc) in accessing items, and time restrictions. In the current volatile eCommerce market, user expectations change rapidly. Reasons for such changes include the vast number of options available, globalization and also the aggressive marketing campaigns and gimmicks for attracting customer interest. These sophisticated marketing strategies cleverly use human behavioral theories and human psychology to 'infiltrate' the thinking process of consumers. This results in individual peculiarities making a bigger impact on purchase decisions. Such behavior could be considered as 'irrational' purchase decisions when they contrast and/or contradict the expected or traditional behavior of an individual.

Due to such 'irrational' behavior of consumers, early consumer behavioral models based on traditional market segmentation are becoming less useful in providing adequate personalization in current electronic markets. Due to these factors, from the vendor's point of view, understanding consumers becomes a challenge. As pointed out by (Fink and Kobsa, 2000), to address this challenge, the traditional user models needs to be complemented by incorporating latest on-line information about consumers in electronic markets. Although eCommerce consumers interact in a different environment, many of the human characteristics and thinking patterns which influence traditional purchase decisions will still be valid. Therefore, even though the focus of this thesis is to develop individualized electronic user models, the ideas developed in the economics/market research and human behavior based models are incorporated to determine the information content of the new user model.

Based on the discussion carried out so far, to determine the information content of the new user model the following two questions are raised.

(a) What behavioral categories should be captured?

If the user's future buying behavior is to be predicted based on the past behavior, it is important to keep track of the past transactions. However, what sort of past behavioral patterns will be able to reveal the future behavior remains a question. For example, should it be long term general buying behavior irrespective of the domain, or domain specific individual behavior, or is it the behavior of groups of people?

(b) What user information is adequate to capture such behavior?

Once behavioral categories are identified, it is important to investigate what information about the user will be adequate to discover such patterns. For example, should user demographics, or information regarding users past purchases or implicit information in log files with user's click streams need to be collected?

To discover the answers to above questions, existing user models needed to be analyzed. User models are developed in both Market Research (MR) and Information Technology (IT) areas of research. The models developed in MR are theoretical models that are mainly meant to be used for advertising and marketing purposes. Consumer behavioral studies have been extensively carried out under market research for this purpose. Since these models are generally based on data collected by surveys over long periods of time, they tend to be quite accurate if there are no drastic changes in trends in the short term.

While buyer behavior analysis is a mature area of research in MR, user modeling is a relatively new area in information technology dating back to late 1970s. The user models in IT are developed to be used as each individual's preference profiles. As such these user models were mainly used to avoid information overload when individuals electronically search for required commodities. Although developed separately, there are similarities in both types of user models since both attempts to capture complex human buying behavior to come up with predictions regarding buyer preferences.

The novel user model presented in this thesis, considered the existing work in both areas IT and MR in its design, by raising and answering the above two questions (a) and (b) in the both backgrounds. Based on the outcome of the solutions to (a) and (b), the information content of the novel user model was determined. In the next section, the derived novel user model architecture is presented. The information content of the user model with regard to the behavioral categories mentioned in above (a) and the user information required to capture such behaviors (which is highlighted in (b)) are explained in section 3.3.

3.2 A Three Layered Model to Represent Consumer/User Behavior

According to the current user modeling attempts in both IT and MR areas, to effectively discover a user model, information should be collected considering multiple factors such as demographics, purchasing history and current requirement. The user models in MR prioritize the usage of segmentation methodologies to identify user groups rather than requirements of individual users. In contrast, IT user models concentrated more on the domain based requirements of individuals. As a result, existing IT user models concentrated on narrower domains when modeling the users. By doing so, such user models expected to increase the accuracy of personalization (Chapter 2).

Purchasing behavior of an individual is based on a combination of factors such as demographics, domain based expectations and impulsive transactions. Therefore, each individual can demonstrate a unique combination of these aspects. To achieve finer personalization, progressing further from stereotyping and/or collaborative filtering, such individuality will have to be captured in user models. To accommodate such requirements, this thesis proposes a novel user model where each information layer captures different categories of information.

The proposed architecture allows modeling highly complex individual purchasing behavior not as an aggregate behavior, but by capturing the elements which contribute to such behavior. Each of this separated information layers can be used to better understand the fluctuations and/or changes in the behavioral categories. In addition, the model focuses on combining the elements together for individuals, creating more individualized profiles. The three layers of information within the novel user model are named as follows.

(i) Personal information (Demographics) based general consumer buying behavior information layer (PI Layer – Personal Information Layer)

(ii) Information layer on buying preferences in specific domains (DI Layer – Domain based Information Layer)

(iii) Information layer on transaction based needs for each interaction (TI Layer – Transaction based Information Layer).

The first and top most information layer captures more general user behavior through slow changing demographics. Since the demographics are used, this becomes the expected behavior of an individual according to his/her actual 'position and ability'. Such information exhibits their general trends and abilities that are valid in any purchasing domain.

The next layer captures the user behavior in a given application domain. This behavior either confirms or contradicts the actual expected behavior in the top layer, which was initially derived on the demographics. This information is formed logically combining the user's individual transaction information. However, combined transactions of a buyer with impulsive purchases will result in less useful domain behavior. Therefore, tracking each transaction details for future references is important. This requires another layer of information.

The last information layer captures the short term needs of the consumer. Information in the last layer may not agree with the more general behavior in the first layer or with the expected domain centric behavior in the second information layer. The reason being, users may deviate from their usual behavior due to impulsive purchases caused by mood changes or unusual circumstances. Recording transaction dependent information in the user model allows future analysis of such information. Such analysis of transaction information may reveal user's tendency to be affected by the impulses caused by the environment (such as advertising). In addition, transaction information is extremely valuable to locate the user's current need.

(i) Figure 3.1: demonstrates how the information layers are formed in the novel user model.

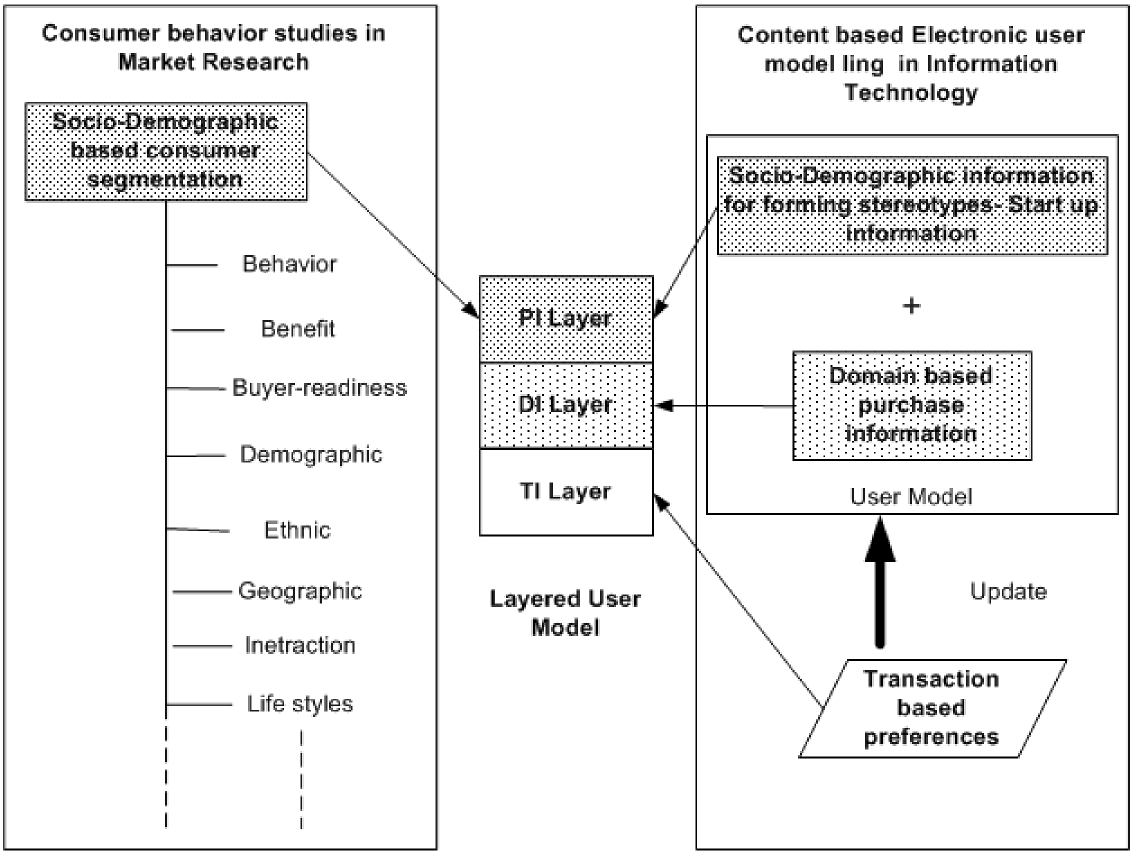

Figure 3.1: Information contribution to the model from MR and IT areas of research

Contributions from the two research areas (market research in MR and user modeling in IT), towards user model layers is reflected as follows.

(i) In market research and consumer behavior studies, user models are based on segmentation.

(ii) IT based electronic user models consists of information such as start-up information, and domain based user preferences. Updating such information is carried out based on user preferences obtained during system-user interactions. After exploiting such information in user model update process, current user models discard the original transaction information without recording in the user model.

Each proposed information layer corresponds to existing work as follows.

(i) Personal information layer is based on the ideas from both areas. In IT based electronic models, need of initial start-up information and successful segmentation and consumer behavior theories in MR background validate the first layer.

(ii) Domain information layer is similar to the domain based information maintained in electronic models.

(iii) Transaction based information is recorded in the third layer (which add more value to the user model as an information source for future analysis).

In the next sections, derivation of the novel model is presented. Selection of each behavioral category and corresponding information type (reference to questions (a) and (b) in section 3.1) to include in the user model are justified highlighting information usage in existing user modeling approaches.

3.3 Analysis and Capturing Consumer Purchase Behavior

In the next two subsections, user models employed in IT and MR areas are discussed and analyzed to argue the strength of the above selected three information layers. The information content of the user models are analyzed with regard to the behavioral categories mentioned in section 3.1 (a) and the user information required to capture such behaviors (which is highlighted in (b)).

3.3.1 Consumer Behavior Analysis and user models in MR Based Research

As a mature area of research, the study of consumers helps firms and organizations to improve their marketing strategies by understanding the behavior of consumers. Since consumer behavior is extremely complex to understand, in dynamic product markets, analyzing consumer behavior has been carried out extensively throughout the past years. Notable examples in the area includes (Patel and Schlijper, 2003, Katona, 1968, Devetag, 1999 and Curtin, 1982).

In the past it was possible to identify the target audience using few demographics such as age, education and income. Consumer markets could also be targeted on the basis of prior purchase behavior. If an item was bought during a previous transaction, "brand loyalty" was assumed as the basis for future consumption patterns. With increasing education, income and social mobility, acceptance of corporate values have decreased during the recent past. Instead, an increase in the degree of individualization, diversity, difference and personal development has been observed.

To capture new trends, several consumer behavioral models have been proposed taking different approaches and focusing on different factors. According to Patel and Schlijper, (2003), the main factors considered are values, lifestyles, life stages, personality, need states (desired benefits), demographics, purchasing patterns, and culture. Since these factors interact to influence consumer buying behavior, some of the models have been based on a combination of factors. During the past years, several segmentation topologies such as Vals[12] (value and lifestyle survey), Roy Morgan Value Segments,[13] and Experian-Global MOSAIC[14] have been born. These topologies are discussed below.

Stanford Research Institute (SRI) has developed the Vals system which defines different segments of the population via a questionnaire measuring primarily values and lifestyle choices, and to a lesser extent cultural and demographic aspects. Vals try to capture the psychological similarities and differences between consumers and how these similarities and differences influence the choices consumers make. There are eight key segments formed describing people and behaviors at group levels. Individuals reflect the characteristic behaviors of these groups in varying degrees. Some people are archetypal, with mindset and behaviors that represent the core of the segment. Others reflect some, yet not all, tendencies of the segment.

Vals survey primarily categorize individuals into the closest category and then present the other possible categories where each individual belongs to each category to a certain

[12] VALS-system, SRI Consulting, The Value and LifeStyle survey, www.srlc-bi.com/VALS/presurvey.shtml

[13] Roy-Morgan-Value-Segments, Developed in conjunction with Colin Benjamin and the Horizons Network, 1997, http://www.roymorgan.com/products/values-segments/values-segments.cfm

[14] Experian-Global-MOSAIC, 2007, http://www.segmenta.no/page?id=2324

degree. For example, one can primarily belong to “Thinker” type and secondarily belong to “Achiever” type. This implies that person has some of “Thinker” behavior and some of “Achiever” behavior.

Roy Morgan Value Segments^{TM*} is a tool with a theoretical foundation which is developed in Australia. It has been tested and found robust internationally, as well as in Australia. They identify ten segments grouping Australians along four human social dimensions (*Individualism, Life Satisfaction, Conservatism, and Innovation*) and two dimensions (*Quality and Price expectations*) that ground the segments in market place reality. These six dimensions form a cross containing the ten mindset segments of the Australian population based on the deeper drivers of choice and change - their values and fundamental ways of approaching the world. This segmentation is derived from a Single Source[15] database. Therefore, it has the ability of profiling the consumers based on the things they do, the brands they choose, the media they consume. Therefore, Roy Morgan Value Segments^{TM}* model can be analyzed and used in two different ways - to examine the responses of individual segments (a place on the map) or to examine the whole map and the way in which the interrelationship of issues has an impact on people saying yes or no. Figure 3.2 shows how the clusters are plotted against the six dimensions.

Clustering systems such as Vals[12] and Roy Morgan Value Segments[13] are useful for identifying different consumer types within similar societies. As the society changes, the clusters and their predictions change. Due to this reason, Vals has introduced country specific systems such as Japan-Vals and UK-Vals (Solomon *et. al.*, 2007).

In contrast Experian-Global MOSAIC analyses consumers in 19 countries. It is a consistent segmentation system that covers over 284 million of the world’s households. It is based on a simple proposition that the world's cities share common patterns of residential segregation. Experian-Global MOSAIC has identified 10 distinct types of residential neighborhoods, each with a distinctive set of values, motivations and consumer preferences, which can be found in each of the countries. The 10 segments are, namely: sophisticated singles, bourgeois prosperity, career and family, comfortable retirement, routine service workers, hard working blue collar, metropolitan strugglers, low income elders, post

[15] See URL http://www.roymorgan.com/products/single-source/single-source_home.cfm

industrial survivors, and rural inheritance. Global MOSAIC used over 600 variables as input information to form the consumer segments, including demographics and socioeconomics values.

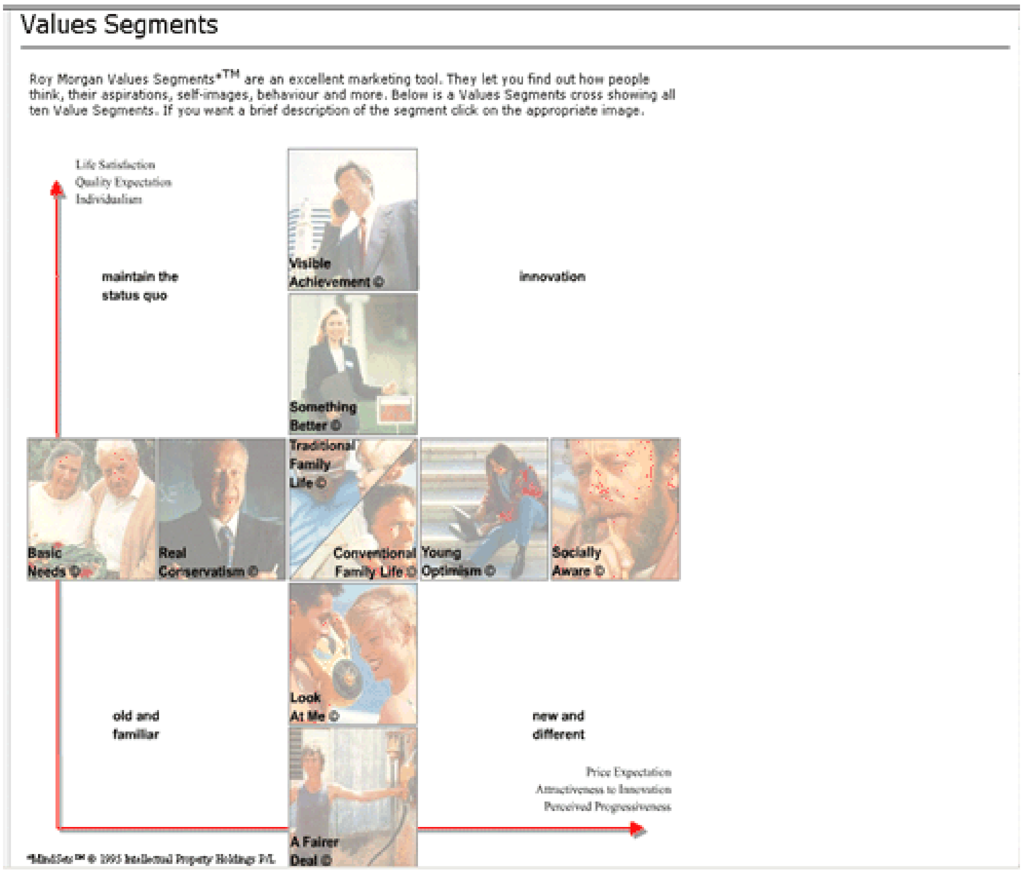

Figure 3.2 : Roy Morgan Value Segments^{TM}* (reproduced from URL: http://www.roymorgan.com/products/values-segments/values-segments.cfm with permission)

In addition to these main models, work described in Patel and Schlijper (2003) mentions a model proposed within Unilever with three segments: short of timers, adventurers and traditionalists. Another approach describes consumers using eight factors (Fashion conscious factor, Leadership factor, Family concern factor, Health consciousness factor, Care-free factor, Community consciousness factor, Cost consciousness factor, and Practicality factor) (Kucukemiroglu, 1997). The information required for segmentation was collected using a questionnaire, under five different sections, where one section of questions was on demographics and socioeconomics such as age, marital status, gender income group and education.

Clustering users into groups according to their behaviors or expectations is a commonality among these systems. Since these systems were successfully used in the commercial world, they can be considered as evidence for the existence of user groups/segments. Furthermore, the possibility of treating the buyers within a group equally, is supported by the performance of such systems. Figure 3.3 depicts the formation of user segments based on demographic, geographic, psychographic information or combination of such information. These segments are considered as valid for any purchasing domain.

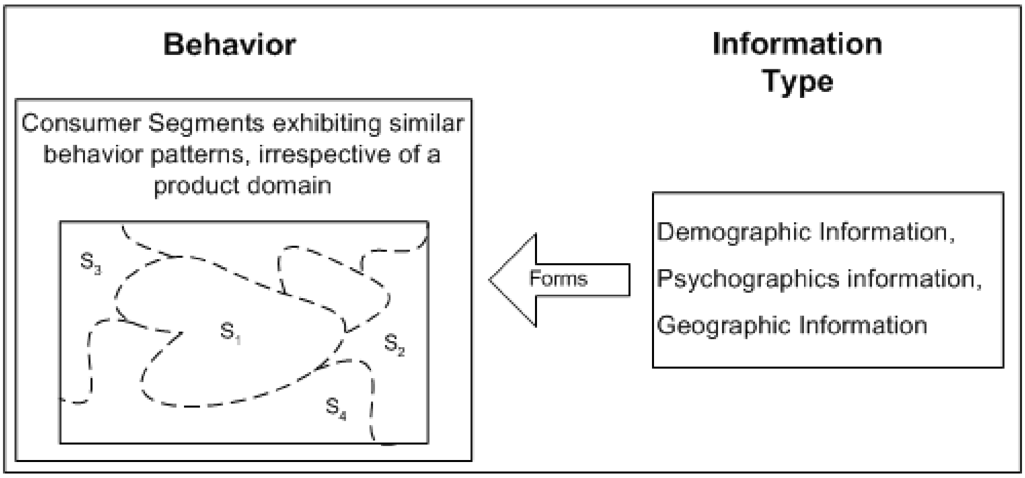

Figure 3.3 : Consumer segments (S_1, S_2, S_3, …) formed in MR user models

Therefore, from the analysis of MR consumer models it is possible to conclude that when demographic, geographic, psychographic information or combination of those are considered, irrespective of the product domains, users show patterns in purchasing behavior. In this context, if the two questions raised in section 3.2 are addressed based on MR consumer models, answer to question (a) is that, consumer behavior irrespective of the purchase domains deliver powerful information about user's purchase expectations. As the answer to question (b), the types of information required to capture purchase expectations of groups of people are their demographic, geographic, and psychographic or combination of such information.

The analysis of MR user modeling approaches illustrates the successful usage of generic user behavior which is independent of the product domain within user models. Since segmentation has the ability to capture an individual's group behavior, such product domain independent general purchasing information is usable as start-up information in a user model. Therefore, as shown in Figure 3.1, such consumer behavioral information is decided to be captured in the first layer of the novel model. By doing so, a rather abstract view of the consumers purchasing requirements (irrespective of the product domain) is captured in the user model.

In the next section, information content in the IT user models is analyzed. Existence of any powerful information categories is utilized to strengthen the justification of information content in the novel user model.

3.3.2 Consumer Behavior Analysis and Models in Information Technology Based Background

The main difference of user models used in IT is that they try to capture the individual preferences of the users without the intervention of humans, while the models used in MR catered for user segmentation for marketing purposes. In marketing, user segments are adequate since the predictions are made by human experts while in electronic user models predictions are automated. User models used in IT was extensively discussed in Chapter 2. It is observed, the main content of the user models have been the individual's current preferences in a given domain. All such user models required start-up information to base the initial predictions about individual's preferences then as the user interacts with the system the initial understanding of the user was updated with transaction information.

In some instances demographics were used as start-up information, to map users into segments (Krulwich, 1997) or to map users into stereotypes (Rich, 1979;Kay, 2000; Ardissono *et. al.*, 1999; Ardissono and Goy, 2000 and Ardissono *et. al.*, 2004). Several other systems used user ratings, towards either items or item descriptions as start-up information. Although user clusters or stereotypes demonstrate a similarity to the segmentation methods employed in MR, in electronic user models, all start-up information obtained were domain specific. For example, stereotyping user behavior in given domain or obtaining user ratings for items or features that belongs to a specific domain.

In addition to the information required at the start-up, all electronic user modeling systems acquire user's current preference as the user interacts with the system. As discussed in Chapter 2, this is carried out either explicitly or implicitly. Data generated after each transaction update the existing information in the user model, where only such updated information is used in consequent interactions to determine the future preferences.

Apart from the early work on shell systems, most of this work has been based on narrow purchasing domains. The shell systems were focused on the idea of information reuse by maintaining a single user model for each individual. But they still had to maintain the domain specific knowledge about the user (apart from the user information that is common across all domains).

The underlying reason for modeling the consumers in a narrower domain is that breaking down or categorization of user information based on the product domain helps to identify more granular user behavioral patterns within the corresponding domain. This approach control the complexity of consumer purchasing behavior and show trends in their buying behavior, demonstrating effective results such as good recommendations.

According to the above discussion it can be concluded that the IT user models basically maintained the current user preferences in a narrow domain where the user models were initialized by domain specific start-up information and then updated using the transaction information. Therefore, as an answer to question (a), (in section 3.1) the key information category captured in the current eCommerce user models is *domain based user preferences*.

As a answer to question (b), (in section 3.1) when analyzing the information required to capture *domain based user preferences*, in general, two main information types have been exploited: namely, user demographics and domain based preferences collected during system-user interactions. Figure 3.4 depicts how domain based user behavior is captured in IT user models using the user demographics and domain based preferences.

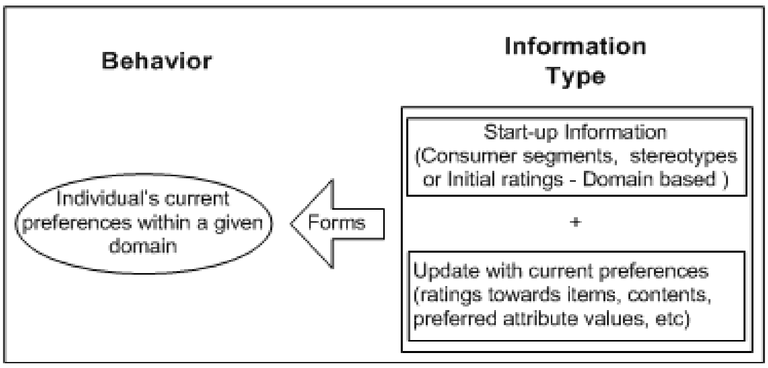

Figure 3.4 : Domain based user preferences captured in IT user models

Apart from using as start-up information, as exemplified by (Vozalis and Margaritis, 2004), demographics has been used along with buying behavior to enhance the outcomes of recommender systems. Pazzani (1999) used demographics to find similarity between two users in collaborative recommendations. Such exploitation demonstrates the value and powerfulness of demographic data. Although a domain based approach is taken, IT user models too justify the power of group based predictions and the effect of demographics on such predictions. As a result the importance of capturing stereotypic or group based information in the proposed model is secondly justified by the user modeling approaches in IT.

In the existing electronic user models, the domain based preferences collected during system-user interactions are captured as ratings towards items in the product domain, or preferences towards item features, as ratings or values. Capturing the domain based preferences are extremely important, since such information indicates the user interests in the product domain and hence is the link between the user requirements and product features. Therefore, the method of collecting such information needs to be carefully selected as to increase the precision and accuracy of the preference. As explained in chapter 2, rather than ratings towards items or product features, a descriptive feature based quantitative measure is more appropriate.

Electronic user models do not keep track of each transaction. Since the interest is only in domain based current user preferences, the transaction information is used only to update the existing user preferences. When updated, although some individuals may show a pattern of behavior within their accumulated transactions, there may be others who only show contradictions. In such occasions, systems find it difficult to make clear predictions and may deliver inaccurate recommendations. As previously explained the number of online buyers that may show such unexpected or irrational behavior can be expected to be high in current eCommerce environment. Therefore, isolation and proper treatment of such individuals is extremely important. If individual transaction data is recorded in the user model, such information can be analyzed to further study the behavior of the users.

3.4 Behavioral Categories and Related User Information

As discussed in section 3.1, in order to capture the complexity of user behavior in dynamic product markets, a user model should be carefully designed to capture appropriate behavioral categories and supporting user information. When formulating the novel user model initially, the behavioral categories to capture and the information required capturing such behavior should be decided. In the section 3.2, important consumer behavior categories and user information required to capture such categories were discussed under the two areas of research, as solutions for the questions (a) and (b). It was observed that several factors were combined when forming user models in the discussed work as to capture the complexity of purchase behavior. The purchase behavior is a result of human decision making. Depending on the decisions they make, individuals exhibit varied and complex purchase behavior in the long-term. Therefore, to further analyze the information that is required to be included in a user model, analysis of human decision making is required. In this section, a *decision frame* is considered to further clarify the important information in user modeling.

Studies by (Tversky and Kahneman, 1981)[16] define a *decision frame* as follows;

We use the term "decision frame" to refer to the decision-maker's conception of the acts, outcomes and contingencies associated with a particular choice. The frame that a decision-maker adopts is controlled partly by the formulation of the problem and partly by the norms, habits and personal characteristics of the decision-maker.

As explained by Tversky and Kahneman (1981), there are two main factors that contribute to a decision: by *formulation of the problem* and the *norms, habits and characteristics of the decision-maker*. In eCommerce context the consumers, makes the purchase decisions based on the available product space, where the personal preferences of the individual comes to effect. Referring back to the above *decision frame,* the personal and behavior information exploited in the existing user models can be interpreted as an attempt to capture the "*norms, habits and personal characteristics of the decision-maker* ". The domain based

[16] Tversky, A. and Kahneman, D. (1981) The Framing of Decisions and the Psychology of Choice. *Science,* VOL 211(4481)pg 453-458.

issues such as product availability and product presentations can be interpreted as the "*formulation of the problem.*"

In order to discover the reasons for user purchases in eCommerce environments, the information required to reveal an individual's decision process needs to be capture in a user model. Therefore, the novel approach, capture the user decision process and thereby understand the user behavior by capturing the following two types of user information in the novel user model.

(i) Personal information based purchase behavior

(ii) Domain centric information purchase based behavior

Capturing the two types of information in the user model is further described below.

3.5 Modeling Personal Information Based Purchase Behavior

As shown in Figure 3.3, consumer modeling in MR greatly relies on the personal information such as demographics, geographic, psychographic or such combinations of user information. At the same time there is evidence of usage of similar information in the IT user models as start-up information. The clear difference between the two usages is the domain consideration, where MR user models cater for market segments irrespective of the domain while the electronic user models strive to provide domain based personalization (see Figure 3.4). The reason for not considering the domain based behavior in MR is that once the segments are formed, the human experts made the domain related purchase predictions, whereas in IT models this was programmed with finer information required for automatic predictions. However, usage in both areas of research, confirms the value of personal information. In other words, a user model can exploit an individual's personal information as start-up information that is general to any domain, and if and when required can be manipulated to capture domain based behavior combining with additional knowledge on product attributes. Therefore, the first category of information to be included in the user model is identified as personal Information.

<u>Layer 1 of the user model</u>: **Capture user *personal information* based purchase behavior.**

Although the category of information is known, deciding the actual information that is required to capture layer 1 information category is still to be investigated. The discussed work in MR emphasized that consumer segments with similar characteristics show similar behavioral patterns. However the work does not point out a particular segmentation method as better than the others. Furthermore, in each of the above discussed work, the segmentation approach is different. According to (Kelly, 2006) there are several segmentation approaches: behavior segmentation, benefit segmentation, buyer-readiness segmentation, demographic segmentation, ethnic segmentation, geographic segmentation, interaction segmentation, lifestyle segmentation, loyalty segmentation, occasion segmentation, profitability segmentation, psychographic segmentation, and usage segmentation.

Furthermore, Kelly (2006) states that the only segmentation method that takes the perspective of the consumer is benefit segmentation which poses the important question 'what is it that the consumer actually buying?' Patel and Schlijper (2003) at Unilever, propose a food benefit framework, where consumer benefit expectations were considered instead of traditional behavioral segmenting methods. They argue that although market segments are traditionally seen as static, in reality it may be a dynamic equilibrium where the same consumer changing their behavioral segments. In other words the same individual can be allocated to various behavioral segments according to the behavior. Patel and Schlijper (2003) try to capture the *needs* of the same consumer where each need may belong to a different segment. The flip side of *need state* segmentation are the benefits that consumers seek in products and services purchased. An example given in Patel and Schlijper (2003)[17] is as follows.

> *Example Need State could be 'I'm organizing a dinner party and need to impress' – the key benefits provided by the product could be great tasting food that you're confident will work every time.*

Patel and Schlijper (2003) identifies nine key *need states* within Unilever food benefit framework as Confidence, Time Saving, Health, Pleasure, Physical Management, Mental

[17] Patel, S. and Schlijper, A., (2003), last sited in 2008, http://www.maths.ox.ac.uk/ociam/StudyGroups/ESGI49/problems/unilever2/unilever2.pdf

Management, Caring (family / world), Socializing, and Fun. The nine need states are shown in the Figure 3.5 (Figure obtained from (Patel and Schlijper, 2003)).

Basing on the 'need states' identified in Patel and Schlijper (2003), we identify eight benefit expectations in consumers, which are more relevant to our example domains; namely; time saver, price sensitivity, quality consciousness, fun spending, health consciousness, family person, socializing, and adventurer. In this thesis, the benefit expectations are referred to as 'Purchase Behavior Characteristics' (*PBC* values), since such benefit expectations (from purchasing) are possible to represent as purchase behavior characteristics of the individual. Depending on the combined effect of the *PBC* values, an individual's expectations towards the product benefits are identified. These *PBC* values can be considered as *general stereotypes* that apply to users irrespective of the purchasing domain. In chapters 4 and 5, *PBC* values and the new concept of *general stereotypes* are introduced and discussed in detail.

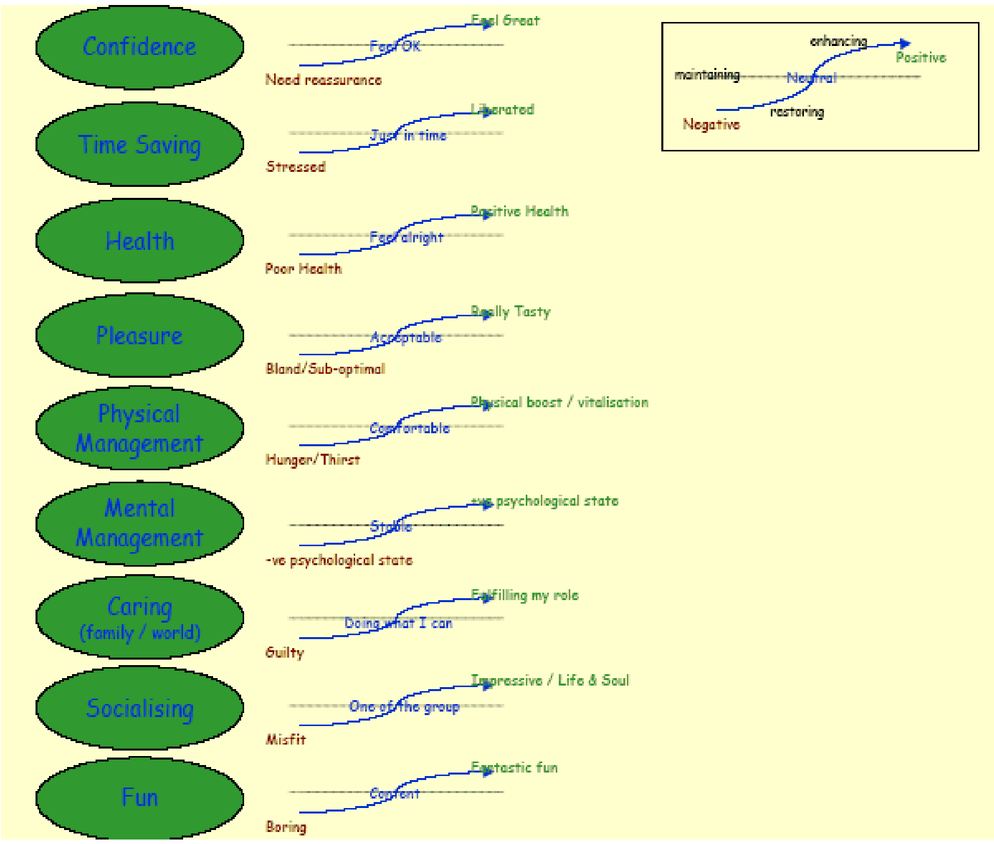

Figure 3.5 : Key benefits in each Unilever category- reproduced from Patel and Schlijper (2003) with permission

3.5.1 Purchase Behavior Characteristics (*PBC*)

The above discussed Purchase Behavior Characteristics (*PBC* values) are indirectly related to the segmentation topologies described in section 3.2.1. Tables 3.1 and 3.2, contains the descriptions of Vals segments,[12] and Roy Morgan Segments[13] respectively. The descriptions of the segments contain the original interpretation of expected behavior of each category of people. To make clear how each segment is related to the proposed set of user characteristics, a third column is added to list the *PBC* values that are related to each segment. Each consumer segment shown in the tables 3.1 and 3.2, demonstrate one or more of the *PBC* values. Both systems discussed in the tables have a proven past performance in clustering consumers and successfully predicting their expectations. Therefore, we argue that the *PBC* values have a basis and a strong foundation. Furthermore, if the *PBC* values of a particular individual are known, deriving conclusions similar to the above systems are possible.

Table3.1: Val Segments and their descriptions

Segment	Description	Purchase Behavior Characteristic (*PBC*) value
Innovators	Reflect cultivated tastes for upscale, niche products and services.	Quality conscious
	Image is important as an expression of their taste, independence, and personality	Socializing
	Their lives are characterized by variety.	Adventurous
	Reflect a cultivated taste for the finer things in life.	Fun Spending
Thinkers	Have a moderate respect for the status quo institutions of authority and social decorum.	Moderately Socializing
	Open to consider new ideas.	adventurous
	Conservative, practical consumers, who look for durability, functionality, and value in the products they buy.	Quality conscious and some Price sensitivity Less fun spending
Achievers	A deep commitment to career and family.	Family person
	Achievers live conventional lives, are politically conservative, and respect authority and the status quo. They value consensus, predictability, and stability over risk, intimacy, and self-discovery.	Less Adventurous
	Image is important to Achievers; they favor established, prestige products and services that demonstrate success to their peers.	Socializing as their income permits, Quality conscious
	Busy lives, and are often interested in a variety of time-saving devices.	Time Saver

Experiencers	As young, enthusiastic, and impulsive consumers, Experiencers quickly become enthusiastic about new possibilities but are equally quick to cool. Their purchases reflect the emphasis they place on looking good and having "cool" stuff.	Adventurous and Fun spending
	Avid consumers and spend a comparatively high proportion of their income on fashion, entertainment, and socializing.	Socializing
Believers	Like Thinkers, Believers are motivated by ideals.	Socializing
	They are conservative, conventional people with concrete beliefs based on traditional, established codes: family, religion, community, and the nation. Believers are predictable; they choose familiar products and established brands.	Family person
Strivers	Strivers are trendy and fun loving.	Fun Spending
	Strivers are concerned about the opinions and approval of others.	Socializing
	Money defines success for Strivers, who don't have enough of it to meet their desires.	Price Sensitive
	They favor stylish products that emulate the purchases of people with greater material wealth.	Of Adventurous nature.
Makers	They express themselves and experience the world by working on it. They are practical people who have constructive skills and value self-sufficiency.	Less Time saving
	They live within a traditional context of family, practical work, and physical recreation and have little interest in what lies outside that context.	Family Person
	Makers are suspicious of new ideas and large institutions such as big business.	Less Adventurous
	Because they prefer value to luxury, they buy basic products.	Price sensitive and less Quality Conscious
	They are unimpressed by material possessions other than those with a practical or functional purpose.	Less Fun Spending and Socializing
Survivors	Survivors live narrowly focused lives. With few resources with which to cope, they often believe that the world is changing too quickly. They are comfortable with the familiar and are primarily concerned with safety and security.	Less Adventurous Less Time Saving
	Because they must focus on meeting needs rather than fulfilling desires, Survivors do not show a strong primary motivation.	Less Fun spending
	Survivors are cautious consumers. They represent a very modest market for most products and services.	Price Sensitive
	They are loyal to favorite brands, especially if they can purchase them at a discount.	Less Quality Conscious

This will also allow the measuring of an individual in several dimensions, along eight different purchase behavior characteristics. Rather than assigning to a single segment, where an individual share only some of the segment characteristic, each individual can now be described with each of the eight *PBC* values. The next challenge is calculation of the

PBC values. The next section describes the calculation of quantitative values for the *PBC* values using a set of *value functions*.

Table 3.2 : Roy Morgan Value Segments and their descriptions

Segment	Description	Purchase Behavior Characteristic (*PBC*) value
A fairer deal	Pessimistic, cynical	Less quality conscious, less adventuress, less fun spending and less socializing
	Financially struggling	Price Sensitive
Basic needs	Generally happy and contended with what they have. They don't looking for more and enjoy watching the world go by. Usually retirees.	Less adventurous
Conventional family life	Trying to give their families better opportunities than what they had in their own childhood.	Family person
	People seeking greater financial security and struggling to improve basic living standards.	Price sensitive
Look at me	Looking for fun and freedom away from their families, and wish to stand out from their parent's generation. Fashion and trend conscious.	Adventurous and Fun spending
	Conscious about what their peer's opinion on them.	Socializing
Real Conservatism	Predictive, disciplined and safe society.	Less adventurous
	They are conservative, conventional people and generally feel that things are not good as they used to be.	Family person
Socially aware	Searching for new and different things. Searching for education and new knowledge.	Of Adventurous nature Quality conscious Time saving
Something better	Competitive, individualistic and ambitious people	Socializing , Adventurous, Quality conscious, Time saving, Less Price sensitive
Traditional family life	Very much similar to conventional family life, but in Australia generally empty-nesters or extended families.	Family person, Less Adventurous
Visible achievement	Despite being successful they retain traditions and family values. Take effort to provide families with high quality environment. Work for financial reward and job stimulation.	Family person, Of Adventurous nature, Quality conscious, Time saving, Less Price sensitive
Young optimism	Long-term thinkers, who are into image and style. Busy planning careers, attending universities.	Adventurous, Quality conscious, Time saving Less Price sensitive

3.5.2 Use of Demographics in Value Functions

As observed in literature, demographics were considered as one of the contributing factors towards the *PBC* values. We admit that in addition to demographics there can be more solid information that contributes to these characteristics such as social influences and life styles. However, *PBC* values are only calculated to be used as initial values when describing the individuals. Therefore, in this thesis demographics are considered to be the only contributing factors towards the *PBC* values. In addition, demographics are simple to request, and convenient to obtain using an online form. Therefore, the value functions (for *PBC* values) were formed using eight demographics: family type, gender, work hours, age group, income, occupation, industry and education in different ratios.

According to the discussions and derivations so far it can be concluded that demographic based value functions generated to represent user *PBC* values are capable of capturing user purchase behavior quantitatively. Therefore, in the new user model, the user purchase behavior is captured in terms of *PBC* values which are calculated using demographic based value functions. These initial estimated values are used as initial start-up information about user behavior and are later updated using actual user behavior. Use of value functions allows to describe each individual in a flexible and a standard manner, since each individual is described using the same set of value functions. Furthermore, rather than segmenting user populations, individual value functions permit easy adjustments. In Chapter 4 formation of value functions are discussed in detail.

3.6 Modeling Domain Centric Purchase Behavior

In the section 3.4, the first most information category captured in the novel user model is identified as the user personal information. As identified in section 3.3, according to Tversky and Kahneman (1981), the second type of information contributing to decision making is highlighted as domain centric information. As explained before, domain centric information contributes to the problem formulation by controlling the product related information such as availability of products, easy access, and variety. For example, if the product desired to purchase is not available, or if certain attributes are missing in the available products, then the user is forced to adapt to the situation: for example, to search for presence of a second best option. In order to include the effect of the *formulation of the problem* (as described in section 3.3) in the user model, information linking the user and the product descriptions need to be captured. For example, user desires and possible products or product attributes that fulfill such desires could be included in the user model.

As observed in section 3.2.2, attention should be paid to the strong domain based approach followed in IT user models. The domain based approach compliments the idea of Tversky and Kahneman (1981) called the *decision frame*. Although socio-demographics are powerful information they alone do not explain the consumer buying behavior due to product related issues such as volatile product markets, modern advertising methods, and vast number of choices. Therefore, a second category of information is required to further clarify the human decision process and thereby capture the reasons for consumer purchase behavior. Most of the user models in IT background capture and analyze user behavior within a single domain since it is reasonable to expect a fairly consistent behavior within a given application domain. Referring back to the *decision frame* by Tversky and Kahneman (1981), the decision makers *norms and habits* are important as well. Such norms and habits can be expected to be consistent in a given domain.

In the literature, elicitation of domain based preferences is carried out either collecting user preferences towards item attributes or items. In the latter approach user characteristics are not considered. Instead, only the items are compared to locate more similar items. Or similarity of users is considered to recommend the same item. In both approaches, since the reason behind the preference is not clear, this might lead to misunderstanding the user expectations. Two individuals may prefer the same item due to two different reasons, such as preference towards two different attributes of the same item (Towle and Quinn, 2000). In such circumstances lack of knowledge about the reason may result in weak cross domain recommendations, where two items from different domains are recommended to similar users. Formulation of the problem becomes more explanative if the product attributes are linked to the user characteristics rather than items. Furthermore, choice available for the consumer is enhanced with decomposing the items into a set of features where selection of preferred features is possible. In such approaches, the search space is expanded due to existence of attribute choices; where in the absence of a particular preference it is replaced by another. Therefore, a second category of information is identified as *domain based preferences*. This category of information, link the consumer and the products as user preferences towards product features.

Layer 2 of the user model: **Capture user's *domain information* based purchase behavior.**

While some consumers may stick to a pattern within the purchase domain, the others may get easily moved by the dynamic environment. Under such circumstances, although 'rational' behavior is assumed, it is possible that certain individuals behave 'irrationally' due to different reasons, in different occasions. As a result, rather than looking at buyer behavior as static throughout all purchasing acts, even though in the same domain, it is more realistic and accurate to expect each individual to change the expected behavior moving from one behavioral segment to another. Although an individual's characteristics specify otherwise, impulsive behavior could occur. In order to capture such volatile buyers it is important to keep track of each and every transaction they perform and thereby capture the frequency of such occurrences. Therefore, a third category of information is identified as *transaction based preferences* which are consumer's preferences towards product features during the current transaction. User provides such information during the product search, indicating the current interest which can be contradict with expected behavior.

Layer 3 of the user model: **Capture user *transaction information* based purchase behavior.**

In addition to the theoretical facts discussed above, commonsense and personal experiences can be used to further confirm above consumer behavior patterns. For example, it is an obvious fact that demographic information such as income plays a big role in peoples buying habits and lifestyles. But still there are people with a low income if they buy any item at all, always go for high cost items. Similarly there are people with high income but also still choose inexpensive options in certain domains. Systems that only rely on user segments are incapable of capturing such deviations.

It can be argued that there is a higher probability that people behave consistently within a given domain to a large extent. It is commonly seen that some people spend most of their income buying expensive clothes but spend little for entertainment or food. This can happen irrespective of the person's demographics. For example, there can be two people with the same job and the same type of family where one is a teetotaler. This small difference will make their behavior very different in a product domain such as alcohol purchase.

Special occasions or impulsive purchases also can make people deviate from their expected pattern. For example, a very price sensitive individual can disregard price completely in a special occasion such as marriage. After switching to a new product a person can either continue to use it or may switch back to what they used before. It is not possible to capture such sudden deviations using demographic or domain specific behavior analysis. Such mood changes are necessary to be separated from other types of behavior and therefore important to keep a record of all user transactions.

Considering the above facts we conclude that once an individual is allocated to one or more clusters, the chance of them belonging to that cluster forever is very low. Therefore, the clustering methods can be considered suitable as initial stereotypes, while dynamic changes needed to be captured using alternative techniques. This way, better personalization is achieved by identifying individual user expectations rather than catering for different consumer segments.

Human purchasing behavior is very complex. Individuals may belong consistently to the same segment, or may exhibit domain based patterns or may not show any kind of pattern in their behavior under any circumstances. Therefore, capturing user information from different perspectives is important. Since none of the existing techniques seems to satisfactorily cater for all above discussed requirements, it is proposed that an ideal user model has to capture the individuality of purchase behavior and thereby provide individualized personalization.

3.7 Summary

This chapter initially discussed the complexity of user behavior in current volatile markets, and how such complexity could be captured within a user model. Then consumer behavior models in recent market research and information contents of electronic user models were analyzed.

When work carried out in both areas are analyzed, three interesting factors were discovered. First we noted the value of measuring user benefit expectations. Instead of segmenting users into clusters, description of each user according to a set of expectations seems more flexible. Although user clustering and stereotypes were utilized in the literature as initial user information, in the electronic user models consumer expectations or market research knowledge has not been used. In addition it was identified that the demographics are adequately contributing to these benefit expectations.

Secondly, existing work demonstrate their user models mostly built for single domains. In other words, most of the models developed in either area are product domain specific. For example, from the MR/economics background, theoretical consumer behavior analysis modeling has been carried out for purchasing durable goods (Katona, 1968). In electronic user models, specific testing domains are chosen to demonstrate the effect of the user model such as restaurants, movies, or news articles. This indicates that rather than looking at an overall behavioral picture, some categorizing such as domain specific behavior helps to understand the user more deeply.

Finally, transaction based information manipulation in content based user models were considered. Such information is usually exploited in updating existing information about user's domain centric preferences. Since this information is expected only to confirm the immerging patterns in existing information, once update is carried out, this information is discarded. Although this information does not support the current pattern in preferences, it may have occurred due to a change in the preferences. Some individuals demonstrate stability throughout their purchasing acts while there are individuals who are easily moved by effects such as advertising. Such consumers may not show any behavior patterns in both upper layers of the user model. Capturing such instability is important in advertising and other product promotions. Therefore, if recorded and analyzed regularly, this information can contain valuable behavior patterns. For example, a certain individual may often demonstrate contradicting behavior. Such observations may lead to new marketing strategies such as personalized marketing. We also noted the importance of users' current needs, which can be contradicting both generally expected characteristics and expected domain based behavior. For example, according to characteristics an individual may supposed to prefer low prices. But his/her current search query, (due to a special occasion such as marriage or for a gift) may indicate preference for the most expensive. In such an instance it is important to capture the difference in the usual price range and cater accordingly.

The above two reasons highlighted the need of acquiring and recording transaction information. Therefore, as mentioned in section 3.4, our analysis pointed out that complex user behavior can be captured in a user model with three information layers; namely, personality based information layer, domain based information layer, and transaction based information layer.

Therefore, it can be concluded, a user model created using the above identified information layers, will be capable of capturing dynamically changing user needs in the long run, and will be ideal for modeling online users. In the next chapter a novel user model architecture that supports the identified three information layers and the framework within which it is implemented is discussed.

Chapter 4

Conceptual Architecture of the Layered User Model

In the previous chapter, several categories of user information made use of in current user models were analyzed. The analysis highlighted the existence of three different information categories that are required to be captured in a user model in order to get a thorough understanding of the user; namely, user personal information, domain centric preferences and transaction level preferences. To capture such categories of information and thereby capture the complexity of human purchasing behavior in a user model, a layered approach was proposed. The proposed novel user model consists of three information layers where each layer captures and maintains the above three information categories. In this chapter, the new user model architecture is discussed in detail and the framework within which it is implemented is presented. The relationships between the *LUM* and its components are described at a higher level of abstraction whereas detailed implementation information and algorithms are presented in Chapter 5.

eHermes is a web based multiagent system which is currently being developed in Monash University. The initial design and the framework, which has been later subjected to changes is described in (Jayaputera *et. al.*, 2003b) and (Alahakoon *et. al.*, 2003; Alahakoon *et. al.*, 2004). eHermes has a flexible and extensible open architecture, which can adapt to changing environments. eHermes helps users with their information needs such as financial services and online shopping for goods and services. The new user model architecture is designed and implemented within the personalization component of the eHermes system.

In section 4.1, overall architecture of the eHermes multiagent system is briefly described under the two main components of the system: the personalization component and the mission processor. Section 4.2, discusses the overall personalization component of eHermes multiagent system. In the section 4.3 the overview of the *LUM* is presented. The next three consecutive sections 4.4, 4.5 and 4.6 present the information layers belonging to the *LUM*. Section 4.7, describes the update strategies and maintenance of information layers. In section 4.8 the approach taken by the model to handle obtrusiveness in user information gathering is discussed. Finally, in section 4.9, a summary is presented.

4.1 The Overall eHermes Architecture

In this section, a brief description of the eHermes multiagent system is presented to highlight the implementation environment of our work. The descriptions are based on early work on eHermes (Jayaputera *et. al.*, 2003a).

eHermes is a web based multiagent system which is currently being developed in Monash University. Initial work on eHermes consists of two major components: *A personalization component* and a *Mission Generation and Execution Component*. The personalization component is meant to generate personalized missions according to a model of the user. Our work was initiated as the personalization component of the eHermes system. Work on eHermes Mission generation and execution component is completed as another PhD project. The publications include (Jayaputera *et. al.*, 2003b; Jayaputera *et. al.*, 2004; Jayaputera, 2005b;2005a).

In the initial proposal, eHermes was designed to operate in ubiquitous environments, in order to provide services for both human users and software agents. But when developing the personalization component, the agent modeling or ubiquitous features were not included. eHermes was supposed to deliver personalized services in many different product and service markets. Therefore, a user model usable over multiple domains was required. When designing such a domain independent generic user model, inclusion of software agents in the first version of the design can dilute the focus on key issues in user modeling. Therefore, at this stage the 'actor' is assumed to be a human user. Usage of the model in ubiquitous environment is also not considered in this version of the implementation. However, due to the generic hardware independent nature of the user model architecture, the usability in ubiquitous environments can be easily demonstrated. Complete initial design of the eHermes system is shown in Figure 4.1.

