ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΝΑΣΤΑΣΙΟΣ ΣΑΒΒΟΠΟΥΛΟΣ

ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΧΡΗΣΤΩΝ ΣΕ ΣΥΣΤΗΜΑΤΑ ΗΛΕΚΤΡΟΝΙΚΟΥ ΕΜΠΟΡΙΟΥ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

ΠΕΙΡΑΙΑΣ **2010**

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Preface

The research of this PhD thesis was conducted at the University of Piraeus, started February of 2005 and concluded in February 2010. During these five years research was conducted in the topic of adaptivity in remote shopping applications. A very important research field in our research is remote shopping applications.

Three prototype applications in three different media were constructed during our research. The three different media were desktop computers, interactive TV and mobile devices. These three applications incorporated methodologies based on user modeling and adaptivity. User models were constructed in order to help users in remote shopping applications in two separate ways. Firstly, by automatically recommending products close to their interests and secondly by helping them adaptively while they used the application. Explicit information, observation of user behaviour and dynamic stereotypes were incorporated into the construction of these user models. The incorporation and combination of these technologies into three different media constitutes a very novel approach on the topic of personalization in remote shopping applications. The three prototypes also incorporated techniques of adaptivity for visualizing the recommendations based on interest and needs and for personalizing the user interaction with the application. The methodology used was a combination of adaptive hypermedia and dynamic user interface and proved to be a very important aspect in the personalization and familiarity of the applications. Furthermore, these methodologies were shown very effectively for the purpose of dynamically assisting users.

A major research field of our research is the field of machine learning. More specifically, recommendation applications

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

that incorporate machine learning algorithms in their reasoning process. These applications recommend products or processes to users according to their interests or similar users' interests. The field of recommendation applications combined with machine learning algorithms has been popular research topic in the recent years and significant results have been achieved. However, a lot of research aims at specific domains and the transition of methodologies to other domains can be very difficult. Our approach provides a novel solution to this problem. The presented approach constructs a software life cycle based on RUP that helps researchers incorporate and use effectively a machine learning algorithm in order to create personalized remote shopping applications. The software life cycle that we have developed is called RESCA-RUP. Through this process a researcher can design and construct a personalized e-commerce system that incorporates a machine learning algorithm and provides personalized responses based on dynamic stereotypes for new and old users.

Two major problems of today's recommendation systems are the following: First, many of these systems are designed for specific domains and can propose products belonging to this domain. Second, the medium that these applications use interferes with their reasoning mechanism thus making them very difficult to be medium independent. In our approach we propose a novel generic architecture that can be combined with the software life cycle mentioned above. The generic architecture presented in this thesis is called PERCOM and consists of three tiers. The first is the User Modelling tier, the second is the User Model Server tier and the Adaptive User Interface tier. A major feature of our approach is the construction of separate components leading to algorithmic independence. The three tier construction leads to medium independence difficulty as the reasoning mechanism functions

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

separately from the user interface components in the server side of our architecture. A very important innovation in our approach on the generic architecture is the construction of a user model server. This server constructs and manages all users' models. Applications can use these models to extract information about inferences and needs. In this way, applications can use user models of other applications that have a richer knowledge base than them and make quick assumptions about their users interests and needs.

This thesis also presents a number of evaluations with real users conducted in all of the three media that we have built our prototypes. The evaluations cover both algorithmic comparisons and evaluation of the methodologies with the help of real users. The algorithmic comparisons have been conducted with several criteria such as effectiveness and resource requirements. The evaluations with real users were conducted with criteria concerning effectiveness in personalization, easiness of use and user friendliness of environment. These evaluations have shown the effectiveness of our methodologies.

I would like to thank my supervisor Professor Maria Virvou. The idea and encouragement for conducting a research are accredited to her. Also Professor Maria Virvou has contributed a lot with her help and great knowledge on the research fields during the whole research. I am also indebted to Professor George Tsihritzis for the valuable collaboration that we had on machine learning algorithms. I would also like to address special thanks to Dr. Eythimios Alepis for his most helpful support and collaboration during my research.

I am also very thankful to the members of my advising committee, Professor Evaggelos Fountas and Assistant Professor Despoina Polemi for their helpful collaboration during my research as a PhD student.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Last but not least, I would like to express my deepest gratitude to my family for their support and encouragement through all these years of my research as a PhD student. Without them this PhD thesis and research would have not been possible to be completed.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

CHAPTER 1
INTRODUCTION

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

1 Introduction

1.1 The problem of user needs and inferences in shopping applications

In recent years, the combination of high speed internet, the growth of mobile phones and new technologies such as digital and interactive TV, have led software engineering companies to create more complex applications concerning various aspects of life. A large number of these applications are relate with the field of shopping, such as shopping products through the internet using your desktop pc, your laptop, your mobile phone or even your television. The large number and variety of these applications is reinforced by the fact that recent studies on customer behavior, which show that computer adoption and internet connections in households are constantly growing (Goy et al., 2007).

The applications that are built for these shopping purposes are called e-commerce applications (Coppel 2000) and attempt to sell various products to various groups of people. The plethora of products that are being sold and the variance of consumers in ages, education, character and buying behaviour create many problems for this kind of applications. The major problem that e-commerce applications are called to solve is satisfaction of their customers, which is the same problem of that traditional shops also face. Traditional shops and e-commerce applications may share the same major problem but the conditions and solution abilities differ a lot. In traditional shops however, people called salesmen are asked to help customers, whereas in e-commerce applications the system must deal with every customer without the help of a human salesman.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

The interaction between a computer application and a human has been proven many times to be very difficult (Kellogg and Breen, 1987, Shneiderman, 1997). In this way the problem of customer satisfaction becomes more difficult in the field of e-commerce applications rather than in traditional shops with human salesmen. A conversation between a human salesman and a customer is always more natural than the interaction between a computer application and a user. Furthermore, e-commerce applications, in contrast to human salesmen, lack in the ability to "sense" a customer's purpose while this specific customer visits the shop and looks at the products being sold. Moreover, a human salesman can use his experience with other customers in order to serve the customer in a more effective way.

All the above human abilities are lost within human-computer interaction. Furthermore, there is a lack of tools supporting the analysis of customers' browsing behavior (e.g., shopping cart abandonment) does not enable vendors to collect feedback useful to redesign and optimize their Web sites (Hall, 2001). This phd thesis emphasizes on this exact problem of customer satisfaction in the field of e-commerce applications. In order to fully understand the problem of customer satisfaction in e-commerce applications we must divide this major problem into to two main sub-problems.