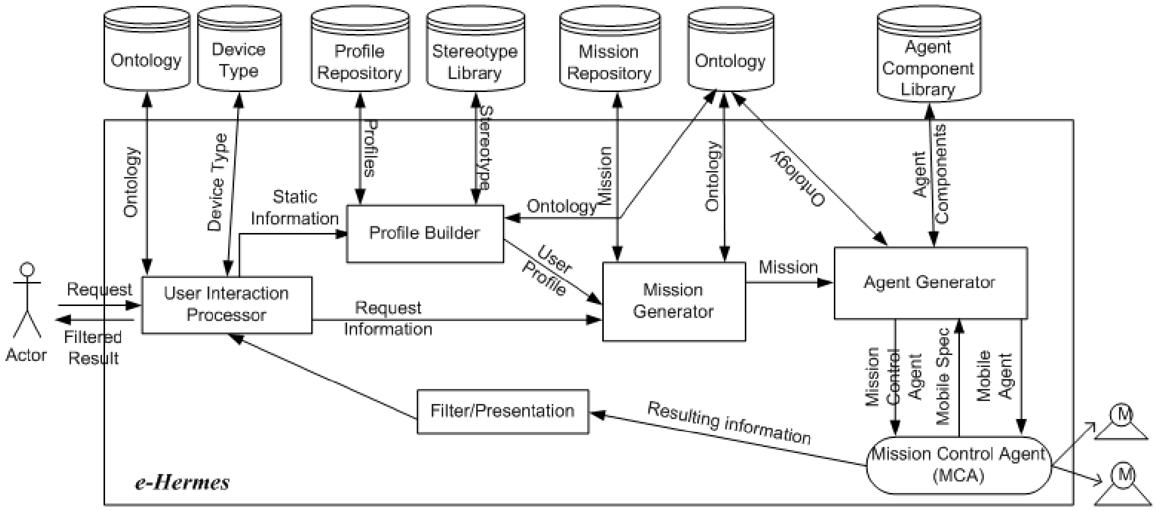

Figure 4.1 : eHermes Architecture

As shown in diagram Figure 4.1, the repository for device type is for the ubiquitous version of the user model. In the Figure 4.1 the main processors are shown as rectangles. Among the main processors, the ***User Interaction*** processor and the ***Profile Builder*** processor belong to the user modeling component. The rest of the processors are out of the scope of work presented in this thesis and will only be briefly discussed.

The ***User Interaction*** processor handles all interactions with the user; including the Personalized Interactive Product Retrieval Process (*PIPRP*). The *PIPRP* is discussed in detail in Chapter 7. The ***Profile Builder*** processor is responsible for building and maintaining the *LUM* using the information retrieved by the interactive interface. The *LUM* is updated after each system-user interaction. Furthermore, the *LUM* is periodically updated to maintain up-to-date user information. Each time user seeks personalized services using eHermes, the *LUM* is employed for personalized interactions.

In the ***Mission Generator*** processor**,** user requests are converted into missions. The system *Mission* is defined as the *goal of the system (perform on behalf of the user)*. When forming the mission, the user model is used for personalizing the outcomes for the individual. The ***Agent Generator*** automatically assembles the suitable agents to carryout the mission. The ***Agent Generator*** is controlled by the Mission Control Agent (MCA). Finally the results of the search carried out by agents are presented to the user. The ***User***

Interaction processor is responsible for presenting the personalized outcomes to the user. The next section describes the actual work carried out on the personalization component.

4.2 The Personalization Component-eHermes personal

eHermes personalization component is named as eHermes PERSONAL. Since eHermes is designed to have an open architecture where both mission and domain knowledge is supplied at run-time, a generic personalization component is required. Figure 4.2 presents the basic high level architecture of the personalization component while Figure 4.3 represents a more detailed picture.

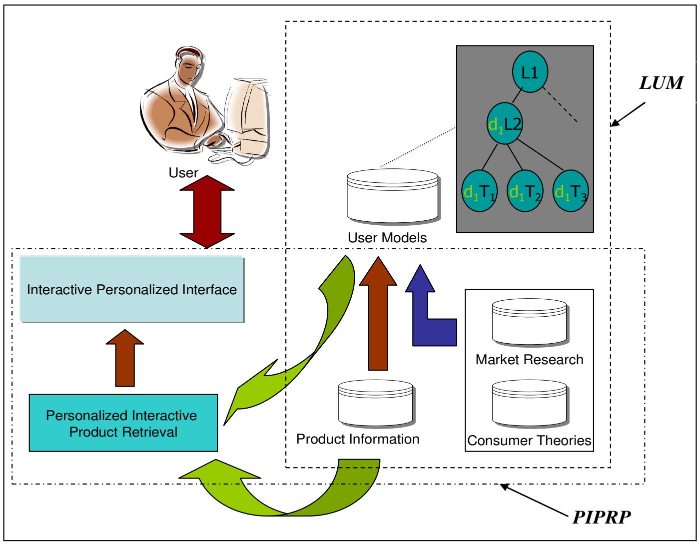

Figure 4.2 : eHermes Personalization Component – Overview of the Architecture

A layered user model is generated for each individual who wishes to obtain personalized services. The first layer (Personal Information layer – *PI* layer) of the user model is created at the time of user registration. Information in the *PI* layer contains user's identification information and personal data obtained during the registration. In addition, this layer also presents the user's stereotype based general buying behavior called "Purchase Behavior Characteristic (*PBC*)" values. These *PBC* values represent user purchase behavior irrespective of the product domains. Each time a user seeks personalized interactions in a new product domain; a new Domain Information layer (*DI* layer) is created. In other words, a new layer 2 of the user model gets attached to the already existing *PI* Layer. This information layer contains user's product feature preferences within that domain.

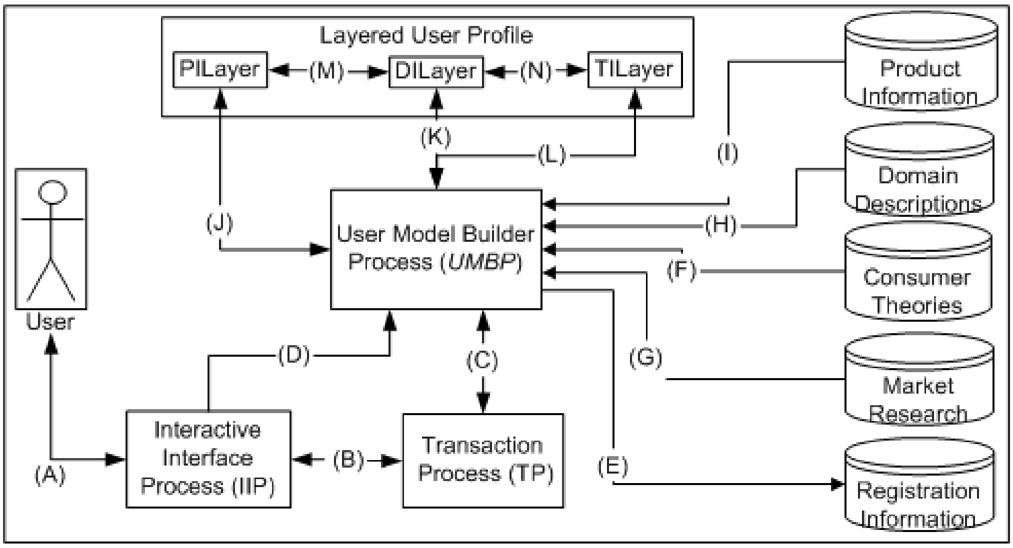

Figure 4.3 : The Overall Architecture of the eHermes personalization component

As shown in the Figure 4.3, when building the *DI* layer, Market Research, Consumer theories and product knowledge are combined with the personal information in *PI* layer. The third layer consists of information related to each individual transaction: Transaction Information layer (*TI* layer). Therefore, each *TI* layer is connected to a *DI* layer. In a given domain, after each transaction, based on user preferences exhibited during the session, *DI* layer feature preferences are updated. Events labeled A-N in Figure 4.3 are explained in Table 4.1.

Users initially interact with the *eHermes* PERSONAL main menu (shown in Figure 4.4). New users are expected to register using the 'Register Now' link in order to obtain personalized services. Once a user becomes a member (by registering), personalized browsing is permitted. When logging in, their identity is verified by providing identification information as in Figure 4.4. The 'Available shopping domains' button, lists out the categories of shopping items from which users can pick the category that they wish to shop in. If the user prefers a non-personalized service that is available under the 'Non Personalized Browsing' link. If this option is selected the user model does not participate in the interaction. "Personalized Browsing for Members" requests user login information as shown in Figure 4.5. Personalized services are available only for registered users. The next section describes the *LUM*.

Table 4.1: Explanation of Figure 3

Event Name	Processors involved	Event
A	*U → IIP*	Registration Information, Product attribute value preferences
	IIP → U	Personalized product list
B	*IIP → TP*	Product attribute value preferences
	TP → IIP	User to system queries, Personalized product lists
C	*UMBP → TP*	System to user queries
	TP → UMBP	New preference data
D	*IIP → UMBP*	Registration information, Attribute preferences in a new domain, Attribute preferences in an existing domain
	UMBP → IIP	Information required to provide personalization
E	*UMB → User Database*	Registration information
F	*Consumer Theories → UMBP*	Consumer theories and market research ideas are combined to form *PI* layer information
G	*MR Theories → UMBP*	
H	*Domain Desc Data → UMBP*	Domain descriptions such as inter-domain, relationships and product features are combined to form the contents of *DI* layer.
I	*Product Data → UMBP*	
J	*UMBP → PI Layer*	Initial characteristic values or updated values are passed on.
	PI Layer → UMBP	Characteristics are passed on when user needs to interact in a new domain
K	*UMBP → DI Layer*	Update preferences in the user model
	DI Layer → UMBP	Preferences in the user model are obtained
L	*UMBP → TI Layer*	Preferences for the current transaction are passed on
	TI Layer → UMBP	Information required for *DI* layer update is obtained.
M	*PI Layer → DI Layer*	Indicate the information updates between layers. But always this happens through the *UMBP*.
	DI Layer → PI Layer	
N	*DI Layer → TI Layer*	
	TI Layer → DI Layer	

4.3 The Architecture of the LUM

This section presents the overall structure, and an overview of the information content of the *LUM*. Furthermore, the capability of the user model with regard to its ability in capturing user behavior from *n*-dimensional perceptions is discussed. The domain hierarchies which support such ability are also described and defined.

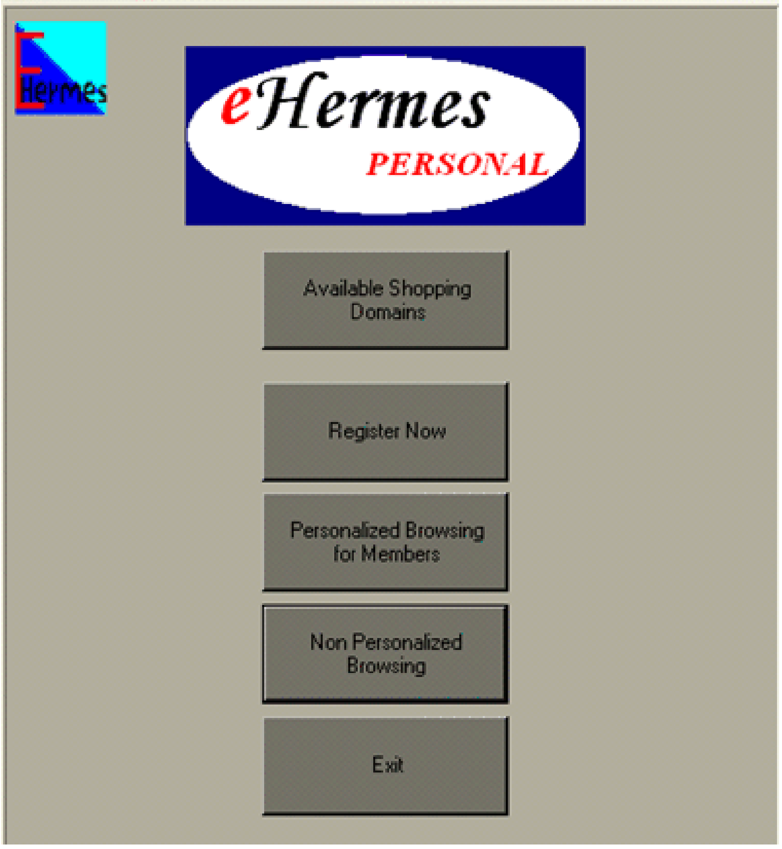

Figure 4.4 : The main screen – eHermes PERSONAL

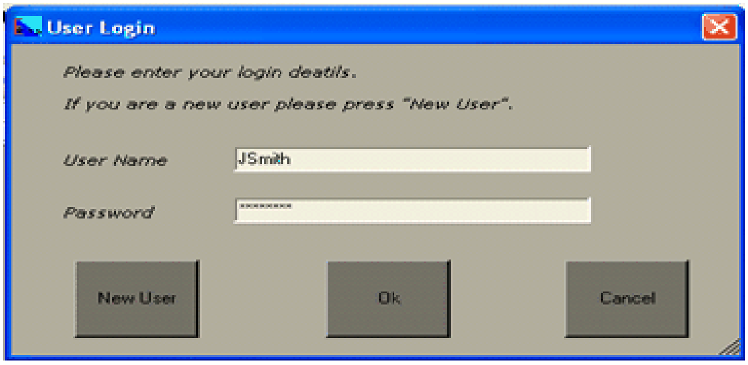

Figure 4.5 : The login form screen

4.3.1 The user model

In this section, the proposed novel user model architecture consisting of three information layers is presented. This layered user model has the ability to capture user needs and dynamically adapt to changes in buying behavior via learning. As described above, information layers are added and accumulated to the profile each time the user seeks personalization. A layered profile of a user with interactions in more than one domain is shown in the Figure 4.6. The three different information layers of the new user model architecture are as follows.

(i) Personal information describing the user such as demographics and general buying characteristics.
(ii) User's product domain specific behavior. (Here the *Domain* is just a specific subject area of knowledge such as restaurants, leg-wear, real estate, etc)
(iii) User's preferences of product related attributes.

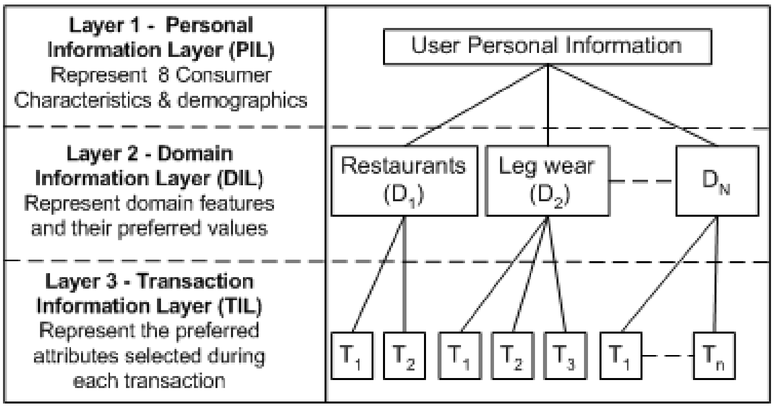

Figure 4.6 : Three layered User model architecture

Personal information about the consumer conveys the general picture of an individual. Domain specific information is captured in the *DI* layer of the user model. Therefore, each time a user interacts in a new domain, a *DI* layer for that particular domain gets created. All the *DI* layer components of the user model are attached to the *PI* layer of the user model. After interacting with several domains, a user will own a user model with several *DI* layers attached to its *PI* layer. The *PI* layer remains common to all *DI* layers, as it is user's general purchasing behavior irrespective of a specific domain.

There are many occasions where users' behavior changes from expected domain behavior. This could be due to the particular 'state of mind' and/or spontaneous decisions. Transactions with 'outlier' patterns can be important to capture the user's current need without solely depending on his previous expected behavior in the domain (as captured in the *DI* layer). For example, a user who is easily excited by advertisements may display changing expectations in each and every transaction, even within the same domain. In addition, this new preferences will reveal the user's current requirements which may be different from his previous expectations. Therefore, this information is suitable as update to the user's current expected behavior in the *DI* layer. In addition, this information is recorded in the *TI* layer of the user model for future references. Each time a user interacts with a domain, preferences declared in that particular transaction are captured in a *TI* layer. For a user who interacted several times in a given domain there will be several *TI* layer components attached to the corresponding *DI* layer component of the user model.

According to the above description, user model consists of three layers. However, individuals tend to interact in various product domains and generate information about there preferences. As described in Chapter 2, if domain centric user models were used, information generated in a domain is not possible to be used in another domain. As mentioned in the introduction (Chapter1), the proposed user model supports information reuse across multiple domains. The *domain hierarchies* which exist within and across product domains provide the environment for such multi-domain functionality. These are described next.

4.3.2 Domain hierarchies and n-layered user model

The above section described the main three layers of the user model. In this section the possible layering within domains are discussed. The *DI* layers are not single layered, but instead consists of n-layers. Hence, the novel user model can be said to consist of n-layers of user information, when product domains are considered.

As mentioned, a *domain* can be a specific subject area of knowledge such as restaurants, leg-wear, real estate, etc. Therefore, the domains in which users interact can be interconnected. For example, it is possible to form a hierarchy of products. In real life such categorization is observed in supermarket aisles. The products can be categorized and the categories can be arranged as an interconnected hierarchy. Therefore, related product domains can be represented in a hierarchy of *n*-layers.

The following definitions, define *domain features* and *domain attributes* within the context of work presented in this thesis.

Definition 1: Domain Feature

If D_i is a domain, and F_j is the set of features describing the domain D_i then d_if_j is a *domain feature*, where i is the index of domain D_i in the domain hierarchy and j is the j^{th} feature that describes the domain D_i.

Then, $\forall D_i\ \exists F_j$, where $F_j = \{d_if_j \mid i \in [1,..,n], j \in [1,..,m]\}$, ($m$, n>0) and where m is the number of features used to describe the domain D_i, and n is the total number of domains.

Definition 2: Domain Attribute

If D_i is a domain, and F_j is the set of features describing the domain D_i then $AT_{d_if_j}$ is the set of attributes used to describe each feature d_if_j.

Then $\forall F_j$ *describing* D_i, $\exists\, AT_{d_if_j}$, where $AT_{d_if_j} = \{at_k | k \in [1,..l], l \geq 0\}$ and where l is the number of attributes used to describe the feature.

Since the *DI* layer components represent user information in a given domain, now this second layer becomes n layered. Therefore, the diagram in Figure 4.6 is re-drawn as shown in Figure 4.7 below. When the domains are related, there are common features and common attributes that describe more than one domain. For example, a child domain in the hierarchy owns a subset of features (and thereby attributes) belonging to the parent domain. Such common attributes makes sharing the attribute values across domains possible. Therefore, existing preference values in the user model that are belonging to parent domains, are reusable by common attributes in the newly interacted child domains.

Definition3: Cross domain common features

If D and *D'* are two domains in the domain hierarchy, where *D'* is a child domain of *D*, then if the set of attributes belonging to *D* is in *S* and if the set of attributes belonging to *D'* is in *S'* then $\exists S''$ containing the common attributes to the two domains, such that $S'' = S \cap S' \neq \phi$

In the implementation of the work, domain hierarchies and cross domain information reuse is not included.

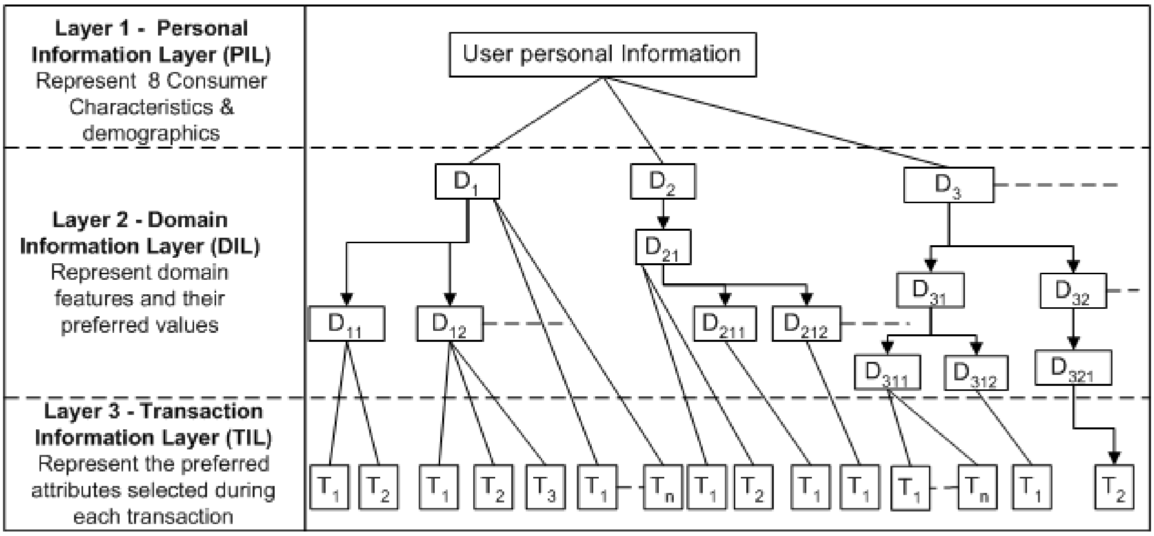

Figure 4.7 : The expanded user model to demonstrate the n- layers

The domains shown in Figure 4.7 belong to different levels of the domain hierarchy; it shows three layers of domains from D_i to D_{ijk} where i=(0,…,n_i), j=(0,…,n_j), k =(0,…,n_k) and the domain position of a given domain in each layer is represented by *i, j,* and *k* respectively. The maximum number of domains in each layer is given by n_i, n_j, and n_k respectively.

Figure 4.8, further clarifies the product hierarchies. According to *Definition 1* and *Definition 2* above, within a domain all the items are described using the same set of attributes. As shown, the upper layers of the hierarchy are super classes of the lower levels. Lower levels are more specific. According to *Definition 3*, all food related categories such as recipe, restaurants and groceries preference may possess common features which belong to the upper layer “Food” category. For example, if “preferred cuisine” is a feature belonging to “Food”, that feature becomes common to all lower level categories. If an individual has “Asian food” as the “preferred cuisine”, that value can be used in all lower level domains.

This is similar to the general ontology described in personalized EPG by Ardissono *et. al.*, (2004). The importance and necessity of ontologies for user modeling is further illustrated in the work of Heckmann *et. al.*, 2005. Their work confirms the possibility of implementing and maintaining ontologies for personalization.

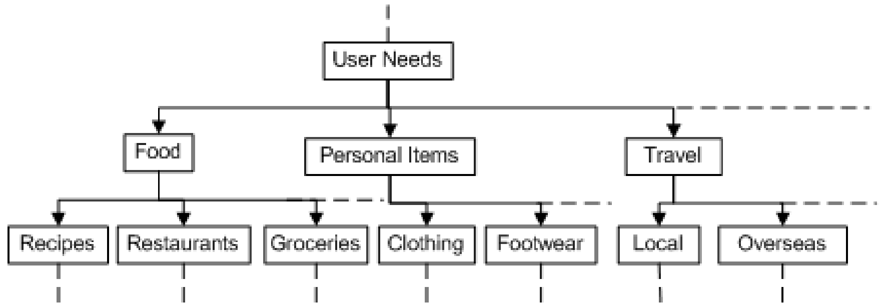

Figure 4.8: Number of domains inter connected as a hierarchy

4.4 The Personal Information Layer (PI Layer)

In order to form the *PI* layer of the user model, users are requested to provide personal details at the registration. The *PI* layer is common for all product domains. Therefore, the start-up information below is re-usable in the future user interactions in all available product domains.

PI layer of the user model contains the following user information.

(i) User identification details such as user name and password.

(ii) User's personal information such as demographics obtained at the registration by filling a form.

(iii) User's Purchase Behavior Characteristics (*PBC* values) calculated using demographics.

As mentioned above, user's personal information is obtained at the registration via filling a form as shown in Figure 4.9. Once the personal information is available, the *PBC* values are calculated. Figure 4.10 demonstrates an example of the *PI* layer of the user "John Smith". The login information is stored under identification details while the demographics obtained at the registration are stored as personal information. The eight *PBC* values shown under "Characteristics", are calculated based on the personal information in the mid section of Figure 4.10.

4.4.1 Identification Information

During the registration, users are requested to provide a user name and a password (Figure 4.8). A user identification number is automatically created during registration for each unique user name and password combination. Similar to any online system, *eHermes*

PERSONAL requests for identity verification, when the user loggs in. Provision of the identification information is mainly required to track consequent interactions of the returning user. In addition, this information is used as a link between different layers of the user model, where each newly created layer 2 for the same individual as well as each layer 3, is identified based on such information.

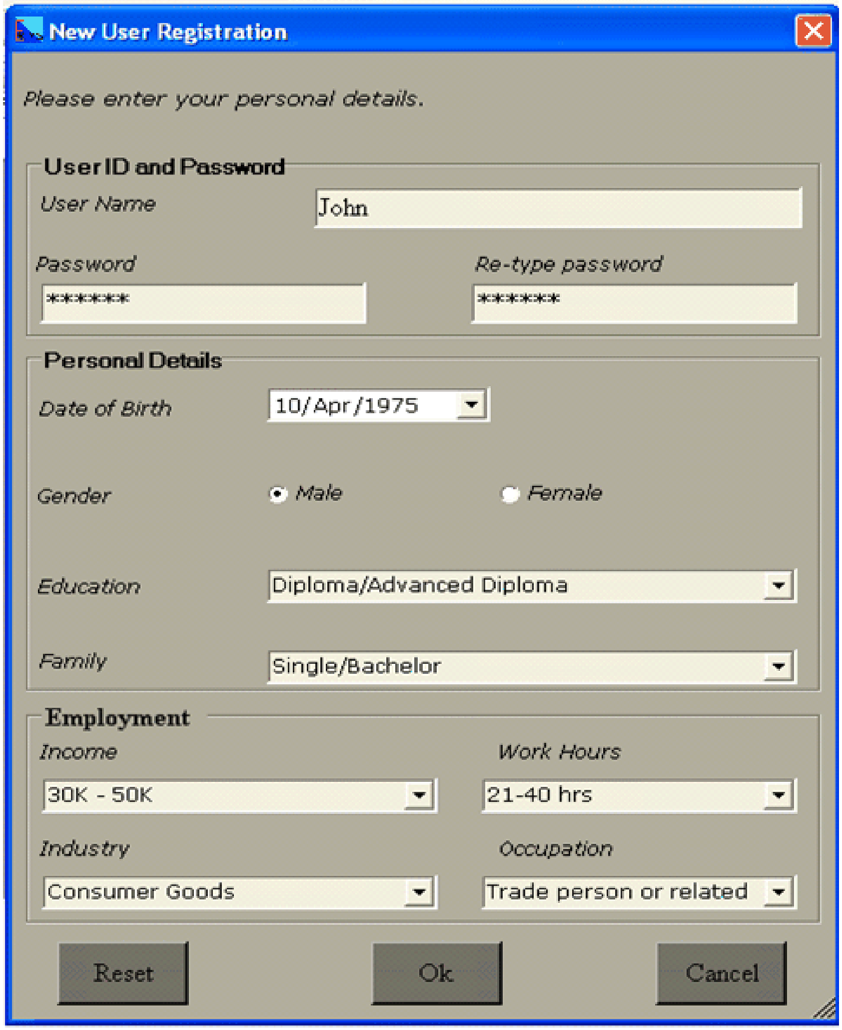

Figure 4.9: Registration Screen

4.4.2 Personal Information

Along with identification information, personal information such as demographic information is obtained at the time of user registration. As shown in Figure 4.8, the information requested are, user's date-of-birth, gender, education, family, income, number-of-hours work, industry and occupation which are exploited in characteristics calculation.

Demographics are powerful information which has the capability of identifying "who the user really is", or more precisely, the abilities of the consumer. For example, one's expenditure depends on the income. On the other hand, if secure handling is provided, obtaining demographics is straightforward by the user explicitly filling out a form. The personal nature of the information can be greatly reduced by the method of collecting them. Rather than directly asking for exact information, users can be asked to select options with ranges of values. For example salary range or age category is less personal than the exact salary or age. Lifestyle Finder (Krulwich, 1997) user modeling system claims, (although they collect demographics) that ninety-three percent of the users surveyed agreed that the questions asked did not invade their privacy.


```xml
<?xml version="1.0" standalone="yes" ?>
<ProfileL1 xmlns="http://tempuri.org/ProfileL1.xsd">
  <IdentificationInfo>
    <Id>30</Id>
    <UserName>John</UserName>
    <Password>jjkkll</Password>
  </IdentificationInfo>
  <PersonalInfo>
    <DOB>10/Apr/1975</DOB>
    <Gender>M</Gender>
    <Family>Single/Bachelor</Family>
    <Education>Diploma/Advanced Diploma</Education>
    <Industry>Consumer Goods</Industry>
    <Occupation>Trade person or related</Occupation>
    <WorkHours>21-40 hrs</WorkHours>
    <Income>30K - 50K</Income>
  </PersonalInfo>
  <Characteristics>
    <TimeSaver>0.53</TimeSaver>
    <PriceSensitive>0.81</PriceSensitive>
    <QualityConscious>0.5</QualityConscious>
    <Fun>0.88</Fun>
    <HealthConscious>0.74</HealthConscious>
    <FamilyPerson>0.36</FamilyPerson>
    <Socializing>0.63</Socializing>
    <Adventurer>0.84</Adventurer>
  </Characteristics>
</ProfileL1>
```


Figure 4.10 : PI layer of the user model belonging to user 'John Smith'.

Deciding what demographics to request from the users is connected with two issues. First, since demographics are highly sensitive to privacy issues, careful attention has to be paid when selecting what demographics to request. Frequently used systems such as mail services, online purchasing guides, and software download sites request some demographic information during the registration. Among the above demographics date-of-birth is commonly requested in most of the web based systems for identification purposes (such as Yahoo_Mail_Service,[18] Hotmail_Mail_Service[19]) as a security measure to avoid unauthorized access. Seta (Ardissono *et. al.*, 2001b), which is a personalized system for dynamically generating web stores, uses demographics such as age group, gender, profession, and the institute to be used in stereotyping. Identification services such as Ms Passport[6] and Liberty Alliance[7] ask for demographic information from their users. The Vals[12] survey, too, use demographic information to decide on an individual's behavior and expectations. Several demographics such as date-of-birth gender and profession are common among most of these systems. Using such commonly requested demographic information will reduce the obtrusive nature of the requests for demographics. Therefore, when deciding on the types of personal data to be requested, the personal information commonly acquired by similar systems was used as a guide.

Secondly, it was necessary to make sure whether this data would truly reveal the characteristics of the users. To select the most powerful demographics, the information used in the Vals survey,[12] LifeStyle Finder (Krulwich, 1997), and Australian Census data[20] were taken into consideration. Demographics used in online datasets such as the Adult dataset[21] were also used as a guide since there is a possibility of using such data sets for future experiments. Finally, ten demographics were selected to be included in the registration form. With regard to obtrusiveness, rather than asking actual values, value ranges make users feel more comfortable to provide true answers. For example, rather than requesting the actual income, requesting to select the appropriate income range from a set of options sounds less obtrusive. Such ranges and demographic categories are also selected as in the above-mentioned systems. Although users are reluctant to reveal their demographics, such information is invaluable in capturing a clearer picture of an individual.

[18] Yahoo_Mail_Service, http://login.yahoo.com/config/login.
[19] Hotmail_mail_service, https://accountservices.passport.net/.
[20] Census_Data, 2001, A Snapshot of Victoria, Census Basic Community Profile and Snapshot. Australian Bureau of Statistics.
[21] Adult_Dataset, Available at http://www.cs.toronto.edu/~delve/data/adult/adultDetail.htm/.

4.4.3 Information on Buying Characteristics

As explained in Chapter 3, in our prototype system, eight *PBC* values in consumers were exploited. Each individual being modeled is described using each of the *PBC* values, where each value ranges in [0, 1]. As mentioned in Chapter 3, these *PBC* values are usable as *general stereotypes* over multiple domains in electronic purchasing. To generate a quantitative value each *PBC* value is represented by a *value function*.

Each value function uses a combination of consumer demographics as inputs and generates a value for the corresponding *PBC* value. In Chapter 5, section 5.2 the value functions are defined using the Multi-Attribute Utility Theory (MAUT) (Schafer, 2001). When forming value functions for *PBC* values, attention has been paid to the following:

(i) how many and which characteristics to use as *PBC*s,

(ii) what demographics contributes to each *PBC* value and

(iii) in what ratios

Selection of *PBC*s to consider was described in Chapter 3. The appropriate demographics to capture each *PBC* value and ratios of the contributing demographics are discussed in Chapter 5 (section 5.2). *PBC* value calculations are illustrated in Chapter 5.

There could be more than eight *PBC*s contributing to consumer buying behavior, but it is not necessary to consider them all to demonstrate the proposed user modeling approach. Since they generate an estimate value for each *PBC* value, the calculated value is an estimate or used only as a start-up value. However, use of more characteristics may results in flexibility describing user preferences towards larger ranges of products, current chosen *PBC* values are appropriate for the demonstration purposes of the thesis. It is obvious that with more knowledge the performance of any knowledge based system could be improved. Therefore, the use of only eight characteristics provides sufficient behavior explanation for many domains, whereas more *PBC* value will definitely explain a consumer better.

As mentioned previously, formulation of *PI* layer is a once-off process. As described in Chapter 3, this layer alone is not capable of providing effective personalization. Having a different *DI* layer for each product domain can capture domain centric consumer needs.

This will allow the consumer to switch among segments as domain changes, rather than directly assigning them to a single static segment. The *DI* layer is explained next.

4.5 Domain Information Layer (DI Layer)

As described previously, the *DI* layer of the user model represents user's requirements and preferences for a given product domain. After registration, if the user is interested in browsing or purchasing in a particular product domain there are two different paths to follow. From the start-up form (Figure 4.4), the user can select either "Personalized Browsing for Members" or "Non Personalized Browsing". The latter option does not involve the user model. If the user is interested in personalized interactions with the system, the interested needs to be selected from the available domains. The available domains are displayed as a list (see Figure 4.11). The selection of the domain initiates the creation of the *DI* layer for that particular domain. For example, if the user seeks personalization in the restaurant domain, then the initial *DI* layer for restaurants domain gets created.

Figure 4.11 : Selecting the domain out of a collection of domains

Information captured in the *DI* layer, is as follows.

(i) User identification

(ii) Product feature numbers in the given domain and attributes under each feature belonging to the same domain

(iii) Current and initial relevance values for each of the above attributes

Once the new layer 2 is created, all the features and their attributes along with calculated relevance values are written into the initial *DI* layer. Part of the *DI* layer generated for user 'John Smith' for the restaurants domain is shown in Figure 4.12. Each of the information types are described below.

4.5.1 Identification Information

As mentioned in section 4.4.1, user identification information is used to relate the *DI* layer to other layers belonging to the same user, by linking the PI layer and any existing *TI* layers to the *DI* layer. In each *DI* layer users are individually identified by their user id. The domain id uniquely identifies the domain and its position in the hierarchy. User John Smith is identified by the unique identification number '30' and the domain id of the restaurants domain is shown as '1,2,2'(Figure 4.12 (a)). The *relevance value* of each attribute is given along with the attribute number and the corresponding feature number (Figure 4.12 (b)).

```
<?xml version="1.0" standalone="yes" ?>
- <ProfileL2 xmlns="http://tempuri.org/ProfileL2.xsd">
  - <Identification>
      <UserId>30</UserId>
      <DomainId>1,2,2</DomainId>
    </Identification>
```

(a)

```
  - <Preferences>
      <FeatureNum>2</FeatureNum>
    - <Feature>
        <Attribute>163</Attribute>
        <Init-Relevance>1</Init-Relevance>
        <Relevance>1</Relevance>
      </Feature>
    - <Feature>
        <Attribute>165</Attribute>
        <Init-Relevance>0.76</Init-Relevance>
        <Relevance>0.76</Relevance>
      </Feature>
    - <Feature>
        <Attribute>167</Attribute>
        <Init-Relevance>0</Init-Relevance>
        <Relevance>0</Relevance>
      </Feature>
    - <Feature>
        <Attribute>169</Attribute>
        <Init-Relevance>0</Init-Relevance>
        <Relevance>0</Relevance>
      </Feature>
    </Preferences>
```

(b)

Figure 4.12 : Sections of the DI layer of the same user "John Smith".

4.5.2 Product features and Attributes

Domain features are product descriptions that are important to consumers when deciding which product to purchase (see Definition 2 in section 4.3.2). For example the restaurant domain consists of 31 domain features such as Cost, Décor, Service, etc. Attributes of a domain are defined in section 4.3.2 under Definition 3. At the creation, *DI* layer refers to the domain hierarchy and obtain the domain feature numbers and their corresponding attribute numbers. Such domain information is written into the freshly created *DI* layer. Since each attribute is relevant to an individual to a certain degree, user preference towards each item attribute is indicated as a *relevance value*. The initial relevance value calculation is described below. Table 4.2, contains a small sample of data showing features, attributes, personal information related status, init-relevance and current relevance. This data is obtained from the *DI* layer belonging to a returning user (Experiments of User41- see chapters 6 and 7). The init-relevance values and the updated current relevance values are given for each attribute. In addition, whether an attribute is related to personal information or not, is also shown.

4.5.3 Relevance Values

The relevance value of an attribute specifies the relevance of a particular attribute to the user within a given domain. In other words, the value represents the importance of the attribute to the user in question. The user model keeps track of two important relevance values for each attribute; the initially assigned one and the current relevance. The initial relevance is based on the start-up information available. The latter is the updated current relevance of the attribute. Availability of both values supports future evaluations such as user trends.

Table 4.2 : A sample set of data with descriptions

Feature	Personal-info Related or not	Attribute	Init-Relevance	Relevance
Cost	Related	below $15	0	0
	Related	$15-$30	0.44	0.65
	Related	over $50	0.56	0.5
	Related	$30-$50	1	0.8
Location	Related	Long Drive	0	0
	Related	Walk	0	0.05
	Related	Short Drive	0.28	0.22
	Related	Central	1	0.8
Atmosphere	Not related	An Out of The Way Find	0	0.05
	Not related	Good Out of Town Business	0	0.05
	Not related	Hip Place To Be	0	0.05

At the creation of the new *DI* layer, relevance values are assigned to only *the attributes that are related to personal information*. Relating certain domain features to an individual's personal information is previously carried out in work employed stereotyping. Notable citations include (Rich, 1989; Kay, 1994; Ardissono et.al. (1999, 2004); Spiekermann, Grossklags and Berendt, 2001). Usually such assumptions are made based on survey results (Ardissono *et. al.*, 1999; Ardissono *et. al.*, 2004) or based on expert knowledge (Spiekermann *et. al.*, 2001) or even based on common-sense (Rich 1979). In our work, when building the prototype system, such assumptions were made supported by the information in existing work and common-sense. In this thesis, such product attributes and features which are related to personal information are referred to as Personal Information Related attributes (PIR-attributes) and Personal Information Related features (PIR-features). In table 4.2 (above), the second column indicates if the given sample of attributes is PIR-attributes or not.

What we call the Influence Matrix (*IM*) is a matrix which contains all such attributes which can be related to personal information/buying behavior characteristics, along with their *influence thresholds*. *Influence threshold* is a range for a given characteristic value within which the corresponding attribute is relevant for the user. More information on the attributes in the *IM*, *IM* itself and *Influence thresholds* are discussed in Chapter 5 in detail.

At the point of a new *DI* layer creation, attributes that appear in the *IM* receives a relevance value. If there are attributes common to the new domain and already existing domains, such attributes inherit their initial relevance values from the existing attributes. The rest of the attributes which are not belonging to either of above types, are considered as *irrelevant to the user* at the start and hence receives a zero relevance value.

Based on user preferences, in consequent interactions the relevance values of all the attributes including the initially irrelevant ones are updated. User preferences are more or less fuzzy. For example, rather than labeling a certain attribute as 'like' or 'dislike', an individual may prefer it to a certain extent. To capture the amount of preference, the work in this thesis follows a fuzzy approach to calculate relevance values. The relevance values in the last two columns of Table 4.2 (above) are obtained from a user (User41) involved in the experimentation process after number of interactions with the system in restaurants selections. The difference between the initial relevance value and the current value indicate the update of the initial value according to the user behavior. A detailed discussion on relevance value calculation methods and algorithms are in Chapter 5. The required transaction information and the layout of the *TI* layer are next discussed.

4.6 Transaction Information Layer (TI Layer)

Although the preferred item attributes along with their relevance values could be found in the *DI* layer of the profile, it is necessary to capture user's current preferences or desires that are important for the current product search since there can be changes in users *usual* buying preferences. The main use of the *TI* layer is to capture and keep track of such new changes in user's preferences for each transaction. The future, analysis of the *TI* layers could reveal important behavioral changes and provide a good understanding of the user.

eHermes PERSONAL provides a personalized interactive product retrieval process, where individuals declare their preferences in different stages of the search. (The product retrieval strategy is discussed in Chapter 7 in detail). The search starts with a user specified start-up query. Then the search process guides the user towards the products he/she may prefer by using the information in the corresponding *DI* layer. During the search process user may acquire more knowledge about the domain and available options. To further constrain the initial query, user may explicitly choose additional options during the search process. Therefore, at the end of the interaction there will be two sets of preferences: the preferences declared in the initial query and the options selected during the search process. Each *TI* layer keeps track of these explicit *choices* made by a consumer during a single transaction.

In the online encyclopedia Wikipedia,[22] a transaction is defined as follows:

> "*a transaction is an agreement, communication, or movement carried out between separate entities or objects, often involving the exchange of items of value, such as information, goods, services and money*."

According to the definition, each time the user interacts with the system is considered to be a single transaction. The Figure 4.13 hows preferences declared by a user during a transaction. The three features shown in the example *TI* layer are either specified by the

[22] Wikipedia URL: http://en.wikipedia.org/wiki/Transaction

user at the initial transaction or they could be answers to the questions presented during the product selection process.


```
<?xml version="1.0" standalone="yes" ?>
- <ProfileL3 xmlns="http://tempuri.org/ProfileL3.xsd">
  - <Identification>
      <UserId>30</UserId>
      <DomainId>1,2,2</DomainId>
      <TxnId>1</TxnId>
    </Identification>
```


(a)


```
<?xml version="1.0" standalone="yes" ?>
- <ProfileL3 xmlns="http://tempuri.org/ProfileL3.xsd">
  - <Identification>
      <UserId>30</UserId>
      <DomainId>1</DomainId>
      <TxnId>1</TxnId>
    </Identification>
  - <Preferences>
      <FeatureNum>1</FeatureNum>
    - <Feature>
        <Attribute>117</Attribute>
        <Relevance>1</Relevance>
      </Feature>
    </Preferences>
  - <Preferences>
      <FeatureNum>2</FeatureNum>
    - <Feature>
        <Attribute>163</Attribute>
        <Relevance>1</Relevance>
      </Feature>
    </Preferences>
  - <Preferences>
      <FeatureNum>11</FeatureNum>
    - <Feature>
        <Attribute>63</Attribute>
        <Relevance>1</Relevance>
      </Feature>
    </Preferences>
  </ProfileL3>
```


(b)

Figure 4.13 : Sections of TI layer of "John Smith" for transaction "TxnId =1"

Information captured in each *TI* layer is as follows.

(i) User identification

(ii) Product feature numbers in the given domain and attributes under each feature belonging to the same domain

(iii) Relevance value of the attributes for the current transaction

The first two types of information are the same as in the sections 4.5.1 and 4.5.2. The *choice* is represented by a set of domain attribute values which were explicitly specified by the consumer during the transaction along with a relevance value. The relevance value is allocated depending on the feature value provision stage and is valid only for the current transaction. For example, a feature specified in the initial query receives a higher relevance value than a feature provided to a system question during the interactive process. The interactive personalized search method employed combining the user model is discussed in Chapter 7 in length.

After each product search, the explicit attribute preferences provided by the user are used to update the existing preferences in the corresponding *DI* layer, by adjusting the existing relevance values. This new preferences obtained during the transactions either confirm or contradict the initial value preferences calculated for the *DI* layer. To capture such dynamism and to gradually update the changing user preferences, a Hebbian network (Hebb, 1949) has been employed. Chapter 5 presents a more detailed version about Hebbian learning and the updating process of the user model layers, while the following section provides a brief description.

4.7 Updating the User Model

In dynamic environments new products arrive frequently to the market and the users tend to change their purchase behavior inspired and affected by advertising and availability of product variety. Therefore, updating the user model according to the changing user behavior is necessary. The proposed user model updates its contents in two different stages:

1. after each personalized user transaction with the system and
2. once the user model is mature where user interacted in several domains then the buying characteristics values needs updating.

We explain these stages below:

4.7.1 Update after each transaction

When a user expresses interest in a new domain, the initial *DI* layer of the user model gets created for that particular domain. Then, any transaction in that domain creates a *TI* layer for that particular domain. If user expectations in a particular domain are changed, that will be gradually captured in the consecutive *TI* layers created for that domain. Since the *DI* layer represents the current user expectations in a given domain, the *TI* layer information is used to update the contents of the *DI* layer.

It is a requirement of a user model to capture user's changing needs as well as remembering the user's past preferences. Therefore, to accomplish both expectations a slow learning approach is employed. This way, the new preferences are added to the user model while not losing a representation of the previous behavior. The details of the update process are described in Chapter 5.

4.7.2 Long-term buying characteristic update

As explained in section 4.4.3, the *PBC* values in the *PI* layer are initially calculated with the user's demographic information. Since there are many assumptions related to such stereotypic calculations, they are prone to many errors. As in any system using stereotypes, our main intention was to use this information as start-up assumptions. Therefore, in the long term, once the user interacted with several domains, the initial *PBC* values are updated using the newly available information. Since the newly calculated values are based on the actual transactional behavior, they are of higher precision. Once the newly calculated values are available, if the user seeks personalization in a domain in which he/she had previously not interacted, the initial preference values for that domain are calculated based on the recalculated *PBC* values. The use of updated *PBC* values increases the accuracy of initial relevance values of the newly interacted domain. Recalculated *PBC* values may deviate from the initial allocated values due to several reasons such as:

(i) if changes of demographics encountered (for example, if an individuals living conditions are changed, such as a bachelor getting married. This may demonstrate a considerable change in his/her purchasing habits).

(ii) if the individual is a "grey sheep" who is not behaving as expected, but contradict the stereotype behavior

(iii) if there are errors in the definition of *value functions* used for initial *PBC* value calculations

The above reasons may show considerable changes in the recalculated *PBC* values. Therefore, this periodical update can be considered as a rectification process to ensure the correctness of the user model. The algorithms employed in the recalculation process are demonstrated in Chapter 5.

4.8 Obtrusiveness in Personalization

As described in the introduction and Chapter 2, unobtrusiveness is an important issue in information gathering. According to the issues discussed in Chapter 2 section 2.4, the obtrusive nature of the information increase up the layers (see Figure 4.14).

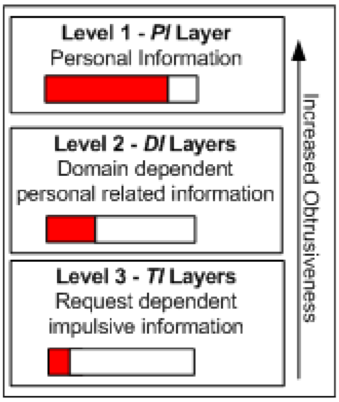

Figure 4.14 : Obtrusiveness of the information along layers of the user model

In the work described in the thesis, there are two instances where system request for user information; during creating the user model and during the *PIPRP*.

4.8.1 During creation of the user model

Although, the nature of the information in the *PI* layer is highly obtrusive to request from a user, such information is kept common to all the domains. Acquisition of personal information is a once-off process and the concept of a single user model over multiple domains results in reuse of user information. In addition, information in *DI* layers is generated based on the information in *PI* layer. Therefore, we believe that the issue of obtrusiveness is adequately addressed during the user model creation via information reuse.

4.8.2 During the *PIPRP*

Information regarding the current user requirement is requested during the product retrieval process (*PIPRP*). The system questions directed to the user are determined based on the user model. Due to such personalization, only the questions that show an importance in the user model are directed to the user. Therefore, the comprehensiveness of the questions increases. This also results in a less number of questions. In addition, the questions that are related to user's personal information are minimized since such information is available with the user model. Therefore, the obtrusiveness of the system is controlled in several ways. Chapter 7 discusses such strategies followed during the product retrieval process.

4.9 Summary

Table 4.3 summarizes the user model information. It shows the number of instances exists in each layer, information content, initially generated values, and when the contents are updated during the user model usage.

Table 4.3 : Summary of the layers of the user model

Profile Layer	Instances	Type of information	Initial values	Value updates
PI layer	Only one	Identification data	At the time of registration	Once-off process
		Personal data/Demographics	At the time of registration.	Once-off process. But the user can update his/her personal details if the information changes.
		PBC values	At the time of registration, initially calculate using the demographic data.	As *DI* layer preferences changes, characteristics are updated periodically.
DI layer	Similar to the number of different domains that the user has interacted.	Identification information and domain number	At the time of creation to link with *PI* layer of the model	Never being updated
		Product features and attributes of the corresponding domain along with user preferences as relevance values.	*IM* fills in the product features and attributes of the corresponding domain. Initial relevance values are calculated based on user characteristics.	Updated after every transaction in the corresponding domain.
TI layer	Similar to the total number of interactions within each domain	Identification details, domain number and transaction number	At the time of creation so as to link with *DI* layer of the model	Never being updated
		User's explicitly specified preferred attribute number along with the relevance value	Obtained from the initial query specification - highest relevance	Never being updated
			Obtained during the question asking process- next highest relevance	Never being updated

In this chapter the new user model architecture and the framework within which it has been employed is presented. The proposed user model captures user behavior in three abstract levels as user's general buying behavior, domain centric behavior and transaction based behavior. Although such abstraction seems to be three layered, due to product taxonomies it is of n-layers.

The next chapter further describes the functionality of the user model with respect to algorithms and the techniques employed.

Chapter 5

Implementation of the Layered User Model

In the previous chapter the architecture of the novel user model was discussed in detail explaining the contents of information layers and their functionality. The discussion included the creation of each layer during different stages of user interactions and the update of user information using the most recent user preferences. In this chapter we explain the algorithms involved in the above process of generating and updating the user model. A number of techniques such as Multi-Attribute Utility Theory (MAUT), fuzzy logic based preference calculations and Hebbian type adaptive learning were utilized. The purpose of this chapter is to provide an understanding of these techniques as well as how these are used and combined to achieve the proposed model.

The flow of the chapter is as follows. Section 5.1 explains the user model generation process. The entire process of user model generation is described as an algorithm. Section 5.2 presents the calculations of purchase behavior characteristics (*PBC*) when forming the PI layer of the user model. The section 5.3 explains the ability of the above calculated *PBC*s to act as a *general stereotype* in online purchases. Section 5.4 describes the influence matrix (*IM*) and its contribution in *DI* layer formation. Section 5.5 describes domain based initial preferences, while section 5.6 presents the techniques and algorithms related to calculation of such preferences. Section 5.7 further discuss the relevance values and define Average Total Relevance (*AVR*) of a domain feature. In section 5.8, the updates carried out in the user model are described. The updates are carried out both after each transaction and periodically. Finally section 5.9 summarizes the chapter.

5.1 User Model Generation Process

In this section, the process of generating, populating and initializing different layers as well as learning and adaptation related to the user model are presented as a sequence of steps.

5.1.1 The Algorithm

The users' initial registration begins the process and generates the *PI* layer of the user model. Once user shows interest in a certain domain for personalized interactions, the *DI* layer for that domain is generated making use of the existing *PI* layer. The *TI* Layer, which is populated from purchase transactions, is used to update the initial values in the *DI* layer. The steps are presented as an algorithm below.

Step 1: User registration; provide personal details/demographics.

Step 2: Calculate *PBC* s from the above information using MAUT.

Step 3: Calculate product feature preferences in the domain of interest using an influence matrix (described later) which combines the knowledge about consumer attribute preferences and the calculated values of the *PBC* s.

Step 4: The interactive product selection process, which is supported by the DI layer of the user model. The selection process is described in detail later in Chapter 7.

Step 5: Populate layer 3 with attribute values for individual purchase transactions (from step 4 selection process)

Step 6: Update domain based user preferences in the corresponding DI layer using Hebbian learning.

Step 7: Once the user has interacted with more than one domain, the domain preferences in DI layers are combined to update the initial *PBC* values calculated in the PI layer.