The first sub-problem is extracting from every different customer their needs and inferences concerning the products being sold. This means that the system must build an effective way of understanding what the customer wants to buy. A customer's needs may vary due to several reasons, from his educational background to his lifestyle. All these criteria must be taken into consideration by the system in order for the e-commerce applications to have a complete image of each customer's needs. The problem of extracting customers' needs and inferences can be altered by many reasons and be complicated in many ways, such as what kind of

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

products are being sold, which group of people these products apply to and what is the medium that these products are sold through. Understanding customers' needs and inferences can be also divided into various sub-problems.

Another sub-problem is to read the customer's aim while s/he visits an e-commerce application. This aim may vary from the one of a customer that has visited the application just to browse through products, to the one of a customer that goes straight to products s/he really wants and buys it directly without deviations. The second sub-problem in understanding customers' needs is the creation of a customer profile based on product interests. The creation of a customer profile (Adomavicius and Tuzhilin, 2001) involves the extraction of customer tendencies to product features with the help of various techniques. The customer profile is in fact a complete image of customer's interests towards products or product features that can help in the selection of products close to this customer's tendencies.

Information for customers' profile can be acquired with two ways, explicitly and implicitly (Rich, 1998). Another significant sub-problem in understanding customers' needs is customer behaviour. Customer behaviour involves observing, recording and extracting information from customer's moves while this customer browses through the application. Customer behaviour is in fact a close observation of the customer's every action and interaction with the e-commerce application. Extracting information from every customer's action can help in providing useful information concerning customers' profile. Today a large number of applications deal with these problems involved around concerning customers' needs.

Many techniques have been used to deal with these sub-problems mentioned below in order to provide effective ways of extracting customers' needs. These techniques involve machine

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

learning, Bayesian networks, cognitive theories etc (Kurokawa, et al. 2007, Katsionis and Virvou, 2004) and are usually used by researchers to discover similarities between customers, create groups of customers with similar tastes or with similar behaviour and aims. The solutions of these techniques concerning e-commerce applications aimed at creating a list of recommended products that were close to specific customer's needs and inferences. In this way systems that deal with customers' needs were called recommendation systems (Resnick and Varian, 1997) because they aimed at in fact the same thing, which is product recommendation. The list of recommended products provided by the application is different for every customer and a degree of closeness to specific customer's interests plays the role of success for every product in this list. Feedback can be provided from this list if the customer chooses to buy a product that belongs to this list or not. In this way recommending systems can re-calculate degrees measured for a specific customer.

The second major sub-problem in user satisfaction is application usage. A product may be recommended successfully by the recommendation system but if a customer finds the application confusing and difficult to use, than this recommendation may prove to be a waste of time as there is a high probability that a frustrated customer may never reach the point of using the recommendation. The roots of the problem of application usage in e-commerce shops can be found in human computer interaction issues (Curran and King, 2008), but in the field of e-commerce these issues may be even more difficult to address. If someone considers the fact that in recent years e-commerce applications have expanded to different media, then the problem of application usage is becoming more important than the one of customer interests.

The problem of computer usage takes here, in the field of e-commerce, the form of e-shop usage or in other words how can a

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

customer use an e-shop application effectively. Many issues have to be considered during the creation of an effective e-shop application concerning usage, like customer familiarity with the medium, customer familiarity with the product, customer age and demographic data in general. For example, a customer who is not familiar with computers may find difficult to comprehend certain user interface elements and their purpose (Mazzoleni et al., 1996, Irizarry et al., 1997).

Other customers may find some products difficult to understand due to complex technical terms (Kurniawan et al., 2006) like mobile phones. Moreover, age can play a significant role in human computer interaction. A study conducted by Zaphiris and Sarwar (2006) that clearly shows many differences in tastes between teenagers and seniors. In addition age can be an obstacle to human computer interaction. For example, older people (Gregor et al., 2002) usually present lessened ability of comprehending tasks and functions that concern computer applications. Other body inabilities of the elderly can prove to impose obstacles to the use of an application. For example, common problems for the elderly, like sight and hearing impairment can raise many difficulties as an older customer interacts with an e-shop application. Furthermore, demographic data can affect human computer interaction (Pazzani, 1999). A customer from a big city with a high education level can accomplish a higher degree of learning on how to use an e-shop application than a customer from a small village with a basic education level.

All the above mentioned problems of e-shop usage can raise the complexity of an e-shop design to a very high level. In order for these problems to be addressed by an e-shop application, intelligent and adaptive techniques must be incorporated in order for the e-shop application to provide to every specific user a friendly and useful application environment. Applications that tried to deal with

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

application usage problems are called intelligent help systems (Krause et al., 1993). Their role is to provide help in smart ways to their users, in order to predict and prevent possible problematic situations concerning user interaction with the application. Many techniques can be used in adaptive help systems in order to gather useful data from users and prevent problematic situations. Such techniques similar to recommendation systems are machine learning, Bayesian networks, cognitive theories etc, (Kurokawa, et al. 2007, Katsionis and Virvou, 2004) frequently used to create profiles of users with similar problems.

Intelligent help systems have been widely used in e-learning environments (Brusilovsky et al., 1996) and computer operating systems usage (virvou and kabassi, 2000). In order to transfer knowledge from these two fields and create an effective adaptive help system for e-commerce, further work must be conducted. More criteria must be considered like the product being sold and its characteristics. Another criterion is the time that a user has been using an e-commerce application which is significantly smaller than the time a user uses an operating system or an e-learning application. Last but not least the group of people that an e-commerce application applies to is larger than a specific group of people that an e-learning application applies to.

1.2 Solutions to the problem of user needs and inferences in remote shopping applications

As we can see from the above issues and problems, users' needs and inferences is not an easy task to be accomplished in e-commerce. The determination of the users' intention is generally complex because they do not always state it explicitly (Ardissono et al., 1995). A solution to this situation can be achieved by personalization techniques that can be extracted from user

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

modelling theory (Rich, 1983). A system that incorporates user modelling can be adaptive to every specific customer's interests and needs and is called product recommendation system. As Ardissono points out, recommendation techniques (Jordan and Weiss, 2002) based on the exploitation of artificial intelligence techniques such as user modeling, content based and collaborative filtering, are thus often presented as a solution to the information overload problem by helping the users to filter relevant items on the basis of their needs and preferences.

With the recent expansion of TV content, digital TV networks and broadband, smarter TV entertainment is needed as well (Ardissono et al., 2003). In order to achieve personalization, the software agent that provides the services must have the capabilities of maintaining a model of the user, containing data about these users' needs, interests, preferences and strategies for adapting its behaviour to each specific user. The choice of such strategies depends on many factors, such as the application task and domain, the goals of the system, the target users and the context (Ardissono et al., 2001). A rich user model (UM) has a strong impact in the interpretation of agents' intentions: when different hypotheses on a user's plans are present, the UM can help to discard some of them, because they are not coherent with the user's plausible intentions encoded in it (Ardissono and Sestero, 1995). Researchers have used various adaptive techniques in order to overcome these problems.

User modeling (Fischer, 2001) has been widely used in order to create a context of every specific customer according to his needs and interests. The customer user model contains data and information about customer behaviour, aims and possible similarities with other customers. Other researchers used stereotypes (Rich, 1998) in order make assumptions about customers and their tastes. Customer stereotypes combined with

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

product characteristics were usually used to group customers to similar groups and extract information from other customers in order to use this information for customers that the system has little knowledge about. A stereotype is an efficient way of providing quick information in order to make quick assumptions for new customers. Another frequently used technique by researchers is adaptive hypermedia (Brusilovsky, 2001). With the help of adaptive hypermedia an application designer can create an adaptive user interface. This user interface can use different fonts, symbols or change every possible media according to a customer's tastes or knowledge. In this way an intelligent user interface can be created to be personalized to every specific customer.

1.2.1 Recommendation Systems

The basic classification of the customer satisfaction problem produced two categories of systems; recommendation and adaptive help systems. As Resnick and Varian pointed out (Resnick and Varian, 1997), in everyday life we often rely on recommendations from other people either by word of mouth, recommendation letters, movie and book reviews or general surveys. Recommending systems assist and augment this natural social process. In a typical recommending system people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients. In some cases the primary transformation is in the aggregation; in others the system's quality lies in its ability to make good matches between the recommenders and those seeking recommendations.

As Schafer (Schafer et al., 1999) says recommender systems are used by E-commerce sites to suggest products to their customers. The products can be recommended based on the top overall sellers on a site, based on the demographics of the customer, or based on an analysis of the past buying behavior of

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

the customer as a prediction for future buying behavior. Broadly, these techniques are part of personalization on a site, because they help the site adapt itself to each customer. Recommender systems automate personalization on the Web, enabling individual personalization for each customer.

An interesting approach in the field of product recommendation has been made in WindOwls (Kazienko and Kolodziejski 2005). WindOwls is a recommending system that uses user modeling techniques to propose products to individual users. WindOwls uses association rules to calculate weights in order to group acquired tastes together. Another interesting approach has been made by Choi, et al. (Choi et al. 2006). They chose a multi-attribute decision making method to find similar products. In their system, the customer must first order a product in order for the system to propose a similar one. Another recommending system that uses clustering techniques in order to group products is the system proposed by Guan (Guan et al. 2005). In their system they use an explicit method of ranking to acquire generic attributes from products and then cluster new attributes into the different groups of generic attributes using the k-NN algorithm.

An important field in recommendation that has a large affinity with interactive TV is TV programs recommendation. Important steps have been made in this field too, e.g. (Maybury et al. 2004, O' Sullivan 2004). Another system that tries to help visually impaired and elderly people is the VISTA Project by Carmichael (Carmichael et al. 2003). This project for the digital TV tries to help visually impaired users find the best TV program for them through a conversation with an animated avatar that has the ability to synthesize speech. This avatar works as a bridge between the electronic program guide of the digital TV and the user and provides user recommendations of programs through the conversation and through the help of the electronic program guide.

1.2.2 Intelligent Help Systems

Carroll (Carroll and Aaronson, 1988) suggests that one of the most important means for facilitating the usability of computer systems is the advisory interface, the training, reference material, on-line help, and other advisory support available to users. Intelligent help systems are categorized in recent bibliography into two types, passive and active. Intelligent assistance can be provided upon user request or initiated automatically by the system, since it diagnosed that the user faces a problem in achieving its objectives.

The systems in the first category in the international literature (Jones & Virvou 1991) are known as a passive intelligent assistance (passive intelligent help systems) and can only help the users understand that they have made a mistake and ask help. Known systems in this category is the AQUA (Quilici 1989), the OSCON consultant (Mc Kevitt 2000), the UNIX Consultant (UC) (Chin 1989, Mayfield 1992, Wilensky et al. 2000) and SINIX Consultant (Hecking 2000, Kemke 2000). The systems in the second category in the international literature (Jones & Virvou 1991) are known as active intelligent help systems and intervene when they consider that there is a problem. Known systems in this category are RESCUER (Virvou 1992, Virvou 1998, Virvou 1999, Virvou & Du Boulay 1999), the CHORIS (Tyler et al. 1991), the USCSH (Matthews et al. 2000), the ACTIVIST (Fischer et al. 1985), the PAL (Silber 1990) and the Office Assistant (Horvitz et al. 1998).

On the field of systems that try to help users with sight and hearing problems, there is significant on-going research work too. In Muller's work (Muller and Wasinger 2002), a multimodal navigational system is presented that learns from the cognitive load of users and then categorizes them into two different stereotypes: elderly and average aged adults. Despite the fact that their

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

application has many modes of functioning, including speech, that can help elderly people, it does not focus on the topic of group categorization. This means that a person may be an elderly but s/he can have different problems from another elderly person. Another system that helps the elderly is made by Zhao and Tyugu (1998). Their system has a personalized web browser that helps people with special needs browse the web. The browser adapts its presentation according to the users' behavior. In Savidis's work (Savidis et al. 2005), a system called Unified User Interface is presented. That system is a framework that can adapt to users depending on their age and kind of incapability by creating polymorphic user interfaces. In their work they apply the Unified User Interface on a health application scenario, namely the MediBridge C-Care web-based EHR system. The polymorphic interfaces are produced through rules of the "tasks" of user performances.

1.3 Methodology created

Our approach was to create a novel methodology for creating e-commerce applications that can accomplish customer satisfaction in two levels; interest and usage. The methodology created targeted three major aims. The first aim was independent user models. Independence of machine learning algorithm used, independence of the product used, independence of the purpose used (user models for recommendation or adaptive help) and independence of the medium used. The second aim was the creation of a novel process for creating such intelligent e-commerce applications based on rational unified process (RUP) (Savvopoulos et al., 2008). The third final aim was the creation of a novel framework that could incorporate various machine learning algorithms.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

In order to accomplish these three major tasks we based our approach into two very effective techniques; machine learning and RUP. The basic reason for choosing machine learning algorithms in order to be used to our methodology is that these algorithms can be easily combined with statistical data usually used in e-commerce applications, in order to measure customer's behaviour about bought products.

Another reason for using machine learning algorithms is that these algorithms present similar input and output forms and in this way different algorithms can be used with minimum changes in the process that uses them. Moreover, these algorithms can provide a very effective way of discovering similarities and tendencies in large groups of customers. Such algorithms have been efficiently used in many applications (Pohl, 1997), (Webb et al., 2001), (Zakrzewska, 2008).