The next section presents this algorithm as a flow chart.

5.1.2 Flow Chart of the Algorithm

In Figure 5.1, the steps involved in the above algorithm are presented as a flow chart. The flowchart starts at the point of user request for personalized services in a domain that the user wishes to interact and ends when the updates to the user model are finalized.

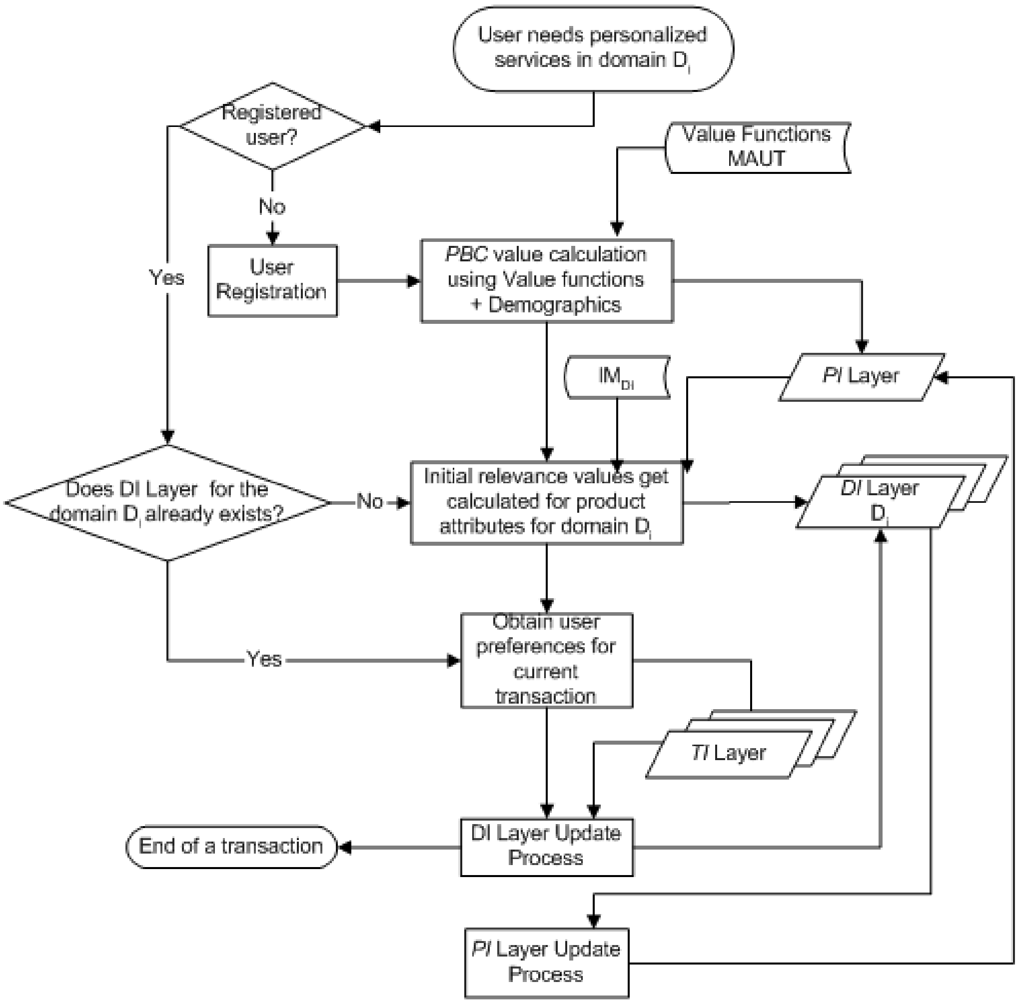

Figure 5.1 : The user model generation process illustrated as a flow chart

However, in the flow chart the *PI* layer update process is isolated from the rest of the flow since it does not occur with each and every transaction. It is a periodical update, which may show effective results after the user has interacted with the system several times. The following subsections describe the techniques used in individual processes of the above algorithm.

5.2 Calculation of User PBC Values Using MAUT

As described in step 1 of the algorithm, during the registration phase, users are requested to provide certain demographic information. Such information is then utilized to calculate the *PBC* values expected in purchasing. As explained in Chapter 4, each *PBC* is represented by a value function based on demographics. There are ten (10) demographic values used in this work for *PBC* value calculations. These demographics are use in eight (8) value functions. The demographics are namely; Family Type, Gender, Work Hours, Age Group, Income, Occupation, Industry, Income and Education. The value functions generate estimated values for time saver, price sensitivity, quality consciousness, fun spending, health consciousness, family person, socializing, and adventurer *PBC* values. Each of the *PBC* value is formed using one or more demographics. Each demographic contribute a certain weight to the connected *PBC* value. The value functions are defined using the Multi-Attribute Utility Theory (MAUT) (see Schafer, 2001). MAUT is explained the next section.

5.2.1 Multi-Attribute Utility Theory (MAUT)

Multi-Attribute Utility Theory is an evaluation scheme which is very popular by consumer organizations for evaluating products. According to MAUT, the overall evaluation $v(x)$ of an object x is defined as a weighted addition of its evaluation with respect to its relevant *value dimensions* (Schafer, 2001). The overall evaluation is defined by the following *overall value function*:

$$v(x) = \sum_{i=1}^{n} w_i v_i(x)$$

Here, $v_i(x)$ is the evaluation of the object on the i^{th} value dimension d_i, and w_i the weight determining the impact of the i^{th} value dimension on the overall evaluation (also called the relative importance of a dimension), n is the number of different value dimensions, and

$$\sum_{i=1}^{n} w_i = 1.$$

In Schafer (2001), the evaluation of $v_i(x)$ for each value dimension d_i is defined as the evaluation of the relevant attributes, as follows;

$$v_i(x) = \sum_{a \in A_i} w_{ai}\, v_{ai}(l(a))$$

(note: If sub-dimensions are involved, the evaluation of the object on a dimension would be defined similar to the overall evaluation, by a weighted addition of the evaluation of the object with respect to its sub-dimensions).

Here, A_i is the set of all attributes relevant for the i^{th} value dimension (d_i), $v_{ai}(l(a))$ is the evaluation of the actual level $l(a)$ of attribute a on d_i. w_{ai} is the weight determining the impact of the evaluation of attribute a on value dimension d_i (also called the *relative importance* of attribute a for d_i). $\forall d_i$ where ($i = 1,.., n$) hold $\sum_{a \in A_i} w_{ai} = 1$.

Use of MAUT in formation of value functions are discussed next.

5.2.2 Derivation of the Formula

For each user x, $\exists\, ch_i(x)$ where (i= 1, 2… n) and $\forall i \; 1 \geq ch_i(x) \geq 0$

Using MAUT, for an individual x, i^{th} *PBC* value $ch_i(x)$ is given by:

$$ch_i(x) = \sum_{j=1}^{n} w_j\, v_j(x) \quad \text{where} \quad \sum_{j=1}^{n} w_j = 1$$

$v_j(x)$ is the evaluation of a given *PBC* value on the j^{th} value dimension d_j. In this case each demographic becomes a value dimension. Therefore, $v_j(x)$ is a demographic value, and w_j is the weight determining the impact of the demographic value on the overall evaluation. In other words, w_j is the relative importance of a demographic type on a given *PBC* value. n is the number of different types of demographics related to the *PBC* value $ch_i(x)$.

In the next section, calculation of each of the *PBC* value is illustrated along with the contributing demographics.

5.2.3 Value Functions for *PBC* Values

This section further explains derivation of value functions for the eight *PBC*s and calculation of quantitative values for individuals. Table 5.1 shows the eight value functions and the demographics used in forming each of them. The 'X' indicate the presence of each demographic. For example, Time Saver *PBC* value function is based on family type, gender, work hours, age group and occupation. Forming a value function for Time Saver *PBC* value is explained in detail below.

Table 5.1: PBC s and contributing demographics

Demographics	***PBC***							
	Time saver	**Price sensitive**	**Quality conscious**	**Fun spending**	**Health conscious**	**Family person**	**Socializing**	**Adventurer**
Family Type (***FT***)	X	X		X		X	X	X
Gender (***G***)	X				X			
Work Hours (***WH***)	X							
Age Group (***AG***)	X	X		X	X	X	X	X
Income (*Inc*)		X	X	X			X	X
Occupation (***Occ***)	X		X				X	
Industry (***Ind***)				X			X	
Education (***Edu***)					X			

As mentioned above, MAUT is used to evaluate the weighted contribution of each demographic value towards individual's *PBC* values. Allocation of demographics to each *PBC* value and their corresponding weight (the impact of the given demographic on a given *PBC* value) needs justification. As highlighted in Chapter 4, section 4.4.2 these issues were handled based on the existing work. Furthermore, as explained in Chapter 3, section 3.5.3 formation of value functions holds a similarity to clustering approaches in existing systems such as Vals system,[12] and Roy Morgan Value segment.[13]

Each of the calculated values is estimates of user behavior which is only used as start-up information. Therefore, less attention is paid to the accuracy of exact demographics that might contribute to the *PBC* values and also the contributing ratios. In the formulas, references are made to demographics using abbreviations as shown in the Table 5.1.

Time Saver *PBC* (*Char*$_{TS}$)

The Time Saver *PBC* ($Char_{TS}$) measures an individual's tendency to save time whenever they make a purchase. For example, purchasing goods with time saving abilities such as easy to use goods, recipes with short cooking preparation times, near by restaurants etc.,. It is assumed that people tend to save time on purchasing or performing tasks due to two reasons; either they are too much involved in their other work (such as job or family activities) or if they are not interested in doing a particular task (e.g. not interested in cooking, gardening etc). Measuring people's family/work involvements could be measured using demographics but interests are only captured by long term behavior analysis. Since only an estimate of $Char_{TS}$ is required, and due to lack of behavior information at the start-up, the behavior contribution is not included in the calculation.

Therefore, $Char_{TS}$ is formed as follows.

$$\mathrm{Char}_{TS} = (w_1 \times \mathrm{family\_involvement}) + (w_2 \times \mathrm{Work\_Involvement})$$

w_1 and w_2 are weights allocated to the attributes "family involvement" and "work involvement" respectively based on their relative importance. For example, based on intuition a higher weight for the work component than the family involvement is allocated ($w_1 > w_2$ and $w_1 + w_2 = 1$).

Again, each value dimension can be broken down into simpler attributes. Substituting the notations given in Table 5.1, in the above formula,

$family\_involvement = (w_3 \times FT) + (w_4 \times G)$ where ($w_3 > w_4$ and $w_3 + w_4 = 1$), and
$work\_involvement = (w_5 \times WH) + (w_6 \times AG) + (w_7 \times Occ)$
where ($w_5 > w_7 > w_6$ and $w_5 + w_6 + w_7 = 1$)

Therefore, the value function for Time Saver *PBC* is given by:

$$Char_{TS} = (w_1 \times ((w_3 \times FT) + (w_4 \times G))) + (w_2 \times ((w_5 \times WH) + (w_6 \times AG) + (w_7 \times Occ)))$$

Here, w_i $\forall i \in \{1,2,\ldots n\}$, is the relative importance of each demographic value, where n is the total number of contributing demographics.

Example: The weights were chosen based on the importance of each demographic towards the *PBC* value. As explained previously, importance of the weights were determined according to the intuition of the authors. The normalized weight values used in the implementation were $w_1 = 0.4$, $w_2 = 0.6$, $w_3 = 0.9$, $w_4 = 0.1$, $w_5 = 0.6$, $w_6 = 0.1$, and $w_7 = 0.3$.

$$Char_{TS} = (.4 \times ((.9 \times FT) + (.1 \times G))) + (.6 \times ((.6 \times WH) + (.1 \times AG) + (.3 \times Occ))).$$

Using the coded values in Tables 5.2-5.6, a 32 year old male bachelor who is a professional, working 30 hrs a week receives a $Char_{TS}$ value of 0.586 which becomes 0.489 $\cong$ 0.49 after normalizing between 0-1.

Table 5.2 : Attribute Codes for FT

Family type	Value
Single/Bachelor	0.3
Couples without kids	0.3
Couples with young kids	0.9
Couples with older kids	0.7
One Parent with young kids	1
One Parent with older kids	0.8
Other	0.4

Table 5.3: Attribute Codes for WH

Work hours	Value
Not employed	0
Less than 20 hrs	0.5
21-40 hrs	0.75
more than 40 hrs	1

Table 5.4: Attribute Codes for G

Gender	Value
Female	0.5
Male	0.4

Table 5.5: Attribute Codes for Occ

Occupation	Value
Managerial/Administration	0.9
Professional	0.9
Associate Professional	0.6
Trade person or related	0.6
Intermediate Clerical/(Sales or Service worker)	0.4
Laborer or related	0.4

Table 5.6: Attribute Codes for AG

Age Range	Value
less than 20	0.5
21-25	0.5
26-30	0.5
31-35	0.5
36-40	0.5
41-45	1
46-50	1
51-55	1
56 - 60	1
more than 60	1

Coding is based on several assumptions. An individual with young kids is believed to have more family commitments than a single person. A female is believed to spend more time on family than a male. The work involvement is considered to be affected from the work hours, age group of the person and the occupation. The high work hours are expected to have a positive effect on the timesaver *PBC*. It was assumed, that people in the middle age groups will work harder to achieve their goals in career, while older and younger people have a relaxed approach. With regard to the occupation, higher the position, greater the responsibility which enforce a positive effect on this particular *PBC*.

Price Sensitive *PBC* (*Char*$_{\text{PS}}$)

Demographics such as age group, family type and the income are assumed to have an effect on the price sensitivity of a person. Large young families with a lower income are considered as highly price sensitive while younger age bachelors are expected to be less price-sensitive. The income of a person considered to be the highest contributing factor towards $Char_{\text{PS}}$, while age group and the family type contribute equally. Hence $w_3 > w_1$ and $w_1 = w_2$.

$$Char_{\text{PS}} = (w_1 \times AG) + (w_2 \times FT) + (w_3 \times Inc)$$

Here, $w_i \ \forall i \in \{1, 2, \ldots, n\}$, is the relative importance of each demographic value, where n is the number of contributing demographics.

Quality Consciousness *PBC* (*Char*$_{\text{QC}}$)

An individual's quality consciousness is less apparent in his/her demographics. However, an individual's income and expected living/social status decide the quality of the goods they purchase. It was assumed that occupation has an indirect influence on the social appearance. Therefore, income and occupation are used as contributing dimensions towards quality consciousness, where $w_1 > w_2$.

$$Char_{\text{QC}} = (w_1 \times Inc) + (w_2 \times Occ)$$

Here, $w_i \ \forall i \in \{1, 2, \ldots, n\}$, is the relative importance of each demographic value, where n is the number of contributing demographics.

Fun Spending *PBC* (*Char*$_{\text{FS}}$)

This *PBC* value shows individual's willingness to pay for fun activities. Since these are not basic needs, individual's income does a larger contribution. Family type shows the available time, and money for fun activities. Also we assume that people of younger age tends to spend more money on fun activities. People involved in certain Industries such as Media/Publishing/Entertainment or Advertising/Marketing/PR can be considered as more involved in fun activities due o the nature of the work they are involved. Therefore, age group, family type, industry and income are considered as contributing demographics towards $Char_{FS}$. Age group, family type and the income are assumed to have a higher impact on the *PBC* value rather than the industry. The weights were chosen as follows; $w_4 < w_1$ and $w_1 = w_2 = w_3$

$$Char_{FS} = (w_1 \times AG) + (w_2 \times FT) + (w_3 \times Inc) + (w_4 \times Ind)$$

Here, $w_i\ \forall i \in \{1, 2, \ldots, n\}$, is the relative importance of each demographic value, where n is the number of contributing demographics.

Health Consciousness *PBC* $Char_{HC}$

For this particular *PBC*, a person's age is considered as the highest contributing factor. Apart from that, educated people are expected to be more health conscious considering their knowledge about the nutrition and healthy food. Again according to commonsense, women are more concern about issues such as weight gain. Therefore, age group, education and gender were assumed as contributing factors where the latter two contribute equally. Hence, $w_3 > w_1$ and $w_1 = w_2$.

$$Char_{HC} = (w_1 \times AG) + (w_2 \times G) + (w_3 \times Edu)$$

Here, $w_i\ \forall i \in \{1, 2, \ldots, n\}$, is the relative importance of each demographic value, where n is the number of contributing demographics.

Family Person *PBC* ($Char_{FP}$)

An individual with a high value for $Char_{FP}$ is considered as purchasing for a family. People with young families are assumed to have higher values. Therefore, both family type and the age group were considered to be contributing towards this *PBC*. Since the family type has a greater effect on $Char_{FP}$, the weights were chosen as follows: $w_1 > w_2$.

$$Char_{FP} = (w_1 \times FT) + (w_2 \times AG)$$

Here, $w_i\ \forall i \in \{1, 2, \ldots, n\}$, is the relative importance of each demographic value, where n is the number of contributing demographics.

Socializing *PBC* ($Char_{SO}$)

This *PBC* evaluates an individual's willingness to spend on socializing. This includes expenditure for maintaining an individual's social status. It is assumed that persons in high ranks of selected industries (such as marketing, media etc) spend more on social status. Therefore, occupation and industry attributes have a combined effect on $Char_{SO}$. In addition, age group and family type are expected to have a less impact than income. Hence, the weights were chosen as $w_1 > w_2 > w_3 > w_4$.

$$Char_{SO} = (w_1 \times AG) + (w_2 \times FT) + (w_3 \times (Occ + Ind)) + (w_4 \times Inc)$$

Here, $w_i \ \forall i \in \{1, 2, \ldots, n\}$, is the relative importance of each demographic value, where n is the number of contributing demographics.

Adventurer *PBC* ($Char_{AD}$)

This *PBC* attempts to capture the individual's tendency towards trying out new products. Although adventurous behavior is expected to be apparent in user's buying patterns rather than in demographic information, age group, family type and income are selected as effective factors. Younger people with less experience in purchasing goods and bachelors with less family commitments are assumed to demonstrate a strong adventurous *PBC*. However, the income can impose restrictions to such people. Therefore, the three demographics were expected to contribute as follows:

$$Char_{AD} = (w_1 \times AG) + (w_2 \times Inc) + (w_3 \times FT), \text{ where } w_1 > w_2 > w_3.$$

Here, $w_i \ \forall i \in \{1, 2, \ldots, n\}$, is the relative importance of each demographic value, where n is the number of contributing demographics.

Each of these *PBC* values is scaled between 0-1. The effect of *PBC* values on user behavior is presented in Table 5.7. The low range of the values (0 - 0.5) and the high range of the values (0.5 – 1) are considered to have opposite effects on the user behavior. The next section describes the importance of the above *PBC* values as start-up information or stereotypes.

5.3 Use of PBC Values as General Stereotypes

As explained in Chapter 3, once put together, the above calculated *PBC* values provide a general picture of an individual's behavior in purchasing. Since each *PBC* imposes a quantitative value for each individual, this value can be considered as a description of the user. On the other hand, the scaling of the value functions makes it a relative value. Therefore, the calculated value becomes a relative description of the individual with respect to the population of the other buyers. For example, an individual with 0.5 values for $Char_{PS}$ can be described as 'relatively medium price sensitive person among the buyer population'.

Table 5.7 : Effect of PBC on relevance of product/item attributes

PBC	**Expected purchasing behavior when low**	**Expected purchasing behavior when high**
Time Saver	Not bothered about saving time	Go for easy to use goods, such as pre-cooked groceries, shops in closer location, may sacrifice the price.
Price Sensitivity	Pay more attention to other aspects of the item.	High tendency for buying cheaper items
Quality consciousness	Other attributes such as price get more attention	Tendency to go for popular brands, not necessarily the most expensive
Fun spending	Concern about the bare minimum	Willing to pay for more than basic needs
Health consciousness	Pay more attention to other aspects of the item.	Generally concern about health
Family person	Buying for themselves	Buying for a family
Socializing	Pay less for outer appearance	Pay for the need to get blended with the others
Adventurer	May stick to the same items rather than moving from brands etc	Try out different options, show less experience in purchasing

Since there is a set of such *PBC* values, each individual is described using several dimensions. Each of the *PBC*s, act as start-up information similar to a stereotype, conveying some information about the user. Although the purchasing behavior is evaluated with the *PBC* values, they do not tie-up the individual to a certain narrow purchasing domain. Instead, the values provide a description that is common to all purchasing domains. Therefore, the combined set of the *PBC* values can be considered as a *general stereotype* of the individual which is valid within the entire eCommerce purchasing domains.

Personal Behavior al Characteristics (*PBC* values) ≡ *General Stereotypes*

As mentioned in Chapter 2, section 2.3.1, work by Kay (1994; 2000) have conducted an extensive survey about stereotypes and discussed different approaches. As they expect, a system using stereotypical reasoning should have a *database of stereotypes* where each stereotype is activated by a *trigger* and the inferences should have some uncertainty. The work described in this thesis attempts to make use of the notion of stereotyping but at the same time keeping the stereotypes as flexible as possible, such that they can be tailored to each individual. Therefore, a database of stereotypes is not maintained; instead the *PBC* values are used as *general-stereotypes*.

The *PBC'*s do not behave as traditional stereotypes. Calculation of *PBC* values is only the start of the stereotyping process. The two processes followed in traditional stereotyping and general stereotyping are depicted in Figure 5.2 and 5.3 respectively.

As shown in Figure 5.2, new approach achieves the total effect of stereotyping in 3 steps. In the **first step**, quantitative values for *PBC*s are calculated for the individual, substituting the demographics in value functions. The **second step** is initiated for a particular domain in which the user seeks personalization. The domain *IM* (which is described in the next section) maps the *PBC* values of the user to the domain attributes, depending on the domain based threshold values given for *PBC* values in the *IM*. Finally, in the **third step,** a value to indicate user preference towards each attribute (in the product domain) is calculated. This value is called the *relevance value* of the attribute to the user. This value indicates 'how relevant is the attribute to the user.' These *relevance values* are used as start-up user preferences in the user model.

In traditional stereotyping, the mapping process is as shown in Figure 5.3. Such domain based stereotypes are built using the expert domain knowledge, and exists as a collection at the start of the mapping process. The expert knowledge determines the preferences of certain people groups, generally based on surveys conducted in the given domain. The people groups are mapped straight to the detailed domain based product information. (e.g. Housewife stereotype prefer watching soap operas). When mapping the user, his/her demographics are mapped to the stereotype to calculate the percentage of stereotypic behavior expected from the user. Based on such expected behavior, user preferences towards domain attributes are calculated.

Compared to traditional stereotypes, the mapping process in general stereotyping are gradual rather than straightforward.

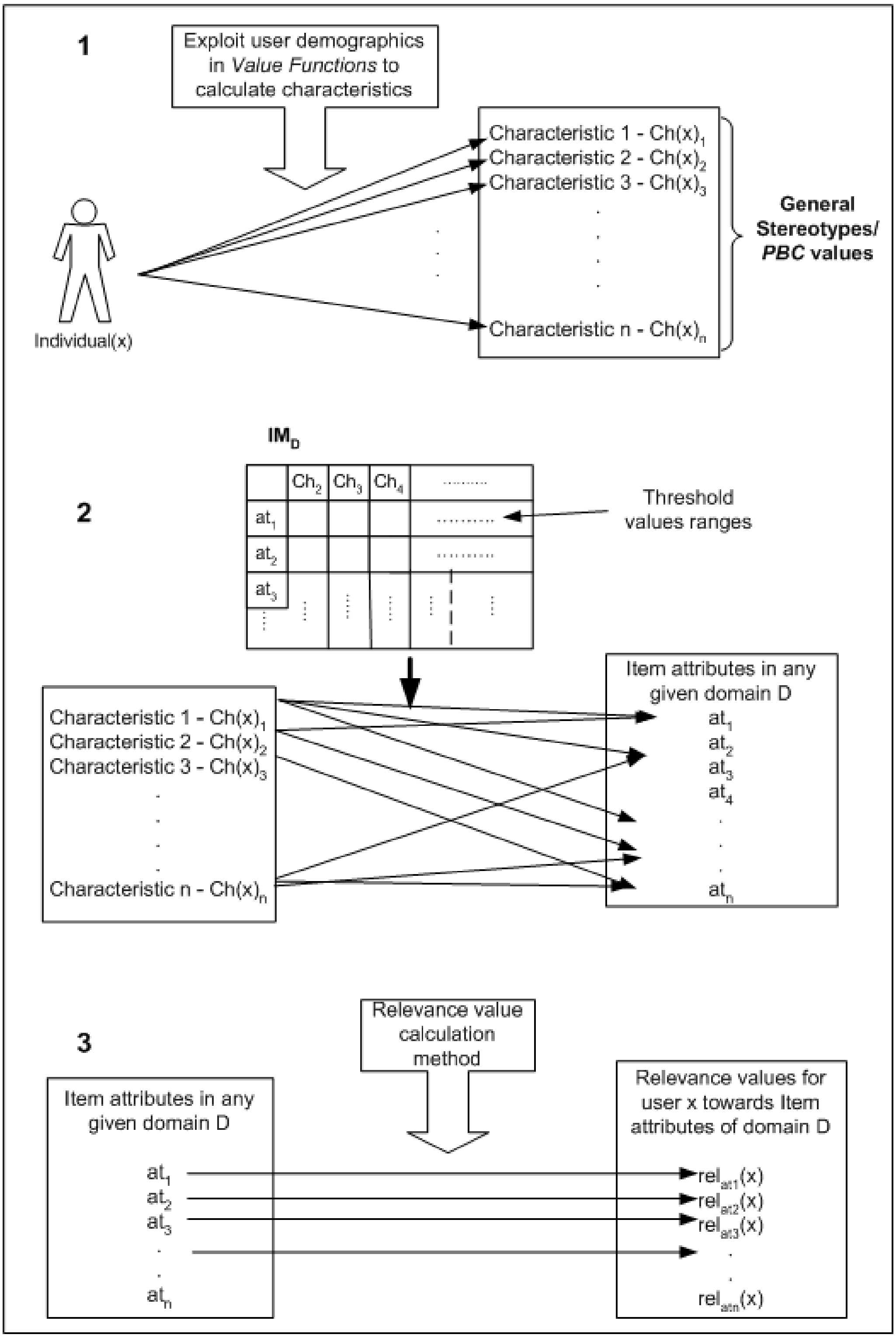

Figure 5.2 : Three step mapping of general stereotypes

Due to the gradual mapping process, general stereotyping is more flexible than traditional stereotyping. The flexibility and gradual mapping allow reuse of demographics obtained from the user in more than single purchasing domains. Since *PBC* values are only based on personal data, once combined with the appropriate domain knowledge they are reusable in any product domain; whereas in traditional approaches the domain information is built into the stereotype at the formation of the stereotypes. Even in automated stereotyping described in Paliouras *et. al.* (1999), the personal information and domain specific information is considered together in the clustering process.

However, at the end of the processes, both approaches calculate predictions for an individual, as percentages of preferences towards personal information related domain attributes.

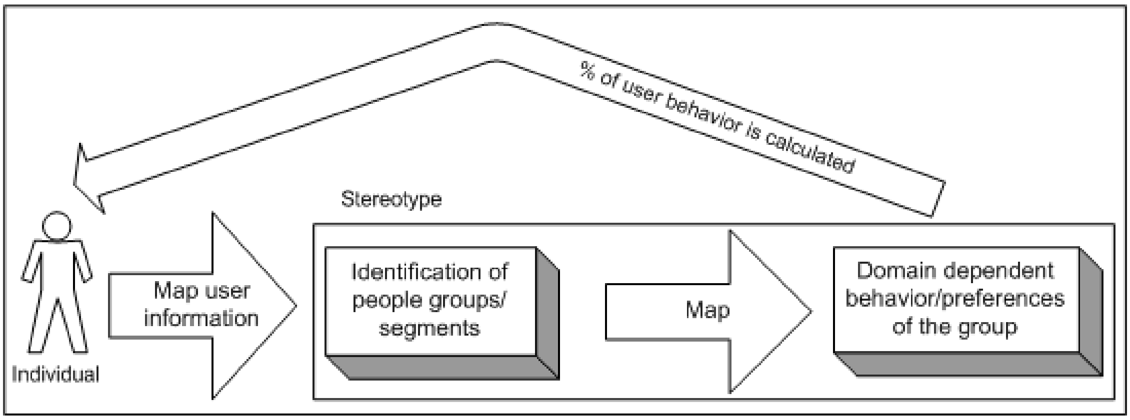

Figure 5.3 : Steps involved in traditional stereotyping

In both traditional and the novel method, expert domain knowledge is required, but for different purposes. Traditional stereotyping uses expert domain knowledge when creating the set of stereotypes (to determine threshold values as *triggers* which activate the stereotype when mapped to user information). This process is carried out separately for each domain, considering the raw demographics that may affect the current domain attributes. The novel approach use expert domain knowledge to determine the threshold values at the time of creating the *IM* for the given domain. Since the demographics are already converted to *PBC* values, the mapping is simplified.

In the novel approach, system knowledge about the stereotypes is easy to manage. For example without affecting the existing framework, new *PBC* s or *IM* s for new domains can be introduced. Furthermore, the update processes described in Chapter 4, section 4.7.2, update the *PBC* values in long-term. This result in rectification of the initial stereotypes generated for individuals. Hence, if a user interacts in a new product domain in the future, (after updating the initial *PBC* values) then more accurate stereotypes will be used for that domain. The *IM* and the relevance value calculation are described in the following sections, further comparing the two stereotyping approaches.

5.4 The Influence Matrix (IM)

IM is domain based. It is a matrix consisting of the threshold values for *PBC* values which influences each of the domain attributes. At the time of creation, it requires expert knowledge about the domain in question, regarding its attributes and expected user preferences towards them. A domain expert needs to identify the connection between *PBC* values and product attributes in the domain. This is demonstrated in the following diagram (Figure 5.4).

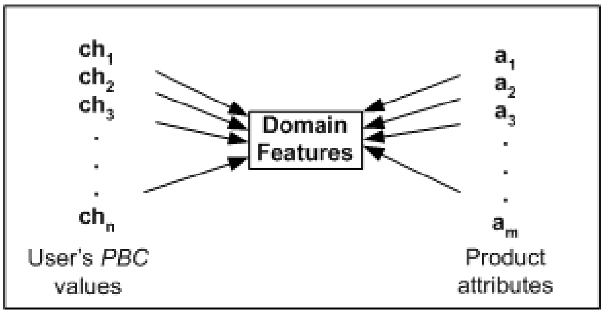

Figure 5.4 : User needs are fulfilled by product attributes

Then the expert has to determine, which of the general-stereotypes (out of the existing collection) influences each of these attributes. In other words, the expert should have the knowledge about general trends within the user population, such as which user *PBC* values are fulfilled with what domain features. This process can be described as a similar but simpler version of traditional stereotype creation process.

Each *IM* is built to deliver such information at the time of forming the *DI* layer of the user model. With regard to the relevance, there are two types of attributes; the attributes that are dependent on the personal information of users and the ones that are independent. The attributes which can be related to personal information, are influenced by the *PBC* values. As previously mentioned in Chapter 4, such product attributes and features which are related to personal information are referred to as Personal Information Related attributes (PIR-attributes) and Personal Information Related features (PIR-features). At the point of

DI layer creation, only PIR-attributes that appear in the *IM* receives a relevance value. The rest of the attributes which are not identified as PIR-attributes, are considered as *irrelevant* at the start and hence receives a zero relevance value. Based on user preferences in consequent interactions the relevance values of all the attributes including the initially irrelevant ones are updated. For example, if the user requests an attribute which is initially considered as irrelevant, then the relevance value of that attribute increases. The *IM* is formally represented below.

Each domain d_i has an *influence matrix* IM_{d_i}, where at_k is a PIR-attribute and $\forall k \in [1,2,..,n]$, where n is the total number of attributes. The columns of the *IM* represent each *PBC* value $Char_l$, and the corresponding weight of the l^{th} *PBC* value towards a given attribute (at_k) where $\forall l \in [1,2,..,m]$ and m is the total number of *PBC* values. The rows of the *IM* represent attributes. As such, *IM* values im_{at_k,ch_l} and w_{at_k,ch_l} represent the influence range and the influence weight of the *PBC* $Char_l$ respectively, towards the attribute at_k. Each im_{at_k,ch_l} is a range with a lower bound lb_{at_k,ch_l} and an upper bound ub_{at_k,ch_l} within which the attribute at_k is influenced by $Char_l$.

The IM_{d_i} is now used to populate the *DI* layer of the user *x*. $Char_l(x)$ (or *PBC*) values for *x* are mapped on to IM_{d_i} to produce relevance values $rel_{at_k}(x)$, for each attribute at_k, and stored in the *DI* layer.

Sections of the $IM_{\text{restaurant}}$ (for the restaurants domain) and $IM_{\text{leg-wear}}$ (for the leg-wear domain) are shown in tables 5.8 and 5.9 respectively. The complete *IM*s for the two domains are attached as Appendix A and B of the thesis. For the convenience in implementation, *IM*'s are stored as database tables. In addition to the threshold value ranges, *IM* store other information such as feature number, attribute number, attribute name, and attribute type. Attribute type can be either 'd' (discrete) or 'c' (continuous).

As shown in Table 5.8, 'cost' is a continuous attribute with four cost ranges; 'below $15', '$15-$30', '$30-$50', and '$50and more'. The Feature Number of 'cost is 2 and it is a continuous feature. Price-sensitivity ($Char_{PS}$) influences the attributes belonging to 'Cost' feature. Since only one *PBC* is influencing the attribute, the 'weight' is 1. Similarly, combined effect of the *PBC* values, $Char_{QC}$ and $Char_{SO}$ influences the Décor feature, where contributing weights of both *PBC*s are 0.5.

Table 5.8 : A section of the Influence Matrix for Restaurants domain

Feature Number	Feature Name	Attribute Name	Attri Type	Price Sensitivity (PS)	PS weight	Quality Consciousness (Qc)	Qc Weight	Health Consciousness (Hc)	Hc Weight	Family Person (Fp)	Fp Weight	Socialising (So)	So Weight
2	Cost	below $15	c	(.75,1)	1	0	0	0	0	0	0	0	0
2	Cost	$15-$30	c	(.5,.75)	1	0	0	0	0	0	0	0	0
2	Cost	$30-$50	c	(.25,.5)	1	0	0	0	0	0	0	0	0
2	Cost	over $50	c	(0,.25)	1	0	0	0	0	0	0	0	0
4	Décor	Fair Decor	c	0	0	(.15,.35)	0.5	0	0	0	0	(.15,.35)	0.5
4	Décor	Good Decor	c	0	0	(.35,.55)	0.5	0	0	0	0	(.35,55)	0.5
4	Décor	Excellent Decor	c	0	0	(.55,.75)	0.5	0	0	0	0	(.55,.7)	0.5
22	Liquor	No Liquor served	d	0	0	0	0	(.65,1)	0.5	(.65, 1)	.5	0	0

If an attribute is not a PIR attribute then such attributes are not influenced by any of the *PBC* values. For example, an individual's preference towards a particular 'cuisine' is totally independent of any of the *PBC* s. Hence, the attributes under feature' cuisine' receives zero relevance for all users. Table 5.10 shows a section of the $IM_{restaurant}$, where none of the 'cuisine' attributes are influenced by the *PBC* values.

Table 5.9 : Section of the Influence Matrix for leg-wear domain

Feature Number	Feature Name	Attribute Name	Attribute Type	Price Sensitivity (Ps)	psWeight	Fun Spending (Fs)	fsWeight	Socialising (So)	soWeight
1	Unit Sale Price	Low Price	c	(.85,1)	1	0	0	0	0
1	Unit Sale Price	Low-Medium Price	c	(.7,.85)	1	0	0	0	0
1	Unit Sale Price	Medium Price	c	(.55,.7)	1	0	0	0	0
1	Unit Sale Price	MedHigh price	c	(.35,.55)	1	0	0	0	0
1	Unit Sale Price	High Price	c	(0,.35)	1	0	0	0	0
4	Basic or Fashion	Basic	c	0	0	(0,.65)	0.5	(0,.65)	0.5
4	Basic or Fashion	Fashion	c	0	0	(.65,1)	0.5	(.65,1)	0.5

Since user preferences are not clear-cut, when calculating the relevance values, a fuzzy approach is taken. According to the Table 5.8 if the user's $Char_{PS}$ value is between (.75,1), then the attribute ‘below $15’ becomes relevant to that individual (attribute “below $15” is a continuous attribute). In addition, price range ‘$15-$30’also becomes relevant due to use of fuzzy membership functions. Calculation of attribute relevance values are shown in the next section.

Table 5.10 : A section of the $IM_{restaurant}$ - non-PIR-attributes

Feature Number	Attribute Number	Attribute Name	Attribute Type	TimeSaver	tsWeight	PriceSensitivity	psWeight	QualityConscious	qcWeight	FunSpending	fsWeight	HealthConsciouss	hcWeight	FamilyPerson	fpWeight	Socializing	soWeight	Adventurer	adWeight
1	96	French	d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	98	German	d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	103	Greek	d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	105	Guatemalan	d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	116	Hungarian	d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	117	Indian	d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	118	Indonesian	d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	120	Irish	d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	121	Italian (North)	d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.5 The Domain Based Initial Preferences

As described in the algorithm in section 5.1, at the end of the step 2, each user will have the *PI* layer of the user model created and the *PBC* values calculated based on the individual’s demographics. When the user exhibits interest in a product domain, the next step is to calculate product feature preferences in the domain of interest combining the *PBC* values and the influence matrix. Since *PBC* values are different from user to user, the relevance of a domain attribute to a user varies from one another. The initial relevance values are assigned either using existing information or calculation methods.

5.5.1 Assigning Existing Relevance Value

As mentioned previously, the *DI* layer consists of multiple (*n*) domain layers. If a particular user has previously interacted with the system, then it is possible that such a user has other *DI* layers apart from the newly created one. In such a situation, with the assistance of the ontology of the domain hierarchy (which is not implemented with our current prototype system) it is possible to trace the availability of super-domains and sub-domains of the current domain. If they exist, then existence of common attributes among the domains is possible (see Definition 4 in section 4.3.2). If relevance values already exist for such attributes, then those values are usable across domains. Such values could be updated values resulted from the actual user behavior in the corresponding domain. Therefore, in a mature user model, existing relevance values are more accurate than the *PBC* based calculated values. In addition to accuracy, this method allows reuse of existing user information across domains, without re-calculating. The calculating approach is explained in the next section.

5.5.2 Assigning Calculated Relevance Values

As described above, the initial relevance values are inherited from the attributes in parent domains if they already exist. If only few of the attributes belonging to the new domain are available then the rest of the values need to be calculated. Relevance values are initially calculated based on user *PBC* values and influence thresholds for each attribute. When more than one *PBC* value is influencing an attribute (see Figure 5.5), influence contribution from each *PBC* is calculated using MAUT. The influencing *PBC* and their contributing weights are obtained from the *IM*. If a single *PBC* is involved then the weight becomes 1. Otherwise the contributing factor becomes the weight. For example, in the Table 5.9, the weight of the single *PBC* $Char_{PS}$ contributing to attributes of feature 1 is '1'. For feature 4, the two contributing *PBC*s ($Char_{FS,}$ $Char_{SO}$) have '0.5' weights.

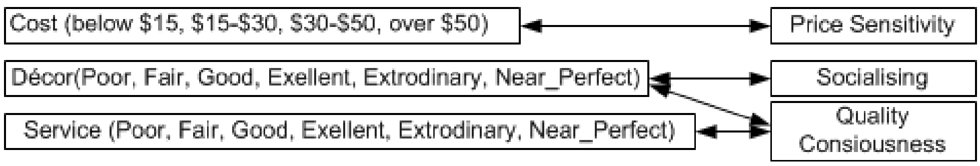

Figure 5.5 : More than one PBC is influencing an attribute

The product features are of two types; continuous or discrete (see column 4, Tables 5.8, 5.9, and 5.10). A feature with a set of continuous attributes is a continuous feature. For example, some of the attributes are inter-related or exist as value ranges under the same feature (Feature 'cost', has four attributes 'below $15', '$15-$30', '$30-$50', and '$50and more'). Cost, Décor, and Service are examples for continuous features in the restaurants domain.

There are attributes that can be grouped under the same feature but yet not continuous. For example, 'liquor preferences' for a restaurant is described by the attributes 'No Liquor Served', 'BYO Liquor', and 'Many Vine Varieties' attributes. Such features as 'Liquor preferences', have mutually exclusive attributes and hence called a discrete feature. Table 5.11 lists out four such features in the restaurants domain along with their features.

The initial relevance values for the continuous attributes, and discrete attributes were calculated using two different approaches. The next section provides the methods employed to calculate relevance values.

Table 5.11 : Discrete Features and their Attributes

Feature	Attribute Names
Cuisine	Have 80 cuisine types. As Asian, European, American, African etc.
Liquor	Fabulous Wine Lists, No Liquor Served, Carry in Wine and Beer, Wine and Beer
Delivery Available	One attribute as 'Delivery Available'
Popularity	Little Known But Well Liked, People Keep Coming Back, Up and Coming

5.6 Algorithm for Calculating Initial Relevance Values

Users select an item (or product) by looking at their attributes. The data we use to describe each item using a large set of binary valued attributes, describing the presence or absence of the attribute in the given item. For example, an item with price \$20 receives '1' for the correct range and '0' for other ranges of the continuous feature (see Table 5.12).

Table 5.12 : Attribute values for an item priced \$20

Continuous attribute ranges	Present or Absent
Low (less than \$15)	0
Medium (\$15 - \$30)	1
Med-High (\$30 - \$50)	0
High (more than \$50)	0

As described above, relevance of an attribute to a user is calculated depending on the *PBC* values. One way of doing this is, to use binary values where 1 or 0 represent "relevance" or "irrelevance" of each attribute. For example, for a person with medium $Char_{PS}$ value (say 0.67) gets relevance 1 for the second attribute and 0 for the other attributes (for the attributes shown in Table 5.12). To capture the fuzziness of user preferences, a fuzzy logic based approach is employed instead of crisp binary values. The next section provides a brief description of fuzzy logic and fuzzy sets.

5.6.1 Fuzzy Logic

Zadeh (1965) suggested a modified set theory in which an individual could have a degree of membership which ranged over continuum of values rather than being either '1' (true) or '0' (false). Instead of crisp boundaries impose by conventional binary logic it uses fuzzy membership functions to assign a value of vagueness, covering the real numbers in the interval [0,1] (Yan *et. al.*, 1994). Therefore, if there is a set of propositions then the *degree of truth*, assigned to each of them may be "absolutely true," "absolutely false" or some *intermediate* truth degree: a proposition may be more true than another proposition (Hajek, 2002).

In Wikipedia definition of a fuzzy set is given as follows.

A fuzzy set is a pair (A,m) where A is a set and m: $A \rightarrow [0,1]$. For each $x \in A$, $m(x)$ is the grade of membership of x. $x \in (A,m) \Leftrightarrow x \in A \wedge m(x) \neq 0$. If $A=\{x_1, \ldots, x_n\}$ the fuzzy set (A,m) can be denoted $\{m(z_1)/z_1, \ldots., m(z_n)/z_n\}$. Therefore, an element mapping to the value 0 means that the member is not included in the fuzzy set, 1 describes a fully included member. Values strictly between 0 and 1 characterize the fuzzy members.

The next section presents the relevance value calculations using Fuzzy logic, which has the ability to capture the 'fuzzy' nature of the user preferences.

5.6.2 Use of Fuzzy Logic in Relevance Value Calculation

To represent the fuzzy sets, it is required to define membership functions. Depending on the fuzzy set, membership function take different shapes such as S-function, $\pi-$function or T (triangular/trapezoid) form (Yan *et. al.*, 1994). According to Negnevitsky (2005) a triangular or trapezoid shape can often provide an adequate representation of the expert knowledge, and at the same time significantly simplifies the process of computations. Yan et. al. (1994) too recommends T-membership functions as suitable for representing properties with non-zero membership. Therefore, to represent relevance values towards attributes, a membership function of type-T has been chosen. The T-function is defined as follows (Figure 5.6).

$$T(u:a,b,c,d)=\begin{cases}0 & u<a\\ (u-a)/(b-a) & a\le u\le b\\ 1 & b\le u\le c\\ (d-u)/(d-c) & c\le u\le d\\ 0 & u>d\end{cases}$$

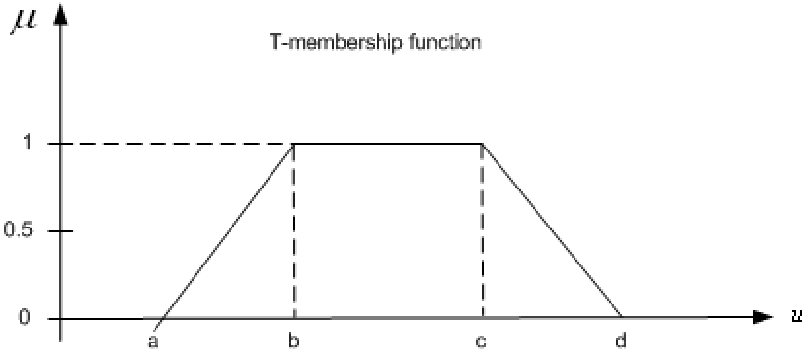

Figure 5.6 : T-membership function corresponding to above function definition

As described previously, each *PBC* value (of an individual) can influence one or more attributes. In other words, attributes become relevant to the user based on his/her *PBC* values. The algorithm to calculate the relevance of an attribute (to a user) is determined by the type of the attribute; discrete or continuous. The algorithm for selecting the appropriate calculation method is shown below.

```
loop: for all Ch_l
    loop: for all at_k   //all attributes influence Ch_l
        read( lb_{at_k,ch_l} , ub_{at_k,ch_l} );//read the upper and lower bound thresholds from the IM
        if ( true)
            If (attri_type(at_k) ==’d’)  //discrete
                calc_rel_discrete(Ch_l, lb_{at_k,ch_l} , ub_{at_k,ch_l} );
            else      // continuous
                calc_rel_continuous(Ch_l, lb_{at_k,ch_l} , ub_{at_k,ch_l} );
            end-if;
        end –if;
    end-for;
end-for;
```

The above algorithm, reads the attribute type of each of the attributes relevant to a given *PBC* from the *IM*. Depending on the attribute type, the appropriate calculation algorithm is used to *PBC* recalculation. The calculation algorithms are discussed next.

Continuous Attributes

When the attributes are ranges it is difficult and unfair to allocate a single range as relevant to the user. Therefore, fuzzy membership functions are used to represent such ranges. As shown in Table 5.13 feature "Cost" has four attributes and hence represented by four membership functions.

Table 5.13 : Ranges and equations of each membership function –"Cost"

Price Sensitivity Value	Range	Equation
high	0<x<=0.25	y=1
	0.25<x<=0.5	y+4x=2
med-high	0<=x<=0.25	4x-y=0
	0.25<x<=0.5	y=1
	0.5<x<=0.75	y+4x=3
medium	0.25<=x<=0.5	4x-y=1
	0.5<=x<=0.75	y=1
	0.75<=x<=1	y+4x=4
low	0.5<=x<=0.75	4x-y=2
	0.75<=x<=1	y=1

Once the $Char_{PS}$ value of an individual is known, the price sensitivity range is acquired. Then by substituting the $Char_{PS}$ value in appropriate equation the relevance of the attribute is calculated. For example, if the $Char_{PS}$ value is 0.67, it belongs to all three ranges darkened in Table 5.13. Three of the attributes become relevant as shown in Table 5.14.

Table 5.14 : Crisp and Fuzzy representations of relevance values

Attribute Name	Crisp Relevance	Fuzzy Relevance
Low (less than $15)	0	0.68
Medium ($15 - $30)	1	1
Med-High ($30 - $50)	0	0.32
High (more than $50)	0	0

Figure 5.7 depicts the relevance value calculation by showing the $Char_{PS}$ value (0.67) intersecting the membership functions Medium ($15 - $30) at 0.32, and Low (less than $15) at 0.68 and finally Medium ($15 - $30) at 1: indicating Medium ($15 - $30) as the most relevant cost range for him/her.

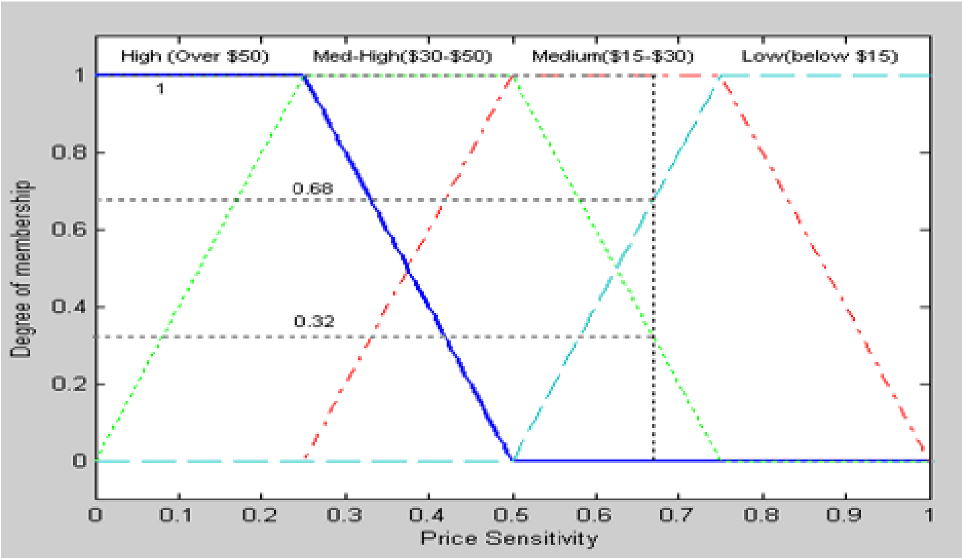

Figure 5.7 : Four membership functions for cost attributes based on Price Sensitivity

Depending on the number of attributes associated with the feature, fuzzy membership functions are changed. In table 5.15, membership functions for feature Décor which has six attributes are shown. Relevance of décor depends on the *PBC Char*$_{QC}$. Figure 5.8 shows the membership functions for a feature consists if six attributes.

Table 5.15 : Ranges and equations for membership functions “Décor”

Quality Consciousness Value	Range	Equations
poor	0<=x<=0.15	y=1
	0.15<x<=0.35	4y+20x=7
fair	0<=x<=0.15	20x-3y=0
	0.15<x<=0.35	y=1
	0.35<x<=0.55	4y+20x=11
good	0.15<x<=0.35	20x-4y=3
	0.35<x<=0.55	y=1
	0.55<x<=0.7	3y+20x=14
excellent	0.35<x<=0.55	20x-4y=7
	0.55<x<=0.7	y=1
	0.7<x<=0.85	3y+20x=17
extraordinary	0.55<x<=0.7	20x-3y=11
	0.7<x<=0.85	y=1
	0.85<x<=1	3y+20x=20
near-perfect	0.7<x<=0.85	20x-3y=14
	0.85<x<=1	y=1

If the number of attributes is equal, and the range of the attribute values are the same, then such features can use the same membership functions in relevance calculations. For example, in the restaurants domain, feature 'Service' or in the leg-wear domain feature "Item Cost" both are associated with five attributes.