RUP was also used in order to accomplish successfully the second aim because it's a very useful tool in software life-cycle. RUP is an object-oriented process that advocates multiple iterations of the software development process. It divides the development cycle in four consecutive phases: the inception, the elaboration, the construction, and the transition phase. Each phase is divided into four procedural steps, namely, requirements capture, analysis and design, implementation, and testing. The phases are sequential in time but the procedural steps are not as mentioned above. Additionally, because RUP is an object-oriented process, it is appropriate for the development of graphical user interfaces such as the one described in our research. Moreover, one important advantage of RUP is the highly iterative nature of the development process.

For the above reasons, RUP can be selected as the basis for presenting adaptive systems too. RUP process has also been used

by many researchers in very effective ways (P Jaferian etc 2005), (Kabassi and Virvou 2006), (Virvou and Kabassi 2000).

1.4 Describing the Architecture

The architecture used in our methodology was based in the philosophy of different and entirely separate components. The novelty of our architecture is that it combines three adaptive techniques in two different ways in order to accomplish customer satisfaction in the level of product recommendation and intelligent personalised help. These three techniques are user modelling, stereotypes and adaptive hypermedia and they are supported by machine learning algorithms that are used in clever ways to provide the architecture with efficient results for every specific customer. A general diagram of our architecture is illustrated in figure 1. Our architecture will be fully presented in later chapters.

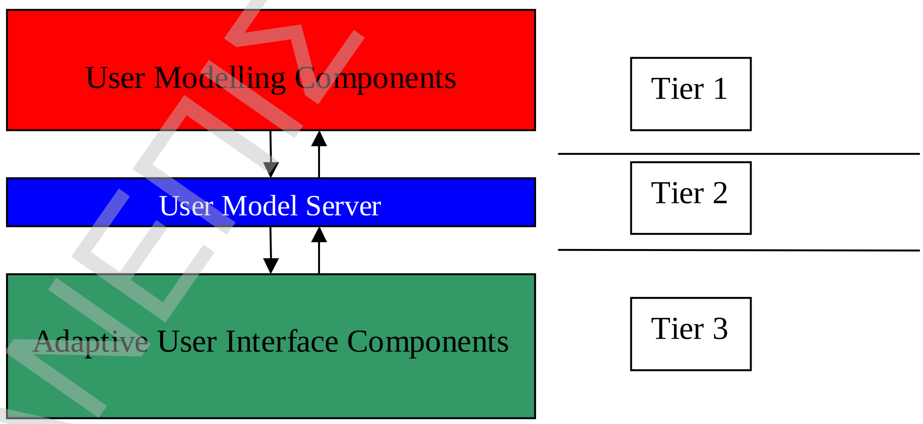


Figure 1 PERCOM General Architecture Diagram

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

As figure 1 illustrates we chose to implement our architecture into three separate tiers. In this way we divided the components that interacted directly with the user and the components that were responsible for manipulating user knowledge. Between these two tiers lies the User Model Server, that contains all the information, implicit and explicit, about a specific user, the stereotype that s/he belongs to and his/her more similar representatives. More specifically the first tier consists of the following components:

1. Explicit Information User Profiles,
2. Monitoring Agent,
3. Clustering Algorithm Process,
4. Double Stereotypes and Server.

The Explicit Information User Profiles component contains all the information in a database that users have provided to the system in an explicit way, either, by answering interview questions or rating products. Every time a new user is registered in the application that incorporates the user modelling server, this component collects this information from demographic data, educational data and interest data that the user provides through the registration process. The next component is Monitoring Agent. This component plays a key role in the construction of the User Model and contains all the information about the user interactions with the system. Moreover, monitors users' actions throughout the usage of the system. The Monitoring Agent component also contains a statistics database of all users' actions and features collected by the systems.

The Clustering Process Algorithm component contains the machine learning algorithm that the system uses to group similar users and extract representative users of these groups. The algorithm takes as input the statistical data of all the explicit and implicit information that the system has inferred and collected about users. Those data are saved on the statistics database of the Monitoring Agent. Those data include visits in products pages, products general categories, search queries made by the user, products moved to the shopping cart and products eventually bought. During this Phd research five algorithms were incorporated in the Clustering Process Algorithm component. These algorithms were k-means (MacQueen, 1967), agglomerative hierarchical clustering (Day and Edelsbrunner 1984), fuzzy c-means clustering (Bezdek et al. 1984), spectral clustering (Jordan and Weiss 2002) and AIS-based clustering (Cayzer and Aickelin 2002), (Morrison and Aickelin 2002).

- k-means is an algorithm used for clustering objects based on attributes into k partitions. It is similar to the expectation-maximization algorithm for mixtures of Gaussians, in that they both attempt to find the centres of natural clusters in the data. It assumes that the object attributes form a vector space. The objective it tries to achieve, is to minimize total intra-cluster variance, or, the squared error function.

1. Given a set of observations (x1, x2, …, xn), where each observation is a d-dimensional real vector,
2. k-means clustering aims to partition the n observations into k sets (k < n) S={S1, S2, …, Sk} so as to minimize the within-cluster sum of squares (WCSS):

$$\underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \left\| \mathbf{x}_j - \boldsymbol{\mu}_i \right\|^2$$

where m_i is the mean of S_i

- agglomerative hierarchical clustering,

1. The agglomerative method starts with a set of n objects to be clustered.
2. The method groups these objects into successively fewer than n sets, arriving eventually at a single set containing all n objects.
3. These sets are hierarchical non-overlapping methods that specify a sequence P0 .... , Pw of partitions of the objects in which P0 is the disjoint partition, Pw is the conjoint partition, and P~ is a refinement (in the usual sense) of Pj for all 0 ~< i < j ~< w.
4. They are sequential methods since the same algorithm is used iteratively to generate P_{i+1} from P_i for all $0 \le i < w$0.