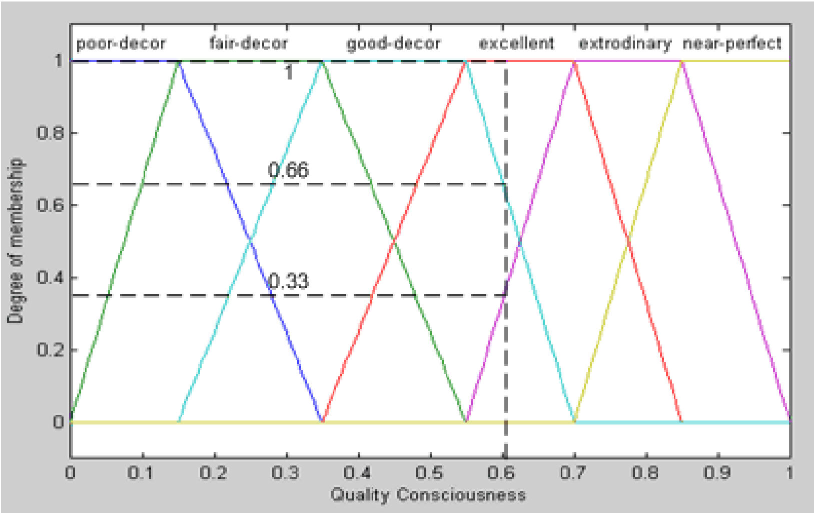

Figure 5.8 : Six relevance fuzzy sets for Quality Consciousness

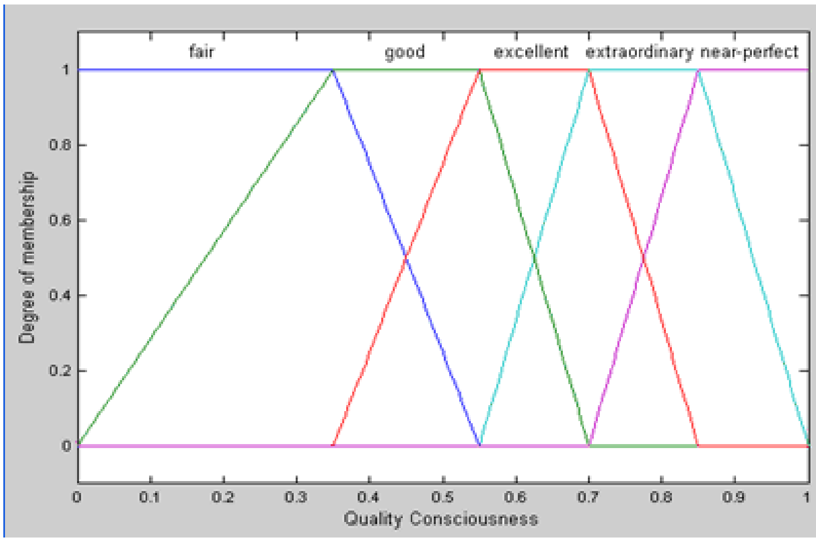

Figure 5.9 : Five relevance fuzzy sets for Quality Consciousness for "Service"

Features "Item Cost" in leg-wear domain and "Service" in restaurants domain, both share the same fuzzy sets as shown in Table 5.16. "Item Cost" is influenced by the price sensitivity of an individual and the "Service" is influenced by the quality consciousness. However, the influence ranges of the attributes are similar as shown in Table 5.16. Therefore, both features can be represented using the same set of membership functions. Figure 5.9 shows the membership functions for a feature consists of five attributes (in this example low, low-medium, medium, med-high, and high or fair, good, excellent, extraordinary, and near-perfect).

Table 5.16 : Range of each membership function/attribute

Price Sensitivity	Quality Consciousness Value	Range	Equations
Low	fair	0<x<=0.35	y=1
		0.35<x<=0.55	4y+20x=11
Low-Medium	good	0<x<=0.35	7y=20
		0.35<x<=0.55	y=1
		0.55<x<=0.7	3y+20x=14
Medium	excellent	0.35<x<=0.55	20x-4y=7
		0.55<x<=0.7	y=1
		0.7<x<=0.85	3y+20x=17
Med_High	extraordinary	0.55<x<=0.7	20x-3y=11
		0.7<x<=0.85	y=1
		0.85<x<=1	3y+20x=20
High	near-perfect	0.7<x<=0.85	20x-3y=14
		0.85<x<=1	y=1

The algorithm described below (initiated by the algorithm given in section 5.1.9), calculates the relevance of a continuous attribute.

1: *calc_rel_continuous*(Ch_l, lb_{at_k,ch_l}, ub_{at_k,ch_l})
2: *{*
3: *if in_range*(Ch_l, lb_{at_k,ch_l}, ub_{at_k,ch_l})
4: rel_{at_k} = *calc_fuzzy_rel*(Ch_l);
5: *else* rel_{at_k} =0;
6: *end_if;*
7: *}*

If the users *PBC* value Ch_l is in the threshold range of the attribute, then the relevance is calculated using the appropriate equations. If the characteristic is out of range, such attributes become irrelevant (relevance =0 as in line 5).

Discrete Attributes

Figure 5.10 depicts the membership functions related to a discrete variable. In the diagram, the same feature ‘Color’ can either be an ‘exciting color’ or it can be a ‘general color’.

Table 5.17 shows the membership functions with their value ranges and equations. As shown in Figure 5.10, if the user has a $Char_{AD}$ value of 0.3, then the relevance values of attributes exiting-colors and general-colors are 0.462 and 1 respectively.

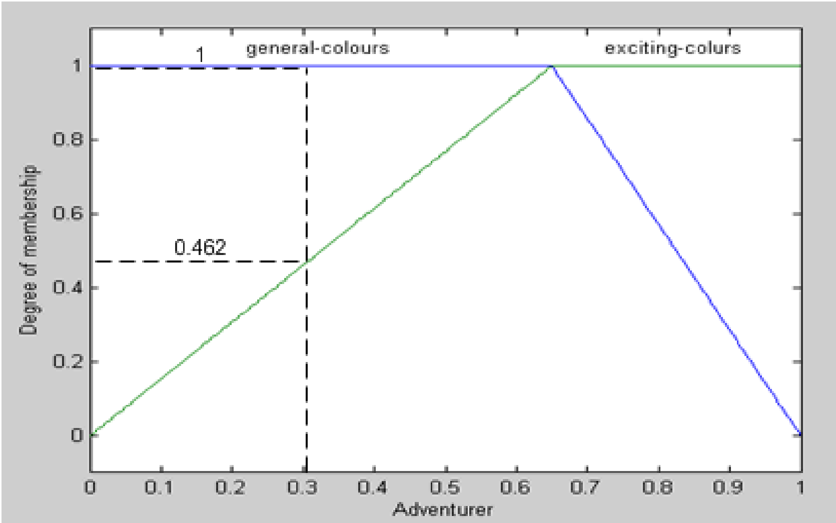

Figure 5.10 : Two relevant fuzzy sets for 'Adventurer' for the feature leg-wear colour

Table 5.17 : Range of each membership function/attribute

Adventurer Value	Range	Equations
Low	0<=x<=0.65	65y =100x
High	0.65<x<=1	35y+100x = 100

However, a discrete attribute cannot be treated as a two valued continuous attribute, since they are mutually exclusive. For example, the attribute 'No liquor served' do not imply that the other restaurants definitely serve liquor. It is just that there are certain restaurants that definitely **do not** serve liquor.

Initial relevance value calculation for two discrete attributes over the same *PBC* is shown in the Figure 5.11. As shown, at_1 is negatively influenced by $Char_{HC}$, while at_2 is positively influenced.

Similar to a continuous attribute, in a situation where more than one *PBC* is influencing, the relevance of a discrete attribute ($Rel_{DiscreteAtribute}$) is calculated as follows.

$$Rel_{DiscreteAtribute} = \Sigma w_i Char_i$$

Where $Char_i$ is the value of the i[th] contributing *PBC* and w_i is the corresponding weight of the PBC values given in the *IM*.

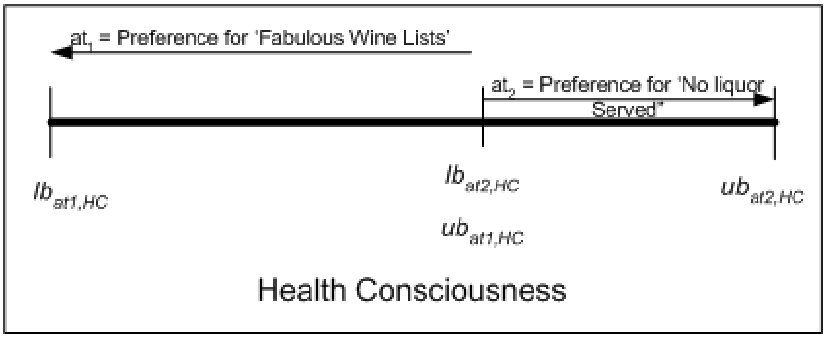

Figure 5.11 : Influence of a PBC over a discrete attribute

In the above example, consider the attribute "No Liquor Served". In Table 5.8 it is shown that both Health Consciousness (HC) and Family Person (FP) *PBC*s contributes to this attribute. Both *PBC*s have the contributing weight 0.5. If the individual has a relevance of 0.67, based on HC value and 0.8 based on his/her FP value the relevance of the attribute is (0.67x0.5 + 0.8x0.5 = 0.735). Therefore, the relevance becomes the weighted sum of the two *PBC*s.

5.7 Average Total Relevance (ATR) Values of Features

The importance of an attribute to a user in a given domain is shown by its relevance value in the corresponding *DI* layer. According to the *Definition 2* in Chapter 4, section 4.3.2, each feature is connected to several attributes. Therefore, the relevance or the importance of a feature depends on the total relevance of the attributes belonging to that feature. May be different attribute values were preferred, but still the total shows the importance of the feature. Since each feature can have different number of attributes, the relevance of a feature is captured using the average relevance of the attributes belonging to the feature. The higher relevance value indicates importance of the feature to the respective user.

The average total relevance of a feature is calculated as follows.

The Average Total Relevance $ATR_{d_i f_j}$ for the j^{th} feature f_j of i^{th} domain d_i is given by:

$$ATR_{d_i f_j} = \frac{\sum_{k=1}^{n} \mathrm{Re}\, l_{at_k}}{n}$$

Here $\mathrm{Re}\, l_{at_k}$ is the relevance value of the k^{th} attribute and $at_k \in d_i f_j$; n is the total number of attributes.

5.8 Updates in the User Model

According to the algorithm steps 6 and 7 in section 5.1, and as explained in Chapter 4, the user model gets updated in two different stages: (i) update the current relevance value of attributes according to new transactions and (ii) in the long-term, to update the values of *PBC* s in the PI layer. In both occasions, a slow update process is desired. Hebbian learning (Hebb, 1949) is a commonly utilized learning methodology in self-organizing neural networks. It is an unsupervised learning method which is suitable to be employed in unexpected and dynamic environments. Hebb's Law provides the basis for learning without a teacher (Negnevitsky, 2005). Learning here is a local phenomenon occurring without feedback from the environment. Therefore, Hebbian learning is suitable to apply in both (above mentioned) update processes.

5.8.1 Hebbian Learning

Donald Hebb (Canadian Psychologist) speculated in 1949 that "When a neuron A repeatedly and persistently take part in exciting neuron B , the synaptic connection between A and B will be strengthened." Therefore, neuron B becomes more sensitive to stimuli from neuron A. So a Hebbian network can be used as an associator which will establish the association between two sets of patterns $\{X_i, i=1,\ldots,L\}$ and $\{Y_j, j=1,\ldots,L\}$ (Figure 5.12).

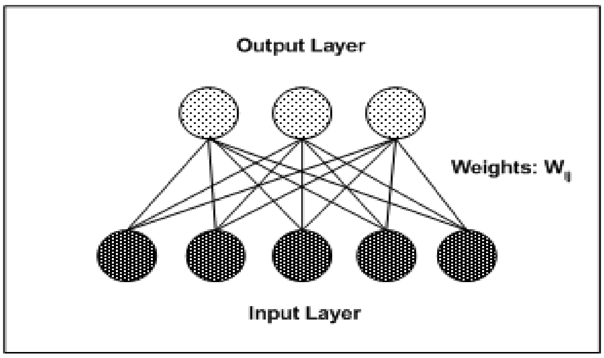

Figure 5.12 : Structure of a Hebbian network

If all output nodes are connected to all input nodes then the strength of a node in the output layer is given by:

$$y_i = \sum_{j=1}^{n} w_{ij} x_j \qquad \{\forall i \mid 1,2,...,m\} \text{ ----------------- } (5.1)$$

As in Negnevitsky (2005) Hebb's Law can be represented in the form of two rules as follows.

- If two neurons on either side of the connection are activated synchronously, then the weight of that connection is increased.
- If two neurons on either side of the connection are activated asynchronously, then the weight of that connection is decreased.

According to the Hebbian learning Law, the new weight after a given iteration is given by,

$$w_{ij}^{new} = w_{ij}^{old} + \eta x_j y_i \qquad (i= 1, 2,..,n; j = 1, ...,m) \text{ ---------------- } (5.2)$$

Here η is the **learning rate parameter**. The positive effect of $\eta x_j y_i$, will drive the new weight (w_{ij}^{new}) into saturation. As exemplified in Negnevitsky (2005), to resolve this problem a non-linear **forgetting factor** is required. The next section explains the application of Hebbian Law and the forgetting factor in updating the user model layers.

5.8.2 Application of Hebbian Learning in Update from *TI* Layer to *DI* Layer

When the user interacts with a certain domain the corresponding *TI* layer component of the current transaction is generated. Such information is exploited in updating the corresponding *DI* layer. In the *TI* layer, allocations of relevance values to product attributes during interactions are carried out as in Table 5.18.

Table 5.18 : Relevance values corresponding to different stages in the retrieval process

Interaction stage	**Possible Relevance Value**
Each attribute in the initial query is considered as of highest relevance	1
Each attribute value explicitly specified by the user during the system interaction is given a slightly low relevance	0.75

In the *TI* layer-to-*DI* layer update process, it is important to change the relevance value of an attribute to indicate its new relevance according to the current user preference. For example, frequent requests for an attribute increases its relevance to the user while not requesting an attribute indicates the unimportance of the attribute to the user. Therefore, as mentioned, Hebbian learning method is suitable for such updating and is employed in relevance calculations as follows.

If features and attributes are considered as output and input nodes respectively, then as shown in Figure 5.13, the feature becomes the single node in the output layer while the attributes belongs to the input layer. Then the *weights* represent the relevance values of the attributes.

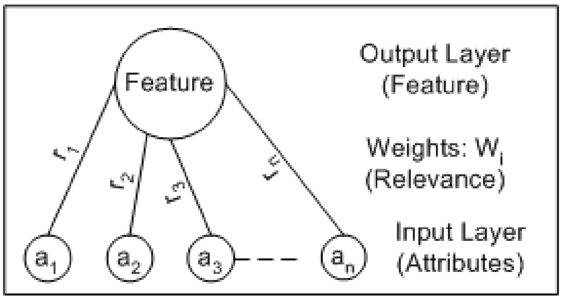

Figure 5.13 : Feature as a single node in the output layer, attributes in the input layer

Therefore, using equation *(5.2)* the new relevance rel^{new} is given by;

$$rel^{new} = rel^{old} + \eta x_j y_i \qquad (i = 1, \ldots n; j = 1, \ldots, m) \text{ ------------- } (5.3)$$

Since a forgetting factor is required, the $\eta x_j y_i$ factor is replaced by $\eta(i\text{-}rel^{old})$, where *i* is the new input value (binary 0/1).

$$rel^{new} = rel^{old} + \eta(i\text{-}rel^{old}) \qquad \text{————— } (5.4)$$

If the input value is 1 then the new relevance value receives a positive update while a '0' input results in a negative update of the relevance value. Furthermore, even if a certain attribute is preferred all the time, still the maximum value reachable for rel^{new} is 1. Figure 5.14, shows an example use of the learning technique.

According to the example in Figure 5.14, the 'Cost' becomes the output while the attributes (cost ranges) represent the input layer. The existing relevance values in the user model, (for each of the attributes) become the weights between the layers. Then, for a given user if feature "cost" repeatedly receives the attribute "less than $15", according to the Hebb's rule, the corresponding relevance value (weight) gets strengthened. The new relevance value is calculated using the equation *(5.4)*.

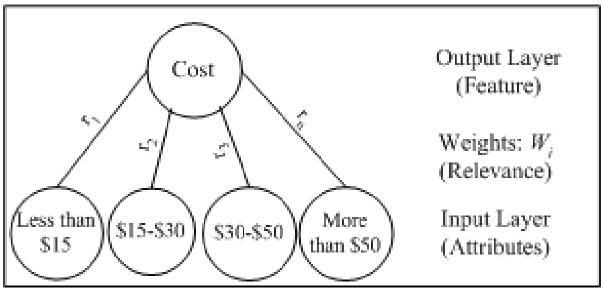

Figure 5.14 : Feature 'Cost' as the output layer and the four attributes as the input layer

5.8.3 Application of Hebbian Learning in Update from DI Layers to PI Layer

As explained before, after each user transaction (within a given domain) the attribute preference values (for that domain) declared during the current transaction update the existing attribute relevance values in the DI layer. As described in section 5.5, these values are initially inherited from a different domain or calculated based on individual's *PBC* values. As a result of updates performed according to the above section, after a number of transactions in a given domain, the initially allocated relevance values may change. For example, an attribute which had a very small initial relevance may have a high current relevance value. If the attribute is initiated using the *PBC* values then, such a difference in the relevance value indicates the unreliability of the initial *PBC* values which was used for predictions. Now the current preference values for the attributes can be employed in a backward calculation method similar to back propagation in Neural Networks (Haykin, 1998), to recalculate the initial data used for predictions. For example, if the user has been interacted with the system for a considerably long period of time in a number of different domains, the update process uses the information from all the *DI* layer's belonging to the user. The initial stereotype based calculations are replaced by the user's freshly calculated actual behavior. Such rectification will result in more reliable start-up values for future transactions especially if the user interacts in a new domain. The process of *PBC* value update is demonstrated in Figure 5.15.

As shown in the Figure 5.15, a single *PBC* value can influence more than one attribute (Ch_A influences both at_1 and at_{11} of domain Ids 01 and 05). After user transactions, several of the initial relevance values are changed (relevance for at_1 changed from r_1 to r'_1). The reasons for this could be any of the following:

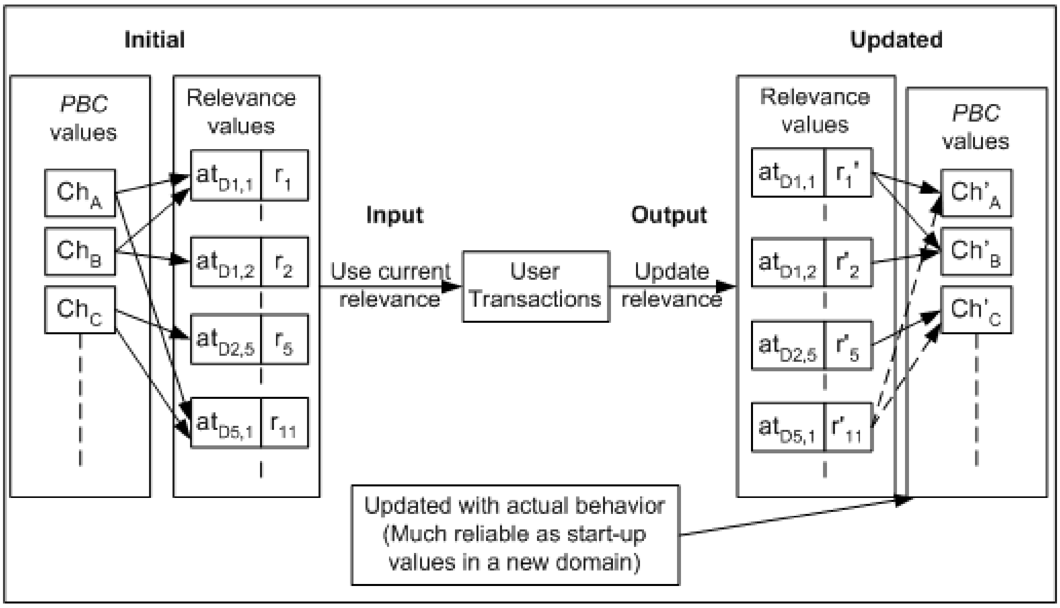

Figure 5.15 : Process of PBC value update

- user exhibits 'grey sheep' behavior.
- user's lifestyle and hence the purchasing behavior has changed from the initial, and
- user behaves as expected only in certain domains, but differently in others (both $at_{D_2,5}$ and $at_{D_5,1}$ are initially based on Ch_c, but the attribute in domain D_2 has changed relevance value while the other attribute still have r_{11} as it's relevance value.

For example, a certain individual may assign a very low price sensitivity *PBC* value ($Char_{PS}$) depending on the registration information. But in the long-term, if he/she seems to sacrifices other qualities of the item to the cheap prices, the calculated $Char_{PS}$ value, needs updating. If not updated, whenever the user seeks personalization in a new domain, the user model is initialized according to the low $Char_{PS}$ value and will result in recommending high priced items at the start-up (in which the user is not interested in).

The technique of updating the *PBC* values needs to consider the attribute type, discrete or continuous. The algorithm employed in recalculation of *PBC* values is given below. Here D_j is the j^{th} domain and at_k is the k^{th} attribute belonging to the domain D_j influenced by the *PBC, Char*$_l$. Where $\forall k \in [0,1,2,..,n]$ and n is the number of attributes in domain D_j and $\forall l \in [1,2,..,p]$ and p is the total number of *PBC* values.

The lines 5 and 7 indicate use of two different methods for updating *PBC*s connected with discrete and continuous attributes. Since the attributes were initialized depending on the type (discrete or continuous), the *PBC* update also require to reverse the same processes. Updating *PBC* values attached to a discrete variable is shown in Figure 5.16.

```
1:  loop: for each D_i
2:      loop:for each Char_l
3:          loop:for each at_k
4:              if (at_k.type ==’d’)
5:                  update_char_dis(at_k num, at_k.oldRel, at_knewRel, at_k.range);
6:              else
7:                  update_char_cont(at_k.num, at_k.oldRel, at_knewRel, at_k.range);
8:              end-if;
9:          end-for;
10:     end-for;
11: end-for;
```

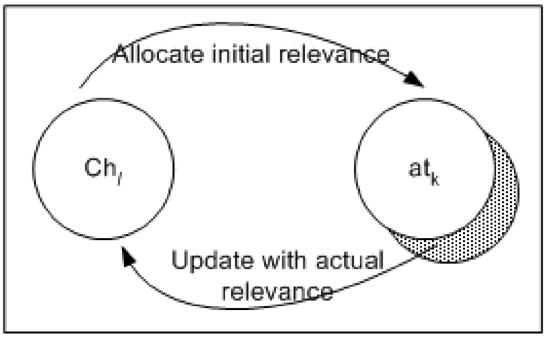

Figure 5.16 : Allocating the initial relevance to individual attributes

The PBC update algorithm for a discrete attribute is as follows.

```
if (at_k.new.rel >0)
        if(at_k.ub == 1)
                if (at_k.old.rel > at_k.new.rel)
                        ch_l.new = ch_l.old +(at_k.old.rel – at_k.new.rel);
                else
                        ch_l.new = ch_l.old - (at_k.old.rel – at_k.new.rel);
                end-if;
        else
                if (at_k.old.rel > at_k.new.rel)
                        ch_l.new = ch_l.old -(at_k.old.rel – at_k.new.rel);
                else
                        ch_l.new = ch_l.old + (at_k.old.rel – at_k.new.rel);
                end-if;
        end-if;
end-if
```

As shown in the section 5.6, for continuous attributes the initial relevance values were calculated using fuzzy membership functions. Therefore, the update process is much complicated. Since the initial calculations were performed considering the attributes as a group under a feature, the attributes cannot be individually considered. The relationship between the attributes and the *PBC* is shown in the Figure 5.17.

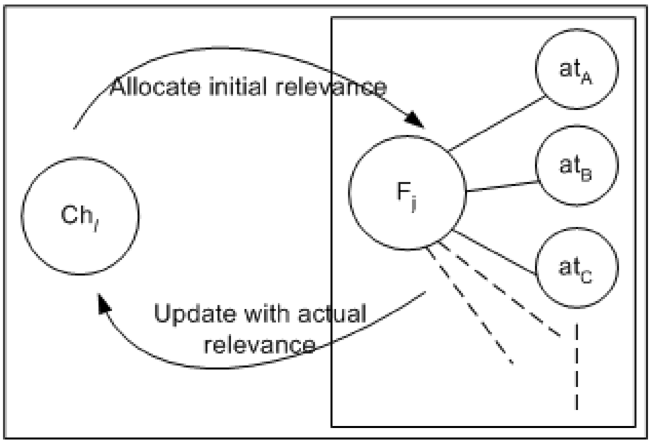

Figure 5.17 : PBC Chl influences the attributes of the feature

For continuous attributes, if a different attribute has become more relevant than the initially allocated highest relevant attribute, this implies the need of rectification to the corresponding *PBC* value. In such situations the mid value of the new range is assigned to the current *PBC* value. Figure 5.18 depicts a feature having three attribute ranges. Depending on the initial *PBC* value ($Char_{old}$), at_k becomes the highest relevant attribute.

Say after several interactions, at_{k+1} become the highest relevant attribute. Therefore, $Char_{new}$ needs to be in the range [a-b[instead of [0-a[.

$$Char_{new} = (c - b)/2 + b$$

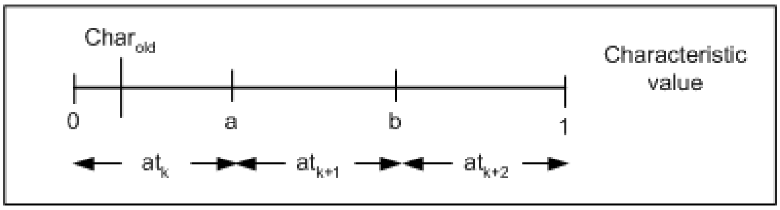

Figure 5.18 : Example calculation of new PBC

The algorithm for updating a *PBC* based on the effect of continuous attributes is as follows. The algorithm is performed for each domain (D_i), for each feature (F_j) in the domain. *At[]* is an array holding the attributes belonging to F_j.

```
1:   loop: for each Char_l
2:        loop: for each D_i
3:             loop: for each F_j
4:                  At[] = get_attributes(F_j);
5:                  rel_old = get_high_rel_Old(At[]);
6:                  rel_new = get_high_rel_New(At[]);
7:                  if (rel_old.range == rel_new.range) //still in the same range
8:                       Char_Fj,new = Charf_old;
9:                  else
10:                      Char_Fj,new = rel_new.lb+(rel_new.ub - rel_new.lb)/2;
11:                 end-if;
12:            end-for;
13:            Char_Di,current = Σw_jChar_Fj,new;
14:       end-for;
15:       Char_Di,current = ΣChar_Di,current/n;
16:  end-for;
```

For each feature $Char_{Fj,new}$ is calculated and combined according to the corresponding weight (w_j). The final total value of $Char_{Fj,new}$ (denoted by $Ch_{DiFj,curretnt}$) is the new *PBC* value for the domain. However, if the user has been interacted in several domains, then the average *PBC* value is calculated (line 15), which becomes the final *PBC* value ($Ch_{curretnt}$).

5.9 Summary

This chapter explained the algorithms and techniques used to implement the architecture described in Chapter 4. The creation of the initial *PI* layer and then creating domain based

DI layers for each user interaction in a new domain is algorithmically explained. How updates are carried out using the transaction based behavior and how such information affects the stereotypic information in *PI* layer was illustrated. Furthermore, this chapter describes the general stereotypes and their role in information reuse and the *IM* and its contribution in *DI* layer formation.

The algorithms presented in this chapter are evaluated in the next chapter. The performances of the algorithms are tested using the available datasets to illustrate and highlight the functionality, value and usability of the novel user model in eCommerce environment.

Chapter 6

Functionality, Value and Usefulness of the Layered User Model

In the previous chapter we discussed the algorithms used in creating and maintaining the *LUM* layers. In addition, the novel *LUM* is capable of handling a range of issues that are required by the current user modeling research. In this chapter we present a comprehensive set of experiments to demonstrate such abilities of the new architecture. The experiments use two datasets to demonstrate functionality, value and the usefulness of the novel model.

The flow of the chapter is as follows. Section 6.1 provides an understanding of the implementation environment. It also discusses the limitations of datasets used in experiments. Section 6.2 demonstrates the functionality and the value of the model using the available data for illustrations. Section 6.4 presents the usefulness of the model. Finally, section 6.5 summarizes the chapter.

6.1 Implementation Environment

In this section implementation environment with regard to the platform, tools and datasets is discussed. Preparation of data and the experiment set up is discussed next.

6.1.1 Platform, Tools and Data Sets

As mentioned before, the eHermes PERSONAL is a part of a larger project which is implemented in a platform independent environment. Therefore, eHermes PERSONAL was implemented in the platform independent .Net environment. The implementation of the prototype system was carried out as an off-line window based project. For programming, about 5000 lines of C# code were used. Data was stored in a SQL Server database which smoothly interacts with .Net and C# environments. SQL Server stored procedures were used to link the data with coding.

There were three datasets used in the implementation of the algorithms. The restaurant dataset Entrée (Asuncion and Newman, 2007), a dataset about leg-wear obtained from KDD-CUP 2000 (Kohavi *et. al.*, 2000) and a mock-up dataset about recipes.

The **Entrée dataset** contains descriptive information about restaurants in a number of cities in the USA. The dataset was intended to use as a knowledge base of a recommender system called Entrée (Burke, 2002a) . Data is available separately for each city and log flies are available indicating user browsing sessions. But the data do not facilitate identification of users. Therefore even though the transaction data are available sessions cannot be identified for the returning user. Each restaurant is described under 256 attributes indicating the presence or absence of each attribute.

In our prototype implementation we modified the data according to our requirements. For example, the 256 attributes were grouped into 31 features (as 'cost', 'décor', etc) to form questions during the search (according to Chapter 4, section 4.3.2, *Definition 3*). In the restaurant domain the 'cost' is a feature. For a given item the feature 'cost' can take one of the four attributes "less than \$15", "\$15-\$30", "\$30-\$50" and "more than \$50". A sample of the product categorization is shown in Figure 6.1. As shown, the feature 'cuisine' consists of sub-features and each sub-feature consists of attributes.

The Entrée dataset was selected due to several reasons.

(i) the data is about restaurants, which is a simple application domain for analysis,
(ii) the data has detailed product descriptions,
(iii) the data had already been tested with a knowledge based recommender system, and
(iv) the descriptive data is suitable for our retrieval strategies with minimum alterations.

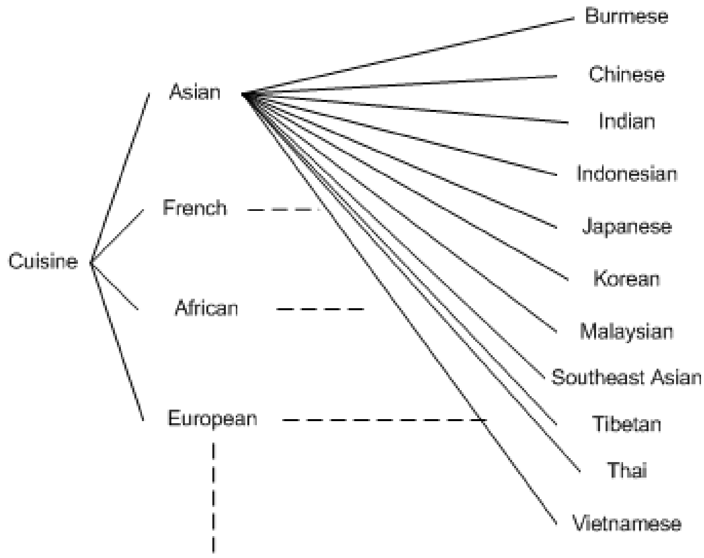

Figure 6.1: Hierarchically organized product attributes

Since the different city locations are not considered in our implementation we combined the data of few cities to form a dataset consist of 1876 restaurants. The size of the database is 51.69 MB.

The leg-wear dataset has demographics of 3466 customers and their actual purchasing transactions. A subset of 1510 (complete and meaningful) transactions out of 3466 transactions was selected to demonstrate our model. The selected transactions are related to 705 different products. To be used with the prototype system, the products data was modified to describe the products as in the Entree dataset. Each product was described under 121 attributes which belongs to 13 features. The importance of the data set is that it consists of both demographics of the users and their transactions. Since the layers of the user model uses demographics, domain based and transaction based user information, it is extremely important to have the demographics connected to the user interaction sessions to be used in experiments. Unfortunately, the footwear dataset is not descriptive enough for demonstrating the product retrieval algorithm. Although there were number of product varieties, the number of products belonging to each variety was often too low. With only few interactions the final product could be retrieved. Therefore, the Entrée dataset was better suited for demonstration in Chapter 7; unlike the footwear there is a large collection of descriptive restaurants. The size of the footwear database is 11.13 MB.

Finally, the recipe dataset consist of 31 recipes. This mock-up dataset was prepared at the early stages of the project. The recipes are described using 62 attributes which belongs to 10 features. Compared to the other datasets this is small in all aspects where the size is only 3.31 MB. This dataset is important especially for future work, when dealing with the domain hierarchies, since this is the closest to the restaurants domain and related to people's food preferences. Although used for early testing purposes, this data set is not presented in any of the demonstrations within this thesis.

6.1.2 Experiment Preparation

The two datasets on restaurants and leg-wear were used in the experiments. As mentioned previously, leg-wear domain provide both demographics and transaction data while restaurants provide isolated sessions of transactions which cannot be linked to any user. For the experiments presented in chapters 6, 7 and 8, the datasets were manipulated as follows.

Leg-wear

The leg-wear dataset contains demographics and transaction history for several users. Using their demographics as registration information the *PI* layers were generated. The testing requires comparison of user purchase behavior in more than one domain. Therefore the same users who interacted in leg-wear domain were selected for the restaurant domain. Combining the *PI* layer and the corresponding domain *IM*, initial *DI* layers were created for the two product domains. Since each user has a transaction history linked to the user-id, the purchased goods are identifiable. Therefore, the attributes presented in each purchased good was considered as a preference explicitly requested by the user. Based on such an assumption, the *DI* layer was updated.

Restaurants

Using (above created) PI layers of the users and the $IM_{restaurants}$, initial *DI* layers for the restaurant domain were created. Since there are no actual users available, we employed a scenario based evaluation for the personalized interactive product retrieval process (*PIPRP* which is explained in Chapter 7). For example User41 was tested for four scenarios (which resulted in four transactions) in selecting restaurants, starting with the initial *DI* layer for restaurants.

User41 already had 15 transactions in the leg-wear domain. The two tables below (Table 6.1 and Table 6.2) provide the initial relevance values (towards sample of the products attributes) and the relevance values at the beginning and the end of the transactions. Most of the experiments, provided in this chapter are based on the data belonging to User41. The user Id's used in the experiments is original from the dataset.

6.2 Functionality of the Layered User Model

In Chapter 4, functionality of the user model was discussed in detail describing each layer, information sharing between the layers and the updates. In Chapter 5, algorithms employed during above process were discussed in detail. In this section an illustration is carried out using the datasets, to further clarify the functionality and algorithms which were discussed in chapters 4 and 5. The novel user model offer number of features that are valuable to current personalization issues. In this section, the functionality of the user model is presented. The corresponding functionality related sections in chapters 4 and 5 are as follows.

(i) Calculation of user's *PBC* values (sections 4.4 and 5.2)

(ii) Initializing user's domain purchasing behavior (sections 4.5 and 5.5)

(iii) Capturing user's transaction bas centric ed buying behavior (section 4.6)

(iv) Updating the user model after each transaction (sections 4.7.1 and 5.8.2)

(v) Long-term update of buying characteristics (sections 4.7.2 and 5.8.3)

6.2.1 Calculation of user's *PBC* values - (Start-up *PI* layer)

As mentioned in Chapter 4, the user model for an individual initiates with his/her demographic data. The *PBC* values are calculated using the demographics and the value functions. In this section;

(a) the information recorded in the *PI* layer, and
(b) the effect of demographics on *PBC* values are demonstrated.

Table 6.1: Restaurant transactions for User 41

Feature	Attribute	Attribute Name	Initial Relevance	Relevance after 4 Transactions
2	163	below $15	0.12	0.08
2	165	$15-$30	1	0.64
2	167	$30-$50	0.88	0.36
2	169	over $50	0	0
3	50	Poor Decor	0.15	0.06
3	51	Fair Decor	1	0.44
3	52	Good Decor	0.85	0.61
3	53	Excellent Decor	0	0
3	54	Extraordinary Decor	0	0
3	55	Near-perfect Decor	0	0
4	73	Fair Food	1	0.41
4	74	Good Food	0.7	0.43
4	75	Excellent Food	0	0
4	76	Extraordinary Food	0	0
4	77	Near-perfect Food	0	0.3
7	203	Fair Service	1	0.41
7	204	Good Service	0.7	0.43
7	205	Excellent Service	0	0
7	206	Extraordinary Service	0	0
7	207	Near-perfect Service	0	0
10	38	Central	0	0
10	137	Long Drive	1	0.41
10	214	Short Drive	0.68	0.54
10	247	Walk	0.32	0.45
17	136	Little Known But Well Liked	0.6	0.32
17	178	People Keep Coming Back	0	0.15
17	242	Up and Coming	0.74	0.47
22	35	Carry in Wine and Beer	0.03	0.14
22	80	Fabulous Wine Lists	0.03	0.02
22	148	No Liquor Served	0	0

Table 6.2 : Leg-wear transactions for User 41

Feature	Attribute	Attribute Name	Initial Relevance	Relevance after 15 Transactions
1	1	Price <= 5	0	0.15
1	2	5 < Price <= 10	0	0.4
1	3	10 <Price <= 15	0.9	0.45
1	4	15 <Price <= 25	1	0.04
1	5	Price > 25	0.1	0.02
2	6	AME	0	0.36
2	7	DAN	0.28	0.05
2	8	DKNY	0.28	0.02
2	9	BER	0.28	0.09
2	10	ELT	0	0.08
2	11	GIV	0.72	0.13
2	13	HPK	0.72	0.1
2	14	HOSO	0	0.23
2	15	NM	0.28	0.02
2	16	Hanes Too	1	0.04
2	17	DON	1	0.04
2	19	Smooth Illusions	0.72	0.02
2	20	ORO	1	0.04
2	21	EVP	0.28	0.02
2	22	HUE	0.72	0.02
2	23	Abs Ultra Sheer	1	0.04
2	24	FAL	0.72	0.02
2	25	Alive	0.72	0.02
2	26	Round the Clock	0.72	0.02
2	27	BB	0.72	0.02
2	28	ANNK	0.72	0.02
4	32	Basic	0.97	0.76
4	33	Fashion	0.77	0.26
5	35	Red	1	0.04
5	36	Black	0.26	0.98
5	37	Pink	1	0.04
5	38	Navy	0.26	0.02
5	39	Grey	1	0.06
5	40	Brown	0.26	0.02
5	41	Khaki	1	0.04
5	42	Nude	0.26	0.02
5	43	Tan	0.26	0.02
5	44	Cream	0.26	0.02
5	45	Beige	0.26	0.02
5	46	Metallic	1	0.04
5	47	Natural	0.26	0.02
5	48	Off White	0.26	0.02
5	49	Taupe	0.26	0.02
5	50	Blue	1	0.04
5	51	Green	1	0.04
5	52	Silver	1	0.04
5	53	Pink/Yel/Grn	1	0.04
10	113	Cotton	0	0.43
10	115	Nylon	0.68	0.06
10	116	Rayon	0.68	0.02
10	117	Luxury	0	0.23

The following Table 6.3 shows the contents of *PI* layer of the user models belonging to six demographically different individuals. Both the demographics and the calculated *PBC* values are presented. The eight *PBC* values for each individual were calculated using the eight value functions described in Chapter 5, section 5.2.

Table 6.3 : Information in *PI* Layer of the user model belonging to six individuals

	User41	User 240	User 256	User 6246	User 14662	User 15244
Age Group	21-30	31-40	41-50	51-60	21-30	31-40
Family	Single/ Bachelor	Couples with young kids	Couples with young kids	Couples with older kids	Couples with young kids	Couples without kids
Gender	M	F	M	M	F	F
Income	30K - 50K	> 200K	50K - 100K	100K - 200K	30K - 50K	> 200K
Occupation	Trade person or related	Professional	Other	Other	Intermediate clerical/Sales /Services	Managerial/ Admin
Work Hours	21-40	20 & less	40 & more	21-40	21-40	20 & less
Characteristics						
Adventurer	0.91	0.45	0.45	0.45	0.81	0.81
Family Person	0.05	1	1	0.86	0.81	0.34
Fun Spending	0.67	0.5	0.17	0.64	0.33	1
H/Conscious	0.21	0.68	0.54	0.86	0.25	0.57
Price Sensitive	0.53	0.36	0.85	0.39	0.62	0.27
Q/Conscious	0.29	1	0.38	0.57	0.19	1
Socializing	0.35	0.92	0.5	0.54	0.02	1
Time saver	0.42	0.93	0.52	0.41	0.52	0.49

According to the above table, demographically different users may still have similar *PBC* values (see Figure 6.2). For example 'Adventurer' characteristic is the same for users 240, 256 and 6246 although they have no common demographic value.

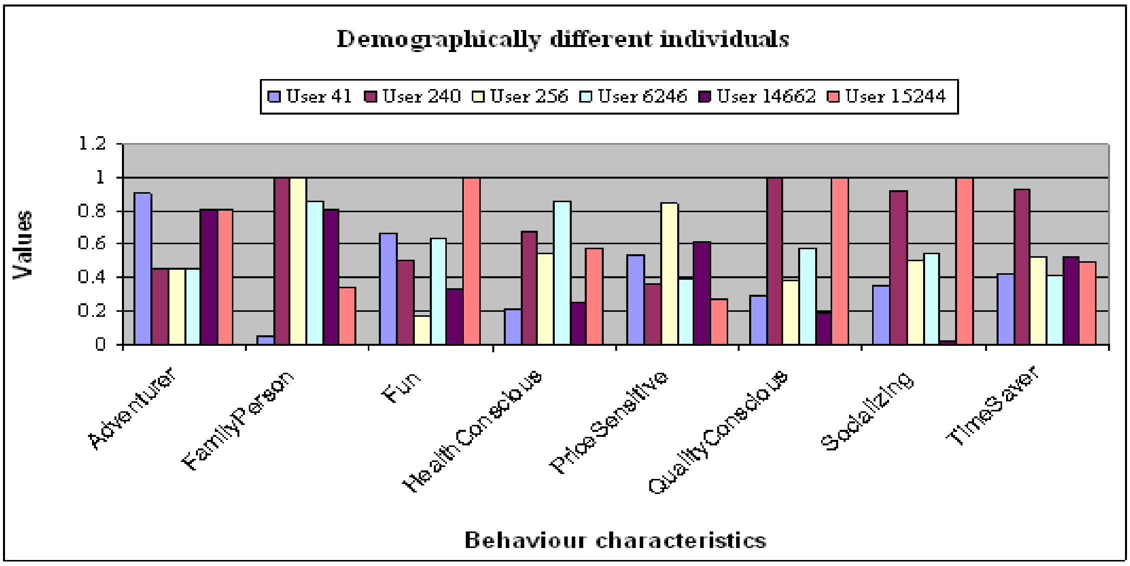

Figure 6.2: Characteristics of six demographically different individuals

As mentioned previously, the *PBC* values are flexible to capture the individuality of the users. The following examples explain the relationship between demographics and the *PBC* values. Figure 6.3 (based on Table 6.4) demonstrates the characteristic values of eight individuals where only the age is a variable.

Table 6.4 : *PI* layer of individuals having the same demographics other than the age

	User 40	User 6594	User 6966	User 8796	User 9684	User 14250	User 14398	User 19168
Age	38	25	39	47	31	29	31	27
Family	Couples with young kids							
Gender	F							
Income	50K – 100K							
Occupation	Other							
Work Hours	21-40							
Characteristics								
Adventurer	0.45	0.9	0.45	0.45	0.72	0.9	0.72	0.9
Family Person	1	0.81	1	1	1	0.86	1	0.86
Fun Spending	0.17	0.33	0.17	0.17	0.33	0.33	0.33	0.33
H/Conscious	0.73	0.25	0.68	0.78	0.57	0.57	0.57	0.57
Price Sensitive	0.85	0.68	0.85	0.76	0.85	0.8	0.85	0.8
Q/Conscious	0.52	0.38	0.38	0.38	0.38	0.38	0.38	0.38
Socializing	0.45	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Time saver	0.78	0.48	0.6	0.6	0.6	0.56	0.6	0.48

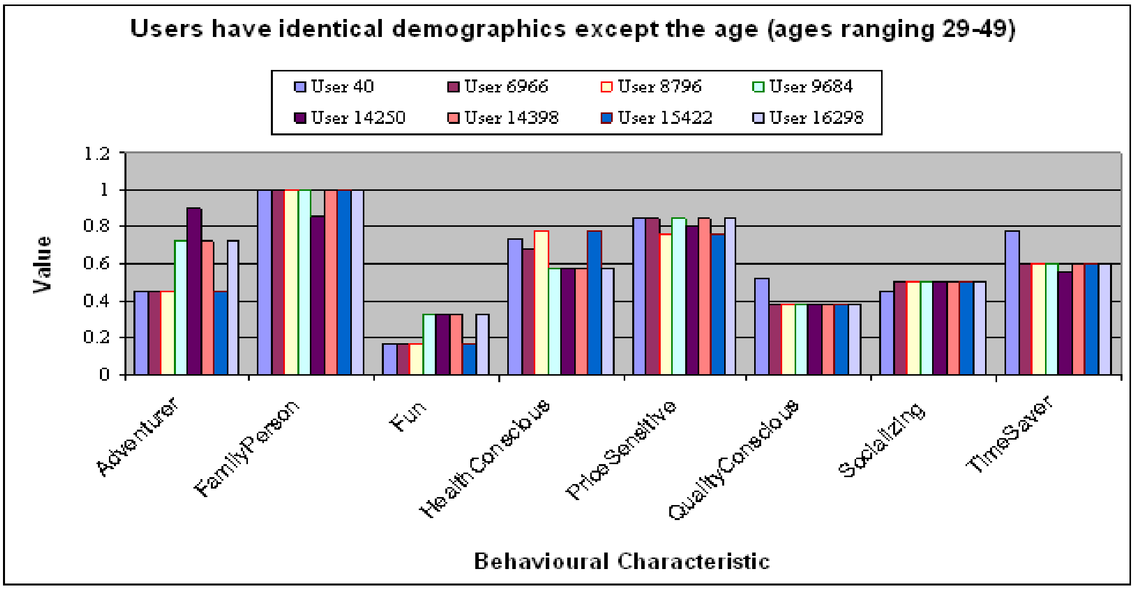

Figure 6.3: Eight demographically similar individuals with varying age

When all the demographics apart from the "age group" are the same, still the eight users possess varying *PBC* values. For example, although users User40, and User6966 have only one dissimilar demographic, they are dissimilar in four *PBC* values out of the eight.

Finally, Figure 6.4 (based on Table 6.5) shows that when only two of the demographics (family type and gender) were kept constant, the characteristic variance is even clearer.

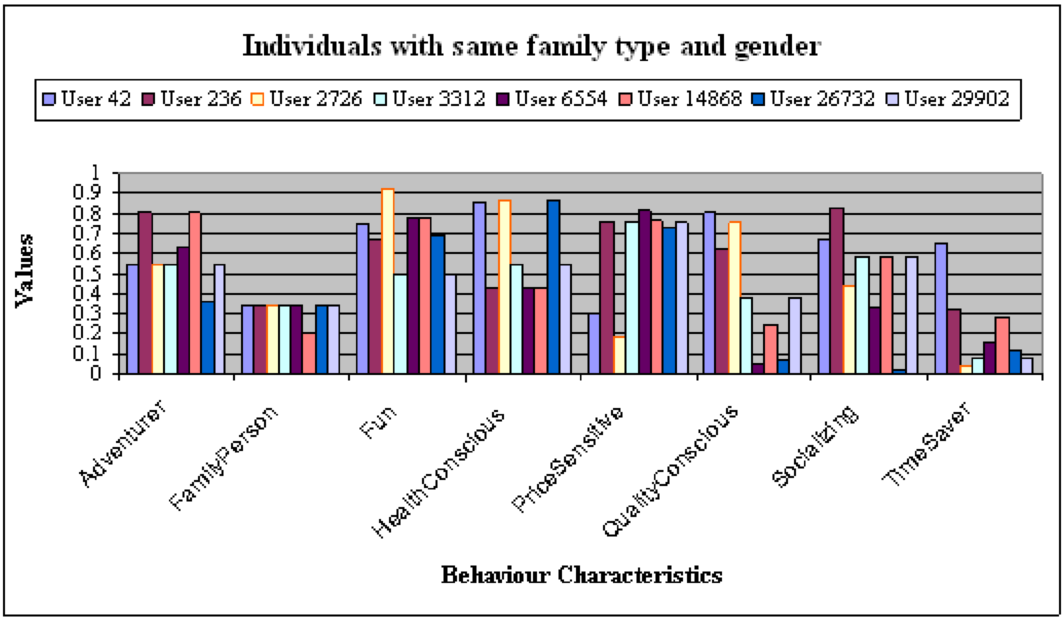

Figure 6.4 : Individuals with same family type and gender

Table 6.5 : *PI* layer of individuals having the same family type and of the same gender

	User 42	**User 236**	**User 2726**	**User 3312**	**User 6554**	**User 14868**	**User 26732**	**User 29902**
Age	56	35	51	44	33	27	55	37
Family	Couples without kids							
Gender	M							
Income	100K - 200K	50K - 100K	200K & more	50K - 100K	30K & less	30K & less	30K & less	50K - 100K
Occupation	Manage/ Admin	Manage/ Admin	Intermediate clerical/Sales /Services	Other	Other	Manage/A dmin	Laborer or related	Other
Work Hours	40 hrs & more	21-40 hrs	40 hrs & more	40 hrs & more	21-40 hrs	21-40 hrs	21-40 hrs	40 hrs & more
Characteristics								
Adventurer	0.54	0.81	0.54	0.54	0.63	0.81	0.36	0.54
Family Person	0.34	0.34	0.34	0.34	0.34	0.2	0.34	0.34
Fun Spending	0.75	0.67	0.92	0.5	0.78	0.78	0.69	0.5
H/Conscious	0.85	0.43	0.86	0.54	0.43	0.43	0.86	0.54
Price Sensitive	0.3	0.76	0.18	0.76	0.82	0.77	0.73	0.76
Q/Conscious	0.81	0.62	0.76	0.38	0	0.24	0	0.38
Socializing	0.67	0.83	0.44	0.58	0.33	0.58	0.02	0.58
Time saver	0.65	0.32	0.04	0.08	0.16	0.28	0.12	0.08

For example, users User14868 and User26732 have all dissimilar *PBC* values. However, with regard to demographics they are dissimilar in only two. In this context, if raw demographics were considered, it is difficult to see the difference in behavior of the two individuals.

6.2.2 Initializing user's domain centric buying behavior - (Start-up *DI* layer)

The above obtained *PI* layer *PBC* values can be used to calculate the initial product attribute relevance values for any number of domains. In this section we intend to demonstrate the following:

(a) How the *PBC* values of an individual are combined with domain based *IM*s to calculate initial preferences towards attributes in different domains.

(b) How the same *PBC* value is used in two different domains.

To demonstrate (a) and (b), *PBC* values of User41 is combined with $IM_{restaurants}$ and $IM_{leg\text{-}wear}$ to calculate his initial preferences in the two domains. Figures 6.5 and 6.6 represent initial preferences of User 41 for the restaurants and the leg-wear domains, respectively. The X-axis represents the feature number, attribute number pairs to show each attribute belonging to the product. These numbers were used to describe the products in the database (eg. attribute number 165 belongs to feature number 2). The Y-axis represents the relevance value of each of the attributes towards the User 41. The (long data) tables Table 6.1 and 6.2 provided in section 6.1.2, shows the initial relevance values calculated for User 41 for the two domains. Since there is no behavior information available for the either of two domains, the preferences are solely based on the *PBC* values. Therefore, if common or similar attributes exists (in the two domains) they should be equally relevant to the user.

Since the two example domains are quite different from one another, features or attributes that are common to the both domains are rare. The only common feature is the 'cost'/'price'. As in the (long data) Tables 6.1 and 6.2 (provided in section 6.1.2), and as shown in the two graphs (Figure 6.5 and Figure 6.6), in both domains user has high relevance values for the middle cost ranges due to his mid price sensitivity of 0.53. In the restaurant domain the user has 0.12, 1 and 0.88 relevance values for the cost ranges (below $15), ($15-$30) and ($30-$50) respectively; shown in the first (2/165), second (2/163) and the third histograms (2/167) of Figure 6.5. Again as shown in Figure 6.6, in the leg-wear domain user possess a high relevance (0.9) to the middle cost range (10 <Price <= 15), and 1 to the next upper cost range (15 <Price <= 25). Highest price range (Price > 25) has only a relevance of 0.1. Therefore, it is clear that the *PBC* value, assigns the user to the upper-mid price range of any product domain.