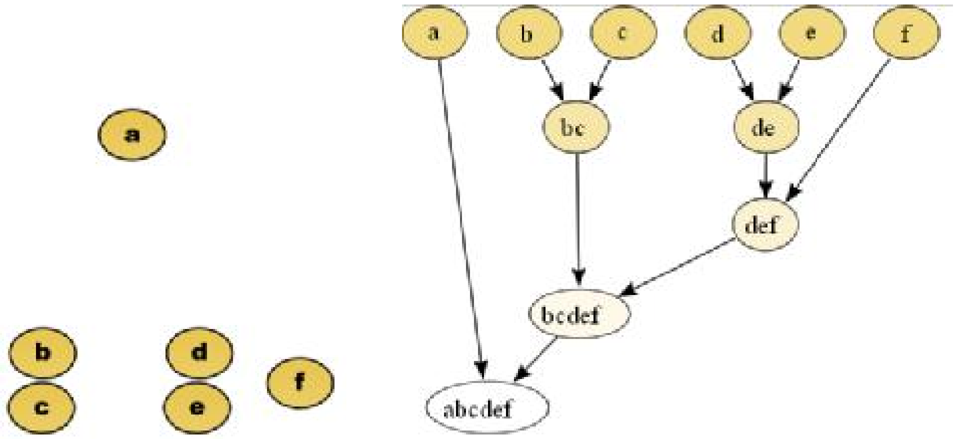


Figure 2 Input and Output of the Hierarchical Clustering Algorithm

- **fuzzy c-means clustering, Fuzzy c-means (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters. It is based on minimization of the following objective function:**

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m \left\| x_i - c_j \right\|^2 \quad , \quad 1 \le m < \infty$$

1. **Initialize** $U = [u_{ij}]$ **matrix,** $U^{(0)}$

2. **At k-step: calculate the centers vectors $C^{(k)} = [c_j]$ with $U^{(k)}$**

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

3. **Update $U^{(k)}, U^{(k+1)}$**

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\| x_i - c_j \|}{\| x_i - c_k \|} \right)^{\frac{2}{m-1}}}$$

4. **If $\| U^{(k+1)} - U^{(k)} \| < q$ then STOP; otherwise return to step 2.**

The algorithm minimizes intra-cluster variance as well, but has the same problems as k-means, the minimum is a local minimum, and the results depend on the initial choice of weights. The expectation-maximization algorithm is a more statistically formalized method which includes some of these ideas: partial membership in classes. It has better convergence properties and is in general preferred to fuzzy-c-means.

- **spectral clustering, spectral clustering techniques make use of the spectrum of the similarity matrix of the data to perform dimensionality reduction for clustering in fewer dimensions.**

Normalized spectral clustering according to Ng, Jordan, and Weiss (2002):

Input: Similarity matrix $S \in R^{nxn}$, number *k* of clusters to construct.

1. **Construct a similarity graph. Let $W(A,B) := \sum_{i \in A, j \in B} w_{ij}.$ be its weighted adjacency matrix.**
2. **Compute the normalized Laplacian L_{sym}.**

$$L_{\text{sym}} := D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$$
$$L_{\text{rw}} := D^{-1} L = I - D^{-1} W.$$

3. **Compute the first k eigenvectors** $u_1 \ldots u_k$ **of** L_{sym} **.**
4. **Let** $U \in R^{nxk}$ **be the matrix containing the vectors** $u_1 \ldots u_k$ **as columns.**
5. **Form the matrix** $T \in R^{nxk}$ **from U by normalizing the rows to norm 1, that is set** $t_{ij} = u_{ij} / (\sum_k u_{ik}^2)^{1/2}$ **.**
6. **For i = 1… n, let** $y_i \in R^k$ **be the vector corresponding to the i-th row of** T **.**
7. **Cluster the points** y_i **(i=1,…,n) with the k-means algorithm into clusters** $C_1 \ldots C_k$ **.**

Output: Clusters $A_1 \ldots A_k$ **with** $A_i = \{ j \mid y_j \in C_i \}$ **.**

- **AIS-based clustering, AIS-based clustering relies on a computational imitation of the biological process of self/non-self discrimination, that is the capability of the adaptive biological immune system to classify a cell as "self" or "non-self" cell. Any cell or even individual molecule recognized and classified by the self/non-self discrimination process is called an antigen. A non-self antigen is called a pathogen and, when identified, an immune response (specific to that kind of antigen) is elicited by the adaptive immune system in the form of antibody secretion. The essence of the antigen recognition process is the affinity (molecular complementarity level) between the antigen and antibody molecules. The strength of the antigen-antibody interaction is measured by the complementarity of their match and, thus, pathogens are not fully recognized, which makes the adaptive immune system tolerant to molecular noise. In this research (Sotiropoulos et al. 2006), the antigenic population consists of the initial set of customer profile feature vectors. The produced set, of memory antibodies, provide an alternative, more compact way of representing the original dataset while conserving their original spatial distribution.**

The three components mentioned above along with the Double Stereotypes component contribute to the construction of the User Model Server component mentioned above. Every category and product characteristic is a calculated by the algorithm. However, this calculation is not based on degrees from products classification; instead the calculation is based on user behaviour while s/he interacts with the system. The data input for the algorithm are calculated based on degrees from the features measured. The algorithm processes these degrees and provides the system with groups based on similarity. From these groups representative feature vectors are extracted. The representatives' vectors work as group leaders and show the groups tendency to specific product features.

These vectors are then compared with the vectors of products' characteristics and the closest vectors are extracted and saved to the Recommendations Database component. In this way the Recommendation Database component is updated dynamically as users navigate through the application. After this procedure the Double Stereotypes component calculates dynamic double stereotypes from these representatives. These stereotypes follow a general to specific hierarchy, meaning that the system constructs a low number of generic stereotypes at first and then continues to construct more specific stereotypes until it reaches a certain point of complexity.

The third tier consists of user interface components. The Incremental Initialization Process component acquires information from the user model and tries to provide the best recommendations to new users or users that the system has little information about. This component uses mostly stereotypic information from the Double Stereotypes component and chooses where the user should belong according to the moves that s/he has made so far.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

The Recommendation component communicates with the Server component and provides the users with recommendation about products and system usage. The Recommendation component contains all system recommendations about products, mistakes or other recommendations in a database. This component can use many techniques in order to make recommendations such as adaptive hypermedia, dynamic annotations and animated agent. The Recommendation component also takes feedback from the users and provides the Server with more useful information.

The next component is the Animated Agent. It is a system component that manages an animated agent that can help users throughout the navigation of the system. This agent can provide useful information about the usage of the system and provide recommendations about products by acquiring information from the user model. This component does not interact with the mobile applications due to hardware limitations. Next, we have the User Interface component. We use a dynamic user interface that not only adjusts to the medium used automatically, but also changes according to the users' interests. The User Interface component can change the whole user interface appearance dynamically and personalise the user interface according to the specific users' preferences and needs. These changes are acquired from the Server that contains all the user models. The User Interface is an entirely separate component and in this way it can adapt on any medium, thus making the user modelling server medium independent. These two functions of the User Interface component create a unique personalised experience for every specific user, resulting in a friendlier and more efficient user interface. Furthermore, the Recommendation and Incremental Initialization component can change the User Interface according to the user model of every user.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Last but not least, we have the Adaptive Hypermedia component that is used to change user interface components according to the users' needs or preferences. This component has the ability to change the user interface according to the recommendations extracted from the user model. This component can annotate products that are recommended and change the position of these products in order to be seen first. It can also, change the symbols of recommended products depending on the degree of the recommendation. Adaptive hypermedia can also change the font of product names or features in order to catch user's attention.

In the lower end of the architecture we have the users' terminals, which can be supported by any medium needed (mobile phones, interactive TV, laptop). The User Interface component is responsible to adapt the application's user interface to the corresponding medium taking into account the appropriate resources.

1.4.1 Overview of the User Models Created

The user models created within our methodology had one major novelty. Instead of basing our user models to the products characteristics sold by the e-commerce applications, we based our user models in users' interests and their behaviour. In this way the user models created were not depended by the product being sold. This was done by basing every specific user's model to their specific actions. There were two types of information taken into consideration in every user model. The first was explicit information and second implicit information.

Explicit information was based on questionnaires based on customers demographic, personal and tastes data. More specifically

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

questions concerning age, education, residence, familiarity with computers and a small set of questions that changed dynamically according the product type being sold. The implicit data was based on statistical data from customer behaviour while s/he used the application. For example this statistical data involved visiting times of the specific product page or times that product was moved to cart or times a specific mistake pattern occurred. Concerning algorithmic independence we formulated our user models in the form of vectors consisted of numerical degrees. The degrees involved either degrees corresponding to tastes or degrees corresponding to mistakes. In this way a large variety of machine learning algorithms could use every user model as an input vector. Furthermore, every machine learning algorithm resulted in an output vector that could also be used as a user model of a new user or a representative user of a customer group. The way our user models are built also resulted in medium independency. Because users' models were considered an entirely different and separate component in our methodology and because there were consisted of only numerical data, they could be easily used by any medium without great cost of data transferring. This is great advantage especially for mobile phones with limited resources data capacity can prove vital to an e-shop application's responses.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

1.5 Architecture's main features

The novelty of the architecture on user modelling creation and management presented in previous section has in fact derived from the philosophy that this architecture is built on. The philosophy of entirely separate component gives the ability of changing parts of the architecture without changing the whole architecture from

scratch. In this way our architecture can adapt to different criteria, domains, products and media.

1.5.1 Different Algorithms

Clustering has been extensively used by researchers in aiding group formulation with the criteria of similarity. Clustering has been used very effectively in recommendation systems by many researchers. However, most recommendation systems incorporate only one clustering algorithm. Changing this algorithm can prove to be a very difficult task for a recommending system. Our methodology through the generalization of the clustering module creates the appropriate conditions that can help the developer change the clustering algorithm used very easily. More specifically, due to the fact that the Clustering Algorithm Process component is separate and follows the numerical vector input and output philosophy, it can be very easily combined with any kind of algorithm that follows a similar input and output process, which of course most of machine learning algorithms follow. This case here is reinforced by the fact that we incorporated five different clustering algorithms in our approach without changing anything in our architecture.

The clustering algorithms used were k-means (MacQueen, 1967), agglomerative hierarchical clustering (Day and Edelsbrunner 1984), fuzzy c-means clustering (Bezdek et al. 1984), spectral clustering (Jordan and Weiss 2002) and AIS-based clustering (Cayzer and Aickelin 2002), (Morrison and Aickelin 2002). Due to the separate module of the Clustering Algorithm Process we were able to run these algorithms in real-time and gather results that proved useful in two different ways. Firstly, on the analysis of the results about the user that already had used our systems based on this methodology, thus finding out useful info

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

about our architecture effectiveness concerning recommendations and help support. Secondly, we were able to create user models, groups and dynamic stereotypes based on the process results of these algorithms targeting new users or users that we had little knowledge about.

1.5.2 Double Stereotypes

The next feature of our architecture lies in the stereotypes creation. Researchers in the field of intelligent systems always considered stereotypes to be a great tool for some first simple assumptions about their users but the static character of stereotypes was always a problem when users changed behaviour in the process of interacting with the application (Rich, 1983). Moreover, users that presented behaviour far different from those framed from the stereotypes were very difficult to be categorized to one. In order to address these problems and give stereotypes a more dynamic character we combined the stereotype creation process with the results of the clustering algorithm incorporated in the corresponding component.

We in fact used the clusters provided by the clustering algorithm to extract groups of users with similar tastes or mistakes, or groups of similar products. This process was conducted many times and resulted in a hierarchy of stereotypes. More specifically, we clustered users based on a small number of features at first, requiring only a very small number of clusters from the algorithm. Then this process was conducted again taking into account more features and requiring a larger number of clusters. These were repeated until a certain number of clusters were achieved. This number of clusters is defined by a rate that measures the intra-cluster consistency.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

This hierarchy of stereotypes resulted in stereotypes of different complexity classified in corresponding level, meaning that more complex stereotypes were classified in the lower levels of the hierarchy and more general stereotypes in the first levels. This process is conducted in real time in every application that incorporates our application. This means that the hierarchy and stereotypes change dynamically as more users become members of the e-commerce application and interact with it.

1.5.3 Different Media

Goy in her work about personalized ecommerce systems, points out the major target factors for adaptivity (Goy et al., 2007). One key factor is information about the device used. As Ardissono says the customer can access an online store by using a desktop PC, a laptop, a mobile phone, a PDA, an on-board device, or other. Every device has different characteristics, with respect to screen size, computation and memory capabilities, I/O mechanism (keyboard, touchscreen, speech ...), type of connection, bandwidth, and so forth. These aspects have also been classified as environment data by Kobsa and his colleagues (Kobsa et al., 2001). These facts mean that the medium used is a very important aspect of e-commerce architecture and needs to be taken under serious investigation.

According to the above facts we have introduced another novelty to our architecture targeting medium independence. The separate component of User Interface is responsible of measuring resources and types of medium and adjusting the user interface according to the user interface used. This means that if a mobile phone is used to access the e-commerce application than the fewer resources are used than the ones of the desktop application. This

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

was accomplished by the User Interface when we followed the ruled of source elements.

All e-commerce applications have a small number of the same source elements. These elements are the shopping cart, the products page, the specific product page, the categories page and the search page. These five source elements are present in all e-commerce applications regardless product, domain and medium. By defining the resources, design and structure for these five source elements we created the bone structure of the user interface for every e-commerce application. Every software designer could take this structure and alter it according to his/her tastes. The generalization of the user interface into these five elements provides the architecture with the ability to process every e-commerce system under the same view. Furthermore, the separate User Interface component is responsible for managing the systems resources according to the medium that the user interface is projected. In this way the user model knowledge remains the same in the User Model Server and only the interaction changes.

The last novelty presented in our architecture is the combination of adaptive hypermedia with the User Interface component. Creating a separate Adaptive Hypermedia component and the data flow between the User Interface component resulted in an effective way of creating a personalised user interface for every specific user either for the purpose of recommending products or intelligent help actions.