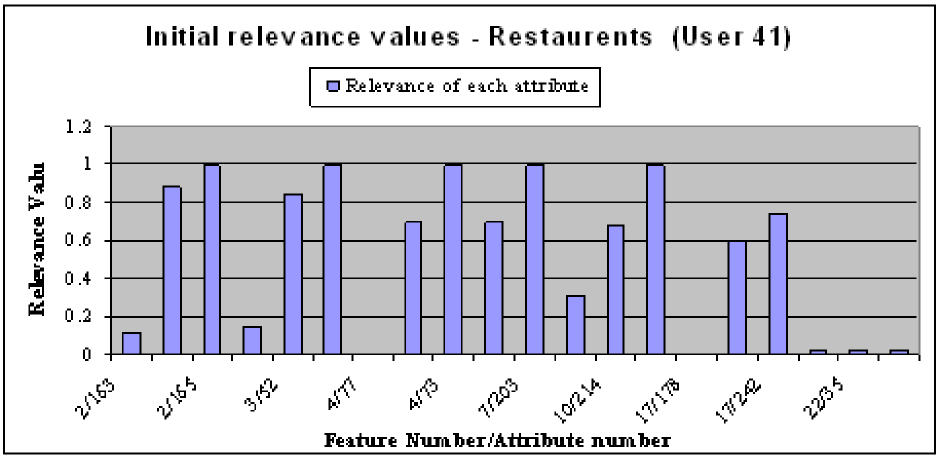

Figure 6.5 : Initial relevance values calculated for the restaurants for User41

This experiment shows how the *PBC* values are usable in multiple domains to generate start-up information in personalization (as claimed in (b)).

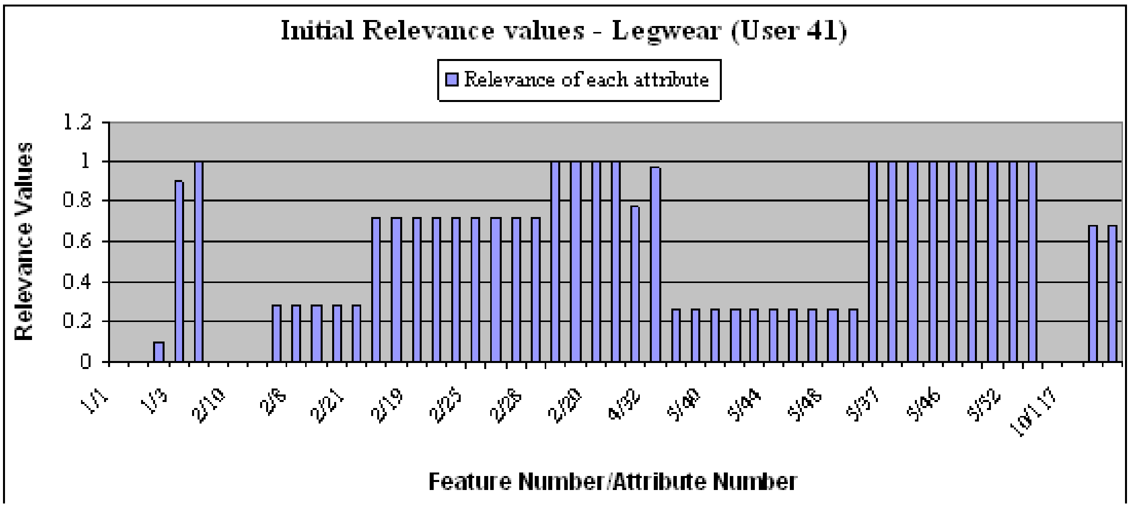

Figure 6.6 : Initial relevance values calculated for the leg- wear domain for User41

6.2.3 Capturing user's transaction specific buying behavior - (*TI* layer)

The *TI* layer holds all information users provided during the interaction about product preferences. Purchase behavior during interactions exhibit how user preferences for certain product attributes vary. This section analyses fifteen transactions for User41 in leg-wear domain. We intend to demonstrate the following:

(a) The effect of each transaction on the current relevance values of attributes

(b) The importance of transaction information to understand the changes in user preferences.

As mentioned in section 6.1.2, the dataset used in this work, only provide a list of items the user purchased in the past as purchase history, instead of actual transaction information. Therefore, in the experiments, all attributes belonging to the purchased items, were considered as explicitly requested by the user. User41 has 15 purchases in the transaction history, for purchasing leg-wear items.

The following (Table 6.7) is a subset of (only nine) attribute preferences obtained during 15 transactions in the leg- wear domain for User41.

Table 6.6 : Relevance value change after each transaction, for nine attributes

Feature number	Attribute number	Relevance value at the start	After T1	After T2	After T3	After T4	After T5	After T6	After T7	After T8	After T9	After T10	After T11	After T12	After T13	After T14	Relevance value at the last
1	1	0	0	0.2	0.36	0.29	0.23	0.18	0.34	0.47	0.58	0.46	0.37	0.3	0.24	0.19	0.15
1	2	0	0	0	0	0.2	0.16	0.13	0.1	0.08	0.06	0.25	0.4	0.52	0.62	0.5	0.4
1	4	1	0.8	0.64	0.51	0.41	0.33	0.26	0.21	0.17	0.14	0.11	0.09	0.07	0.06	0.05	0.04
2	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0.36
3	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	32	0.97	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.78	0.62	0.7	0.76
5	36	0.26	0.41	0.53	0.62	0.7	0.76	0.81	0.85	0.88	0.9	0.92	0.94	0.95	0.96	0.97	0.98
10	115	0.68	0.74	0.59	0.47	0.38	0.3	0.44	0.35	0.28	0.22	0.18	0.14	0.11	0.09	0.07	0.06
12	121	0	0	0.2	0.36	0.49	0.59	0.47	0.38	0.3	0.44	0.55	0.64	0.71	0.77	0.62	0.5

Figures 6.3-5 demonstrates the changes in attribute relevance values. For clarity, each graph demonstrates three attributes out of the total nine. The graphs are named with feature/attribute to distinguish one from another. Graph 1/1 shows the relevance value change for the attribute "1" of feature "1".

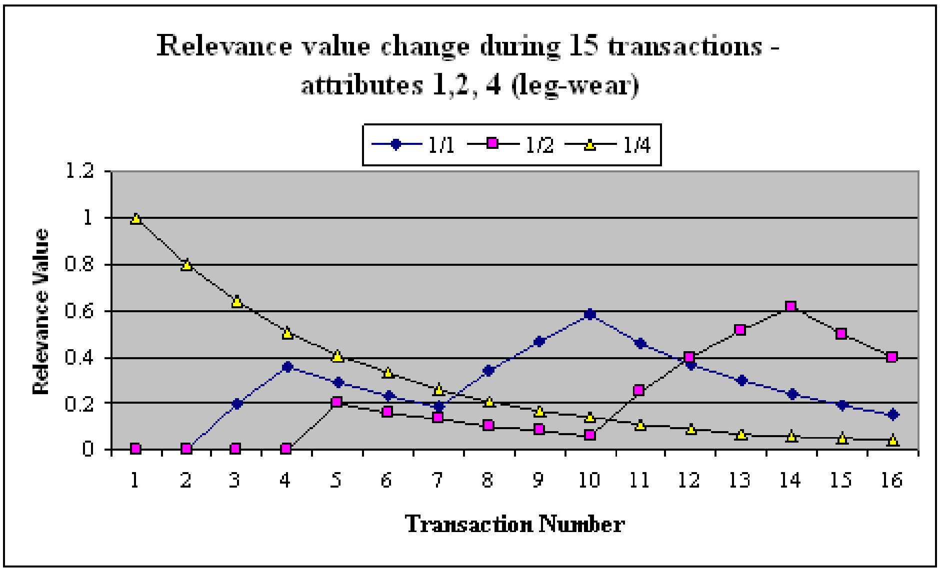

Figure 6.7: User preferences towards feature/attribute 1/1, 1/2 and 1/4

Attributes 1/1 and 1/2 were considered to be irrelevant at the start-up; the initial decision becomes false, due to the user requesting the attribute in a later transaction. Attribute 1/4 was highly relevant depending on user's *PBC* values, but declined in importance since the user never requested for it. In Figure 6.8, attribute 3/31 was never required, while attribute 4/32 was always requested by the user.

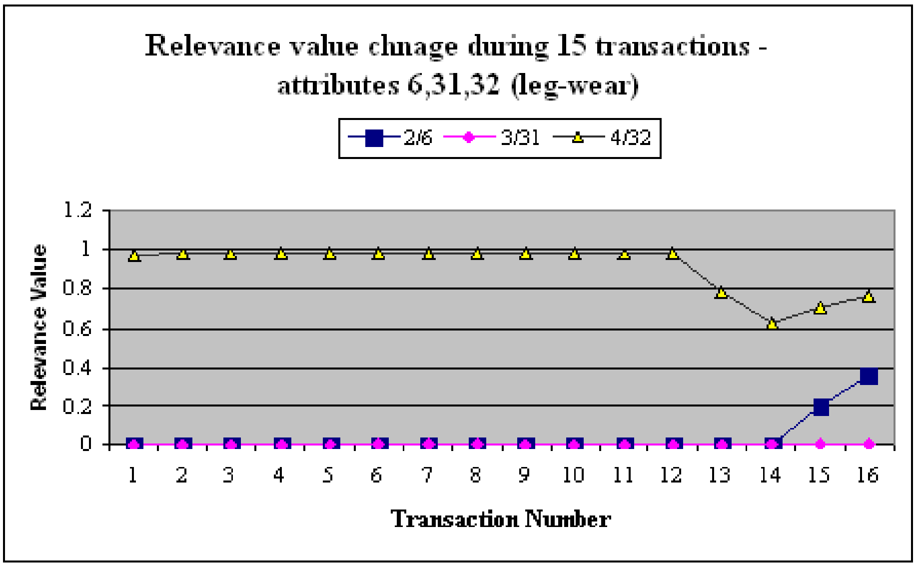

Figure 6.8 : User preferences towards feature/attribute 2/6, 3/31 and 4/32

In Figure 6.9, attribute 5/36 was initially considered to be less important; however, values later reached the highest relevance level since the user requested the attribute in all transactions. For both attributes 10/115 and 12/121, varying preferences are observed.

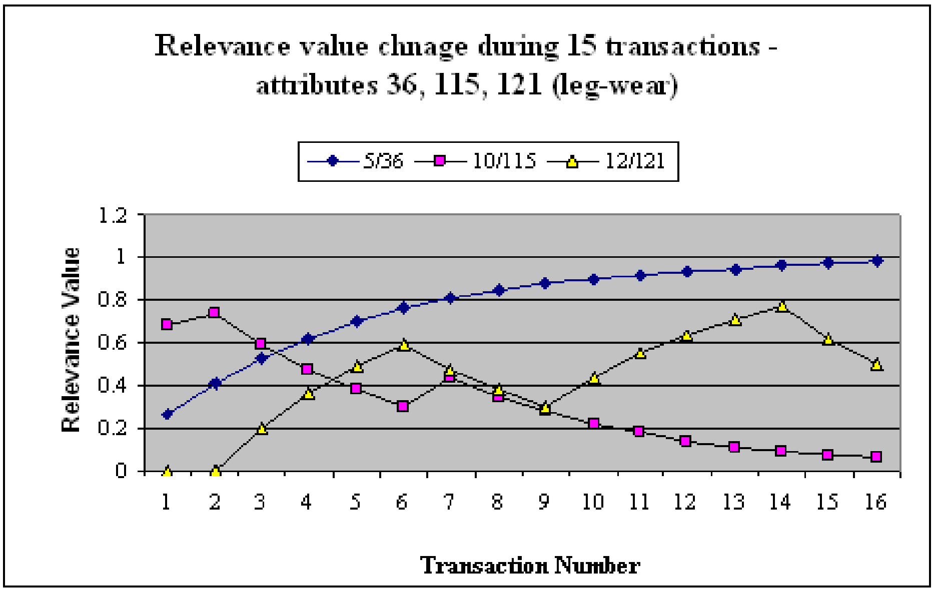

Figure 6.9 : User preferences towards feature/attribute 5/36, 10/115 and 12/121

The discussion on charts shows that rather than looking at relevance of an attribute at a certain point of time, it is important to follow up each transaction over time. The reason being, relevance values of occasionally requested attributes (such as 1/1,1/2, 10/15, and 12/21) do not explain the reason for increase or decrease. However if transaction information is analyzed, there is a high chance of recovering meaningful occurrence patterns such as time based requests.

6.2.4 Updating the user model after each transaction - (*DI* layer based on *TI* layers)

This section demonstrates the models ability to adapt to user's changing requirements. Updates to the initial relevance values after 15 transactions in the leg-wear domain and four transactions in the restaurants domain for User41 is provided in (section 6.12) Table 6.2 and Table 6.3 respectively. Figures 6.10 and 6.11 present the data in above two tables. It shows how the initial predictions did not represent the reality.

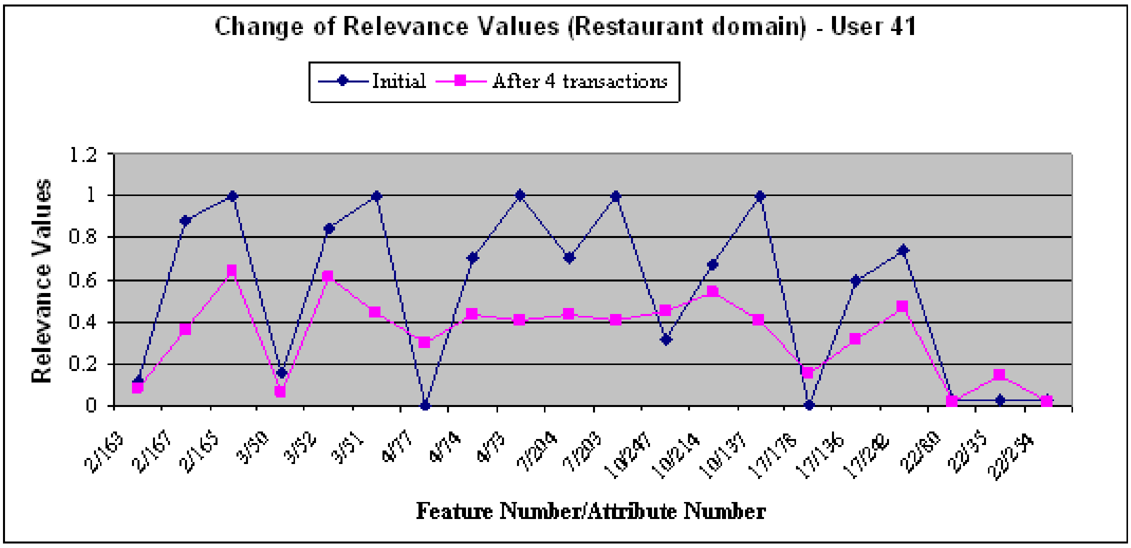

Figure 6.10 : Attribute relevance values from the DI layer for Restaurants

For example, in the Table 6.2, attributes belonging to feature 3, were initially assigned relevance values to attributes 51, 52 and 50 in the most relevant order. However after transactions the order of importance changed to 52, 51 and 50. Similar changes are observed in the Table 6.3, for Leg- wear domain.

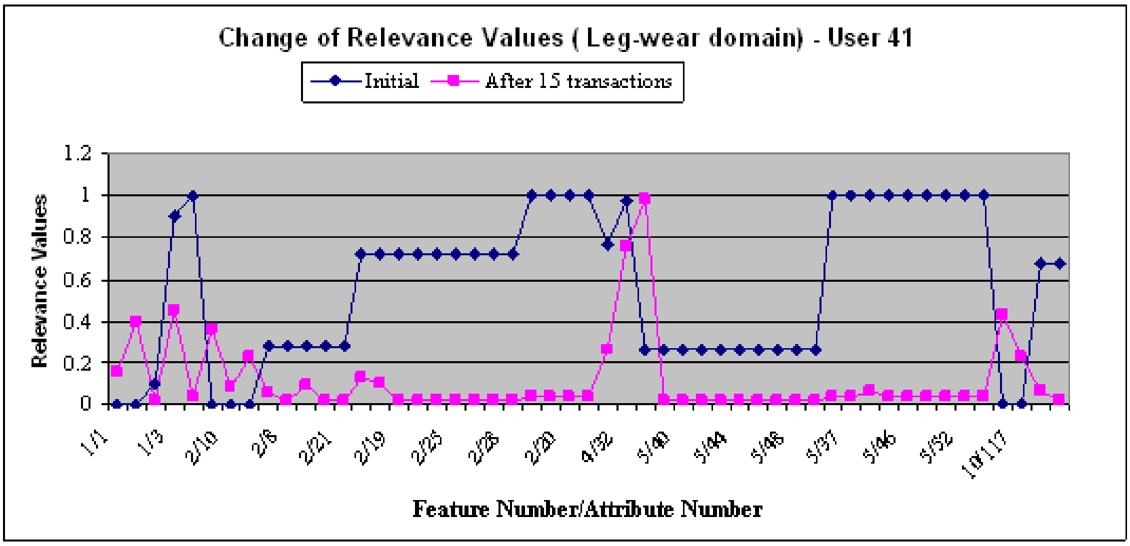

Figure 6.11 : Attribute relevance values from the DI layer of leg-wear

If the user is a 'grey sheep', then *PBC* value based predicted user behavior in the start-up, can be different from the user behavior resulted after transactions. If the user rather behaves according to the stereotypes, then a more continuing and confirming new relevance values are observed. For example, for user preferences for feature 2, cost ranges are preferred in the same order (165,167, 163 then 169). However, the value of the preference has decreased for all three attributes. Therefore, according to the above evidence the model is able to capture the changing user preferences.

6.2.5 Long-term update of buying characteristics - (*PI* layer based on *DI* layers)

As described in Chapter 4 section 4.7.2, after the user interacts in several domains during a long period of time, the user's purchase habits may change. In addition, the initially calculated *PBC* values can be invalid for certain users, since they were stereotypic values. The actual user behavior is continuously captured in the *DI* layer of the *LUM*. Therefore, based on such information (instead of the initial stereotypic based *PBC* values) more accurate user behavior based *PBC* values can be calculated.

Table 6.7 shows the initial *PBC* values calculated for User41. Then based on User41's transactions in leg- wear and restaurant domains, new *PBC* values for those domains were calculated. According to the algorithm provided in Chapter 5, these values are combined to calculate the new *PBC* values. In order to improve the accuracy of the *LUM*, the *PI* layer contents are replaced by the new values. In the future, if the user interacts in new application domains, then the start-up *PBC* values for such domains can be considered more accurate than the (stereotype based) values utilized for the previous two domains as start-up information. The graphs in Figure 6.12 and 6.13 plot the initial *PBC* values against the later calculated ones.

Table 6.7 : The initial, updated based on the domain and current PBC values

PBC	**Initial *PBC* values**	**Updated *PBC* values**		**Current re-calculated *PBC* values in PI Layer**
		Leg- wear Domain	**Restaurant Domain**	
Time Saver	0.42	0.42	0.51	0.465
Price Sensitive	0.53	0.57	0.53	0.55
Q/Conscious	0.29	0.39	0.54	0.465
Fun Spending	0.67	0.5	0.67	0.585
H/Conscious	0.21	0.79	0.21	0.5
Family Person	0.05	0.19	0.05	0.12
Socializing	0.35	0.5	0.87	0.685
Adventurer	0.91	0.53	0.79	0.66

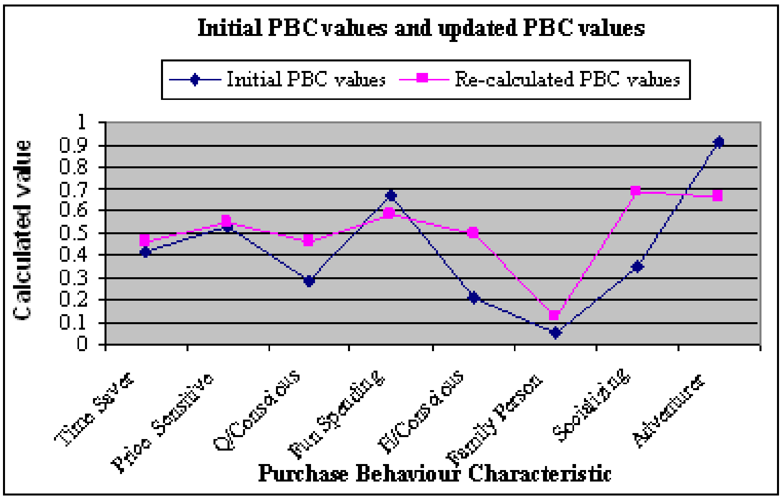

Figure 6.12 : Initial and recalculated PBC values

In both domains, few of the *PBC* values remains unchanged from the initial calculations; indicating the reliability of the initial *PBC* values. Only 'Time Saver' in leg-wear domain remains the same. However, 'Price Sensitive', 'Fun Spending', 'H/Conscious' and 'Family Person' values remains unchanged in the restaurants domain.

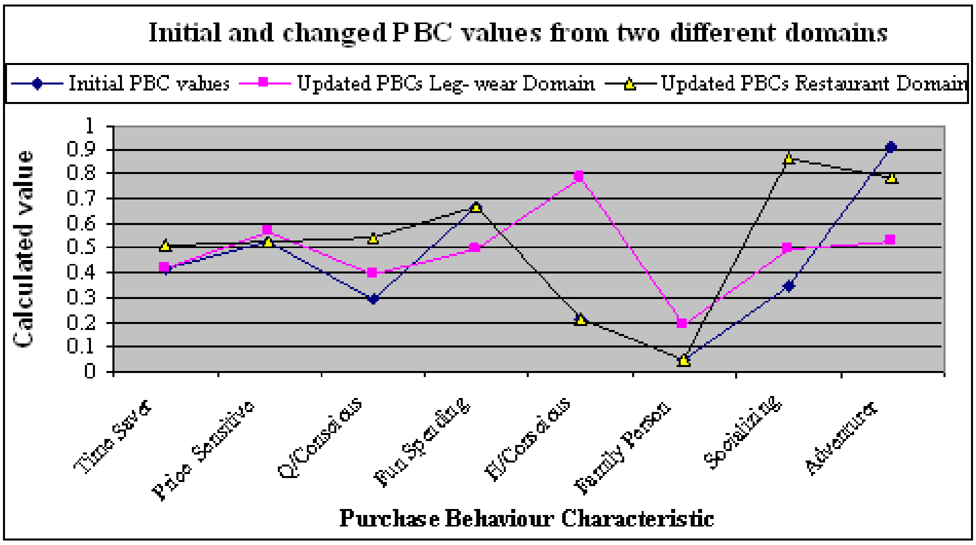

Figure 6.13 : Re-calculated domain based PBC values and the initially calculated vales

The reason could be either of the following:

(a) The user behaves according to his stereotype in leg-wear domain but not in the restaurants domain.

(b) The data set allows only considering very low number of (four) transactions for Restaurants domain which is inadequate for capturing user's domain behavior whereas fifteen transactions in the leg-wear domain capture the actual user behavior.

The initial *PBC* values are replaced by the re-calculated values resulted from combining the domain based values. Figure 6.12 shows the difference of the later calculated values from the initial stereotypic values. Figure 6.13 shows how each domain based behavior value deviate from the initial values.

6.3 Value of the Layered Model

The novel user model architecture we present in this dissertation contributes valuable features that are major issues in the user modeling research. In this section, we use the data to demonstrate six such features which are discussed under the following topics.

(i) Ability of the model to capture individuality of the user using PBC values instead of raw demographics.

(ii) Ability to provide users with personalized services in the initial interaction-solves the 'New User' problem.

(iii) Ability to handle the dynamic product markets by solving the 'New Item' problem.

(iv) Ability to understand the reasons for preferences and hence the ability to capture the individuality of the user.

(v) Ability of the user model to provide personalization in multiple domains.

(vi) Ability of the user model to enforce control over irregular transactions

6.3.1 Ability of the model to capture individuality of the user

As mentioned before, the first layer of the user model captures the user's general buying expectations, irrespective of the application domain. We justified demographics as suitable information to capture such expectations. As shown in Figure 6.14, the raw demographics are able to describe user requirements and abilities once they are combined with some market research heuristics.

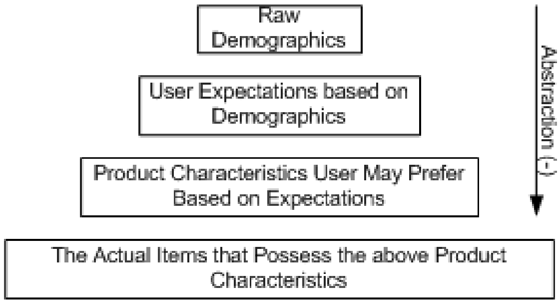

Figure 6.14 : Highest to lowest abstraction, when identifying user behavior

For example, knowing an individual's income allows to judge the price ranges one possibly could accept. According to the figure, down the hierarchy the abstraction reduces. But it maps the individual to the items that he/she may accept. Although raw demographics are flexible to use, for a broader domain like eCommerce, a set of general rules can be easily introduced. For example, deriving the individual's price sensitivity is general to all eCommerce activities. In addition, such one level of reduced abstraction will lead to forming higher number of measurements than the few demographics that are available. For example rather than clustering consumers according to six demographics, these six values could be combined to form ten different mind sets or characteristics to describe the consumer in a more flexible way. This kind of approach was taken by the consumer topologies adapted by organizations such as Vals,[12] Roy Morgan Value segments,[13] and Global MOSAIC.[14]

The examples provided in the section 6.2.1, illustrate the relationship between the *PBC* values and raw demographics. It shows the flexibility and importance of using PBC values instead of raw demographics.

6.3.2 Ability to provide personalized services from the initial interaction

In current eCommerce environments new users are expected frequently. One of the major problems that recommender systems face is identifying their preferences for immediate provision of personalization. Many systems require to be trained by the user, rating several items prior to beginning recommendations. But users are reluctant to spend their searching time on training a system.

In *eHermes* PERSONAL the demographic information collected at the registration is a once-off process. The *PBC* values, calculated using the demographics are then used in number of domains to calculate initial preferences. Therefore, the user model is capable of providing users with personalized services even during their initial transaction. Even though the user has never interacted within the given domain or with any of the other domains, still domain specific preferences can be calculated as initial knowledge about the user.

In the section 6.2.2, Figures 6.5 and 6.6 show start-up values for two domains, restaurants and leg- wear. As the user interacts with the system these relevance values may change, but still provide some understanding about the user as start-up information. Hence, this solves the 'New User' problem at the start-up.

6.3.3 Ability to handle the dynamic nature of the product markets

In the current dynamic eCommerce product markets, new items arrive frequently. Recommender systems that rely on user ratings are unable to recommend a new item that is not yet rated by any of the users. Therefore, the chances of recommending such items stay rare and popular items show even more growing popularity. Due to the content based approach taken in the new user model the user preferences are represented as relevance values towards product attributes. The item search is conducted using item description rather than user similarity. Therefore, all items that satisfy a user query have an equal chance of retrieval. This valuable feature qualifies the new user model architecture to be utilized in dynamic environments where new items frequently arrive.

6.3.4 Ability to capture the individuality of the user

Individuals with exact demographics end up with identical set of *PBC* values (shown in Figure 6.15, based on Table 6.8 for the users User29160 and User2240).

Table 6.8 : Similar demographics resulted in identical *PBC* values

User 29160	User 2240
40	38
F	F
Couples with young kids	Couples with young kids
High School	High School
Education	Automotive
Trade person or related	Trade person or related
20 hrs & less	20 hrs & less
50K - 100K	50K - 100K
Characteristics	
0.52	
0.86	
0.38	
0.17	
0.68	
1	
0.5	
0.45	

If only the demographical similarity is considered, these two users are expected to demonstrate identical preferences. However, by capturing the transaction specific user behavior and then updating the initial model, deviations in their preferences can be observed.

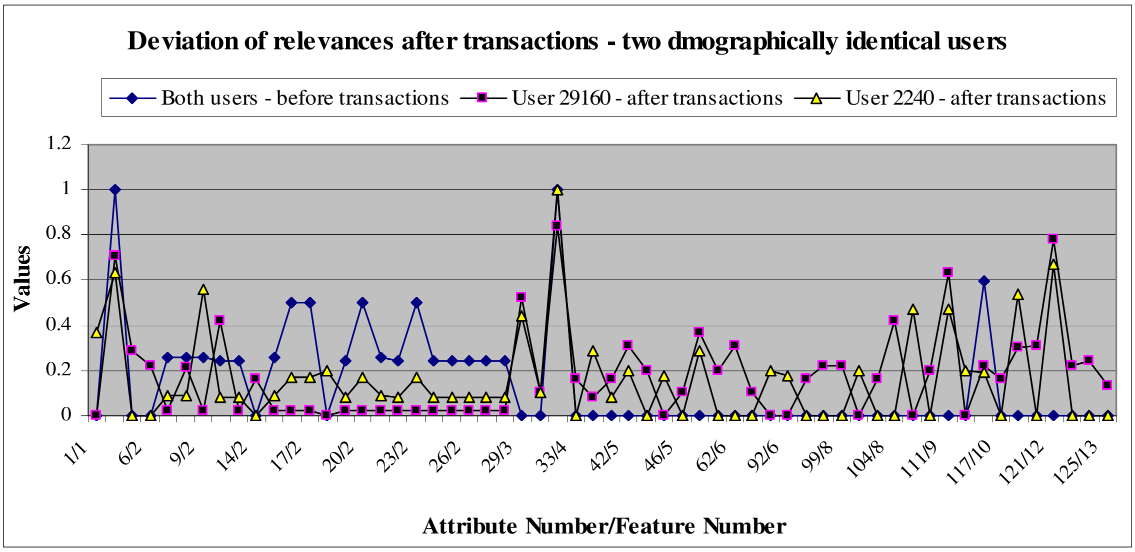

Figure 6.15 : Initial and after transaction relevance values - User 29160 and User 2240

As shown in Figure 6.15, the two users seem to follow the predicted behavior with regard to some attributes (attributes belonging to feature 10 - much closer behavior) and sometimes both deviate from the prediction (attributes belonging to from features 4-8). Sometimes both demonstrate a preference towards the same attribute but behave differently from predicted expectations (attributes belonging to features 4, 5 and 6).

This example demonstrates the ability of the model to identify users' individuality rather than labeling them as identical just based on the demographics. In consumer segmentation approaches such users get grouped together and remain in the same segment.

6.3.5 Ability to provide personalization in multiple domains

The *LUM*, allows usage over multiple domains. The Figures 6.5 and 6.6 show the use of same *PI* layer information for the two domains restaurants and leg- wear. In addition, if domain hierarchies were used, then direct use of preferences among common attributes is possible. This is more specifically described in Chapter 4, section 4.3.2.

6.3.6 Ability to enforce control over irregular transactions

The *TI* layer of the user model allows observing the immediate changes due to user transactions. For example, if the user exhibit seasonal changes (such as special purchases during Christmas) then certain attributes may become important during that period. If the behavior corresponds to the 'season', after a time period the behavior will return back to normal. Using such transaction data in updating the corresponding *DI* layer, may add 'noise' to the actual user behavior pattern. The current user models do not provide the suitable structure to identify and isolate such transactions. The novel *LUM* architecture provides the necessary requirements to notice such transactions. Furthermore, the update methods in *LUM* allow control on such updates by manipulating the "learning rate parameter" (see Chapter 5 section 5.7.1) to control the changes in *DI* layer. As such, the impact of the seasonal kind of transactions is minimized in the *DI* layer.

6.3.7 Ability to support user scrutinize interfaces

Scrutable user models are considered as a solution to privacy issues in user modeling. When online systems provide users with personalized interactions or recommendations, often users are eager to know how they come up with such outcomes. Again when users feel that the system holds erroneous assumptions about them there should be a way of handling it. As a solution many researchers claim need of self scrutable user models where users are free to access and modify the information contained in their own model. One major problem with providing such facility is the information recorded within the user models. Often such information is either incomplete or not in human understandable form. Our layered user model consists of linked modules (components) and presented in XML. Both layers 1 and 2 which contain the information used for predictions are readable and easy to understand by the human user. Although not implemented in the current work, a user scrutable interface could be easily coupled to the existing structure and the presentation of the user model. Then, the users are free to adjust any of the model contents such as their characteristic values in the layer 1 or attribute relevance values in domain centric layer 2 components.

6.4 Usefulness of the User Model

In addition to the above discussed functionality related value, the novel *LUM* is important in data mining. Current data mining approaches use customer transaction data and payment data to discover important behavioral patterns within user segments. Data mining is carried out by organizations for the reasons such as improving target marketing campaigns, customer relationship management (CRM) and customer retention and avoiding churn.

Target marketing is identification of consumer segments and use suitable marketing strategies that maximize the results for the particular community. Data mining is employed to discover behavioral segments and further study their behavior by projecting the data to existing customer bases consists of transactions and billing histories.

CRM mainly deal with cross-selling, up-selling (recommending) products to consumer segments. Cross-selling, is identifying the possible items that a certain consumer population will interested in, based on their transaction histories. Up-selling is selling similar but more expensive items to the consumers who showed interest in certain items. Data mining is used to analyze existing consumer bases to identify groups of people who preferred the same items and then to apply the marketing strategies for recommendations.

Another important application of data mining is identifying customer churn. By analyzing data, customers who exhibit behavior of leaving the company (churn) can be identified. In such situation strategies for customer retention is used.

Consumer trend analysis is extremely important for marketing and advertising purposes. The detailed information content in the *LUM* architecture is ideal for such studies. The *PI* layer information about the individual and his/her domain centric preferences in the *DI* layer of the user model can be combined to predict user personality and trends in purchasing. Again when considered as a collection of user models, trend analysis is possible within diverse product domains as a population or community of users. Two types of possible trend analysis are described below.

6.4.1 Trend analysis for the individual

With regard to each individual, his/her behavior in each product domain can be captured (see 6.3.3). Therefore, if high preferences for certain attributes are observed such tendencies can be identified as a user trend in the domain. Advertising and target marketing can be utilized to take advantage of such identified trends.

As shown in section 6.2.3, domain based user information can be exploited to discover time dependent user behavioral changes. For example, preference changes for individual product attributes over time can be observed. If reasons for such changes are analyzed, time dependent user needs can be discovered. For example, users may exhibit seasonal behavioral changes such as different purchase patterns during the Christmas season. As previously mentioned (section 5.36), such information can be utilized in user model update. In addition, suitable advertising can be used to help both user and the vendor.

6.4.2 Trend analysis for the entire user population

As a result of the *LUM* architecture, the *DI* layer of the user model holds domain based user behavior. If such information is analyzed, users can be grouped based on their actual purchase behavior. Time dependent behavior of such identified groups can be analyzed long term. When doing so, facts such as their trends over time can be observed and treated accordingly. For example, change of behavior in user groups over different seasons of the year in different domains are interesting to observe. Furthermore users can be grouped according to the *PI* layer, and analyze the trends pertaining to *PBC* values or raw demographics. The *LUM* architecture provides market annalists with a number of useful user information within the single user model.

6.4.3 Usability in target advertising

The *LUM* architecture facilitates categorizing users according to behavior shown in the layers. In long term, each user can be categorized into one of the three categories: (i) *Traditionalist*, (ii) *Domain Traditionalist*, or (iii) *Volatile buyer*.

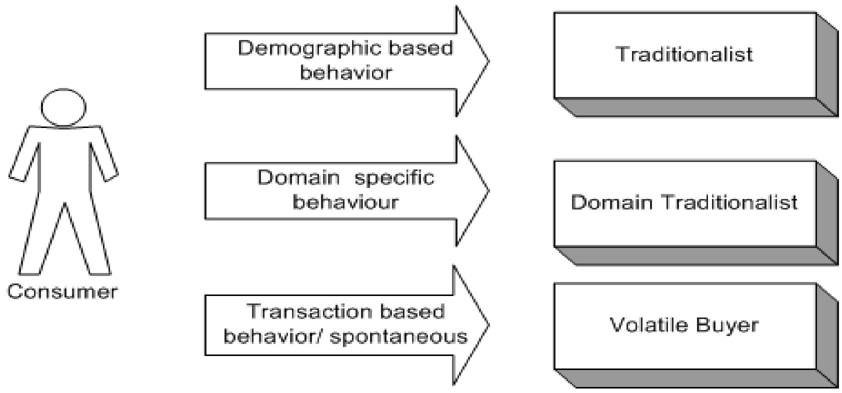

Figure 6.16: Consumer types based on the PBC values

A buyer who confirms the demographic based behavioral predictions is called a *traditionalist.* A *traditionalist* demonstrates continuous behavior patterns across domains. For example, a low income earner who always goes for the low price, (irrespective of the product domain) paying less attention to the other aspects of the items is a traditionalist with respect to expenditure. A *Domain Traditionalist* demonstrates consistent purchasing patterns within a single domain. For example, an individual who purchase expensive clothing during all transactions, and at the same time spending less in groceries is said to be a domain traditionalist with respect to expenditure. A *Volatile buyer* always shows an unpredictable preference during each transaction. Such people could be considered to be more reactive to effective advertising.

An important functionality of the model is that it does not tie down a person into a specific category. The transaction based incremental update capability allows a person to be moved from one classification to the other. This is very important in the dynamic commercial environment where changing user needs are fulfilled in a volatile product market. The reason for such movement can be explained as follows.

1. Initial misclassification due to insufficient information.

2. Due to the user being non traditionalist who does not adhere to expected demographic based behavior.

In the advertising tasks, these three types of behavior can be treated separately. A shift in the behavior of an individual can be an interesting observation for marketing. If trends within groups of users are identified, such groups can be treated with the same strategies in advertising and marketing.

6.5 Summary

The chapter provided the functionality, value and the usefulness of the new *LUM* architecture using available datasets. Section 6.3 on functionality proved the ability of employing the proposed methodologies to the *LUM*. The section on value of the *LUM*, demonstrated the importance of layering user information and how it facilitate the abilities of the user model. Finally, the discussions on usefulness of the user model (in section 6.4) pointed out the existing and possible future improvements. The next chapter demonstrates the application of the user model in eCommerce product retrieval.

Chapter 7

Layered User Model for Personalized Interactive Product Retrieval

In the previous chapter, the functionality, value and usefulness of the user model was discussed and demonstrated using datasets. This chapter consists of two major parts. The first part of the chapter explains and discusses the application of *LUM* in the Personalized Interactive Product Retrieval Process (*PIPRP*). Involvement of the *LUM* provides personalization in all three phases of the online product retrieval: requirement elicitation, product search and product presentation, and thereby, minimizes the problems in online product retrievals such as null retrieval, retrieving unmanageable number of items, and the retrieving unsatisfactory items. At the end of the demonstration, the results are discussed. And then, evaluation of the model is carried out using a set of criteria, which is based on existing evaluation methods.

The organization of the chapter is as follows. Section 7.1 discusses the related work in online interactive product retrieval and search methods. Section 7.2 describes the process of eCommerce activity when purchasing products, explaining the phases of the *buyer decision process in purchasing*. It shows the importance of the system-user interactions to clearly identify the user need. Section 7.3 presents the *PIPRP* algorithm and explains the similarity measure calculation involved in product selection. Section 7.4 presents the experimental plan for the *PIPRP* and presents the scenarios utilized in experiments. Then, the steps of the algorithm are demonstrated using a scenario, according to the experimental plan. At the end, the results of the demonstration are discussed. Section 7.5 presents the related work in evaluating personalized systems and explains the necessity of evaluating the combined performance of the *LUM* and the *PIPRP*. Then, a novel set of evaluation criteria is proposed (based on existing work) and the thesis work is evaluated using the proposed criteria. Finally, in section 7.6, a summary of the chapter is presented.

7.1 Related Work

Personalized services are provided in current eCommerce websites in different situations such as during product retrieval, product recommendations and advertising. One of the crucial tasks for today's eCommerce systems is to help the users find products that satisfy their preferences with minimum search effort. Therefore, we concentrate on personalized product retrieval where the novel *LUM* can be effectively used. Two key areas which provide the background for this chapter is discussed in this section.

(i) Online product retrieval in eCommerce

(ii) Search methods

7.1.1 Online Product Retrieval in eCommerce

Based on the work described in (Burke, 2002b; Schmitt and Bergmann, 2001 and Bergmann and Cunningham, 2002), product retrieval approaches in eCommerce websites can be categorized as shown in Figure 7.1.

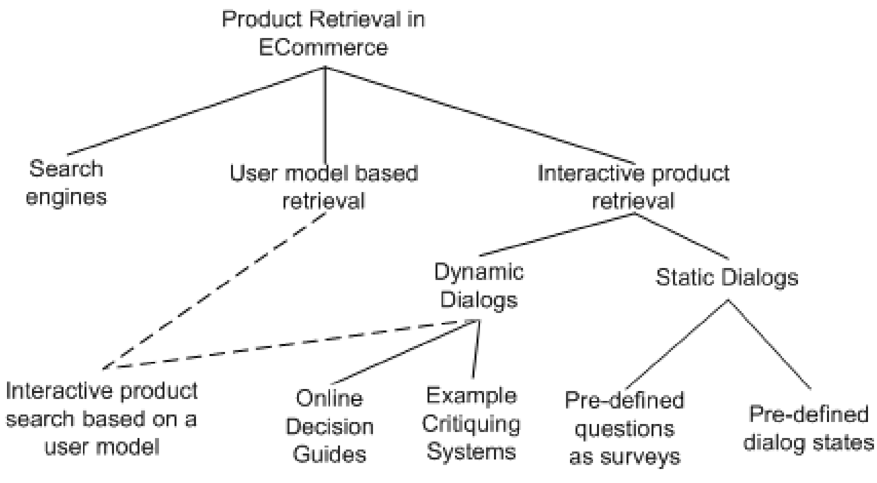

Figure 7.1 : Product retrieval methods in eCommerce

There are three main approaches employed in online product retrieval; search engines, user model based retrieval and interactive product search. As shown using dotted lines, we identify a new category of systems that has features common to both user model based retrieval and dynamic dialog systems. In the following sections, the three main product retrieval approaches in eCommerce (as shown in Figure 7.1) are discussed. Under interactive product retrieval, each of the dynamic dialog systems is discussed.

Search Engines Based Retrieval

Most of the websites use simple search engines, and do not provide personalized interactions. Although, only searching is provided, different approaches of such search engines can be distinguished depending on their additional features. Search engines are further discussed in detail in section 7.1.2.

User Model Based Retrieval

In the literature, there are instances where product retrieval is carried out by employing user models. Then, the retrieval strategy depends on the information in the user model and the underlying user modeling technique. As described in Chapter 2, to employ content based searching methods, user's preferred attribute values are matched against the product descriptions, whereas in collaborative methods descriptive product information is not used. If the user model is content-based, then the retrieval strategies are as shown in Figure 7.2. Such retrieval strategies strive to reduce the number of system-user interactions. In the collaborative filtering approach, the issues involved in system-user interactions are seen from a different angle. For example, *new users* need to rate a set of items for the system to understand their preferences. In this approach, selecting the *best* list of items for a new user for initial ratings (so as to reduce the number of ratings required to start recommendations) is crucial. Rashid *et. al.* (2002) present some interesting and important work carried out in such collaborative type online product selections. In their work, different techniques are followed to learn about a new user within a minimum number of interactions. The work investigates techniques to provide the new user with a list of products that the user is most likely to have an opinion about. In addition, the items presented should maximize the system utility by learning both about the new user and the information useful for the entire user population.

Interactive Product Retrieval

Alternatively, interactive services are provided with interactive dialog systems. Static dialogs are unable to provide personalized interactions. Such approaches obtain user requirements using survey style questions. More flexibility is offered in dialogs using pre-defined dialog states. With them, user interaction starts at a certain point of the dialog, and proceeds in a pre-defined path. In contrast, dynamic dialogs have the ability to adapt to different users, and the answers users provide; in other words, they can provide personalized interactions. Online dynamic dialogs are employed by;

i. online decision guides,
ii. example critiquing systems, and
iii. dialog systems using models of the user.

In the literature, the initial **online decision guides** were used for various applications such as fault diagnosis (Cunningham and Smith, 1994) and eCommerce product search (Cunningham *et. al.*, 2001). Later such systems were improved incorporating an Incremental Case Based Retrieval (ICBR) method. The technique is incremental in the sense that it does not need the complete query specifying all the user preferred attributes of the item, but in fact build it up by asking focused questions from the user.

The ICBR process starts as an incomplete query. The first pass retrieves a subset of the initial case base that is similar to the query. Then, subsequent refining queries are directed to the user to reduce the size of the initial retrieval. The refining queries use a simple information theoretic approach to find the item feature that best discriminates between the current set of retrieved cases. Rather than searching for the most suitable product, this method tends to reduce the dialog length. More recent work uses the facts such as *similarity* and *variance of products distribution* and hence pays more attention to user satisfaction in the retrieval (Bergmann and Cunningham, 2002; Schmitt *et. al.*, 2002).

The idea of **example critiquing systems** is to guide the user to the product he/she prefer by eliminating the products that do not comply with the query. This approach is initiated by the user specifying an example item, which is known in the past. The system considers the provided example as the starting query and provides the user with a list of similar items.

Then, the user gets to criticize the provided item list on their features which are not satisfying. Based on each criticism, the system presents a list of items, until the user is satisfied. As previously mentioned in Chapter 2, the early example critiquing systems were known as ‘Find me’ systems (Burke, 2002b). There has been more work carried out in the area to improve the initial outcomes (Pu *et. al.*, 2006; Viappiani *et. al.*, 2006; Reilly *et. al.*, 2007). The pros and cons of such systems are discussed along with the new algorithm in section 7.5.

As shown in Figure 7.1, there is a new category of interactive product retrieval systems: dialog systems using models of the user. This category of systems inherits from both user modeling and dialog systems research. They employ a model of the user to provide personalization during product retrieval. Adaptive Place Advisor (Thompson *et. al.*, 2002) is a notable example for maintaining a content based long-term user model for facilitating personalized online product retrieval.

If the work proposed in this thesis is placed within the categories shown in Figure 7.1, it will belong to the same category as “Dialog systems using models of the user” category. Therefore, among the systems described above, systems such as Adaptive Place Adviser (Thompson *et. al.*, 2002), and critique based systems described in (Pu *et. al.*, 2006; Viappiani *et. al.*, 2006; Reilly *et. al.*, 2007) can be considered as the closest to the thesis work. In the algorithm discussion (section 7.5) such work is compared to the performance of the proposed method.

7.1.2 Searching Methods

In the literature, there are two main retrieval methods suitable for catalogue navigation (Figure 7.2): filter based and similarity based. When searching for the user preferred attribute value in an item this is carried out by following two different techniques, either considering the similarity of the request to a given item or by matching parameter values. Filtering the content using matching parametric values is called filter based retrieval, whereas use of similarity metrics is called similarity based methods.

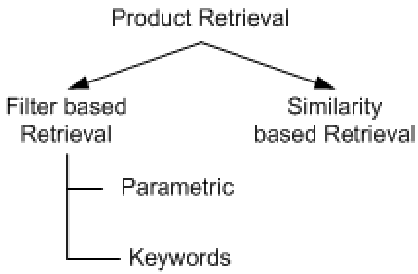

Figure 7.2 : Categorization of product retrieval methods

Filter Based Retrieval

Filtering the catalogue contents is carried out either based on keywords or by assigning values to product attributes as parameters.

In **keyword search**, the user provides the keywords to be searched within the item description as a query. The search engine retrieves and ranks the products according to the frequency of keyword occurrence within the item description. As mentioned in Burke (2002b) item descriptions are invariably short, most of the time less than hundred words or even less. A shorter description has less chance of containing a diversity of occurrences of the same term (e.g., "copy", "coping", "copies" etc). Due to this reason, in keyword search, terminological mismatch occurs when the user's language does not match with the vocabulary of the catalogue (Foskett, 1980). In addition, misspelling results in null retrieval. In certain search engines some of these faults are handled. For example, the DIESELPOINT[23] search engine offers misspelling correction and linguistic stemming (e.g., "copy" will also retrieve "coping", "copies" etc) to decrease the possibility of terminological mismatch. Keyword search in eBay is supported by suggesting similar terms (e.g., searching for "armchair" will suggest "arm chair" as a similar search). But still, keyword search either ends up with null retrieval or with a massive number of outcomes. To locate the correct keyword it takes time to test several different keywords in order to find the one that the user needs. And it is unlikely to retrieve all the outcomes using a single keyword. For example, in eBay, to retrieve "arm chair" needs at least two searches such as "armchair" and "arm chair". Since broadening the search space by using "chair" still misses "armchair".

[23] http;//www.dieselpoint.com

In **parametric search** the user query is represented as a set of constraints on item descriptions. Generally the user is asked to fill a form specifying values for product attributes. Then, a database query is formed where the query consists of features or feature ranges. Parametric search organizes the user query along the features and a vocabulary with which products are described, and do not encounter some of the problems seen in keyword search. But similar to keyword search, this also narrows down the search space according to set constraints, and therefore, has a higher chance of null retrieval.

Another drawback in parametric search is that its system defined parameters forces some of the similar items to go into two different searches. For example, if product price is set to a certain range, then the items which are slightly expensive are not included, unless the next range is included in the search, which increases the search space (Burke, 2002b).

Similarity Based Retrieval

Similarity based retrieval has its roots in case based reasoning (CBR). In CBR new problems are solved based on the solutions that were used to similar problems in the past. In the eCommerce scenario the available product descriptions represent the cases. User request is not considered as a strict set of constraints. The products are considered as cases and the user request is compared to items and ranks the outcome depending on how they score according to given *similarity metrics*. This results in less vocabulary mismatches and null retrievals. A similarity metric can be any function that takes two entities and returns a value reflecting their similarity with respect to a given goal. For example, in the restaurants domain, numeric attributes such as *cost* or *décor* have ranges of quantitative values, for which similarity can be easily used. For descriptive and qualitative attributes, forming and implementing a similarity metric is complicated, time consuming, and inefficient at runtime.

The case retrieval approach has performed well in the eCommerce context and several commercial case retrieval products are available (Burke, 2002b). It is clear that incorporating knowledge in the product retrieval as similarity matrices is the reason for success in CBR methods. Although similarity based retrieval is successful, building similarity matrices is a time consuming knowledge engineering task. As in any knowledge based system, the success of similarity based methods greatly depends on the quality of the similarity matrices formed. In addition, similarity computations during product retrievals slow down the entire process.

Entrée (Burke, 2000;2002a) is a critique based restaurant retrieval system. In Entrée matrices such as *niceness* and *quietness* of restaurants are computed combining several attributes. As an example, each restaurant gets a value computed for its *niceness*. At the selection process, this value is compared to a threshold value to determine if a given restaurant is a *nicer* restaurant. Features such as *cuisine* are even more complicated. According to Burke (2000) the Entrée system used a semantic network to present similarity of *cuisines*, which is a complex matching problem.

As explained in Burke (2001) for product retrieval two kinds of similarity matrices are required. For example, Entrée used *local* and *global* similarity metrics. The local similarity metric defines the similarity between two items while in the global similarity metric local similarity measures are combined in a priority ordering. For example, Entrée assigns maximum similarity rating to any two restaurants with the same price based on the local metric of *price*. The global similarity measure ranks the system selected features in priority order. For example, in Entrée, the global similarity metric is formed applying cuisine, price, quality and atmosphere in ranked order. According to Burke (2000) in the family of findMe systems, when forming global similarity matrices for different domains, the designer of the system selected the most important feature. For example, "cuisine" for restaurants, "grape variety" for wines and "genre" for movies were prioritized above the rest of the features. Therefore, when pre-defined similarity matrices are used, individuality is not reflected in the product retrieval. For example, the Entrée restaurant recommender, assumes for all users that the cost of a restaurant as second priority to cuisine. Moreover, similarity is goal based. For example, different users may request the same attribute value for two different needs. As explained in Burke (2000) to encounter different priorities of users, a knowledge-based system needs several global similarity measures. For example, PickAFlick (Burke *et. al.*, 1997) created multiple preference lists based on three different features, one based on the genre of the movie, another focusing on the actor, and lastly, a focus on the movie director. This highlights the fact that apart from the difficulties in designing and implementing similarity measures, it also introduces system designer's bias ideas into the system recommendation process. The next section discusses the issues encountered in the product retrieval process.