1.6 The RESCA-RUP software life cycle process

Adaptive applications present a greater difficulty in applying e traditional software life cycle techniques. A very useful tool in software life-cycle is the Rational Unified Process (RUP) (Kabassi

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

and Virvou 2006). RUP is an object-oriented process that advocates multiple iterations of the software development process. It divides the development cycle in four consecutive phases: the inception, the elaboration, the construction, and the transition phase. Each phase is divided into four procedural steps, namely, requirements capture, analysis and design, implementation, and testing. The phases are sequential in time but the procedural steps are not. Moreover, one important advantage of RUP is the highly iterative nature of the development process. For the above reasons, RUP can be selected as the basis for presenting adaptive systems too.

Our research introduces an RUP based software life cycle on incorporation of a clustering algorithm into a prototype e-shopping system called RESCA-RUP (Remote Shopping Clustering Algorithm RUP). The process has four major steps. First, designing and building the prototype system that does not include any clustering techniques. Second, evaluating the system and through this process obtaining data for the clustering algorithms. Third, comparing several clustering algorithms with the above data as input and choosing the most efficient algorithm. Fourth, incorporating the clustering algorithm into the system and building stereotypes based on this algorithm.

In adaptive applications it's very difficult to apply a general life cycle. A very useful tool in software life-cycle is the rational unified process (RUP) (Virvou and Kabbassi 2000). RUP is an object-oriented process that advocates multiple iterations of the software development process. RUP divides the development life cycle in four consecutive phases: the inception, the elaboration, the construction and the transition phase. Each phase is divided into four procedural steps, namely, requirements capture, analysis and design, implementation, and testing. The phases are sequential in time but the procedural steps are not.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Additionally, RUP is an object-oriented process; thus, it is appropriate for the development of graphical user interfaces such as the one described in our research. Moreover, one important advantage of RUP is the highly iterative nature of the development process. For the above reasons, RUP can be selected as the basis for presenting adaptive systems too. An implementation of RUP life –cycle into systems has been researched by Jaferian (Jaferian et al. 2005), which presented extensions on Business modelling and Requirement discipline of RUP. RUP has been used to present extensions that concern possible security threats and attacks.

Significant work in this field has also been done by Virvou and Kabbassi. Their work showed that developing a Graphical User Interface incorporating intelligence concerning in files and folders managing and modification like Windows Explorer. The system called IFM (Virvou and Kabassi 2000) presents an object - oriented approach in knowledge based software engineering of an intelligent GUI. In their second work (Virvou and Kabassi 2003) experimental studies were conducted for the IFM. Lastly, their third work (Kabassi and Virvou 2002) they extended the knowledge-based software life-cycle framework and they incorporated a multicriteria analysis. The common use of a modified RUP life –cycle as a tool for the design and development of IFM proved to be very efficient enhanced the software life –cycle process of IFM.

Similarly to IFM we chose to implement a parameterized version the standard RUP table for the purposes of the clustering algorithm incorporation. RESCA-RUP is based on iterations but does not specify what sort of requirements analysis has to be conducted for adaptive systems and what kind of prototype has to be produced during each phase or procedural step. This table follows the phases and procedural steps of RUP but the difference is that we specify what prototype has to be constructed in each phase and what kind of experiment has to be conducted. Consequently, here is presented

a modified solution to the problem of clustering incorporation and how this procedure can be generalized and be applied in an entirely different medium or product.

The resulted systems through this life cycle process were very effective and helped its customers choose the right product. This research shows that the RUP life cycle process built in our previous research can also be applied in other domains with roughly the same steps. Furthermore, through the developing and evaluating procedures, we proved that RUP can be selected as the basis for presenting adaptive systems too. The evaluation was conducted in two different systems, selling different products through different technologies. The evaluation results showed that the basic steps of the proposed by out RUP life-cycle remained the same. This means that the different domain and media did not play a sufficient role in changing the procedure thus resulting in medium and domain independent procedure.

Our research on RUP software life-cycle did not stop on the creation of a software life-cycle process but continued on the combinations of RUP steps with the UML technique. By combing UML and RUP we created a fully modelled procedure for incorporating a clustering algorithm into a personalized recommending e-commerce system. UML helped us model the architecture and procedures of the software thus making them clearer for the average programmer. Furthermore, the repeatability of RUP created a series of general to more complex UML diagrams of every program procedure and in this way a diagrammatic image of the whole research was created; a tool very useful for future programmers of adaptive ecommerce systems.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

1.7 Phd Thesis Contents

This Phd thesis is organized in seven chapters. Chapter 2 focuses on the related work done in the fields of recommending systems, intelligent help systems and combined of the above systems. Chapter 2 also gives comparisons of our methodology on these fields against the methodologies of the systems presented in this chapter. Chapter 3 presents our methodologies on different media and machine learning algorithms developed for the adaptive remote shopping applications. Chapter 4 presents RESCA-RUP the software life cycle based on the rational unified process for adaptive remote shopping applications. This chapter focuses on presenting how a researcher can successfully incorporate the methodologies developed in this phd.

Chapter 5 presents a generic architecture consisted of three tiers, the user modelling components, the user modeling server and the adaptive interface components. This generic architecture incorporates the innovative methodologies based on different media and machine learning algorithms. Chapter 6 presents evaluation studies on remote shopping applications incorporating the innovative methodologies presented in the previous chapters. Chapter 7 seven discusses contributions and innovations made through this phd in the research fields that this phd falls under. Furthermore, presents the reader with general conclusions of this researcher and discusses open research issues and future amplifications.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

CHAPTER 2
RELATED WORK

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

2. Related Work

This phd research covers many fields of software engineering and informatics. In this chapter we will present researchers that have conducted similar work in the fields that our research applies to. The main topic that our research addresses is the field of e-commerce as our main purpose of our research is the construction of effective intelligent remote shopping applications. Adaptivity and user modelling are two very wide spread concept in software engineering of intelligent applications in general. Likewise, our methodology uses such techniques to have the ability to be adaptive to users' behaviour. An indirect but affecting field of informatics to our research is e-learning. Many e-learning researchers have conducted significant works in the field that were taken into consideration during our research. For example, stereotypes are widely used in the field of stereotypes and similar techniques are used in our research too.

This chapter follows with the presentation of works belonging to fields of recommendation systems and intelligent sales assistants. These two fields are the most relevant with our work and a direct comparison with works conducted in these fields is mandatory and will provide us with a better picture of the differences between work conducted by other researchers and us. In the research field of recommendation systems special emphasis has been given to application that incorporate machine learning algorithms as these applications are directly similar to our methodology. Mobile and TV recommendation must also be addressed as an affecting field to our work as our research applies to different media and more specifically applies to mobile phones, interactive TV and internet shopping. Our research during this phd has expanded not only in creating methodologies that would recommend products to users in effective ways but also support

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

while they buy these products. In this way a thorough research of the field of intelligent help systems was conducted during this phd in order to extract information and effective ways of supporting users while they interacted with the system. The research field of animated agents was also greatly researched during this phd. Animated agents are a great tool for presenting users with a more human-like interface and help in supporting users in a friendlier way. Animated agents can interact with user directly, giving users the feel of human-like conversation.