7.2 The purchasing process in eCommerce

As mentioned above there are massive numbers of items available for sale from electronic web sites. Users without proper product knowledge need personalized guidance in order to select the products and services that meet their needs within a reasonable time. To handle this issue, it is important to provide an environment similar to the bricks-and-motar stores where the consumer and a sales person communicate to find target items. According to Bergmann and Cunningham (2002), the entire process of product selection can be broken down into three main stages:

(i) Requirement elicitation
(ii) Product search
(iii) Product presentation

This maybe implemented as three independent processes or in tandem with each other. For example, product search may happen interleaved with requirements elicitation. It is also possible to use product presentation as a mechanism to guide the user in the requirement elicitation process.

7.2.1 Requirement Elicitation

In the current dynamic product markets user requirements change rapidly. A user requirement can be much different from the needs identified during the previous transaction. Therefore, it is important to elicit current user needs for each and every transaction. In a study about user attitudes towards eCommerce-websites by (Alpert *et. al.*, 2003), it has been shown that users prefer to have content filtering based on the information or criteria provided explicitly by the user for the current situation. Currently available systems employ different strategies to capture user requirements. Mainly this is carried out by asking questions from the user about preferred attributes by filling out forms, or by employing software agents (see (Jameson, 2001) variety of systems).

When eliciting user requirements, systems can guide the user by providing them with a form to fill out the preferences. Filling in the form is not very time consuming. However, product retrieval using the filled form, results in long lists of search results, or null retrievals. Users have to go back and adjust the initially filled form to include any new ideas or relaxed constraints. This forces the user to repeatedly provide the preference information. In the literature, such approaches are improved using interactive product retrieval strategies, which were described in the section 7.1.4. We believe that such approaches can be further enhanced by providing personalization by employing models of the users.

7.2.2 Product Search

Product search is an essential and crucial task of eCommerce sites. Product search is carried out based on the preferences expressed by the user. The product descriptions are stored as a relational database or as xml files with the site. The product collection or list is called the product catalogue. Hierarchically arranged product catalogues and unclassified product listings are used in websites such as eBay and Yahoo. Hierarchical product catalogues also play a role in product retrieval. If an unmanageable large set of results were retrieved after a search query, users can narrow down the search space by querying within a lower level category of the hierarchy. However, an understanding of the vocabulary used by the catalogue is required to correctly identify the suitable product category. For example, in eBay, if an antique furniture item is listed only under *antiques*, then a user browsing under *furniture* will never come across it.

As previously discussed in section 7.1.2, the main searching methods used in the literature are either filter based or similarity based.

7.2.3 Product Presentation

Finally the selected products should be presented to the user. As stated before, the product presentation cannot be often separated from the other two processes: requirement elicitation and product retrieval. In Bergmann and Cunningham (2002) there are three issues discussed related to item presentation. They are: choosing an appropriate presentation form, presenting appropriate amount of information and enabling high interaction speed.

An item needs to be described appropriately since the user does not physically see it. Depending on the item characteristics, different presentation forms may suit. For example, text, pictures, sound or videos maybe used. The ideal amount of information should be to present exactly what the user asked and not to give anything that the user already possesses. Since different customers may have different information needs, the product descriptions can be presented using a customized display. For example, SETA (Ardissono *et. al.*, 2001b) uses a user model to display only the features that are interesting to the user. Another useful feature of presenting items would be ranking or sorting the selected items in order of importance to the user.

In online product retrieval, products may present to the user several times before a satisfactory product is found. For example, in critique based systems, initially the user is presented with a ranked list of preferences from highest suitable to the lowest, which is selected based on the initial user input and similarity measures. Then, the user is allowed to provide "tweaks" to guide the system to further filter out unwanted selections. In the Adaptive Place Advisor (Thompson *et. al.*, 2002), which is a conversational recommender system, a similar personalized approach is taken.

In addition to choosing an appropriate presentation form and presenting an appropriate amount of information, interaction speed of the system needs attention. If the user has to be idle while the system searches for his/her needs, users get bored. It is important to maintain a balance between keeping the user involved and at the same time being unobtrusive. Long database access delays and similarity calculations for each item maybe time consuming and make the user wait.

7.2.4 Problems Encountered

As highlighted in the discussion on the three phases, retrieving the desired products out of a large product catalogue is a tedious task. A search query may fail due to one or more of the following.

- The query ends up with an unmanageable long list of outcomes
- The query ends up in null retrieval
- The final recommendations or the list of products retrieved are unsatisfactory

As explained in previous sections, unacceptable (zero or massive) number of retrievals may occur due to several reasons. In such situations there should be a way of constraining the unconstrained criteria or adjust constrained criteria to retrieve a manageable set of records. Null retrievals may occur due to the user trying to specify all constraints at the start-up or due to vocabulary mismatches. As a solution, users can be allowed to provide partial queries where user specifies only the most important features at the start-up. Then, the system should be able to guide the user towards target items by asking questions to further constrain the search gradually. This way the user is able to monitor the size of the result set and determine the point where to stop further constraining. Furthermore, if null retrievals occur the search method should be able to carry out strategies such as using its knowledge about the products distribution to use alternative (but similar) values for user constraints.

To avoid vocabulary mismatches, the system can provide the user with an interface where the user is allowed to select options out of a given set, instead of open ended questions. Such an interface will familiarize the user with the system vocabulary and thereby minimize the vocabulary mismatches. However, these approaches can lead to too many questions and annoy the user. Therefore, it is important pay attention to system obtrusiveness.

The problem of unsatisfactory recommendations can be improved by personalizing the search. However, personalization requires further information about the user. Obtaining user information by explicit methods improves the accuracy of data required for personalized product retrievals. Unfortunately, such queries for explicit user inputs will add to the system's intrusiveness.

Therefore, it is important that the system maintains a balance between the two issues: keeping the user involved in the selection process and at the same time not be too obtrusive. The next section provides a comprehensive definition of system obtrusiveness.

7.3 A new technique for personalized product retrieval

In this section we introduce a new algorithm: Personalized Interactive Product Retrieval Process (*PIPRP*) algorithm, which utilizes the *LUM* as described in earlier chapters. In the prior sections, several key product-retrieval techniques were described and the main problems and limitations were highlighted. Due to functionalities such as long-term acquisition of information, ability of updating the model of the user, layered separation of information and domain focus in the *LUM* it has been possible to successfully address several key problems in online product retrieval. The problems addressed in online product retrieval were as follows:

- The problem of null retrieval
- The problem of retrieving unmanageable number of items
- The problem of retrieving unsatisfactory items

In solving the above problems, the *PIPRP* combined with the *LUM* provides personalized services during all three phases of eCommerce activity: personalized requirement elicitation, personalized product search and personalized product presentation. The rest of this section describes the new algorithm based on the *LUM*.

7.3.1 *PIPRP* - Algorithm

The *PIPRP* is given below as an algorithm consisting of a sequence of six steps. The process is initiated by the user providing the initial query. We assume there is a *LUM* exist for the interacting user.

Step 1: Obtain the most important attribute values for the current query (user input).

Step 2: Expand the above query based on the preference in the *DI* layer and retrieve products from the database using the expanded query. Return the retrieved products set. If this resulted in null retrieval, prompt the user for relaxed constraints.

Step 3: Filter the product set using the *personal information related attributes* (PIR-attributes). According to the definition in Chapter 4, section 4.5.3, PIR-attributes are the attributes in the *IM*. If this resulted in null, relax the filtering by removing the least relevant attributes.

Step 4: Select all the non PIR-attributes in the *DI* layer and form questions based on their features. Filter out the unwanted records according to the user preferences.

Step 5: Calculate the similarity (as described in section 7.4.2 below) between the query and the remaining set of items and produce a list of the items in descending order of similarity.

Step 6: Display the results as recommendations starting from the item in the top of the list.

Step 2 of the algorithm retrieves items satisfying the constraints provided in Step 1, and adds more items through query expansion. Step 3, filter out any items that do not match with the user model. However, only the constraints that are not specified in Step 1 are used for personalized filtering in Step 3.

Therefore, as shown in Figure 7.3 the (results of Step1) $\cap$ (results of Step 2) $\cap$ (results of Step3) = ϕ. The records resulting after Step 3 is maintained as a record set, which is used in the proceeding steps. The question asking process do not filter out the results in the Step 3 record set but just perform selections on them. Therefore, in the event of unsatisfactory outcomes or if user preferences change, this allows the user to track back and perform different selection criteria on the same records.

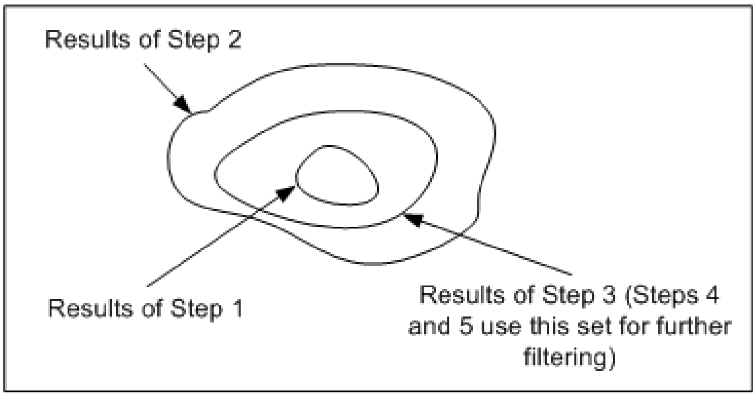

Figure7.3 : Sets of records resulting from each step

Figure 7.4 presents the above algorithm as a flowchart, showing each main step in bold. As shown, after completion of each step, the user gets the opportunity to view the results if required (Step 6). The items that best match the user query are displayed as the result set.

Therefore, if the user requests to view the results, a list of matching items is presented in the order of similarity to the user query. Similarity calculation is described next.

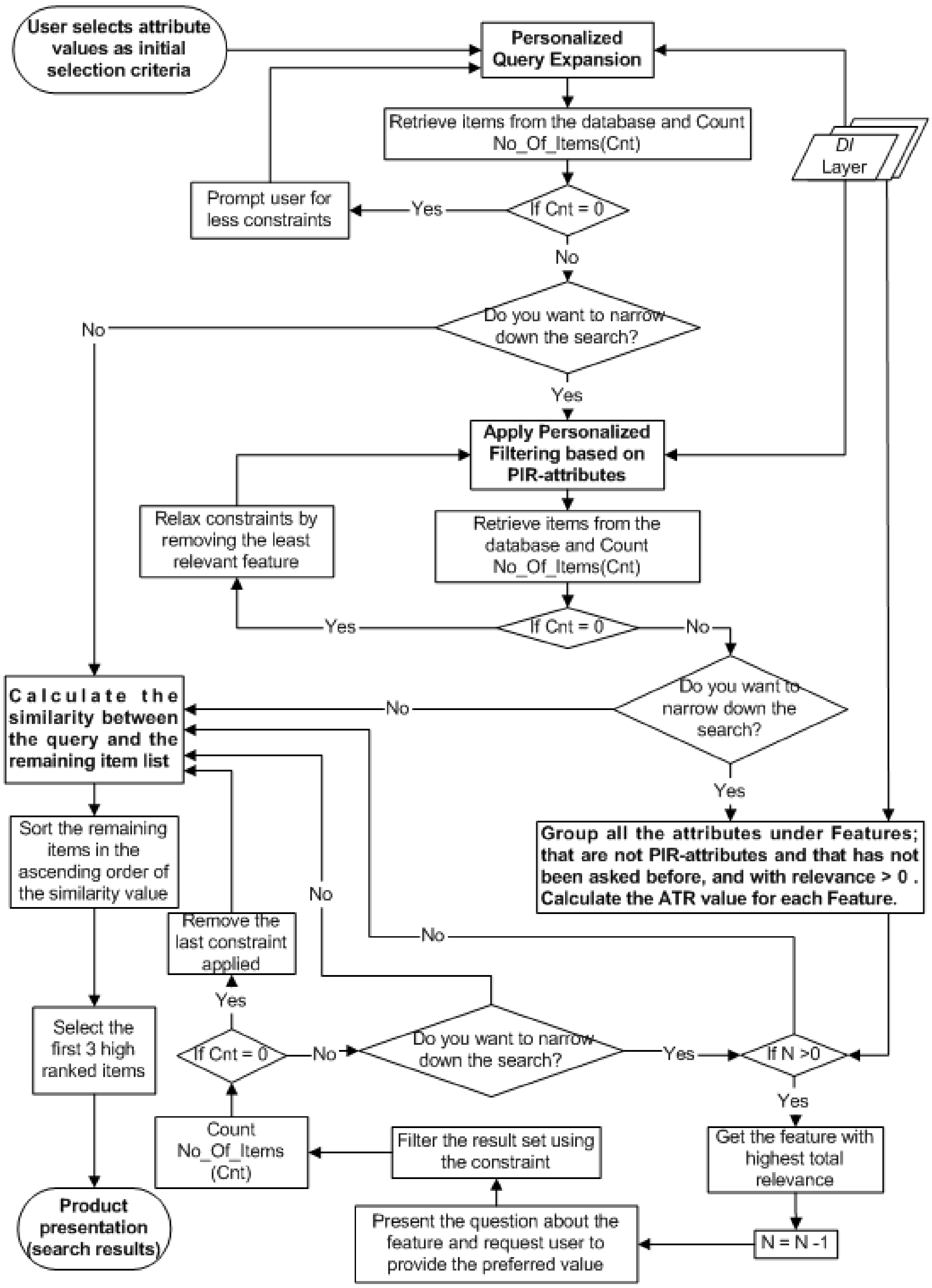

Figure 7.4 : Product selection algorithm as a flow chart

7.3.2 Similarity Calculation

This section describes the similarity calculation used to identify the closest matching items to the query. The requirement of a search is to identify and retrieve the closest matching items to the user requirement. In *PIPRP* user requirement is obtained using both the user model and the initial query. The product retrieval process begins by the user providing an initial query as a partial query, by specifying the most important features for the current search. During the search process, the partial query is refined using the user model and further questioning of the user. The refined query is used in Step 6 for results retrieval. Since being refined, this query more clearly expresses the current user requirement. Therefore, to further reduce the remaining item list, the similarity of each item to the user query (resulted in Step 6) is considered. The items with the closest similarity to the query are presented to the user as the most suitable outcome.

The items in the product base are represented as an array of attributes with values 0, or 1, indicating the existence (1) or absence (0) of each attribute. The initial query can consist of a few key preferences as attribute values. The user preferences (in the query) will also be in the form of (0, 1) for each attribute with '1' indicating preference. The user model contains a 'relevance value' for each attribute (based on the past behavior etc). The technique introduces an additional relevance value to capture the importance of the attribute for the current transaction. The similarity of an item to the user query is calculated based on the total relevance value of the attributes present in the item. Therefore, each attribute present in the item, contribute a sum of two relevance values to the total similarity.

The two relevance values contributing to the total similarity are:

(i) The importance of the attribute to current query.

a. If the current attribute is among the constraints provided with the initial query (Sim_{iq}) - assign a relevance value of r_1.

b. If the current attribute is among the user provided constraints during the product selection process – provided during step 4 (Sim_{psp}) – assign a relevance value of r_2.

Here ($r_1 > r_2$), therefore, in the implementation r_1 and r_2 were assigned as 1 and 0.75.

(ii) If the current attribute has a relevance value in the *DI* layer of the user model (Sim_{um}) – read the actual relevance value from the corresponding *DI* layer

The total similarity of an item *I* ($I_{Sim_{Tot}}$) which is described using *n* attributes to the user query is given by;

$$I_{Sim_{Tot}} = \sum_{i=1}^{n} \left(\mathrm{Sim}_{iq}(i) + \mathrm{Sim}_{um}(i) + \mathrm{Sim}_{psp}(i)\right) \text{ ------------ } \quad (7.1)$$

Using the formula (7.1), $I_{Sim_{Tot}}$ is calculated for each item remaining in the result set. The algorithm is as follows.

```
loop:  for each item in []item_list
            item = get_item([]item_list);
            loop:for each attribute in item
                        current_attri = get_attribute(item);
                        Sim_um  =  readRel_DILayer(current_attri);
                                    If InitQeryAttri(current_attri) == true
                                                Sim_init= r_1;          // r_1> r_2>0
                                    Else   Sim_init= 0;
                                    End-if
                                    If PSPAttri(current_attri) == true
                                                Sim_psp =  r_2;     // r_1> r_2>0
                                    Else   Sim_psp =  0;
                                    End-if
                        I_SimTot = I_SimTot  +  Sim_init (i)+  Sim_um(i)+  Sim_psp(i);  //total relevance of the item
            end-for
```

Then, the result set is sorted in the descending order of the similarity value. Finally, the resulting items are displayed to the user starting from the items with highest similarity/importance.

7.3.3 Example of Similarity Calculation

A query Q_u and four items to compare are presented in Figure 7.5. The query Q_u is the *complete user query*[24] of user U, in a domain D. For simplicity, assume that domain D has only ten attributes. I_A, I_B, I_C, and I_D, are four items each described using 10 binary-values to indicate the presence or absence of the ten attributes. According to the query Q_u, the user shows interest in attributes a1, a3, a5, a11 and a15.

	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15
Q_u =	1	0	1	0	1	0	0	0	0	0	1	0	0	0	1
I_A =	1	0	1	0	1	1	0	1	0	0	1	0	0	0	1
I_B =	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0
I_C =	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1
I_D =	0	0	0	1	0	0	0	1	0	0	0	1	1	1	0

Figure 7.5 : Comparison of four different items IA, IB, IC, and ID to a query Qu

The query Q_u is formed starting from the initial specification and then through the steps 2, 3, and 4 of the algorithm. U requested attributes a1, a3 and a11 (circled) when specifying the initial query (algorithm Step1) and a5 and a15 (underlined) were provided as preferences during Step 4. The rest of the attributes that appear in the items have various relevance values in U's *DI* layer for domain D. Assume the relevance values in the *DI* layer of U is as shown in Table 7.1.

Table 7.1 : Relevance values of attributes from U's *DI* layer

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
0.69	0	0.42	0	0.18	0.81	0.46	0.2	0.02	0	0	0.08	0.24	0	0.27

Table 7.2, summarizes the details of the ten attributes of items I_A-I_D. The total relevance of each item is calculated using equation (7.1).

For example, item I_A has seven attributes out of all the ten attributes in the domain. According to Q_u, a1, a3 and a11 were requested in the initial query (user's most important attributes). Item I_A has all three of them present. I_A also has both attributes the user showed

[24] Here "complete user query" is the query resulted at the end of the question asking process. This query consists of user's initially specified preferences, the preferences obtained as answers to the system questions and the preferences in the user model.

an interest in during Step 4. In addition, there are extra attributes. Although those two attributes were not explicitly mentioned in the current query, they show a positive relevance value in the user model. When the total similarity to the query is calculated, I_A shows the highest similarity.

Table 7.2 : Summary of attributes and their calculated relevance using the formula 7.1

Item	**Initial Query**	**During Retrieval**	**Other**	**Total Similarity=** $\sum_{i=1}^{n}(Sim_{iq}(i)+Sim_{um}(i)+Sim_{psp}(i))$
I_A	a_1,a_3, a_{11}	a_5,a_{15}	a_6, a_8	(1+0.69+0)+(1+0.42+0)+(1+0+0)+(0+0.18+0.75)+(0+ 0.27+0.75)+(0+0.81+0)+(0+ 0.2+0) = 6.7
I_B	a_3	a_5	a_8	(1+0.42+0)+(0+0.18+0.75)+ (0+ 0.2+0) = 2.6
I_C	a_1	a_{15}	a_6	(1+0.69+0)+ (0+0.27+0.75)+(0+0.81+0) = 3.52
I_D	none	none	a_4, a_8, a_{12}, a_{13}, a_{14}	(0+0+0)+(0+0.2+0)+(0+0.08+0)+(0+0.24+0)+(0+0+0) = 0.5

I_C with one attribute from the initial query and then one from the selection process and two highly relevant attributes to the user, scores the next highest similarity. I_B, is similar to the item I_C with regard to the number of attributes present from each of the three types: from initial query, during retrieval and other. For example, I_B has a3 from initial query whereas I_C, has a1; and I_B has a5 during Step 4, whereas I_C has a15; they both have one more additional attribute present in them. However, the relevance values of the attributes present in I_B are lower than of item I_C.

If the relevance values from the user model are not considered, both items I_B and I_C ends up in identical similarities, which is not true. Usually, when calculating similarity measures, vector based methods such as cosine similarity[25] or Pearson correlation[26] are required to avoid such errors. Due to the use of relevance values from the user model, our approach of similarity calculation does not require such methods.

7.4 Experimentation

As mentioned in Chapter 6, section 6.1, the work carried out does not have the resources to demonstrate the combined performance of the user model and the product selection process. Therefore, to demonstrate the outcomes, a scenario based approach is taken. In this section, first the experimental plan and the scenarios used are discussed. Then, for a

[25] Cosine similarity Wikipedia, URL: http://en.wikipedia.org/wiki/Jaccard_index

[26] Pearson Correlation Wikipedia URL: http://en.wikipedia.org/wiki/Correlation#Pearson.27s_product-moment_coefficient

selected scenario, the algorithm steps are demonstrated showing outcomes of each step. After each step, the performance of the related work is compared. Finally, the outcomes of the above demonstration are compared and discussed according to the experimental plan.

7.4.1 Experimental Plan

To demonstrate the strength of the model, some form of testing is favorable. Therefore, we decided to compare the performance of eHermes PERSONAL with the performance of filter based parametric search (see section 7.1.2 for details on parametric search). The reason for selecting parametric search is, even though the retrievals are personalized (using the *LUM*), the underlying searching method in *PIPRP* is parametric search. Therefore, to compare the quality of personalized retrievals of the model, simple parametric search is suitable. Furthermore, implementation of parametric search is no additional work, since it is easily usable for the same scenarios.

In the following sections three scenarios are selected to compare the performance of the two methods; parametric search and *PIPRP*. Sections starting from 7.4.3, presents the demonstration of *PIPRP* steps using one of the scenarios. Section 7.4.8 presents the results of the demonstrated scenario and the other scenarios. Section 7.4.9 compares the outcomes of the two approaches.

Five restaurant search scenarios were selected to demonstrate the performance of the prototype system. In Table 7.3, the scenarios are presented with an Id and a description.

Table 7.3 : Possible scenarios related to restaurant search

Scenario Id	Description
a	To have lunch during work break.
b	To have dinner with the family at the weekend.
c	To meet a colleague over lunch for a business talk.
d	To organise a gathering of friends to celebrate a special occasion.
e	To meet a personal friend over dinner.

Since the experimentation is conducted using synthetic data, scenarios were constructed for only one user (User41). Information in *PI* layer of User41 is given in Table 7.4.

The real life scenarios constructed for User41 is given in Table 7.5. Since User41 is a bachelor, the scenario ‘b’ was not created (as the family type is not known). Due to his employment, scenario ‘c’ was considered as trivial.

In all three scenarios constructed, the user is concerned about the price. However, realistically he may have other preferences that are not apparent from only the demographic information (such as interest in food, and a desire to taste different culinary disciplines).

Table 7.4 : PI layer information belonging to User41

Age	26
Family	Single/Bachelor
Gender	M
Income	30K - 50K
Occupation	Trade person or related
Work Hours	21-40 hrs
Adventurer	0.91
Family Person	0.05
Fun	0.67
Health Conscious	0.21
Price Sensitive	0.53
Quality Conscious	0.29
Socializing	0.35
Time Saver	0.42

Table 7.5 : Possible Scenarios of interactions for user 41

User Id	Scenario Id	Description
User 41	**41-a1**	The young bachelor with lower income searching a restaurant to have lunch during the work break. He is interested in food, and would like to taste different culinary disciplines for an affordable price.
	41-d1	To organize a gathering of young bachelor friends to celebrate his new promotion over dinner and a drink. Price is still a concern but more attention to the friendly atmosphere and distance.
	41-e1	To have dinner over a chat. Price is a concern as paying for both but the atmosphere and the quality of the restaurant should be adequately good. At the back of his mind plans for watching a movie with the friend before the meal. Better if parking available.

The initial *DI* layer for restaurants is created (based on the *PI* layer of User41) to be used in the first run of the *PIPRP*.

7.4.2 Demonstration of Algorithm Steps

In this section, the algorithm steps are demonstrated explaining the screens and results. Each algorithm step is compared with approaches followed in related work. For comparison, we selected three closely related online product retrieval systems: Entrée (Burke, 2002a), Adaptive Place Advisor (Thompson *et. al.*, 2002), and Smart client (Viappiani *et. al.*, 2006). The *PIPRP* is initiated using the main menu of eHermes PERSONAL shown in Chapter 4, section 4.2, Figure 4.5. Then, as described in section 7.5.1, the algorithm steps are carried out.

7.4.3 Step 1: Personalized Initial Query Specification

The initial preference elicitation process provides the user with a flexible interface where the user is allowed to start with an incomplete query. The later steps further refine the query adding or removing constraints.

As described in Chapter 6, section 6.1.1, the products are hierarchically categorized under features such as 'Cuisine' and within a feature sub-features such as 'Asian'. There can be several layers within a feature. For example, sub-features, sub-sub-features and so on until the final layer consists of atomic attributes. The user is able to specify the preferred features, sub-features or attributes. This is done using an interface as shown in Figure 7.6. Since the system is supported with a product categorization, this kind of option selection strategy is possible. In case people find it difficult to articulate what they need, this interface will become handy to provide guidance to declare preferences.

The interface provides the following advantages.

(i) The user has the opportunity to reveal the most important preference in the initial query.

Each time a user needs to browse for a suitable item, it can be for a different reason. As given in scenarios, if looking for a restaurant, once it could be for a family outing and at other times it could be for a meeting over lunch with a colleague. Depending on the occasion the most important attributes may change. For example, User41 is more concerned about price in scenario **a**, but he pays attention to other aspects when friends are present (in scenarios **d** and **e**). He may need to impress them. Therefore, although previous preferences are important, it is necessary to capture the current requirement of the user.

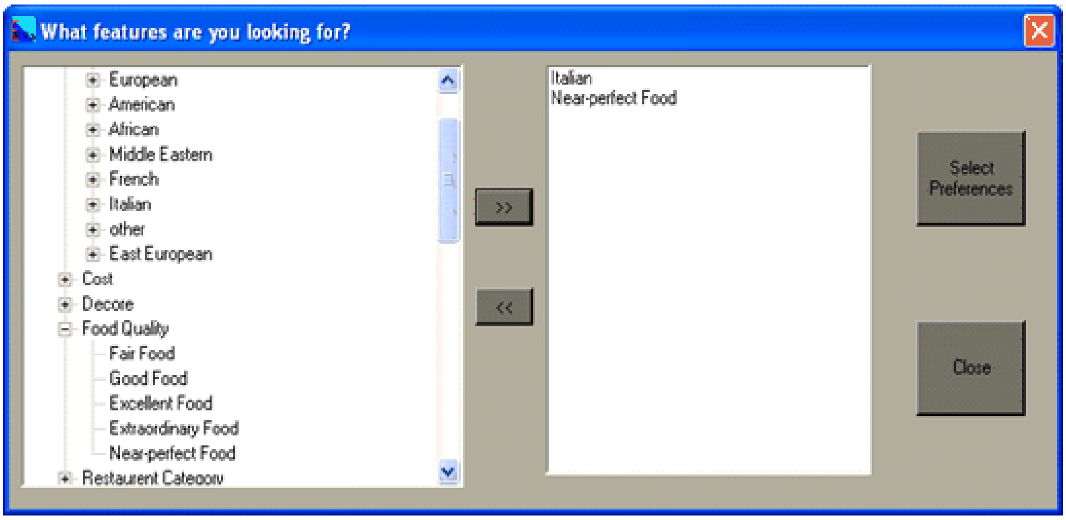

Figure 7.6 : Screen – Users initial query selection

(ii) Hierarchical product presentation allows easy specification of the query and provides an idea about the available product characteristics and the vocabulary.

If the user is happy to accept a restaurant offering any type of Italian food, he/she can select the subcategory 'Italian'. But if the constraint is more specific (say looking for 'Italian (Southern)'), then a more definite constraint under the 'Italian' category is selected. This interface supports such a layered set of categories from general to more specific.

(iii) The user is allowed to select more than one value for a given feature.

If the user prefers more than one attribute value for a given feature, specifying such additional attributes are possible. For example, if the user preferred both 'Italian' and 'Indian' cuisine, then both are considered equally important. Furthermore, the algorithm allows the user to:

(iv) Select any important features

(v) Provide any preferred attribute value for the above selected feature

The interface allows selection of any feature that is important to the user and then to provide a constraint on it. For example, if the user is interested in (even rather insignificant feature such as) 'opening hours', still it can be constrained using the preferred attribute value (such as 'dinner after theatre is preferred'). Comparison with other approaches (with regards to Step 1) is given in Table 7.6.

Figure 7.7, shows the interactive screen used in prototype implementation. To facilitate the comparisons and to monitor the outcomes after each step the algorithm steps were assigned to single procedures. For example, the first button in Figure 7.7 "Results-Initial Query", use the initial user inputs as constraints to do a parametric search in the database. This does not use the *LUM*, and hence is not personalized. The results are used to compare with the final personalized outcome of the algorithm.

The other details of the screen in Figure 7.7 are as follows: "Results- Expanded Initial Query" button corresponds to the Step 2 of the algorithm. Step 3 of the algorithm is executed using "Results- Personalized Filter" button. Finally, "Further Filtering" button, executes the Step 4 and 5. The grid box shows the retrieved restaurants after executing each procedure. "Cancel" button terminates the entire selection process.

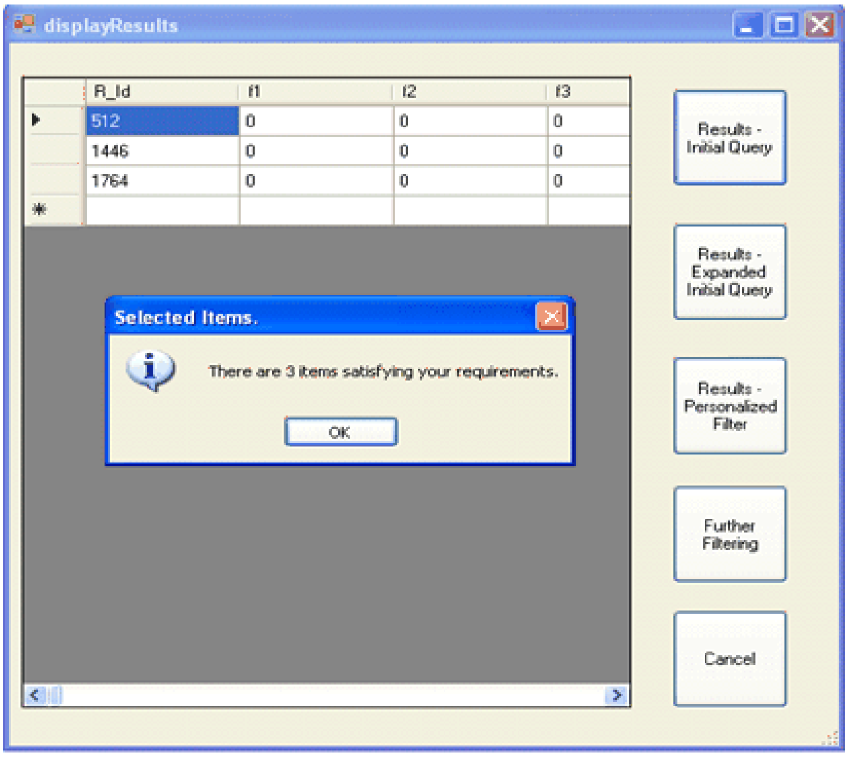

Figure 7.7 : Screen – Interactive product selection

Once the preferred values for the initial query are selected, search is invoked by pressing the "Results-Initial Query" button. The constraints provided in query selection screen (Figure 7.6) are used to filter data using parametric filtering. The results are displayed in the grid showing the restaurant Id and the features present. In scenario 41-a1, two constraints are selected as shown (Figure 7.6); "Italian cuisine" and "Near perfect food". 'Italian' is a category of 'Cuisine' and it has six subcategories. Choosing 'Italian' include all six attributes in the query. 'Near perfect food' is a single attribute. If these two constraints are used in parametric search, only the restaurants offering 'Near perfect food' of Italian culinary discipline will be selected. The screen resulted from parametric search is given in Figure 7.7. *Query_Step1* below shows the Step 1 query generated for User41 (scenario 41-a1).

Table 7.6 : Comparison with existing systems with respect to Step 1 of the algorithm

PIPRP	**Adaptive Place Advisor (Thompson *et. al.*, 2002)**	**Entrée (Burke, 2002a)**	**SmartClient (Viappiani *et. al.*, 2006)**
The user gets to reveal the most important preference in the initial query.	Yes. (The user provides the initial query during a dialog with the system. Preferences on item attributes; as answers to system questions, or items as known examples)	Yes. (As preferred attributes or as an example)	Yes. (As preferred attributes or as an example)
Interface allows an understanding of the system-vocabulary.	No. (The system gives some guidance which is very primitive)	Yes. (Preferred option selection from the system provided answers)	Yes. (Preferred option selection from the system provided answers)
In the initial query, user is allowed to select more than one value for a given feature.	Yes.	Yes. (But only if the user provides a known example item as the preference)	Yes. (But only if the user provides a known example item as the preference)
User is allowed to select any important feature and provide any preferred attribute value for the above selected feature	No. (User is allowed to constrain a few pre-selected features)	No. (The user is allowed to constrain only system-selected features. Such features are ordered in importance, according to a system-introduced global similarity measure. In addition, Entrée allows the user to specify a known example as the requirement)	Yes. (Provide users with an interface to select both desired features and attributes. However, there are only a limited number of features available. Importance among features are not considered)

***Query_Step1* - "*Search for the items offering CUISINE type [‘Italian (North & South)’ OR ‘Italian (North)’ OR ‘Italian (Northern)’ OR ‘Italian (Southern)’ OR ‘Italian’ OR ‘Italian Nuova Cucina’] AND with FOOD QUALITY [‘Near-perfect Food‘]”*

If such strict parametric selection is carried out, the restaurants that are closer to the user’s expectations may not be selected. For example, it excludes slightly lower and upper ranges of the attributes, which the user may accept. To avoid such loss, personalized query expansion is employed.

7.4.4 Step 2: Personalized Query Expansion

In personalized query expansion the user entered constraints are not used as is. Instead the profile contents are utilized to relax the given constraints. By relaxing the constraints, items with similar attributes also get selected by the query. For example, User41’s request for “Near perfect food” excludes the restaurants that offer slightly upper or lower ranges of food quality from the query. As a result of personalized query expansion, if there are any other attributes (in the user model) that belongs to the same feature (as the selected one and has positive relevance values), such attributes are also included in the query. Figure 7.8 demonstrates the query expansion algorithm.

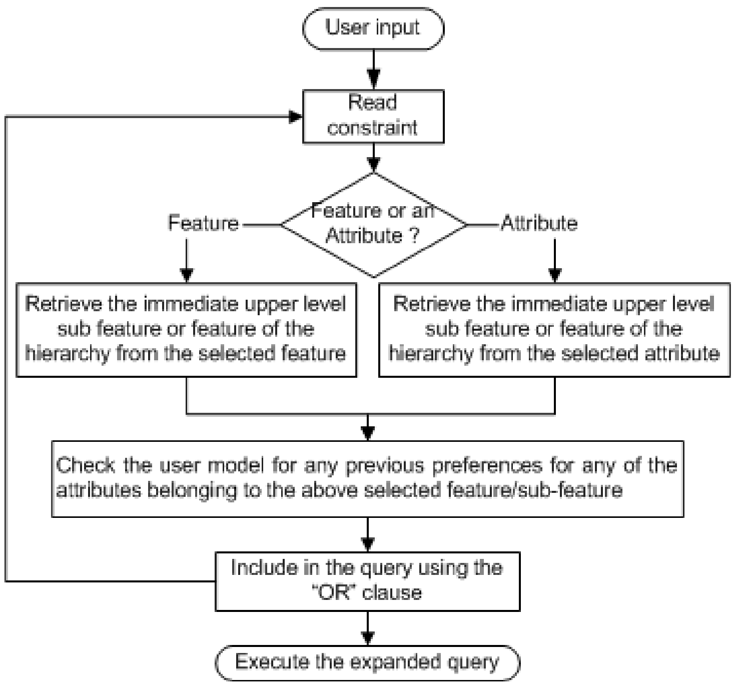

Figure 7.8 : Steps involved in personalized query expansion

In the literature, to include the items with similar characteristics (in the query results), similarity based methods have been used. Similar items to the current user request are identified and included in the result set using the following methods:

(i) Use of similarity matrices to identify the items with similar attributes as in Entrée (Burke, 2001).

(ii) Inclusion of items similar to previous user preference; item-item correlation as in Adaptive Place Advisor (Thompson *et. al.,* 2002).

(iii) Inclusion of items preferred by like-minded users; user-user correlation as in collaborative filtering (Resnick *et. al.*, 1994).

However, as explained in Burke (2001) two items maybe similar due to several different reasons. For example, in the 41-a1 scenario, User41 is described as a person interested in food and different culinary disciplines. He may have selected 'Italian' cuisine due to reasons such as, he has never experienced 'Italian' food; he has experienced Italian food before and like the Italian flavor, or maybe he prefers a certain Italian dish. However, the underlying reason is not visible to the selection algorithm. Therefore, *PIPRP* do not look for *similar* items, but instead uses the previous preferences in the user model. For User41, if there were other types of cuisine he preferred in the past, such cuisines also get included in the query. However, being the initial interaction with the system there are no preferences for cuisine in the User41's user model.

For a quantitative attribute such as "Near perfect food" if a single constraint is provided, the restaurants in the slightly upper or lower ranges will not get a chance to be included in the results set. Although the user has selected only 'Near perfect food', the user model shows high relevance values for both 'Fair Food' and 'Good Food'. Therefore, as a result of the personalized query expansion, the query is expanded using all additional attribute values in the user model as follows.

Query_Step2 - "*Search for the items that offer CUISINE type ['Italian (North & South)' OR 'Italian (North)' OR 'Italian (Northern)' OR 'Italian (Southern)' OR 'Italian' OR 'Italian Nuova Cucina'] AND with FOOD QUALITY ['Near-perfect Food' OR 'Fair Food' OR 'Good Food']"*

A preference for these additional attributes in the user model results due to the initial impact of the user *PBC* values or implicit user preferences during product selections. For example, if the *PBC* values describe the user as a less price sensitive person, then the higher cost range is highly relevant to that user. Or else if the user explicitly expresses preference for higher cost ranges during previous interactions, then again the user model indicates high relevance for higher cost. If the user selects the lower cost range as preferred value, still the higher range gets included in the personalized query expansion.

Since the query is expanded using the user model, in addition to the items satisfying query constraints, all the items relevant to the user are included in the result set. This item set is used in the Steps 3, 4 and 5 of the algorithm to obtain the most suitable results. The results of the expanded query are shown in Figure 7.9. Due to the query expansion, this step of the algorithm always retrieves a greater number of records than the previous step; hence, it never results in null retrieval.

Related work is compared with respect to Step 2 of the algorithm in Table 7.7.

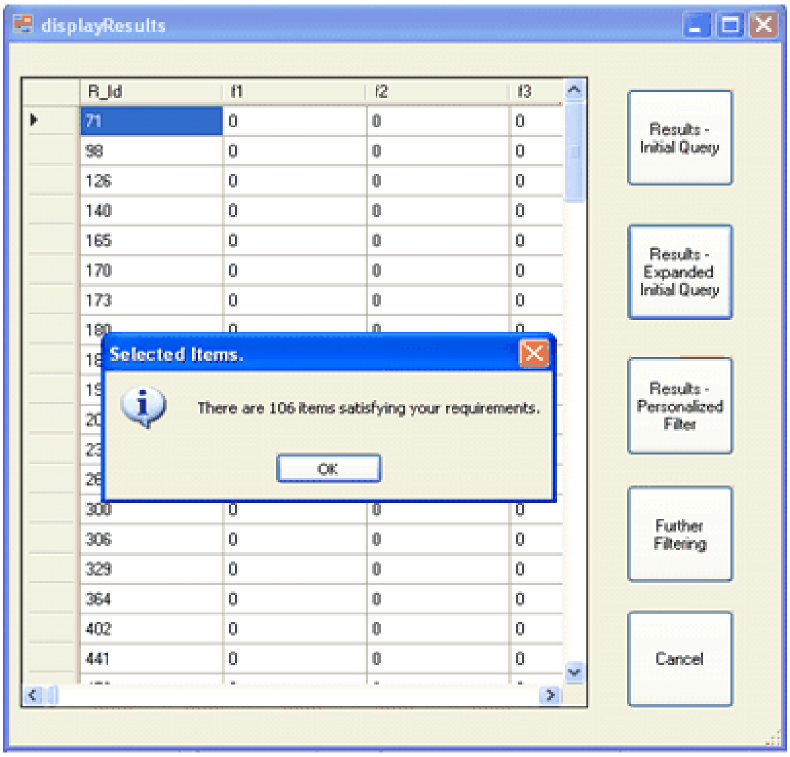

Figure 7.9 : Records retrieved using the expanded query

Table 7.7 : Comparison with existing systems with respect to Step 2 of the algorithm

PIPRP	**Adaptive Place Advisor (Thompson *et. al.*, 2002)**	**Entrée (Burke, 2002a)**	**SmartClient (Viappiani *et. al.*, 2006)**
Partial fulfillment of the items considered	No. (But, Indirectly consider the items that were not retrieved by the initial query. All the items that were not retrieved by the initial query are ranked according to their *similarity* to the previous user preferences. If no matching items resulted from the initial query, system queries the user based on the features of the highest similarity ranking item)	Yes. (Query expansion is carried out according to local similarity matrices. For example, according to the 'price' local similarity measure, the immediate lower price tab receives the highest score while the immediate higher price tab gets a lesser similarity score)	Yes. (Form penalty functions based on the initial query. If the penalty is less, even the items that are not exactly matching to the initial query are displayed. However, the penalty functions do not consider similarity between attributes. The similarity of an item to the query is determined based on the number of attributes that fulfils the penalty function and the number of attributes that do not comply with the penalty function)
Personalization provided	Yes. (Since Questions are formed based on item similarity to the users past preferences, personalization is provided. However, all the users have the same start-up user model)	No. (The local similarity metric is common to all users)	No. (No user model is involved. The penalty functions are common for all users)

If the user opted to further filter the records, *personalized filtering* is carried out. The records are filtered using the PIR-attributes. Such filtering further reduces the results set by removing any items that do not match the user's personal preference.

7.4.5 Step 3: Further Filtering Based on the User Model – Personalized Filtering

To further filter out the unwanted results, there are two important facts to consider.

- What is the next most important feature of the item to ask from the user?
- What is the preferred value for that feature?

To control the obtrusiveness, the facts discussed in the section 7.3 are important. In the literature, the information gain of a question was utilized to unobtrusively determine the next best attribute to ask (Doyle and Cunningham, 2000). This method only focused on reducing the number of questions. Although less number of questions were asked, if the user is asked to provide his/her preference for unimportant features, one of the following can happen.

- If the user answered the question and the records were filtered depending on the answer given, important records may be lost.
- If the user did not answer the question, then asking an unimportant question adds to the obtrusiveness of the system.

For instance, say at a certain point of filtering the feature ‘wheelchair access facility’ (in restaurant domain) was requested from the user. Even though it is of least interest, the user may answer the question positively. This leads to removal of all the restaurants without ‘wheelchair access facility’. Among the filtered restaurants, there maybe ones that offers more desired features, although not offering the wheelchair facility. Therefore, filtering records based on un-important features leads to low quality recommendations.

The PIR-attributes in the user model are considered as important to the user (since they are related to user’s personal information based preferences). However, due to the personal nature of PIR-attributes, (to maintain unobtrusiveness) *PIPRP* never queries about such attributes. (Note – Since user preferences for PIR-attributes are never requested, the user is able to declare preferences for such attributes only in the initial query). Therefore, instead of asking the user for the preferred values of these PIR-attributes, the personalized filtering is carried out using the existing values of PIR-attributes in the corresponding *DI* layer of the user model. Since only the relevant PIR-attributes are to be considered as preferred or applicable to the user, ones that are with a greater than zero relevance are used in the filtering.

In scenario 41-a1, User41 interacts with the system for the first time. Therefore, only a few PIR attributes has greater than zero relevance values. When personalized filtering is applied the resulting query is for scenario 41-a1 is as below.

Query_Step3 - "*Search for the items with CUISINE type [‘Italian(North & South)’ OR ‘Italian(North)’ OR ‘Italian(Northern)’ OR ‘Italian (Southern)’ OR ‘Italian’ OR ‘Italian Nuova Cucina’] AND with FOOD QUALITY [‘Near-perfect Food’ OR ‘Fair Food’ OR ‘Good Food’] AND the DÉCOR is from poor to good AND the SERVICE is from fair to good”*

As shown in Figure 7.10 the outcome of Step 2 (106) items was reduced to 85 items.

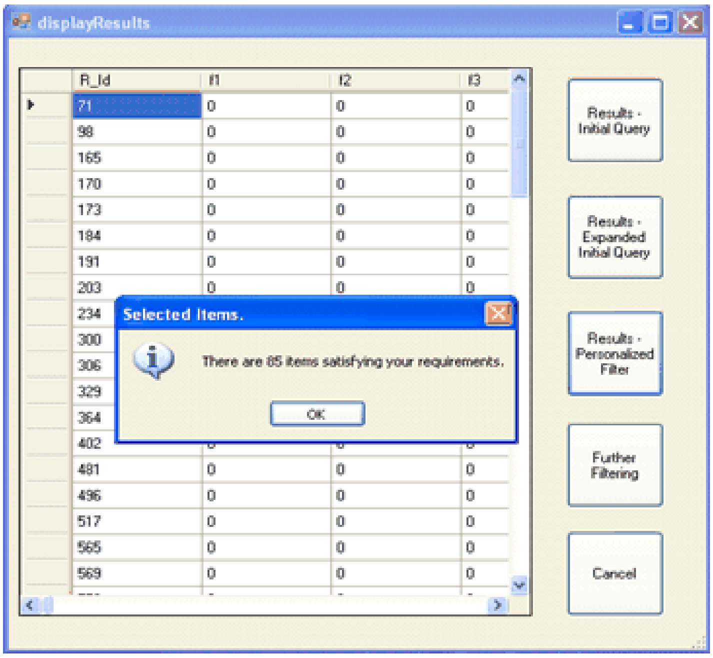

Figure7.10 : Records available after personalized filtering

After filtering, if still an unmanageable number of records are available, *PIPRP* prompts the user asking for further filtering. The user is allowed either to browse through the available result set or to provide constraints for further filtering. Related work is compared with respect to Step 3 of the algorithm in Table 7.8.

Table 7.8 : Comparison with existing systems with respect to Step 3 of the algorithm

PIPRP	**Adaptive Place Advisor (Thompson *et. al.*, 2002)**	**Entrée (Burke, 2002a)**	**SmartClient (Viappiani *et. al.*, 2006)**
Provide personalized recommendations from the initial step	No. (At the start, all users has the same user model) Yes. (Once the user model is updated later)	No. (Since no user model is used, user has to provide all required information each time using the system)	No. (since no user model is used, user has to provide all required information each time using the system)
Automatic filtering involvement	Yes. (The system queries the user based on the user model for any filtering on the result set)	No. (Since no user model is used, any automatic filtering is not possible without user intervention)	No. (Since no user model is used, any automatic filtering is not possible without user intervention)
Tries to avoid personal information related questions	No. (All system directed questions are treated the same)	N/A. (System does not ‘ask’ any questions)	N/A. (System does not ‘ask’ any questions)

7.4.6 Step 4: Further Query the User

To further filter the results set, the item features which were not constrained in the previous steps needs to be consider. This can be achieved by asking the preferred attributes values for such features from the user. However, as mentioned in section 7.5.6, the most important item features need to be constrained first (to avoid losing more suitable results).

As explained in Chapter 5, section 5.7, the Average Total Relevance (*ATR*) of an item feature is the summed average of all relevance values of the attributes belonging to that feature. The features with high *ATR* values are the ones that were important to the user in the previous queries. Therefore, to find out the importance of the features, the *ATR* values of the remaining (so far unconstrained in Steps 2 or 3) features are calculated. Since the ATR value represents the importance of a feature to the user, the features are listed in the descending order of the ATR value. The feature with the highest *ATR* value becomes the highest relevant feature.

The algorithm for ordering the item features according to *ATR* values is shown in Figure 7.12. Each feature is represented as Fj, $\forall j \in [1,2,..,J]$, and J is the total number of features exploited to describe items in the current interacting domain d_i.

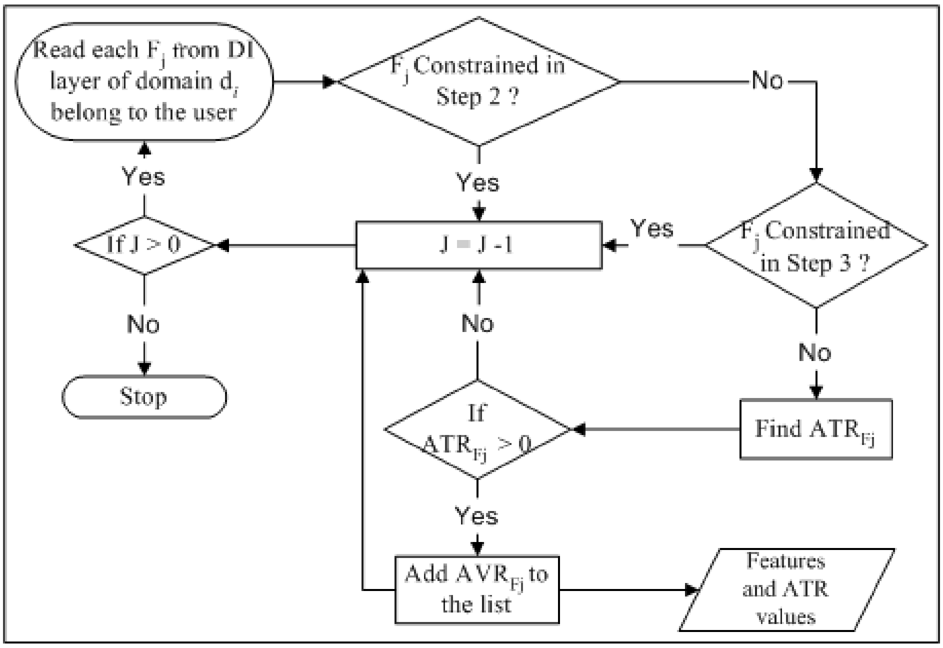

Figure 7.11 : Feature selection for further querying

The features and ATR value are in the list “Features and ATR values”. The feature in the top of the list is the most important (relevant) feature of the user. Now (starting from the highest important feature on the top of the sorted list) each feature is directed to the user as a question. All the attribute values are provided as options for the question.

For scenario 41-a1, in User41’s *DI* layer, the ‘Popularity’ attribute tops the list with the highest *ATR* value. Figure 7.12 shows the three possible attribute values as options to obtain a value for the feature ‘Popularity’. In the scenario User41 is looking for ‘Near perfect food’, and is looking forward to enjoy the meal. Therefore, to further assure the quality of food, and at the same time to keep the cost low, he selects the options as shown in Figure 7.12. Selection of the options ‘little known but well liked’ and ‘people keep coming back’ may choose inexpensive restaurants due to wide spread popularity. The current result set is then filtered according to the provided user preferred attribute values.

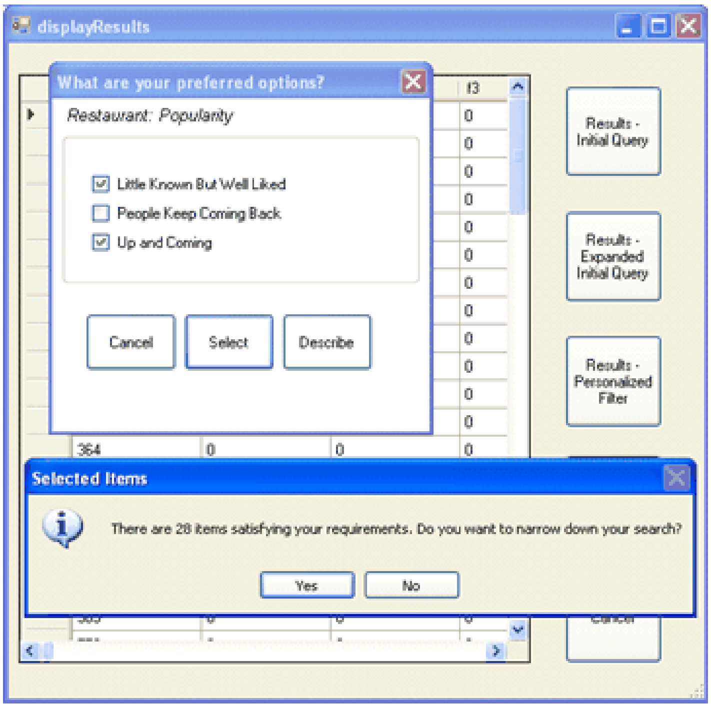

Figure7.12 : Filtered according to ‘Popularity’, constraint

Even though the user model indicates ‘popularity’ as important, the user may not be interested in the feature for the current scenario. To capture such situations the selection method allows flexibility in answering questions. As shown in Figure 7.12, when a question is presented to the user, either selection of an option or cancellation (“Cancel”) is possible.

This way the user gets the chance not to filter out the items depending on an unimportant constraint. The "Describe" button provides a description of any of the selected attributes, providing any sub-features of the attribute or a simple description held by the system.

After filtering based on 'Popularity' constraint the resulting item list consist of 28 restaurants (Figure 7.12). Then, the user can choose either to browse or further constrain the result set. If further constraining is required, the feature with second high ATR value (which is 'Location' in User41's user model) is presented as a question (Figure 7.13).

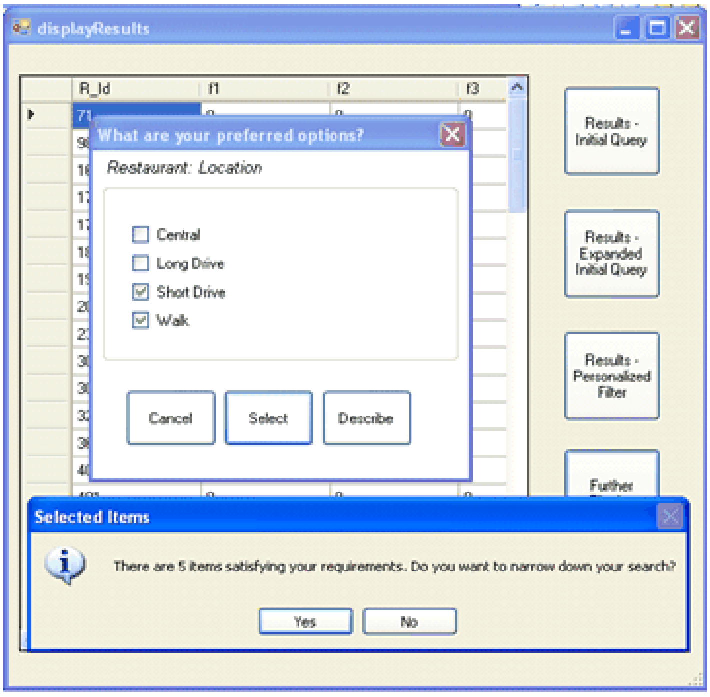

Figure 7.13 : Filtered according to 'Location' constraint

After filtering according to the second important constraint, ('Location') only five records remains in the results set. If User41 chose to view the recommendations, (after Figure 7.14) the selected restaurants appear one after the other.

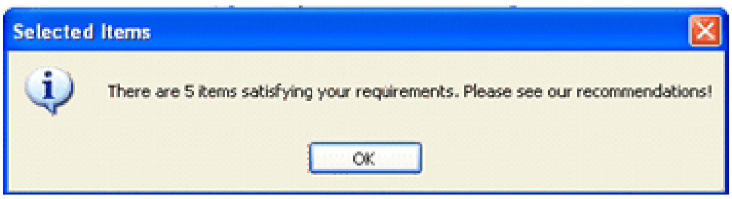

Figure 7.14 : Displaying a selected Restaurant

Figure 7.15 demonstrates how the prototype system displays a selected restaurant. The picture of the restaurant gives the user a more descriptive idea about what to expect. By showing the attributes explicitly requested by the user separately, the item presentation is personalized.

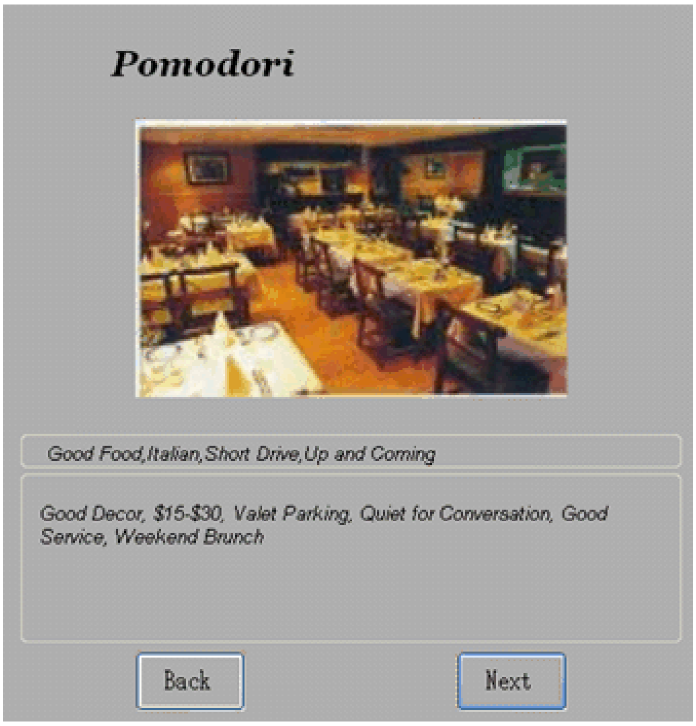

Figure 7.15 : Item display screen

If still the number of records is unmanageable, the question asking process continues until all the features with positive (greater than zero) ATR values are presented to the user as a question. Related work is compared with respect to Step 4 of the algorithm in Table 7.9.

7.4.7 Step 5: Calculate Similarity between the Query and the Items

If a further unmanageable set of records exists, the system needs to filter out the less important items. For this purpose, the similarity measure (discussed in section 7.4.2) is used to rank the remaining list of items. The highest ranking items are displayed to the user. In the prototype system the first three top ranking items are displayed. For the scenario 41-a1, the best three outcomes are shown in Figure 7.16.

Table 7.9 : Comparison with existing systems with respect to Step 4 of the algorithm

PIPRP	**Adaptive Place Advisor (Thompson *et. al.*, 2002)**	**Entrée (Burke, 2002a)**	**SmartClient (Viappiani *et. al.*, 2006)**
Use personalized attribute ranking	Yes. (According to the user model, the most important attribute to ask is chosen. However, as mentioned previously all users starts with the same user model with initialized attribute preferences. Therefore, users who interact for the first time have identical importance to all unconstrained attributes)	No. (The system does not ask any questions the user performs further tweaks on the system provided features)	No. (The system does not ask any questions. The user is able to query on the system provided features)
Present results after each question – Items and their details	Yes. (Only the best three items are presented one after the other. No detailed descriptions are provided)	Yes. (After each tweak the best item is presented with detailed descriptions out of a list of ten. The names of the next nine suggestions are listed without details, which the user can select to view in more details)	Yes. (All the detailed results are presented for the user to make further trade-offs)

As mentioned previously, similarity calculations are time consuming if used with a large data set. In *PIPRP,* the similarity calculation is utilized only if required, after several filtering steps. Therefore, the *PIPRP* can handle a smaller number of records, when the result set has being filtered.

Figure 7.16 : Final outcome for User41's scenario a1

At the end of viewing the results, the user may decide to change preferences provided during Step 4. This can be done after viewing the last results and then selecting "Yes" in Figure 7.17.

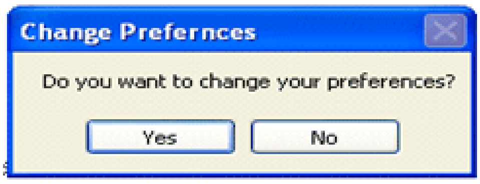

Figure 7.17 : Message - start of Step 4 with the record set resulted after Step 3

The question asking process starts from the beginning of Step 4 using the same item set which resulted after Step 3. As such, the user is allowed to change the preferences provided during the interaction with less effort and time. Related work is compared with respect to Step 2 of the algorithm in Table 7.10.

Table 7.10 : Comparison with existing systems with respect to Step 5 of the algorithm

PIPRP	**Adaptive Place Advisor (Thompson *et. al.*, 2002)**	**Entrée (Burke, 2002a)**	**SmartClient (Viappiani *et. al.*, 2006)**
Number of items displayed – Three if viewed in Step 5 else user desired number of outcomes is viewable.	Three – the three top ranking items are presented to the user starting from the most suitable until the user is satisfied with the answer.	Ten – the top ranking item out of the list of ten is presented to the user in detail. Each tweak changes the selected list of items.	Long list – in ranked order. Each trade off change the ordering of the records.
Possibility of changing the previous selections.	Difficult - User has to suggest if knowledge permits. Require similarity calculation to be carried out for the entire item set.	Not possible – after each tweak the rest of the items are removed from the result set. Need to start from the beginning.	Possible without much effort.
Possibility of comparing results	Not possible	Possible by critiquing the items one after the other	All items are represented using raw data in a list to compare.
Picture included in the display	No.	No.	Yes. (The locations of the cities are marked on a map)

The following section demonstrates and discusses the experimental results of *PIPRP*.

7.4.8 Results

Detailed outcomes/results of the experiments conducted for scenarios presented in Table 7.5 of section 7.5.1 are given in Table 7.11. S-Id is the scenario Id. For each scenario, the initial query inputs ("Initial Query Inputs") and answers to questions ("Questions and Answers") are given in two columns. Under "Initial Query Inputs", the features of interest and the preferred attributes for such features provided during the initial query specification (Step 1) are given. "Questions and Answers" shows the questions asked in Step 4 and the answers provided. The presentation of data follows the underlined format shown in Table 7.11 heading. S-id 41-a1 represents the demonstrated scenario in section 7.5.2.

Table 7.11 : Inputs and explicit answers provided by User41 during four interactions

S-id	**Initial Query Inputs**		**Questions and Answers**	
	Feature (Feature Number)	**Attribute (feature num, sub-feature num, attribute num)**	**Feature (Feature number)**	**Attribute name(Attribute number)**
41-a1	Cuisine (1)	Italian (1,7)	Popularity (17)	Little known but well liked (136), Up and coming (242)
	Food Quality (4)	Near-perfect Food (77)	Location (22)	Short drive(214) Walk(247)
41-d1	Atmosphere (11)	Dining Outdoors (63)	Popularity (17)	Cancelled
	Liquor (22)	Liquor (22)	Location (10)	Short drive(214), Walk(247)
41-e1	Cost (2)	$15-$30 (165)	Location (10)	Cancelled
			Popularity (17)	Little known but well liked (136), Up and coming (242)
			Meal Times (6)	After hours dining (4), Dining after the theatre (62)
	Décor (3)	Good décor (52)	Special Menus (28)	Cancelled
	Atmosphere (11)	Warm spots by the fire (248)	Liquor (22)	Cancelled
			Parking (12)	Parking lot available (171)
41-a2	Cuisine (1)	Italian (1,7)	Location (10)	Short drive (214), Walk (247)
	Food Quality (4)	Near-perfect Food (77)	Popularity (17)	People keep coming back (178)

For example, in S-id, 41-a1 two features “Cuisine” and “Food quality” are constrained. The preferred attribute values are “Italian cuisine” and “Near-perfect Food”. The value “Italian” is a sub-feature (sub-feature number 7) consists of six attributes (such as “Italian (northern)” etc.) and belongs to feature “cuisine”. Under the “Questions and Answers” user was asked two questions: the “popularity” and the “location” preferred.

For the first question two answers were provided by the user; ‘Little known but well liked’ and ‘Up and coming’. For the second question again user selected two preferences; ‘Short Drive’ and ‘Walk’. Initial preferences for scenarios 41-d1, 41-e1 are different from 41-a1. Scenario 41-a2, is the same scenario as 41-a1 performed (for the second time) using the updated user model after implementing the scenarios 41-a1, 41-d1, and 41- e1. For each scenario, the number of records retrieved after each step is given in Table 7.12. The test result column (“Parametric Filtering”) is for comparison with the model outcomes. It shows the number of restaurants retrieved using the parametric search for the initial inputs. In the table, for scenarios 41-a1, 41-d1 and 41-a2 parametric filtering retrieved only 3 records, while the number of records retrieved for scenario 41-e1, is 20. The column “Expanded Query” shows the result generated after the initial query is expanded using the *LUM*. The next column shows the number of records after Step 3. In Step 4, items are further filtered according to the user’s answers to the personalized questions. In scenario 41-a1, after providing the answer for the first question (in other words after answer A1), 28 records remained in the result set (A1-28). Then after the second answer, 5 records remained in the result set (A2-5) and so on. The last column (“Final Outcome”) shows the best three recommendations after each session.

Table 7. 12 : Results of the interactions with User 41 after each algorithm Step

I-id	**Parametric Filtering (Step 1)**	***PIPRP***			
		Expanded Query (Step 2)	**Personalized Filtering (Step3)**	**Items after Questions (Step 4)**	**Final Outcome (Rid) (Step 5)**
41-a1	3	106	85	A1-28, A2- 5	928, 1692, 1281
41-d1	3	51	3	A1-2	1712, 1407, 1384
41-e1	20	185	119	A1-33, A2-11, A3-10	904, 928, 764
41-a2	3	106	85	A1-22, A2-7	1384, 928, 1778

7.4.9 Comparison of results

As mentioned previously, testing of the outcomes to measure accuracy is difficult to do without a proper dataset. Therefore, we employ a method to compare the outcomes of parametric search and *PIPRP* with regard to user preferences.

Initially, at the beginning of the interactions the only user preference expressed is the initial query. However at the end of *PIPRP* more user preferences are available to the system (obtained from the user model and by question asking). Therefore, we argue that the actual user requirement is the total preferences available at the end of the *PIPRP*. In this work the actual requirement is referred to as the *complete user query.*[24]

Complete user query = preferences in the initial query + preferences from the user model + preferences expressed as answers to system questions.

The *complete user query* describes the most preferred items of the user. It consists of the information from the user model and user preferences obtained during the interaction as well as the initial preferences. Therefore, the complete user query is more complete and comprehensive. If an item fulfils such requirements, then surely it should satisfy the user.

Therefore, the testing of the quality of results is carried out as follows.

(i) Select two sets of results (restaurants lists) for the same scenario using the two methods; parametric search, and *PIPRP*.

(ii) For each scenario, calculate the similarity of each item in the database (in this case 1875 restaurants) to the *complete user query*.[24]

(iii) Compare the similarity values/ranks of items retrieved using parametric search with the items retrieved using *PIPRP*.

These steps were carried out for each scenario. The Table 7.13 shows the results of each scenario. In three of the scenario, parametric search retrieved 3 results except once (20). For each scenario, same number of outcomes from *PIPRP* is shown in the table for comparison.

Table 7.13 : Similarity based relevance of restaurants to the User41 during the four transactions

Interaction	Parametric Filtering			*PIPRP*		
	R_Id	Rank	Value	R_Id	Rank	Value
41 – a1	1764	18	5.45	928	1	6.96
	1446	32	5.15	1692	2	6.4
	512	811	2.39	1281	3	6.31
41 – d1	1407	2	5.78	1712	1	6.55
	1313	5	5.56	1407	2	5.78
	1674	20	5.14	1384	3	5.7
41 – e1	757	4	7.95	904	1	8.64
	1017	21	7.15	928	2	8.4
	910	27	6.86	764	3	8.29
	867	34	6.77	757	4	7.95
	178	63	6.45	863	5	7.86
	551	67	6.41	821	6	7.78
	162	88	6.19	1839	7	7.71
	404	89	6.19	261	8	7.7
	785	90	6.19	1028	9	7.51
	787	91	6.19	175	10	7.46
	906	92	6.19	883	11	7.46
	377	112	6.01	1040	12	7.46
	85	133	5.85	1758	13	7.34
	134	160	5.69	48	14	7.26
	1044	253	5.31	1825	15	7.26
	997	337	5.1	792	16	7.24
	298	362	4.94	1566	17	7.23
	1015	471	4.56	25	18	7.21
	233	535	4.4	565	19	7.16
	1170	537	2.7	800	20	7.16
41 – a2	1764	30	4.65	1384	1	6.37
	1446	47	4.42	928	2	6.15
	512	809	2.39	1778	3	5.62

The "Parametric filtering" column presents the information on the outcomes of parametric search while the "*PIPRP*" column shows the information on the outcomes of *PIPRP*. "R-Id" gives the Id of the restaurants retrieved as final result. The "Value" is the calculated similarity value (to the user query) using equation 7.1. (See the example in section 7.3.3 for details of calculation). "Rank", shows its similarity rank to the *complete user query*.[24] This rank is based on the similarity calculation. A restaurant that has a close similarity to the complete user query receives a lower value as the rank (example, highest rank possible is

1). If the same restaurant appears in both results sets, then they are highlighted in the table. Although the final result set obtained in *PIPRP* is only three items, for comparison purposes similar number of records as parametric search are shown in Table 7.13.

As shown in Table 7.13, the restaurants selected in the parametric search are less similar to the *complete user query,* than the outcomes of *PIPRP*. This confirms that the *PIPRP* retrieval has provided a better fitting set of restaurants to the user query, than the ones chosen using parametric search (in all of the scenarios).

The reason is as follows. When specifying the initial query, user selected only a few attributes. If parametric search is employed only such preferences will be included as search criteria. In *PIPRP*, the interactive product selection, guides the user to specify additional product features. The user gets to learn more about the product features which resulted in the user being able to better describe the expectations. In addition, the attribute relevance values in the user model also facilitate capturing user preferences. As a result, the *PIPRP* recommendations show a greater similarity value to the complete user query. In fact, *PIPRP* generate the complete user query step-by-step during the interaction.

According to Table 7.13, there are two occasions where the same restaurant is retrieved by both retrieval methods (highlighted). However, in both scenarios *PIPRP* managed to retrieve other restaurants better than the initial query selection. For example, in 41-e1, the restaurant number 757 has a similarity of 7.95 to the complete user query. In parametric search 757 is the top similarity restaurant while in *PIPRP* there are three more restaurants with higher similarities exists (where the top similarity is 8.64).

When scenarios 41-a1 and 41-a2 are compared, the results of the *PIPRP* changed while the initial parametric results remained the same. The reason is initial queries are similar in both scenarios. After implementing each of the scenarios, the user model gets updated. Interaction 41-a2 uses the updated user model. Therefore, although the query is the same the results produced are different from the initial interaction.

7.5 Evaluation of eHermes PERSONAL

In Chapter 6, within the limits of available resources, experiments were conducted to demonstrate the functionality and usefulness of the *LUM*. In section 7.5, performance of the

PIPRP and the role of the novel *LUM* were demonstrated using the available data. In addition, the step by step performance of the novel model was compared to the existing work. However, lack of proper resources such as real users, calls for further experimentations in order to confirm the positive outcomes of the model. In addition, it is worth evaluating the model against criterion critical to requirements of the current eCommerce needs. Therefore, in this section the novel model is evaluated according to a set of criteria.

7.5.1 Background

There are two major evaluations to perform. Namely: evaluation of the user model and the evaluation of the interactive product search. As far as the user model is concerned, the most important measure is its ability to generate accurate recommendations. However, the evaluation method depends on various factors such as available data and the testing environment. With regard to the interactive product selection processes the evaluation method varies according to the type of the system. However, in all these systems the number of interactions to isolate a manageable results set was the major issue. The existing methods employed in user model and interactive product search evaluations are discussed below.

7.5.2 Evaluation of User Model Accuracy

One of the most important issues of a user model is its accuracy in recommendations. Although, there is no generally accepted methodology for evaluating the performance of a user model, there are few methodologies that are introduced to the user modeling community from other backgrounds such as AI (Zukerman and Albrecht, 2001). According to Zukerman and Albrecht (2001) and Billsus and Pazzani (1999) three such methods employed to evaluate user models are as follows:

1. Precision, Recall and F_1-measure
2. Predicted probability and accuracy
3. Utility

More often, the evaluation methods depend on the availability of data and testing environments. However, the most suitable would be to set up a testing environment where user based evaluation is obtained over a long period of time for a large database of various items. The actual user opinion would be the most trustworthy evaluation. This seems quite unachievable especially for academic prototype systems.

7.5.3 Evaluation of Online Product Selection

As previously discussed, there are three online product selection approaches such as interactive decision guides, example critiquing system and user modeling systems. Evaluation of such approaches is performed mostly by counting the number of interactions required to obtain the final recommended product list. For example, in interactive decision guides such as Schmitt *et. al.* (2002) the number of interactions is the number of features specified by the user. In critique based systems (Burke, 2002b; Viappiani *et. al.*, 2006) initial query and the number of critiques applied to direct the search becomes the total number of interactions. Both these types do not use a model of the user. In user model based systems, such as Adaptive Place Advisor (Thompson *et. al.*, 2002), the number of times explicit information acquired from the user during a transaction is considered as the number of interactions. Adaptive Place Advisor was evaluated by comparing the number of system-user interactions, with and without the user model.

All interactive product selection systems pay very little attention to other aspects contributing to the system obtrusiveness; apart from the number of interactions. An alternative set of principles to guide online interactions for product search is described in Pu (2004). In contrast to the previously discussed work, these principles investigate a broader spectrum of options to handle during an online interaction. These principles are based on the assumption that the user is not fully aware of his/her need to the minor detail. In Pu (2004), to support user's product selection process, a three-way strategy has been employed:

(i) provide users with domain knowledge - provision of domain knowledge to help user make the selections

(ii) avoid means objectives - capturing the user's actual objectives, and

(iii) convince the user – provide user with verification of system selections

The three-way strategy is enforced using a set of principles. As explained in Pu (2004), Table 7.14 shows the three aspects and the six principles defined to achieve them.

Table 7.14 : Interaction principles for online product navigation given in Pu (2004)

Principle	**Explanation**
Provide users with domain knowledge	
Principle 1: Elicit preferences within context.	*A search tool should ask questions with reference to a complete and realistic context, not in an abstract way.*
Principle 2: Allow partial satisfaction of user preferences.	*When no solutions exit that satisfy all preferences, show solutions that satisfy a maximal subset.*
Avoid Means Objectives	
Principle 3: Allow partial preference models.	*Do not force the user to provide any specific preferences.*
Principle 4: Any preference	*Allow users to state their preferences on any attribute rather than a fixed subset.*
Principle 5: Any order	*Allow users to state their preferences in any order they choose.*
Convincing the user	
Principle 6: Support tradeoff navigation	*The search tool should provide active tradeoff support for the user to compare examples shown.*

For a detailed explanation of each principle and examples please see Pu (2004). In the next section, need of our own criteria is discussed.

7.5.4 Proposed Evaluation Criterion

The goal of this thesis, is to control a broader spectrum of issues related to system-user interactions; namely 'systems obtrusiveness in system-user interactions'. Therefore, none of the above described criterion is directly applicable to our work. The following evaluation criterion attempts to evaluate the combined effect of *LUM* and the *PIPRP* in achieving the above goal. The novel evaluation criterion consists of five requirements are shown in Table 7.15. In addition, this criterion, satisfy the above described interaction principles given in Pu (2004).

The overlapping between the interaction principles (Pu, 2004) and the proposed evaluation criterion is given in the Table 7.16. As shown in Table 7.16, 3 and 4 of the criterion is not related to the principles. The additional criterions were enforced to establish the obtrusiveness related requirements discussed in section 7.3.

Table 7.15 : List of evaluation criterion

Criterion	Description
Criterion 1	*Ease and Flexibility of querying*
Criterion 2	*Capable of retrieving items similar to the query rather than exact matching items.*
Criterion 3	*Comprehensive questions*
Criterion 4	*Minimize personal questions*
Criterion 5	*Convincing the user*

Table 7.16 shows that proposed criteria cover all six interaction principles by Pu (2004). Therefore, if eHermes PERSONAL satisfies the proposed criteria, the interactive principles by Pu (2004) are also satisfied. In the next sections, each of the criteria is discussed relating the corresponding principles.

Table 7.16 : Interaction principles (Pu, 2004) and the proposed evaluation criterion

<table>
<tr><th>Criterion</th><th>Principle</th></tr>
<tr><td>Criterion 1: Flexibility of querying</td><td>Principle 1: Elicit preferences within context.
Principle 4: User should be allowed to specify any preference
Principle 5: User should be allowed to specify his/her preferences in any order.</td></tr>
<tr><td>Criterion 2: Capable of retrieving similar items to the query rather than only exact matching items.</td><td>Principle 2: Allow partial satisfaction of user preferences.
Principle 3: Allow partial preference models.</td></tr>
<tr><td>Criterion 3: Comprehensive questions</td><td rowspan="2">Facilitates less obtrusive system-user interactions. Not covered by the interaction principles by Pu (2004)</td></tr>
<tr><td>Criterion 4: Minimize personal questions</td></tr>
<tr><td>Criterion 5: Convincing the user</td><td>Principle 6: Support tradeoff navigation</td></tr>
</table>

7.5.5 Evaluation Using the Proposed Criterion

In this section each of the above criteria, are evaluated against eHermes PERSONAL. Under each criterion, performance is analyzed with respect to corresponding principles.

Criteria 1: Flexibility of Querying

As shown in Table 7.9, criterion 1 covers 3 of the interaction principles given in Pu (2004): principles 1, 4 and 5.

In a dynamic world user preferences are volatile; especially the segments of users who use modern technology such as eCommerce who are expected to have varying preferences. Therefore, eHermes PERSONAL provides an easy to use flexible interface, facilitating the users to specify their changing needs. In addition, users are often unable to articulate their complex needs, due to a lack of catalogue knowledge. The criteria one is set to measure the flexibility of the interface at the initial query specification.

According to section 7.5.3, Step 1 and 4 of the algorithm obtain user preferences as inputs.

Principle 1*: Elicit preferences within context*

In step 1 of the algorithm, use of the interface shown in Figure 7.8, (where all the features and attributes are displayed) helps the user to concentrate on the context. This corresponds to the above principle 1, which request user inputs to be more specific to the context. Since all the possible options are provided, specifying preferences needs less effort than providing inputs for an open-ended question. The items are thoroughly described and the options are provided to help specify the user need.

In Step 4, system generated questions are presented along with all possible attribute values (see Figures 7.12 and 7.13).

Principle 4*: Any preference*

As explained in the Step 1 of the algorithm, the user is free to specify preference towards any feature in the order of importance they prefer. For example, in the restaurant domain, the cuisine seems to be extremely important. However, for an individual with a walking disability, a restaurant with wheelchair access will be more important. The interface permits selection of the most important feature and then selecting the preferred attribute value for that particular feature without restricting the user to a fixed set of priorities.

In Step 4, the user is supposed to select preferred attributes for system selected features which were directed to the user as questions. Users have the choice of answering each question or avoid answering by canceling. Although the features are directed to the user by the system, these features are chosen based on the user model. As explained (in Chapter 5, section 5.7) ATR values determine the total relevance of a feature to the user. Therefore, features with high ATR values become important features to the user. As a result, even though the questions were directed by the system, still they are the user's most important features. The following example clarifies the question sequence selection.

Figure 7.18, shows the *ATR* values calculated for a sample set of features in the *DI* layer of User41 after the scenario 41-e1. As shown, the user model indicates greater concern (highest *ATR* value) for the *popularity* of the restaurant and least attention is paid to the *restaurant category*.

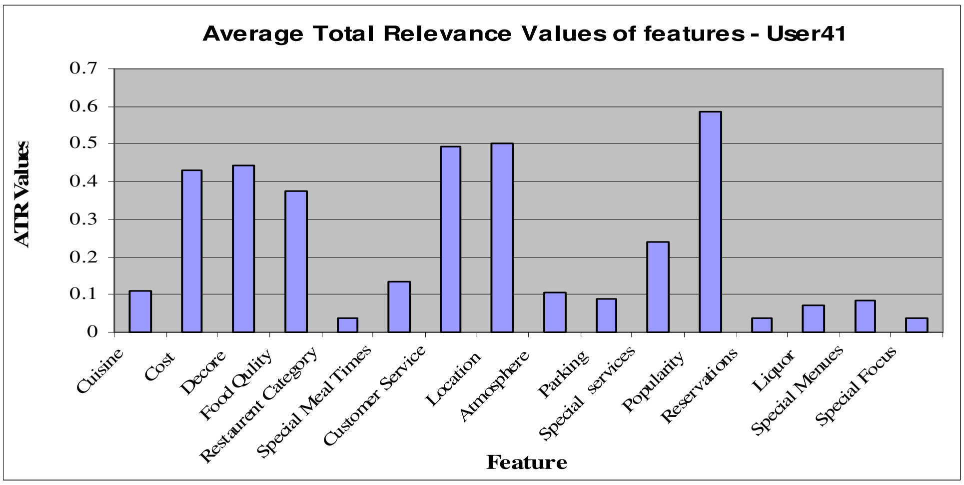

Figure 7.18 : ATR values calculated for User41's DI layer after the scenario 41-e

Questions to obtain user preferences are generated in the order of the *ATR* values. For example, according to Figure 7.18 User41 shows the highest concern about the *popularity* of the restaurant. Therefore, the first question in Step 4 was on *popularity*. Since *ATR* of the *location* is the next highest value, this question was directed to the user after *popularity*. (Refer to section 7.4.8, Table 7.11, scenario 41-a2, the questions directed to the user are *popularity* and *location*).

__Principle 5__: Any Order

When specifying the initial query, in Step 1, the user is free to select any of the features and in any order.

In Step 4, the question sequence corresponds to the priority order of the preferences in the individuals' user model. Therefore, even though the questions were asked by the system, the sequence is decided by the individual's preferences in the user model.

Users are also allowed to select more than one preference from the given options (Figure 7.19), where the user is not particular about a single option. All selected options are included in the query. After each user selection, the results set get filtered retaining the records that fulfill the user request.

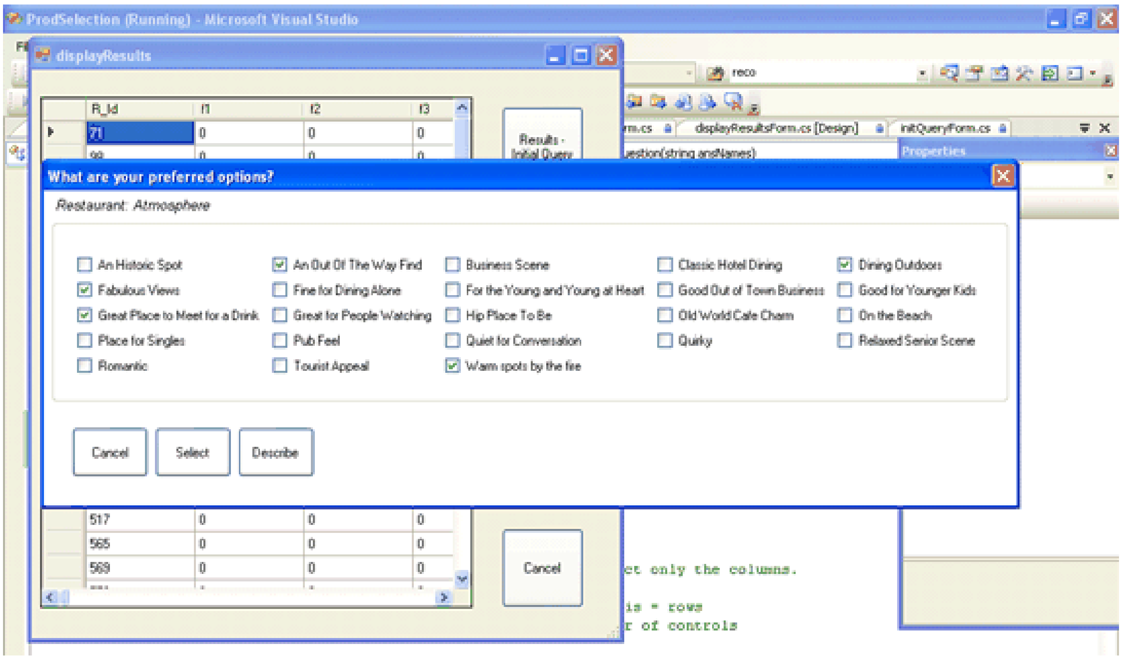

Figure 7.19 : User preference for the restaurant category

Based on above discussion, it can be claimed that the proposed model fulfils the requirements in criterion 1 by providing a flexible interface to capture actual user need.

Criteria 2: Capable of Retrieving Similar Items to the Query Rather Than Exact Matching Items

As shown in Table 7.12, criterion 2 covers two of the interaction principles given in Pu (2004): principles 2 and 3.

A search engine could often result in null retrievals: or it misses out on many good items that might suit a user. For example, strict constraints leave out closer outcomes such as slightly greater or lesser price ranges. In our approach, the preference information in the user model is used to accommodate such partially suitable items in the recommendations.

Since the relevance values are allocated in a fuzzy manner, more than one attribute become relevant under the same feature. As explained in section 7.5.3, Step 2, search query, personalized query expansion is applied to the initial query according to the user model. In other words, the query is expanded using the other relevant attributes in the user model. For example, if presented as a query, the initial attribute selection for scenario 41-e1 (Table 7.6) is as follows:

> *"Search for the items that has price in range \$15-\$30, has good décor and has the atmosphere as warm spots by the fire"*

At the time of the interaction the user model had more values under all three features. Therefore the following query resulted after the personalized query expansion.

> *"Search for the items that has price in range (\$15-\$30) or (\$15 and less) or (\$30-\$50)*
> *and has poor décor or fair décor or good décor*
> *and has the atmosphere as (warm spots by the fire) or (Dining Outdoors) or (Place for Singles) or (Quiet for Conversation)"*

For clarity, the contents of the *DI* layer of user41 are presented in Table 7.17. According to the table; positive relevance values for more than one 'Cost', 'Décor' and 'Atmosphere' values are available. The following query was resulted after the personalized query expansion with the effect of the user model information in Table 7.17.

> *"Search for the items that has price in range (\$15-\$30) or (\$15 and less) or (\$30-\$50)*
> *and has poor décor or fair décor or good décor*

and has the atmosphere as (warm spots by the fire) or (Dining Outdoors) or (Place for Singles) or (Quiet for Conversation)"

Table 7.17 : A summary of the above user model content

Feature Name & Number	Attribute number	Attribute Name	Relevance	
			Initial	Current
Cost (2)	169	\$50 and more	0	0
	163	\$15 and less	0.1	0.12
	167	\$30-\$50	0.45	0.88
	165	\$15-\$30	0.74	1
Décor (3)	53	Excellent Decor	0	0
	54	Extraordinary Decor	0	0
	55	Near-perfect Decor	0	0
	50	Poor Decor	0.08	0.15
	52	Good Decor	0.7	0.85
	51	Fair Decor	0.55	1
Atmosphere (11)	63	Dining Outdoors	0	0.2
	184	Place for Singles	0	0.04
	196	Quiet for Conversation	0	0.04

Therefore, the expanded query is capable of retrieving more items than the items fulfilling exact constraints.

Principle 2*: Allow partial satisfaction of user preferences*

The query expansion results in a larger number of outcomes. These items may not satisfy all the constraints the user specified in the initial search query, but the system still produces adequately matching outcomes. The top three high ranking restaurants retrieved for the above query are as in Figure 7.20. Therefore *principle* 2 is fulfilled.

Figure 7.20 : Result retrieved for the above query

Principle 3*: Allow partial preference models*

In the *PIPRP* the user can start product navigation by simply specifying the most important attributes that comes the mind. Consider scenario 41-e, User41 initially had three constraints in mind: the price, décor and the atmosphere. Suppose he also appreciates that parking is available since he plans to go to the cinema if the friend agrees. Keeping additional requirements in mind User41 starts *PIPRP* with partially declaring his preferences. Therefore, the model complies with *principle* 3, and hence fulfils the requirements for criterion 2.

Criterion 3: Comprehensive Questions

This criterion evaluates the interactive process with respect to its ability to ask comprehensive questions. Comprehensibility of a question depends on not only its simple and easy-to-understand nature but also the sequence in which they were asked. For example, static surveys include related questions in a group box or in the same page for the interviewee to understand the questions which are related to the same issue. But in dynamic dialogues, if the intention of the process is only to reduce the number of questions it is hard to maintain a proper sequence. These systems chose the most discriminating feature among the result set to get a value from the user. Therefore, online interactive product navigation processes often puzzle users with the order of their questions.

In *PIPRP*, the questions are generated according to the average total relevance of a feature. For example if the ATR values are 0.325 and 0.222 for *cost* and *décor* respectively, then $ATR_{COST} > ATR_{DÉCOR}$. This results in asking the user what price range is important, before asking the preference for *décor*.

Since the questions have a relevance to the user, they can be expected to be "appreciated" by the user and most useful for the user to answer. On the other hand, high relevance ensures that this question is information that the user should reveal in order to filter the results set. Due to the personalized nature of the questions, the comprehensibility can be ensured.

Criteria 4: Minimize Personal Information Related Questions

This criterion claims the model's ability to minimize or completely remove any personal questions. During an eCommerce product search requesting personal questions can be alarming to the user, especially if the user is not concerned about the particular feature, requesting a preference for that feature maybe intrusive. As discussed in Step 3, in addition to demographics any questions that reveal the user's personal information can be considered as personal questions. For example, if the user does not indicate a concern about the price of the item, but certain other aspects, asking the user about his/her price range preference may appear to be intrusive. It may appear as intrusively trying to figure out the user's income. Similarly system questions related to any PIR-attribute may look intrusive trying to capture user's personal information. Therefore, we believe such questions should not be directed to the user unless the user specified them willingly as an initial preference. As a result *PIPRP* employs 'personalized filtering' as described in Step 3 of the algorithm. Such question reduction strategies further improves the quality of the interactions by indirectly encouraging a fewer number of questions.

Criterion 5: Convincing the User

According to Pu (2004), if the system provides the user with solutions that is supposed to be the best, the users might not be convinced. Therefore, users need to verify the quality of the system results. They need to compare the final results and find the true best answer by considering any affordable tradeoffs.

Principle 6*: Support trade-off navigation*

Once the final results are displayed to the user, there should be a facility to go back and compare the results or even to do slight changes to the previous selections. In critique based systems such as Entrée (Burke, 2002a), Adaptive Place Advisor (Thompson *et. al.*, 2002) or SmartClient (Viappiani *et. al.*, 2006) this is possible. However, the filtering mechanism used in Entrée, removes any un-matching items from the results set after each tweak. If the user selection criterion is significantly changed then there is no way of finding the items that match the new constraints unless the user re-starts the entire product selection process.

PIPRP does not provide critique based selection. However, the results set retrieved in Step 3, are maintained as a record set out of which the final results are obtained. If the user is not satisfied with the outcomes, he/she can go back and invoke the same set of system questions on the same large set of results obtained in Step 3. This record set contains the data satisfying the initial query and the user model. Therefore, the user does not have to start from the beginning. In such circumstances, the user is able to modify his/her preferences and observe the effects on the final results.

Although *PIPRP* does not directly support trade-off navigation, we believe the possibility to re-perform selection helps adequately. Furthermore, the interface permits the user to go back and forth in the system selected result set (using 'Next' and 'Back' buttons). As explained previously, the system output is in the highest to least relevant order. The user can use the navigation facility to traverse the list and compare the recommendations. User will be convinced that the items in the top of the list are preferred than the items later in the list.

7.6 Summary

Interactive product search in a personalized context is extremely important for the growing consumer requirements in eCommerce. There are different approaches demonstrated in the literature with regard to interactive product search. However, the personalization provided with such approaches do not seems adequate for the expected growth of eCommerce. Search algorithms have traditionally been optimized to find everything that might be relevant solely based on search criteria. Therefore, as highlighted in (Hagen *et. al.*, 2000) to deliver successful services in growing eCommerce activities, personalized product retrieval is valuable and timely.

As explained in the chapter the *PIPRP* combined with the *LUM* provides personalization in all three phases of the online product retrieval: requirement elicitation, product search and product presentation. Such personalization can help minimize the problems encountered in online product retrievals such as null retrieval, retrieving unmanageable number of items, and the retrieving unsatisfactory items.

The experiments presented in the chapter, are subject to the limitations of the datasets used. It is an obvious and known fact that the retrieval process greatly depend on the available data, since the distribution of products in the search space determine how easily they could be located. For example, if most of the available items belong to the same price range, then retrieving items belonging to other price ranges cannot be constrained as much as constraining the items belonging to the common price range, as this might lead to null retrievals.

Although the restaurants were described using 256 attributes, some of the attributes are not presented in any of the restaurants. In fact, out of the 1875 restaurants used in experiments only 295 had cuisine descriptions, where others seem not to be offering any cuisine type. Furthermore, it is observed that certain features such as *restaurant category* which is extremely important from *PIPRP* point of view, is completely ignored for most of the restaurants. Among the 256 attributes there are several restaurant types such as "Bakeries", "Bar-B-Q", "Cabin", "Cafe/Espresso Bars", "Cafe/Garden Dining", "Cafeterias", "Coffee Houses", "Coffee Shops", "Coffeehouses", "Deli", "Diners", "Fast Food", "Fountain and Ice Cream", "Noodle Houses", "Noodle Shops", "Oyster Bars", "Pastry Shops", "Pizzerias", "Steakhouses", and "Yogurt Bar". Most of the restaurants are not described under any of these attributes. In the experiments, these attributes were grouped as *restaurant category*, whereas in the original system these were possibly used in similarity calculations. In *PIPRP, restaurant category* is an excellent indicator to specify user's requirement with respect to the situation. For example, an individual taking a friend for dinner would not consider a "Cafeteria" or a "Pizza parlor". Therefore, the selection should be made out of "Diner's restaurants". However, with existing data, such specification may result in null retrieval.

In *PIPRP* approach, ideally the vendor needs describing the commodity under each feature in order to clearly describe the item for sale. We believe this approach is necessary since the customer does not physically see the item. For example, in eBay, sellers try their best to describe the items with supporting photographs and even diagrams clearly elaborating positive as well as negative features. The approach taken in the thesis allows the vendor to describe the item with regard to large number of features. This helps vendor to identify all the item features consumers may interested in. Therefore, minimizes the chances of vendor forgetting to describe the item with regard to features that may interest the consumer. At the time of item retrieval consumers benefit by choosing items described using a wide variety of features. In a domain where user lacks much knowledge, this may help to identify the crucial features pertaining to that domain.

This chapter demonstrated the role of the user model in personalized product retrieval from online catalogues. As shown in experiments, by combining the descriptive and up-to-date user model with the interactive process, items that are similar to the user query can be retrieved.

Chapter 8
Conclusion and Future Work

This thesis has focused on developing a new method of modeling eCommerce users in order to address key challenges in the field. The new method consists of a three layered user model as well as a new interactive product retrieval algorithm. The layered user model was developed based on the need for accumulating, storing and adapting user information identified in personalization research in IT as well as MR. In this final chapter, we present a summary of the research outcomes in the thesis work and suggest some areas and problems which have emerged from our work, with potential for future research and expansion. In section 8.1, a summary of the contributions and outcomes of this research thesis is presented. Section 8.2 discusses the future work and possible extensions to this research project.

8.1 Research Summary

As described in the introduction, the contributions of the thesis facilitate user modeling and user adapted interactions in eCommerce. This research has been initiated by the requirement of a single user model to unobtrusively provide personalization to eCommerce consumers in multiple application domains. As our literature review progressed, the additional motivations (such as capturing dynamics of preferences, handling different ontologies, representing fuzzy preference and need for techniques to learn or capture preferences) described in the Introduction were identified. With supporting evidence, we identified the main challenge in user modeling as complexity of user behavior.

However, with regard to our goal, the idea of a single user model for multiple applications demanded a complete and comprehensive user model. On the other hand the idea of a complete user model demands explicit user information when building the user model, which is an obstacle to achieving unobtrusiveness. The idea of the layered user model, with components, compasses both ideas without confliction; where the complexity of the user is captured in components and wrap-up as a single complete user model and the unobtrusiveness is accomplished via re-use of components.

Furthermore, our literature review revealed the strengths of approaches taken in the MR; for example the success of the segmentation topologies described in Chapter 3. Therefore, we exploited similar methodologies in layer one of the user model while maintaining the other two layers to capture the dynamism in online markets. In our work, we recognize exploitation of MR approaches in an electronic user model as a novelty.

In order to exhibit its strengths, the novel user model needs to be utilized in providing personalized services to users. Motivated by the information overload problem in eCommerce sites, we employed the novel model in interactive product retrieval. As explained in Chapter 7, the novel product retrieval algorithm utilizes the user model in personalized interactions during all three phases of eCommerce activity.

In the thesis work, one of the major concerns were the datasets used in experimentation. The current datasets in use mostly confirm the ability of the model with regard to its functionality. We realize that the experiments with regard to system performance need to be strengthened; for example increase the number of users involved. We believe a proper dataset consisting of both user demographics and interaction details for those returning consumers would help to yield clearer and impressive experimentation results. Alternatively, if a set of real users were used in the evaluation then the actual scenarios would be used in evaluating both the user model and the product selection process successfully.

This research does not focus on the following issues and they are considered to be outside of the scope of this research project.

- *Privacy concerns in personalization*. When user information is handled online, there are concerns about the security and privacy of user data. We assume that such issues are taken care of.

- *Maintenance of ontologies*. In the thesis work we do not generate or maintain product ontologies. However, in the future work we mention and discuss possible methodologies to handle ontology issues.

The summary of contributions is presented below.

8.2 Summary of Contributions

The major contributions of this thesis can be divided into four parts and they are summarized as follows:

1. Justification of the three layered architecture.

2. Design and implementation of the novel layered user model.

3. Development of a novel interactive product retrieval algorithm which exploits the above user model.

4. Evaluation of the user model and its application to product retrieval using a new evaluation criterion.

The next sections discuss the contributions in detail highlighting the initial aims of the project and the outcomes.

8.2.1 Justification of the three layered architecture

A new user model architecture was proposed designed and justified based on the existing research in MR and IT. The layered model combines the advantages of existing segmentation topologies (in MR) and domain based individual user models (in IT). We validated the proposed architecture both by argument and experimentation. Our study revealed that the user models in MR, are based on user segmentation, where an individual's purchase behavior is predicted irrespective of an application domain. On the other hand IT user models strive to predict user behavior based on past purchases minimizing the use of personal data. The novel model uses a combined approach of the above two approaches taken in IT and MR.

8.2.2 Design and implementation of the novel layered user model

The main contributions of the thesis are the design and development of the new layered user model (*LUM*). The novel user model architecture consists of three information layers each capturing the identified information categories in (i): the personal information in Personal Information Layer (*PI* layer), domain behavior information in Domain Information Layer (*DI* layer) and transaction information in Transaction Information Layer (TI layer). The *LUM* encapsulates several new features to provide it with ability to capture, accumulate and adapt user information as well as to derive further information. The following issues in the user modeling research area have been addressed with the techniques employed

Capture changes in user preferences in dynamic product markets over time - The layered architecture facilitates the use of impulsive and short term behavior in the lower layers of the user model to update the much stable long term behavior in the upper layers of the user model. The Hebbian learning technique is employed to capture changes in user preferences over time. The learning technique also considers the "forgetting factor" to maintain more accurate information in the user model layers. In our experiments it was noted that even if certain attributes were considered as relevant to a given user based on the *PBC* values, the learning technique decays the relevance if that attribute is not explicitly requested by the user in consequent transactions.

The layered approach allows content-based information in the user model. The relevance values for each attribute in the user model allows searching the entire product base for preferred features solving the *new item* problem encountered in personalization. Therefore, the model captures user preferences towards product features allowing it to include all products in the product-base in the search space. Therefore, the model is capable of locating new products that arrive in dynamic product markets, and hence, do not encounter the 'new item' problem.

The design of the model supports reuse of preference information facilitating usability in multiple product domains - General Stereotypes/*PBC* values is a set of quantitative values which describe the user behavior generally in purchasing. *PBC* values describe individual users along different dimensions rather than assigning them to broad segments and are therefore, more flexible in describing the user behavior. We introduce a new concept called the Influence Matrix (*IM*) to map domain information to user interests. When combined with the *IM*, the quantitative *PBC* values in the *PI* layer of the user model are usable as start-up information for any application domain. Therefore, the user model facilitates information reuse among multiple domains. In addition, this also solves the start-up information problem which is known as the "new user" problem in recommender systems.

Capture the fuzziness in user preferences - The user model represent the preferences for each attribute by a relevance value in the range 0-1. Rather than allocating a value from a given scale (e.g. such as a five point scale from 'love it' to 'hate it') this allows the user preferences towards attributes to be described more accurately. In addition, such quantitative values are used in calculating fuzzy relevance values for more than one attribute at the same time. In the experimentation, we noticed that the fuzziness of preferences made more attributes relevant to the user, and hence, indirectly contributed to avoid null retrievals.

Able to use existing information to generate/infer knowledge about the user rather than explicitly requesting information from the user - The existence of the general stereotypes in the upper *PI* layer provides the start-up information for any *DI* layer. The latter updates the *PI* layer (based on *DI* layers) resulting in more accurate and up to date user information.

Able to provide reduced obtrusiveness when building the user model - The acquisition of personal information is a once-off process, for generating *PBC* values. Since *PBC* values are re-used for multiple domains a significant reduction in obtrusiveness is achieved. The ability to derive relevance values and the automatic update of these values from transactions also results in reduced obtrusiveness.

8.2.3 The new product retrieval algorithm

Traditional product retrieval techniques face problems such as null retrievals, retrieval of unmanageable or unsuitable products. The new technique called *PIPRP* (Personalized Interactive Product Retrieval Process) introduced in the thesis manages to address these issued by providing personalization in all three phases of product purchase; requirement elicitation, product search and product presentation. The new algorithm makes use of the *LUM* to generate a personalized sequence of questions. By doing so, *PIPRP* avoids requesting any personal information related product preferences. The algorithm also controls the number of questions directed to a user by using already available information in the user model. At the same time the user is provided with a personalized interactive interface where he/she is able to specify the current need. Therefore, we claim that the new product retrieval algorithm produces a reduced number of questions to the user as well as reduced level of obtrusiveness in the questions, thus resulting in a less obtrusive interface.

8.2.4 Evaluation of *LUM* and *PIPRP*

Due to unavailability of complete user and related transaction data for evaluation, an evaluation criterion has to be defined. The new criteria were based on the *interaction principles* proposed by Pu (2004). The original principles were extended to evaluate the thesis work with regard to obtrusiveness. As discussed in the evaluation section, although the novel product retrieval algorithm (*PIPRP*) fulfills both the original and extended criteria, it does not directly support one of the criterions (trade-off navigation). However, we believe the possibility to re-perform product selection from the middle of the process (rather than from the very beginning) is a decent alternative.

8.3 Future Work

Due to the technological growth and the growing number of web users, user modeling and personalization is expected to become an interesting area of research. Therefore, the work described in this thesis opens up several interesting areas of future research.

As mentioned in Chapter 4, the proposed model was initially designed as part of a larger project called eHermes. eHermes is designed to provide users with search facilities for online commodities. eHermes PERSONAL; the personalization component was tested in two domains (restaurants and leg-wear) within the limits of this thesis. In addition, the model was proposed to provide personalized services in a completely different domain, Helpdesk Services. A framework was proposed to provide personalized services for helpdesk service-requests, employing the *LUM*. The work on this project is published in (Zaslavsky *et. al.*, 2007).

Although layered architecture supports cross domain applicability of the user model, due to a lack of proper datasets it was not properly demonstrated within the work of the thesis. As described in Chapter 6, section 6.3.5, apart from reuse of *PI* layer information, reuse of relevance values of domain attributes in *DI* layer is also possible. As described in Chapter 4, section 4.3.2 this can be achieved using domain hierarchies. However, as the number of product categories (domains) increases, manual generation of product hierarchies become a challenge. Therefore, we propose automatic generation of product hierarchies (ontologies) to support reuse of cross domain references as interesting future work. We intend to employ hierarchical clustering algorithms such as GSOM (Alahakoon, 2000) in automating ontology generation (Chen *et. al.*, 2005).

From the implementation point of view, eHermes PERSONAL needs linking with the original main project (eHermes) as part of the front end. This is another possibility of immediate future work. This could be achieved by implementing the current window based offline system as an online system. By putting the system up on the Web will allow the online users to employ the user model in their purchasing and thereby facilitate the proper evaluation of the model. Furthermore, such implementation will provide the necessary grounds for demonstrating the model in a distributed environment. Implementing in a distributed environment will demonstrate the maximum capabilities of the *LUM*, where each *DI* layer is generated in different points of the system and referring to the common *PI* layer.

We also intend to employ more sophisticated mechanisms to further refine the update strategies of the user model. We may also investigate the possibilities of further refining the value functions used in *PBC* value generation.

Bibliography

Alahakoon, L. D., Halgamuge, S.K., Srinivasan, B.,, (2000), Dynamic Self Organising Maps with Controlled Growth for Knowledge Discovery, *IEEE Transactions on Neural Networks,* VOL 11, No3, pg 600-614.

Alahakoon, O., Loke, S. and Zaslavsky, A., (2003), Modular and hierarchical user profile instantiation for e-commerce applications using mobile software agents, The 14th Australasian Conference on Information Systems (ACIS 03),Perth, Pg 26-28.

Alahakoon, O., Loke, S. and Zaslavsky, A., (2004), Unobtrusive acqusition of user information for e - commerce applications, *6th International Conference on Enterprise Information Systems (ICEIS 2004),* VOL 5, pg 3-8.

Alpert, S. R., Karat, J., Karat, C.-M., Brodie, C. and Vergo, J. G., (2003), User Attitudes Regarding a User Adaptive eCommerce Web Site, *User Modeling and User-Adapted Interactions,* VOL 13, (4), pg 373-396.

Annacker, D., Spiekermann, S. and Strobel, M., (2001), E-privacy: A new search cost dimension in online environments, Proceedings of 14th Bled Conference of Electronic Commerce, Bled, Slovakei.

Ardissono, L., Barbero, C., Goy, A. and Petrone, G., (1999), An Agent Architecture for Personalized Web Stores, Proceedings of 3rd International Conference on Autonomous Agents, Seattle, WA, Pg 182 -189.

Ardissono, L., Console, L. and Torre, I., (2001a), An adaptive system for the personalised access to news, *AI Communications,* VOL 14, (3), pg 129 -147.

Ardissono, L., Gena, C., Torasso, P., Bellifemine, F., Difino, A. and Negro, B., (2004), User Modelling and Recommendation Techniques for Personalized Electronic Program Guides, IN Ardissono, L., Kobsa, A.and Maybury, M. (Eds.) *Personalized Digital Television: Targeting programs to individual users*. Kluwer Academic, VOL 6, Pg. 71-101.

Ardissono, L. and Goy, A., (2000), Tailoring the Interaction with Users in Web Stores, *User Modeling and User-Adapted Interaction,* VOL 10, (4), pg 251--303.

Ardissono, L., Goy, A., Petrone, G. and Segnan, M., (2001b), A software architecture for dynamically generated adaptive Web stores, 17th International Joint Conference on Artificial Intelligence (IJCAI'01), Seattle, Morgan-Kaufman, Pg 1109-1114.

Asuncion, A. and Newman, D. J., (2007), Last sited in 2008, University of California, Irvine, School of Information and Computer Sciences, url:"http://www.ics.uci.edu/$\sim$mlearn/{MLR}epository.html".

Balabanovic, M. and Shoham, Y., (1995), Learning Information Retrieval Agents: Experiments with Automated Web Browsing, Proceedings of the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Resources Stanford, CA.

Bergmann, R. and Cunningham, P., (2002), Acquiring Customers' Requirements in Electronic Commerce, *Artificial Intelligence Review, Special Issue on User Requirements Elicitation in E-Commerce,* VOL 18 3-4, pg 163-193.

Billsus, D. and Pazzani, M., (1999), A Hybrid User Model for News Story Classification, In Proceedings of the Seventh International Conference on User Modeling, Banff, Canada, June, Pg 20-24.

Blum, A., L.Hellerstein and Littlestone, N., (1995), Learning in the Presence of Finitely or Infinitely Many Irrelevant Attributes, *Journal of Computer and System Sciences,* VOL 50(1), pg 32-40.

Boone, G., (1998) Concept feature in Re:Agent, an intelligent e-mail agent, Proceedings of the 2nd International Conference on Autonomous Agents,

Borges, J. and Levene, M., (2007), Evaluating Variable Lenght Markov Chain Models for Analysis of User Web Navigation, *IEEE Transactions on Knowledge and Data Engineering,* VOL 19, (4), pg 441-452.

Brajnik, G. and Tasso, C., (1992), A Flexible Tool for Developing User Modelling Applications with Nonmonotic Reasoning Capabilities, In Proc. of the Third International Worrkshop on User Modeling, Dagstuhl, Germany, Pg 42-66.

Breese, J. S., Heckerman, D. and Kadie, C., (1998), Empirical Analysis of Predictive Algorithms for Collaborative Filtering, Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI), Madison WI: Morgan Kauffman, Pg 43-52.

Brusilovsky, P., (2001), Adaptive Hypermedia, IN Kobsa, A. (Ed.) *User Modeling and User Adapted Interaction, Ten Year Anniversary Issue*. VOL 11 (1/2), Pg. 87-110.

Brusilovsky, P., (2004), KnowledgeTree: A Distributed Architecture for Adaptive E-Learning, Thirteenth World Wide Web Conference WWW 2004, New York, Pg 104 -113.

Burke, R., (2000), Knowledge-based recommender systems, IN Kent, A. (Ed.) *Encyclopedia of Library and Information Systems*. New York, Marcel Dekker, VOL 69, Pg. 370-379.

Burke, R., (2001), Ranking Algorithms for Costly Similarity Metrics, Aha D., Watson I.and Yang, Q. (Eds.), Case-Based Reasoning Research and Development (4th International Conference on Case-Based Reasoning), New York, Springer Verlag, Pg 105-117.

Burke, R., (2002a), Hybrid Reccommender Systems: Survey and Experiments, *User Modelling and User-Adapted Interaction,* VOL 12, pg 331-370.

Burke, R., (2002b), Interactive Critiquing for Catelog Navigation in E-Commmerce, *Artificial Intelligence Review,* VOL 18(3-4), pg 245-267.

Burke, R., Hammond, K. and Young, B., (1997), The FindMe Approach to Assisted Browsing, *IEEE Expert,* VOL 12, (4), pg 32-40.

Chen, S., Alahakoon, D. and Indrawan, M., (2005), Background Knowledge Driven Ontology Discovery, Proceedings of IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05), Pg 202-207.

Chin, D. N., (1989), KNOME:Modelling what the user knows in UC, IN Kobsa, A.and Wahlster, W. (Eds.) *User models in dialog systems*. Springer, VOL 1, Pg. 74-107.

Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D. and Sartin, M., (1999), Combining Content-Based and Collaborative Filters in an Online Newspaper, SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation, Berkeley, CA.

Conati, C., Gertner, A. and Vanlehn, K., (2002) Using Bayesian Networks to Manage Uncertinity in Student Modelling, *user Modeling and User Adapted Interaction,* VOL 12, pg 371-417.

Condliff, M. and Lewis, D., (1999), Bayesian Mixed-Effects Models for Recommender Systems, Proceedings of the ACM SIGIR Workshop on Recommender Systems: Algorithms and Evaluation, 22nd Intl. Conf. on Research and Development in Information Retrieval, Berkeley, California, USA.

Cunningham, P., Bergmann, R., Schmitt, S., Traphöner, R., Breen, S. and Smyth, B., (2001), WEBSELL: Intelligent sales assistants for the World Wide Web, *KI - Künstliche Intelligenz,* VOL 1, pg 28-31.

Cunningham, P. and Smith, B., (1994), A comparison of model based and incremental case based approaches to electronic fault diagnosis, In the proceedings of the 7th Irish Artificial Intelligence and Cognitive science Conference, Belfast, Ireland.

Curtin, R. T., (1982), Indicators of Consumer Behavior: The University of Michigan Surveys of Consumers, *The Public Opinion Quarterly,* VOL 46, (3), pg 340-352.

Devetag, M. G., (1999), From utilities to mental models: A critical survey on decision rules and cognition in consumer choice, *Journal of Industrial and Corporate Change,* VOL 8, pg 289-350.

Doyle, M. and Cunningham, P., (2000), A Dynamic Approach to Reducing Dialog in On-Line Decision Guides, E.Blanzieriand Portinale, L. (Eds.), 5th European Workshop on Case-Based Reasoning (EWCBR 2000), Springer Verlag, Pg 49-60.

Finin, T. W., (1989), GUMS:A General User Modelling Shell, Kobsa, A.and W.Wahlster (Eds.), User Models in Dialog Systems, Berlin, Heidelberg, Spinger-Verlag, Pg 411-430.

Fink, J. and Kobsa, A., (2000), A Review and Analysis of Commercial User Modeling Servers for Personalization on the World Wide Web, *User Modeling and User-Adapted Interaction,* VOL 10 (2/3), pg 209-249.

Foskett, D. J., (1980), Thesaurus, IN A. Kent, H. Lancourand Daily, J. E. (Eds.) *Encyclopedia of Library and Information Science*. VOL 30, Pg. 416-462.

Goldberg, D., Nichols, D., Oki, B. M. and Terry, D., (1992), Using Collaborative filtering to weave an information tapestary, *Communications of the ACM,* VOL 35, pg 61-70.

Graef, G. and Schaefer, C., (2001), Application of ART2 Networks and Self-Organizing Maps to Colaborative Filtering, In revised papers from the Workshops OHS-7, SC-3, AH-3 on Hypermedia: Openness, Structural Awareness and Adaptivity , London, UK, Springer, Pg 296-309.

Guttman, R. H. and Maes, P., (1999), Agent-Mediated Integrative Negotiation for Retail Electronic Commerce, *Lecture Notes in Computer Science,* VOL 1571, pg 70--90.

Guttman, R. H., Moukas, A. G. and Maes, P., (1998), Agent-mediated electronic commerce: a survey, *Knowledge Engineering Review,* VOL 13(2), pg 143--152.

Hagen, P. R., Manning, H. and Paul, Y., (2000), Last sited in 2008, The Forrester Report, url:"http://www.cse.iitb.ac.in/banks/Private/RelatedPapers/intuifind-stink.pdf".

Hajek, P., (2002), Last sited in 2008, Stanford Encyclopaedia of Philosophy, url:"http://plato.stanford.edu/entries/logic-fuzzy/".

Haykin, S., (1998), *'Neural Networks a comprehensive foundation'*, Published by Second Edition, Published by Prentice-Hall, Inc.

Hebb, D., (1949), *The Organization of Behavior: A Neuropsychological Theory,* NewYork, Published by Wiley.

Heckmann, D., (2003), A Specialized Representation for Ubiquitous Computing and User Modeling, *First Workshop on User Modeling for Ubiquitous Computing, in conjunction with UM 2003, University of Pittsburgh at Johnstown, USA,* VOL pg.

Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M. and von-Wilamowitz-Moellendorff, M., (2005), GUMO - the General User Model Ontology, *Proceedings of the 10th International Conference on User Modeling (UM'2005), Edinburgh, UK, Springer-Verlag Berlin Heidelberg, LNAI 3538,* VOL pg 428-432.

Jameson, A., (2001), Personalization for E-Commerce, Half-day tutorial presented at UM 2001, The Eighth International Conference on User Modeling, Sonthofen, Germany

Jayaputera, G., Alahakoon, O., Cruz, L., Loke, S. and Zaslavsky, A., (2003a), Assembling Agents On-Demand for Pervasive Wireless Services, Mahmoud, Q. H. (Ed.) The Second International Workshop on Wireless Information Services (WIS 2003), Angers-France, ICEIS Press, Pg 21-30.

Jayaputera, G. T., Loke, S. W. and Zaslavsky, A., (2003b), Mission Impossible? Automatically Assembling Agents from High-Level Task Descriptions, The 2003 IEEE/WIC International Conference on Intelligent Agent Technology (IAT 2003), IEEE Computer Society, Halifax, Canada, Pg 161-167.

Jayaputera, G. T., Loke, S. W., Zaslavsky, A., (2005a), Just in Time Mobile Agent Generation and Management, The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-05), Utrecht, Netherlands, ACM Press, Pg 1237-1238.

Jayaputera, G. T., Zaslavsky, A. and Loke, S. W., (2004), Run-Time Mission Evolution in Mobile Multiagent Systems, IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT2004), Beijing, China, IEEE Computer Society, Pg 79-85.

Jayaputera, G. T., Zaslavsky, A. and Loke, S. W. (2005). An Approach to, (2005b), An Approach to Dynamically Generated User-Specified MAS, The Sixth International Workshop on Agent-Oriented Software Engineering (AOSE-2005), LNCS 3950, Berlin, Germany, Springer-Verlag, Pg 139-153.

Jennings, A. and Higuchi, H., (1992) A personal news service based on a user model neural network, *IEICE Transactions on Information and Systems,* VOL 75, (2), pg 198-209.

Jennings, A. and Higuchi, H., (1993), A User Model Neural Network for a Personal News Services, *User Modeling and User Adapted Interactions, Springer Verlag,* VOL 3, (1), pg 1-25.

Jin, R. and Si, L., (2004), A study of method for normalising user ratings in collaborative filtering, SIGIR '04: Proceedings of the 27th annual International ACM SIGIR Conference on Research and Development on Information Retrieval, Sheffield, UK, Sheffield, UK.

Katona, G., (1968), Consumer Behavior: Theory and Findings on Expectations and Aspirations, *The American Economic Review,Papers and Proceedings of the Eightieth Annual Meeting of the American Economic Association.,* VOL 58, (2), pg 19-30.

Kay, J., (1994), Invited keynote address - Lies, damned lies and stereotypes: pragmatic approximations of users, Kobsa, A. a. D. L. (Ed.) Proceedings of the Fourth International Conference on User Modeling, Mitre, Pg 175-184.

Kay, J., (1995), The UM toolkit for reusable, long term user models, *User Modeling and UserAdapted Interaction,* VOL 4(3), pg 149 -196.

Kay, J., (2000), Stereotypes, Student Models and Scrutability, IN Gauthier, G., Frasson, C.and VanLehn, K. (Eds.) *In Proceedings of the 5th international Conference on intelligent Tutoring Systems (June 19 - 23, 2000),Lecture Notes In Computer Science*. Springer-Verlag, London, VOL 1839, Pg. 19-30.

Kay, J., Kummerfeld, R. J. and Lauder, P., (2002), Personis: A Server for User Models, IN De Bra P, Brusilovsky Pand Conejo R (Eds.) *Adaptive Hypermedia*. Springer-Verlag, VOL.2347, Pg. 203-212.

Kay, J., Kummerfeld, R. J. and Lauder, P., (2003), Managing private user models and shared personas, Workshop on User Modelling for Ubiquitous Computing in conjunction with User Modeling 2003 Pittsburgh, PA, USA, Pg 22-23.

Kelly, S., (2006), *Customer Intelligence from Data to Dialogue,* West Sussex, England, Published by John Wiley & Sons Ltd.

Kobsa, A., (2007), Generic User Modelling Systems, IN Brusilovsky, P., Kobsa, A.and Nejdl, W. (Eds.) *The Adaptive Web: Methods and Stratergies of Web Personalization*. Berlin, Heidelberg, NewYork, Springer Verlag, VOL 4321/2007, Pg. 136-154.

Kobsa, A. and Pohl, W., (1995), The user modeling shell system bgp-ms, *User Modeling and User-Adapted Interaction,* VOL 4(2), pg 59--106.

Kohavi, R., Brodley, C., Frasca, B., Mason, L. and Zheng, Z., (2000), KKD_Cup 2000 Organizers' Report: Peeling the Onion, *SIGKDD Explorations,* VOL 2, (2), pg 86-98.

Konstan, J. A., Miller, B.N, Maltz, D., Herlocker, J.L., Gordon, L.R., Riedl, J., (1997), GroupLens: Applying Collaborative Filtering to Usenet News, *Commun. ACM,* VOL 40(3), pg 77-87.

Krulwich, B., (1997), Lifestyle finder: intelligent user profiling using large-scale demographic data, *AI Magazine,Summer,* VOL 18(2), pg 37-45.

Kucukemiroglu, O., (1997), Market segmentation by using consumer lifestyle dimensions and ethnocentrism: An empirical study, *European Journal of Marketing,* VOL 33,5/6, pg 470-487.

Lawrence, R. D., Almasi, G. S., Kotlyar, V., Viveros, M. S. and Duri, S. S., (2001) Personalization of supermarket product recommendations, IN Kohavi, R.and Provost, F. (Eds.) *Data Mining and Knowledge Discovery*. VOL 5, Pg. 11-32.

Lieberman, H., (1995), Letizia: An Agent that Assists Web Browsing, Proceedinds of the 1995 International Joint Conference on Artificial Intelligence, Montreal, Canada.

Littelestone, N. and Warmuth, M., (1994), The Weighted Majority Algorithm, *Information and Computation,* VOL 108(2), pg 212-261.

Maes, P., Guttman, R., Moukas, A. and Zacharia, G., (1989), Last sited in url:"http://web.media.mit.edu/~pattie/ECOM/sld018.htm".

Middleton, S. E., Alani, H. and Roure, D. C. D., (2002), Exploiting Synergy Between Ontologies and Recommender Systems, In Proc. of the World Wide Web Conference (WWW2002) International Workshop on the Semantic Web, Hawaii, USA.

Middleton, S. E., Roure, D. C. D. and Shadbolt, N. R., (2001), Capturing Knowledge of User Preferences: Ontologies on recommender systems, In Proceedings of the First International Conference on Knowledge Capture (K-CAP2001), Victoria, B.C., Canada.

Negnevitsky, M., (2005), *Artificial Intelligence,* Harlow, England, 2, Published by Addisson Wesley.

Orwant, J., (1991), Doppelgnger goes to School: Machine Learning for User Modeling, IN *Program in Media Arts and Sciences.* MIT.

Orwant, J., (1995), Heterogeneous Learning in the Doppelgnger User Modelling System, *User Modelling and User-Adapted Interaction,* VOL 4, pg 107-130.

Paiva, A. and Self, J., (1994), TAGUS- A User and Learner Modeling System, Proceedings of the 4th International Conference on User Modeling , UM Inc, Cape Cod.

Paliouras, G., Karkaletsis, V., Papatheodorou, C. and Spyropoulos, C., (1999), Exploiting Learning Techniques for the Acquisition of User Stereotypes and Communities, Proceedings of the Seventh International Conference on User Modelling (UM '99), Banff, Canada, Pg 169-178.

Patel, S. and Schlijper, A., (2003), Last sited in 2008 url:"http://www.maths.ox.ac.uk/ociam/StudyGroups/ESGI49/problems/unilever2/unilever2.pdf".

Pazzani, M., Muramatsu, J. and Billsus, D., (1996), Syskill & Webert:Identifying interesting web sites, IN Weld, D.and Clancey, B. (Eds.) *Proceeding of the 13th National Conference on Artificial Intelligence*. Menlo Park; NJ, AAAI Press, VOL 1, Pg. 54-61.

Pazzani, M. J., (1999), A Framework for Collaborative, Content-Based and Demographic Filtering, *Artificial Intelligence Review,* VOL 13, (5/6), pg 393-408.

Prasad, B., (2005), HYREC: A Hybrid Recommendation System for E-Commerce, *Case-Based Reasoning Research and Development, Lecture Notes in Computer Science,* VOL 3620/2005, pg 408-420.

Pu, P., (2004), Effective Interaction Principles for Online Product Search Environments, Proceedings of the IEEE/WIC/ACM international Conference on Web Intelligence (WI'04), Beijing, China.

Pu, P. and Kumar, P., (2004), Evaluating example-based search tools, EC '04: Proceedings of the 5th ACM conference on Electronic commerce, New York, NY, USA, ACM, Pg 208--217.

Pu, P., Viappiani, P. and Faltings, B., (2006), Increasing User Decision Accuracy Using Suggestions, In Proceedings of the SIGCHI conference on Human factors in computing systems (CHI'06) Montreal, Canada, Pg 121-130.

Rashid, A. M., Albert, I., Cosley, D., Lam, S. K., McNee, S., J.A.Konstan and Riedl, J., (2002), Getting to know you: Learning New User Preferences in Recommender Systems, In Proceedings of the International Conference on Intelligent User Interfaces, San Francisco, CA, Pg 127-134.

Reilly, J., Zhang, J., McGinty, L., Pu, P. and Smyth, B., (2007), Evaluating Compound Critiquing Recommenders: A Real-User Study, In Proceedings of ACM Conference on Electronic Commerce (EC'07), San Diego, USA.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J., (1994), GroupLens: An Open Architecture for Collaborative Filtering of Netnews, Conference on Computer Supported Coorperative Work, Chapel Hill, ACM, Pg 175-186.

Rich, E., (1979), User modeling via stereotypes, *Cognitive Science,* VOL 3, pg 329-354.

Rich, E., (1983), Users are Individuals: Individualizing User Models., *International Journal of man machine studies,* VOL 18, pg 199-214.

Rich, E., (1989), Stereotypes and user modeling, IN Kobsa, A.and Wahlster, W. (Eds.) *User Models in Dialog Systems*. Berlin, Heidelberg, Springer-Verlag, VOL 1, Pg. 35 - 51.

Russel, S. and Norvig, P., (1995), *Artificial Intelligence: a modern approach*, Published by Prentice Hall.

Salton, G., (1989), *Automatic Text Processing*, Published by Addison-Wesley.

Salton, G. and McGill, M., (1983), An Introduction to Modern Information Retrieval, VOL pg.

Sandvig, J. J., Mobasher, B. and Burke, R., (2007), Impact of Relevance Measures on the Robustness and Accuracy of Collaborative Filtering, IN *Lecture Notes in Computer Science*. Springer, VOL 4655/2007, Pg. 99-108.

Sarwar, B. M., Karypis, G., Konstan, J. A. and Riedl, J., (2001), Item-based collaborative filtering recommendation algorithms, In Proc. of the 10th International World Wide Web Conference (WWW10), Hong Kong.

Schafer, J. B., Frankowski, D., Herlocker, J. and Sen, S., (2007), Collaborative Filtering Recommender Systems, IN Brusilovsky, P., Kobsa, A.and Nejdl, W. (Eds.) *The Adaptive Web:Methods and Strategies of Web Personalization*. Berlin Heidelberg NewYork, Springer Verlag, VOL 4321/2007, Pg. 291-324.

Schafer, J. B., Konstan, J. A. and Riedl, J., (2001), E-Commerce Recommendation Applications, *Data Mining and Knowledge Discovery,* VOL 5, pg 115-153.

Schafer, R., (2001), Last sited in 2008, Online proceedings of ABIS-Workshop, url:"http://www.kbs.uni-hannover.de/%7Ehenze/ABIS_Workshop2001/ABIS_2001.html".

Schmitt, S. and Bergmann, R., (2001), A Formal Approach to Dialogs with Online Customers, Proc. of the 14 th Bled Electronic Commerce Conference, Bled, Slovenia.

Schmitt, S., Dopichaj, P. and Domínguez-Marín, P., (2002), Entropy-Based vs. Similarity-Influenced: Attribute Selection Methods for Dialogs Tested on Different Electronic Commerce Domains, Proceedings of the 5th European Confernce on Case Based Reasoning (ECCBR), Aberdeen, Scotland, UK, Pg 380-394.

Schwab, I. and Kobsa, A., (2002), Adaptivity through Unobtrusive Learning, *KI, Special Issue on Adaptivity and User Modeling,* VOL 3, pg 5-9.

Shardanand, U. and Maes, P., (1995), Social Information Filtering: Algorithms for Automating ``Word of Mouth", Proceedings of the Computing-Human interaction Conference (CHI 95), Denver,CO, Pg 210--217.

Shearin, S., (1999), Developing a Data Representation for Electronic Profiles, IN., MAS 964, Special Topics in Media Technology (E-markets).

Solomon, M. R., Dann, S., Dann, S. and Russell-Bennett, R., (2007), *Consumer Behaviour:Buying, Having, Being*, Published by Pearson Education Australia.

Spiekermann, S., Grossklags, J. and Berendt, B., (2001), E-privacy in 2nd generation E-Commerce: Privacy Preferences versus actual Behavior, Proceedings of ACM Conference on Electronic Commerce (EC'01), Tampa, Florida.

Thompson, C. A., Göker, M. and Langley, P., (2002), A personalized system for conversational recommendations, *Journal of Artificial Intelligence Research.,* VOL 21, pg 393-428.

Towle, B. and Quinn, C., (2000), Knowledge Based Recommender Systems Using Explicit User Models, Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, AAAI Technical Report WS-00-04, Menlo Park, CA, AAAI Press, Pg 74-77.

Tversky, A. and Kahneman, D., (1981), The Framing of Decisions and the Psychology of Choice, *Science,* VOL 211(4481), pg 453-458.

Viappiani, P., Faltings, B. and Pu, P., (2006), Preference-based search using example-critiquing with suggestions, *Journal of Artificial Intelligence Research(JAIR),* VOL 27, pg 465-503.

Vozalis, M. and Margaritis, K. G., (2004), Enhancing collaborative filtering with demographic data: The case of item-based filtering, Proceedings of the Fourth IEEE International Conference on Intelligent Systems Design and Applications (ISDA 2004), Budapest, Hungary, Pg 361-366.

Wahlster, W. and Kobsa, A., (1989), User Models in Dialog Systems, IN Kobsa, A.and Wahlster, W. (Eds.) *User Models in Dialog Systems*. Berlin, Heidelberg, Springer-Verlag, VOL 1, Pg. 4-33.

Yan, J., Ryan, M. and Power, J., (1994), *Using Fuzzy Logic:Towards intelligent systems*, Published by Prentice Hall International (UK) Limited.

Zadeh, L. A., (1965), Fuzzy Sets, *Information and Control,* VOL 8, pg 338-353.

Zaslavsky, A., Bartolini, C., Boulmakoul, A., Alahakoon, O., Loke, S. W. and Burstein, F., (2007), Enhancing the IT service desk function through unobtrusive user profiling, personalization and stereotyping, HP Softwear University Association, 14th Workshop, Garching/Munich, Germany.

Zukerman, I. and Albrecht, D., (2001), Predictive Statistical Models for User Modeling, *User Modeling and User-Adapted Interaction, Invited paper,* VOL 11, (1-2), pg 5 - 18.

Appendix A

Influence Matrix for Restaurant Domain

Feature Number	Attribute Number	Attribute Name	Attri Type	Time Saver (Ts)	Ts Weight	Price Sensitivity (Ps)	Ps weight	Quality Consciousness (Qc)	Qc Weight	Fun Spending (Fs)	Fs Weight	Health Consciousness (Hc)	Hc Weight	Family Person (Fp)	Fp Weight	Socialising (So)	So Weight	Adventurer (Ad)	Ad Weight
2	163	below $15	c		0	0.75,1	1		0		0		0		0		0		0
2	165	$15-$30	c		0	0.5,0.75	1		0		0		0		0		0		0
2	167	$30-$50	c		0	0.25,0.5	1		0		0		0		0		0		0
2	169	over $50	c		0	0,0.25	1		0		0		0		0		0		0
3	50	Poor Decor	c		0		0	0,0.15	0.5		0		0		0	0,0.15	0.5		0
3	51	Fair Decor	c		0		0	0.15,0.35	0.5		0		0		0	0.15,0.35	0.5		0
3	52	Good Decor	c		0		0	0.35,0.55	0.5		0		0		0	0.35,0.55	0.5		0
3	53	Excellent Decor	c		0		0	0.55,0.7	0.5		0		0		0	0.55,0.7	0.5		0
3	54	Extraordinary Decor	c		0		0	0.7,0.85	0.5		0		0		0	0.7,0.85	0.5		0
3	55	Near-perfect Decor	c		0		0	0.85,1	0.5		0		0		0	0.85,1	0.5		0
4	73	Fair Food	c		0		0	0,0.35	1		0		0		0		0		0
4	74	Good Food	c		0		0	0.35,0.55	1		0		0		0		0		0
4	75	Excellent Food	c		0		0	0.55,0.7	1		0		0		0		0		0
4	76	Extraordinary Food	c		0		0	0.7,0.85	1		0		0		0		0		0

4	77	Near-perfect Food	c		0		0	0.85,1	1		0		0		0		0		0
7	203	Fair Service	c		0		0	0,0.35	1		0		0		0		0		0
7	204	Good Service	c		0		0	0.35,0.55	1		0		0		0		0		0
7	205	Excellent Service	c		0		0	0.55,0.7	1		0		0		0		0		0
7	206	Extraordinary Service	c		0		0	0.7,0.85	1		0		0		0		0		0
7	207	Near-perfect Service	c		0		0	0.85,1	1		0		0		0		0		0
10	137	Long Drive	c	0.25,0.5	1		0		0		0		0		0		0		0
10	214	Short Drive	c	0.5,0.75	1		0		0		0		0		0		0		0
10	247	Walk	c	0,0.25	1		0		0		0		0		0		0		0
11	100	Good for Younger Kids	d		0		0		0		0		0	0.65,1	1		0		0
27	111	Health Conscious Menus	d		0		0		0		0	0.65,1	1		0		0		0
27	112	Health Food	d		0		0		0		0	0.65,1	1		0		0		0
24	59	Delivery Available	d	0.65,1	1		0		0		0		0		0		0		0
12	174	Parking/Valet	d		0		0	0.65,1	1		0		0		0		0		0
28	193	Prix Fixe Menus	d		0		0		0		0		0		0		0	0.65,1	1
17	136	Little Known But Well Liked	d		0		0		0		0		0		0	0,0.65	0.5	0.65,1	0.5
17	178	People Keep Coming Back	d		0		0		0		0		0		0	0.65,1	1		0
17	242	Up and Coming	d		0		0		0		0		0		0		0	0.65,1	1
21	150	No Smoking Allowed	d		0		0		0		0	0.65,1	0.5	0.65,1	0.5		0		0
22	80	Fabulous Wine Lists	d		0		0		0	0.65,1	0.5		0		0	0.65,1	0.5		0
22	148	No Liquor Served	d		0		0		0		0	0.65,1	0.5	0.65,1	0.5		0		0
22	35	Carry in Wine and Beer	d		0		0		0	0.65,1	0.5		0		0	0.65,1	0.5		0
22	254	Wine and Beer	d		0		0		0	0.65,1	0.5		0		0	0.65,1	0.5		0
16	1	Authentic	d		0		0		0		0		0		0		0		0
1	2	Afghanistan	d		0		0		0		0		0		0		0		0
1	3	African	d		0		0		0		0		0		0		0		0
6	4	After Hours Dining	d		0		0		0		0		0		0		0		0
1	5	American (Contemporary)	d		0		0		0		0		0		0		0		0
1	6	American (New)	d		0		0		0		0		0		0		0		0
1	7	American (Regional)	d		0		0		0		0		0		0		0		0

1	8	American (Traditional)	d		0		0		0		0		0		0		0		0
1	9	American	d		0		0		0		0		0		0		0		0
11	10	An Historic Spot	d		0		0		0		0		0		0		0		0
11	11	An Out Of The Way Find	d		0		0		0		0		0		0		0		0
1	12	Argentinean	d		0		0		0		0		0		0		0		0
1	13	Armenian	d		0		0		0		0		0		0		0		0
1	14	Asian	d		0		0		0		0		0		0		0		0
1	15	Austrian	d		0		0		0		0		0		0		0		0
5	16	Bakeries	d		0		0		0		0		0		0		0		0
5	17	Bar-B-Q	d		0		0		0		0		0		0		0		0
1	18	Belgian	d		0		0		0		0		0		0		0		0
12	171	Parking Lot Available	d		0		0		0		0		0		0		0		0
13	253	Wheelchair Access	d		0		0		0		0		0		0		0		0
9	19	Brasserie	d		0		0		0		0		0		0		0		0
1	20	Brazilian	d		0		0		0		0		0		0		0		0
30	21	Buffet Dining	d		0		0		0		0		0		0		0		0
1	22	Burmese	d		0		0		0		0		0		0		0		0
15	23	Burritos	d		0		0		0		0		0		0		0		0
11	24	Business Scene	d		0		0		0		0		0		0		0		0
16	25	Creative	d		0		0		0		0		0		0		0		0
5	26	Cabin	d		0		0		0		0		0		0		0		0
5	27	Cafe/Espresso Bars	d		0		0		0		0		0		0		0		0
5	28	Cafe/Garden Dining	d		0		0		0		0		0		0		0		0
5	29	Cafeterias	d		0		0		0		0		0		0		0		0
1	31	Californian	d		0		0		0		0		0		0		0		0
1	32	Cambodian	d		0		0		0		0		0		0		0		0
1	33	Canadian	d		0		0		0		0		0		0		0		0
1	34	Caribbean	d		0		0		0		0		0		0		0		0
26	36	Catering for Special Events	d		0		0		0		0		0		0		0		0
15	37	Caviar	d		0		0		0		0		0		0		0		0
10	38	Central	c	0.75,1	1		0		0		0		0		0		0		0

1	39	Chinese	d		0		0		0		0		0		0		0		0
11	40	Classic Hotel Dining	d		0		0		0		0		0		0		0		0
5	41	Coffee Houses	d		0		0		0		0		0		0		0		0
5	42	Coffee Shops	d		0		0		0		0		0		0		0		0
9	43	Coffee and Dessert	d		0		0		0		0		0		0		0		0
5	44	Coffeehouses	d		0		0		0		0		0		0		0		0
16	45	Continental	d		0		0		0		0		0		0		0		0
20	46	Credit cards are not accepted	d		0		0		0		0		0		0		0		0
1	48	Cuban	d		0		0		0		0		0		0		0		0
14	57	Dancing	d		0		0		0		0		0		0		0		0
5	58	Deli	d		0		0		0		0		0		0		0		0
15	60	Dim Sum	d		0		0		0		0		0		0		0		0
5	61	Diners	d		0		0		0		0		0		0		0		0
6	62	Dining After the Theater	d		0		0		0		0		0		0		0		0
11	63	Dining Outdoors	d		0		0		0		0		0		0		0		0
6	66	Early Dining	d		0		0		0		0		0		0		0		0
1	67	Eastern European	d		0		0		0		0		0		0		0		0
16	68	Eclectic	d		0		0		0		0		0		0		0		0
1	69	Egyptian	d		0		0		0		0		0		0		0		0
1	70	English	d		0		0		0		0		0		0		0		0
14	71	Entertainment	d		0		0		0		0		0		0		0		0
1	72	Ethiopian	d		0		0		0		0		0		0		0		0
11	79	Fabulous Views	d		0		0		0		0		0		0		0		0
5	81	Fast Food	d		0		0		0		0		0		0		0		0
1	82	Filipino	d		0		0		0		0		0		0		0		0
11	83	Fine for Dining Alone	d		0		0		0		0		0		0		0		0
31	84	Focus on Dessert	d		0		0		0		0		0		0		0		0
9	85	Fondue	d		0		0		0		0		0		0		0		0
11	86	For the Young and Young at Heart	d		0		0		0		0		0		0		0		0
5	87	Fountain and Ice Cream	d		0		0		0		0		0		0		0		0
1	88	Franco-Russian	d		0		0		0		0		0		0		0		0

9	89	Frankfurters	d		0		0		0		0		0		0		0		0
1	90	French (New)	d		0		0		0		0		0		0		0		0
1	91	French	d		0		0		0		0		0		0		0		0
1	92	French Bistro	d		0		0		0		0		0		0		0		0
1	93	French Classic	d		0		0		0		0		0		0		0		0
1	94	French Contemporary	d		0		0		0		0		0		0		0		0
1	95	French Nouvelle	d		0		0		0		0		0		0		0		0
12	172	Street Parking Available	d		0		0		0		0		0		0		0		0
1	96	French-Japanese	d		0		0		0		0		0		0		0		0
14	97	Game	d		0		0		0		0		0		0		0		0
1	98	German	d		0		0		0		0		0		0		0		0
11	99	Good Out of Town Business	d		0		0		0		0		0		0		0		0
11	114	Hip Place To Be	d		0		0		0		0		0		0		0		0
11	101	Great Place to Meet for a Drink	d		0		0		0		0		0		0		0		0
11	102	Great for People Watching	d		0		0		0		0		0		0		0		0
1	103	Greek	d		0		0		0		0		0		0		0		0
9	104	Grills	d		0		0		0		0		0		0		0		0
1	105	Guatemalan	d		0		0		0		0		0		0		0		0
9	106	Hamburgers & Beer	d		0		0		0		0		0		0		0		0
9	107	Hamburgers	d		0		0		0		0		0		0		0		0
31	113	High Tea	d		0		0		0		0		0		0		0		0
9	115	Hot Dogs	d		0		0		0		0		0		0		0		0
1	116	Hungarian	d		0		0		0		0		0		0		0		0
1	117	Indian	d		0		0		0		0		0		0		0		0
1	118	Indonesian	d		0		0		0		0		0		0		0		0
16	119	International	d		0		0		0		0		0		0		0		0
1	120	Irish	d		0		0		0		0		0		0		0		0
1	121	Italian (North & South)	d		0		0		0		0		0		0		0		0
1	122	Italian (North	d		0		0		0		0		0		0		0		0
1	123	Italian (Northern)	d		0		0		0		0		0		0		0		0
1	124	Italian (Southern)	d		0		0		0		0		0		0		0		0

1	125	Italian	d		0		0		0		0		0		0		0		0
1	127	Jamaican	d		0		0		0		0		0		0		0		0
1	128	Japanese	d		0		0		0		0		0		0		0		0
1	129	Jewish	d		0		0		0		0		0		0		0		0
1	130	Korean	d		0		0		0		0		0		0		0		0
1	131	Kosher	d		0		0		0		0		0		0		0		0
28	132	Late Night Menu	d		0		0		0		0		0		0		0		0
1	133	Latin	d		0		0		0		0		0		0		0		0
1	134	Lebanese	d		0		0		0		0		0		0		0		0
1	135	Lithuanian	d		0		0		0		0		0		0		0		0
1	139	Malaysian	d		0		0		0		0		0		0		0		0
1	140	Mediterranean	d		0		0		0		0		0		0		0		0
13	141	Menus in Braille	d		0		0		0		0		0		0		0		0
1	142	Mexican	d		0		0		0		0		0		0		0		0
1	143	Middle Eastern	d		0		0		0		0		0		0		0		0
1	144	Moroccan	d		0		0		0		0		0		0		0		0
11	153	Old World Cafe Charm	d		0		0		0		0		0		0		0		0
23	146	Need To Dress	d		0		0		0		0		0		0		0		0
1	147	Nicaraguan	d		0		0		0		0		0		0		0		0
19	149	No Reservations	d		0		0		0		0		0		0		0		0
5	151	Noodle Houses	d		0		0		0		0		0		0		0		0
5	152	Noodle Shops	d		0		0		0		0		0		0		0		0
9	154	Omelettes	d		0		0		0		0		0		0		0		0
11	155	On the Beach	d		0		0		0		0		0		0		0		0
18	156	Open for Breakfast	d		0		0		0		0		0		0		0		0
18	157	Open on Mondays	d		0		0		0		0		0		0		0		0
18	158	Open on Sundays	d		0		0		0		0		0		0		0		0
31	159	Other Quick Food	d		0		0		0		0		0		0		0		0
5	160	Oyster Bars	d		0		0		0		0		0		0		0		0
9	173	Pancakes	d		0		0		0		0		0		0		0		0
8	175	Parties and Occasions	d		0		0		0		0		0		0		0		0
9	176	Pastries	d		0		0		0		0		0		0		0		0

5	177	Pastry Shops	d		0		0		0		0		0		0		0		0
1	179	Persian	d		0		0		0		0		0		0		0		0
1	180	Peruvian	d		0		0		0		0		0		0		0		0
9	181	Picnics	d		0		0		0		0		0		0		0		0
9	182	Pizza	d		0		0		0		0		0		0		0		0
5	183	Pizzerias	d		0		0		0		0		0		0		0		0
11	184	Place for Singles	d		0		0		0		0		0		0		0		0
1	186	Polish	d		0		0		0		0		0		0		0		0
1	187	Polynesian	d		0		0		0		0		0		0		0		0
1	188	Portuguese	d		0		0		0		0		0		0		0		0
6	190	Pre-theater Dining	d		0		0		0		0		0		0		0		0
8	191	Private Parties	d		0		0		0		0		0		0		0		0
13	192	Private Rooms Available	d		0		0		0		0		0		0		0		0
11	194	Pub Feel	d		0		0		0		0		0		0		0		0
11	196	Quiet for Conversation	d		0		0		0		0		0		0		0		0
11	197	Quirky	d		0		0		0		0		0		0		0		0
11	198	Relaxed Senior Scene	d		0		0		0		0		0		0		0		0
1	199	Romanian	d		0		0		0		0		0		0		0		0
11	200	Romantic	d		0		0		0		0		0		0		0		0
1	201	Roumanian	d		0		0		0		0		0		0		0		0
1	202	Russian	d		0		0		0		0		0		0		0		0
1	209	Salvadoran	d		0		0		0		0		0		0		0		0
1	210	Scandinavian	d		0		0		0		0		0		0		0		0
1	211	Scottish	d		0		0		0		0		0		0		0		0
9	212	Seafood	d		0		0		0		0		0		0		0		0
14	213	See the Game	d		0		0		0		0		0		0		0		0
16	216	Soul Food	d		0		0		0		0		0		0		0		0
16	217	Soulfood	d		0		0		0		0		0		0		0		0
1	218	South American	d		0		0		0		0		0		0		0		0
1	219	Southeast Asian	d		0		0		0		0		0		0		0		0
1	222	Southwestern	d		0		0		0		0		0		0		0		0
1	223	Spanish	d		0		0		0		0		0		0		0		0

28	224	Special Brunch Menu	d		0		0		0		0		0		0		0		0
5	225	Steakhouses	d		0		0		0		0		0		0		0		0
15	226	Sushi	d		0		0		0		0		0		0		0		0
1	227	Swiss	d		0		0		0		0		0		0		0		0
1	228	Swiss-French	d		0		0		0		0		0		0		0		0
16	229	Traditional	d		0		0		0		0		0		0		0		0
25	231	Takeout Available	d		0		0		0		0		0		0		0		0
1	235	Thai	d		0		0		0		0		0		0		0		0
1	236	Tibetan	d		0		0		0		0		0		0		0		0
11	237	Tourist Appeal	d		0		0		0		0		0		0		0		0
1	238	Tunisian	d		0		0		0		0		0		0		0		0
1	239	Turkish	d		0		0		0		0		0		0		0		0
1	240	Ukrainian	d		0		0		0		0		0		0		0		0
1	241	Ukranian	d		0		0		0		0		0		0		0		0
29	243	Vegetarian	d		0		0		0		0		0		0		0		0
1	244	Venezuelan	d		0		0		0		0		0		0		0		0
19	245	Very Busy - Reservations a Must	d		0		0		0		0		0		0		0		0
1	246	Vietnamese	d		0		0		0		0		0		0		0		0
11	248	Warm spots by the fire	d		0		0		0		0		0		0		0		0
6	249	Weekend Brunch	d		0		0		0		0		0		0		0		0
6	250	Weekend Dining	d		0		0		0		0		0		0		0		0
14	251	Weekend Jazz Brunch	d		0		0		0		0		0		0		0		0
6	252	Weekend Lunch	d		0		0		0		0		0		0		0		0
5	255	Yogurt Bar	d		0		0		0		0		0		0		0		0
1	256	Yugoslavian	d		0		0		0		0		0		0		0		0

Appendix B

Influence Matrix for Leg-wear Domain

Feature Number	Attribute Number	Attribute Name	Attri Type	Time Saver (Ts)	Ts Weight	Price Sensitivity (Ps)	Ps weight	Quality Consciousness (Qc)	Qc Weight	Fun Spending (Fs)	Fs Weight	Health Consciousness (Hc)	Hc Weight	Family Person (Fp)	Fp Weight	Socialising (So)	So Weight	Adventurer (Ad)	Ad Weight
1	1	Low (UnitPrice <= 5)	c	-	0	0.85,1	1	-	0	-	0	-	0	-	0	-	0	-	0
1	2	Low-Medium(UnitPrice > 5 AND UnitSalePrice <= 10)	c	-	0	0.7,0.85	1	-	0	-	0	-	0	-	0	-	0	-	0
1	3	Medium(UnitSalePrice > 10 AND UnitSalePrice <= 15)	c	-	0	0.55,0.7	1	-	0	-	0	-	0	-	0	-	0	-	0
1	4	Med_High(UnitSalePrice > 15 AND UnitSalePrice <= 25)	c	-	0	0.35,0.55	1	-	0	-	0	-	0	-	0	-	0	-	0
1	5	High(UnitSalePrice > 25)	c	-	0	0,0.35	1	-	0	-	0	-	0	-	0	-	0	-	0
2	6	AME	c	-	0	-	0	0.75,1	0.5	-	0	-	0	-	0	0.75,1	0.5	-	0
2	7	DAN	c	-	0	-	0	0.5,0.75	0.5	-	0	-	0	-	0	0.5,0.75	0.5	-	0
2	8	DKNY	c	-	0	-	0	0.5,0.75	0.5	-	0	-	0	-	0	0.5,0.75	0.5	-	0
2	9	BER	c	-	0	-	0	0.5,0.75	0.5	-	0	-	0	-	0	0.5,0.75	0.5	-	0
2	10	ELT	c	-	0	-	0	0.75,1	0.5	-	0	-	0	-	0	0.75,1	0.5	-	0
2	11	GIV	c	-	0	-	0	0,0.25	0.5	-	0	-	0	-	0	0,0.25	0.5	-	0
2	13	HPK	c	-	0	-	0	0,0.25	0.5	-	0	-	0	-	0	0,0.25	0.5	-	0

2	14	HOSO	c	-	0	-	0	0.75,1	0.5	-	0	-	0	-	0	0.75,1	0.5	-	0
2	15	NM	c	-	0	-	0	0.5,0.75	0.5	-	0	-	0	-	0	0.5,0.75	0.5	-	0
2	16	Hanes Too	c	-	0	-	0	0.25,0.5	0.5	-	0	-	0	-	0	0.25,0.5	0.5	-	0
2	17	DON	c	-	0	-	0	0.25,0.5	0.5	-	0	-	0	-	0	0.25,0.5	0.5	-	0
2	18	Silk Reflections	c	-	0	-	0	0.75,1	0.5	-	0	-	0	-	0	0.75,1	0.5	-	0
2	19	Smooth Illusions	c	-	0	-	0	0,0.25	0.5	-	0	-	0	-	0	0,0.25	0.5	-	0
2	20	ORO	c	-	0	-	0	0.25,0.5	0.5	-	0	-	0	-	0	0.25,0.5	0.5	-	0
2	21	EVP	c	-	0	-	0	0.5,0.75	0.5	-	0	-	0	-	0	0.5,0.75	0.5	-	0
2	22	HUE	c	-	0	-	0	0,0.25	0.5	-	0	-	0	-	0	0,0.25	0.5	-	0
2	23	Absolutely Ultra Sheer	c	-	0	-	0	0.25,0.5	0.5	-	0	-	0	-	0	0.25,0.5	0.5	-	0
2	24	FAL	c	-	0	-	0	0,0.25	0.5	-	0	-	0	-	0	0,0.25	0.5	-	0
2	25	Alive	c	-	0	-	0	0,0.25	0.5	-	0	-	0	-	0	0,0.25	0.5	-	0
2	26	Round the Clock	c	-	0	-	0	0,0.25	0.5	-	0	-	0	-	0	0,0.25	0.5	-	0
2	27	BB	c	-	0	-	0	0,0.25	0.5	-	0	-	0	-	0	0,0.25	0.5	-	0
2	28	ANNK	c	-	0	-	0	0,0.25	0.5	-	0	-	0	-	0	0,0.25	0.5	-	0
3	29	Sheer	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
3	30	Opaque	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
3	31	Ultra Sheer	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
4	32	Basic	c	-	0	-	0	-	0	0,0.65	0.5	-	0	-	0	0,0.65	0.5	-	0
4	33	Fashion	c	-	0	-	0	-	0	0.65,1	0.5	-	0	-	0	0.65,1	0.5	-	0
5	35	Red	c	-	0	-	0	-	0	-	0	-	0	-	0	-	0	0.65,1	1
5	36	Black	c	-	0	-	0	-	0	-	0	-	0	-	0	-	0	0,0.65	1
5	37	Pink	c	-	0	-	0	-	0	-	0	-	0	-	0	-	0	0.65,1	1
5	38	Navy	c	-	0	-	0	-	0	-	0	-	0	-	0	-	0	0,0.65	1
5	39	Grey	c	-	0	-	0	-	0	-	0	-	0	-	0	-	0	0.65,1	1
5	40	Brown	c	-	0	-	0	-	0	-	0	-	0	-	0	-	0	0,0.65	1
5	41	Khaki	c	-	0	-	0	-	0	-	0	-	0	-	0	-	0	0.65,1	1
5	42	Nude	c	-	0	-	0	-	0	-	0	-	0	-	0	-	0	0,0.65	1
5	43	Tan	c	-	0	-	0	-	0	-	0	-	0	-	0	-	0	0,0.65	1
5	44	Cream	c	-	0	-	0	-	0	-	0	-	0	-	0	-	0	0,0.65	1
5	45	Beige	c	-	0	-	0	-	0	-	0	-	0	-	0	-	0	0,0.65	1
5	46	Metallic	c	-	0	-	0	-	0	-	0	-	0	-	0	-	0	0.65,1	1

5	47	Natural	c	-	0	-	0	-	0	-	0	-	0	-	0	-	0	0,0.65	1
5	48	Off White	c	-	0	-	0	-	0	-	0	-	0	-	0	-	0	0,0.65	1
5	49	Taupe	c	-	0	-	0	-	0	-	0	-	0	-	0	-	0	0,0.65	1
5	50	Blue	c	-	0	-	0	-	0	-	0	-	0	-	0	-	0	0.65,1	1
5	51	Green	c	-	0	-	0	-	0	-	0	-	0	-	0	-	0	0.65,1	1
5	52	Silver	c	-	0	-	0	-	0	-	0	-	0	-	0	-	0	0.65,1	1
5	53	Pink/Yellow/Green	c	-	0	-	0	-	0	-	0	-	0	-	0	-	0	0.65,1	1
6	54	One Size	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	55	S	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	56	M	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	57	L	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	58	A	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	59	B	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	60	C	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	61	D	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	62	E	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	63	F	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	64	T	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	65	XT	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	66	AB	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	67	CD	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	68	EF	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	69	S/M	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	70	M/L	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	71	L/XL	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	72	XL	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	73	6-8 1/2	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	74	5-6 1/2	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	75	P1	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	76	P2	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	77	P3	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	78	A/B	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0

6	79	C/D	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	80	1X	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	81	2X	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	82	3X	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	83	4X	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	84	PP	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	85	3P	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	86	2P	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	87	1P	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	88	P	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	89	MAXI	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	90	Q-Petite	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	91	1x-2x	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	92	3x-4x	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	93	5x-6x	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	94	I	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
6	95	2+	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
7	96	Textured	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
7	97	Flat	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
8	98	Ridgeview	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
8	99	American Essentials	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
8	100	HAN	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
8	101	ORO	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
8	102	HOSO	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
8	103	Kayser-Roth Corp\.	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
8	104	Peneco	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
8	105	Berkshire	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
8	106	Donna Karan Company	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
8	107	High Point Knitting Inc\.	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
8	108	Paul Lavitt Mills Inc\.	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
8	109	Easton International; Inc\.	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
8	110	Belly Basics	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0

9	111	SF - Sandal Foot	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
9	112	RT - Reinforced Toe	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
10	113	Cotton	d	-	0	-	0	0.65,1	0.4	-	0	0.65,1	0.6	-	0	-	0	-	0
10	114	Lycra	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
10	115	Nylon	d	-	0	-	0	-	0	-	0	0,0.65	1	-	0	-	0	-	0
10	116	Rayon	d	-	0	-	0	-	0	-	0	0,0.65	1	-	0	-	0	-	0
10	117	Luxary	d	-	0	-	0	0.65,1	0.5	-	0	-	0	-	0	0.65,1	0.5	-	0
10	118	Silk	d	-	0	-	0	0.65,1	0.5	-	0	-	0	-	0	0.65,1	0.5	-	0
11	119	CT - Control Top	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
11	120	STW - Sheer To Waist	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
12	121	Women	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
12	122	Men	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
12	123	Children	d	-	0	-	0	-	0	-	0	-	0	0.65,1	1	-	0	-	0
13	124	Solid	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
13	125	Conversational	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
13	126	Plaid	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
13	127	Floral	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
13	128	Stripe	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
13	129	Herringbone	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0
13	130	Pique	d	-	0	-	0	-	0	-	0	-	0	-	0	-	0	-	0