The last paragraphs of this chapter cover two very important fields of our research. Generic architectures are greatly discussed here in order to present the need for a generic architecture in the field of shopping applications and the novelty of our methodology towards other work conducted in the field. Research work conducted in the field of software life cycle and rational unified process (RUP) is also discussed in this chapter. We have used RUP in order to create a methodology for incorporating machine learning algorithms and user modellig techniques into remote shopping applications. Combining our generic architecture and RUP approach we wanted to give future researchers an integrated tool for developing intelligent remote shopping applications.

2.1 E-commerce

In e-commerce there are mainly two types of commerce. There is B2B (business to business) commerce and B2C (business to customer) commerce. The first applies between companies and the second is the more traditional commerce and targets individual customers. However, both types the mainly aim in selling products effectively. Nowadays many researchers and companies have tried to create systems that manage content in a more general way thus creating a kind of framework for products. In e-commerce industry

these systems are called CMS (content management system) and CRM (customer relationship management) systems. Such a system has been created by Fensel and his colleagues. (Fensel et al. 2001) and its aim is extract product content in order to help business create an efficient CMS system. Its reasoning is based on creating a structure of classes based on information extraction from products. Our system on the other hand focuses on customer behaviour than product information extraction thus basing its reasoning in an entirely different approach. Our system creates customer groups by incorporating a clustering algorithm in its reasoning mechanism.

Another approach, on B2B market is GoldenBullet by Ding (Ding et al. 2002). GoldenBullet is a system that applies techniques from information retrieval and machine learning to the problem of product data classification. The system helps to mechanize an important and labour-intensive task of content management for B2B Ecommerce. Again, our system also uses machine learning techniques, but these techniques are used on user behaviour and not product classification. An interesting approach has also been created by Albert (Albert et al. 2004). They created a model for design and management of content and interactivity of customer-centric web sites called GIST. The key to their approach is the identification of nanosegments and nanoflows, the specific gap analyses for these groups of visitors, their preferred paths, and the consequent actionable findings that are derived from the process.

Here, we present an entirely different approach that is based on user models created by machine learning techniques. These techniques create groups of similar user based on their behaviour dynamically as the customer navigates through the system. An approach that tries to generalize content management for products has been followed by Trappey (Trappey 2004). Their global content management services for product providers and purchasers is based on xml and proposes a suitable content search engine, implements

a Web user interface for a variety of content users, and provides Web tools for e-content creation, maintenance and management. On the other hand, our system is based on a database server and not xml schemas. It creates user group that present similar behaviour and it can be used as a module on any system instead of forcing the developer to use its own web tools.

2.2 Adaptivity and User Modelling

Despite the fact, that the above system try to generalize their role and create independent systems concerning products they do not address user adaptivity issues. On the other hand many systems have tried to incorporate adaptive mechanisms in order to help customers. These systems are called recommendation systems. The use of recommendation systems is widely spread and can be found in many fields such as e-commerce, TV-commerce, program recommendation and many others. The recommendation techniques usually involve the construction of user models that are either based on explicit user information or on data about the user behaviour that is collected implicitly by the system.

User models can be constructed using many techniques adaptive hypermedia (Brusilovsky, 2001) or individual user models (Rich 1983, Agichtein et al. 2006). One powerful way also for creating user models is based on stereotypes. Stereo-types were originally invented for the system called Grundy (Rich 1998) that recommended books to users based on their preferences. In this system it was defined that a stereo-type represents a collection of attributes that often co-occur in people and thus they enable the system to make a large number of plausible inferences on the basis of a substantially smaller number of observations.

Building stereotypes involves mainly defining the triggering conditions (the conditions that enable a specific stereotype), and

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

the inferences (what can be assumed for users belonging in the triggered stereotype). As Kay (Kay 2000) points out there are two ways for constructing stereotypes, one is hand-crafted and the other one is empirically-based. In the hand-crafted case the designer of the system makes assumptions about the stereotype groups whereas in the empirically-based stereotypes there is an important role of machine learning techniques in acquiring them. However, in the majority of the existing commercial personalization systems, the personalization process involves substantial manual work and most of the time significant effort on the part of the user; despite these problems, the number of personalized web pages is increasing (Pierrakos et al. 2003).

Clustering algorithms are also a common type of algorithmic implementation of adaptive systems. An adaptive and personalized system according to Brusilovsky is a system that is able to adapt its behavior according to changes of a user's behavior. (Brusilovsky 2004). Recommender systems (Resnick and Varian 1997) are specific adaptive systems that use a type of information filtering (IF) technique and aim to present to the user information items (movies, music, books, news, web pages) that the user may be interested in. To do this the user's profile is compared to some reference characteristics. These characteristics may be from the information item (the content-based approach) or the user's social environment (the collaborative filtering approach). When building the user's profile a distinction is made between explicit and implicit forms of data collection.

Examples of explicit data collection include the following:

1. Asking a user to rate an item on a sliding scale.

2. Asking a user to rank a collection of items from favourite to least favourite.

3. Presenting two items to a user and asking him/her to choose the best one.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

4. Asking a user to create a list of items that he/she likes.

Examples of implicit data collection include the following:

1. Observing the items that a user views in an online store.
2. Analyzing item/user viewing times
3. Keeping a record of the items that a user purchases online.
4. Obtaining a list of items that a user has listened to or watched on his/her computer.
5. Analyzing the user's social network and discovering similar likes and dislikes

Our system uses implicit techniques in order to create user profiles according to their interests. The recommender system compares the collected data to similar data collected from others and calculates a list of recommended items for the user. Recommender systems are a useful alternative to search algorithms because they help users discover items they might not have found by themselves.

Adaptive help systems (also called intelligent help systems) are a specific kind of help systems and are recognized in research direction on the crossroads of Artificial Intelligence (AI) and Human-Computer Interaction (HCI). The goal of adaptive help systems (AHS) is to support users in an adaptive way and dynamically change their behaviour as the user interacts with the system (Oppermann 1994). This goal can be achieved in many ways. For example (Virvou and Kabassi 2004) a cognitive theory like Human Plausible Reasoning can be implement in order to create an intelligent user interface to help users recover from their mistakes and perform tasks easily.

Unlike traditional "static" help systems that serve by request the same information to different users, AHS attempt to adapt to knowledge and goals of individual users offering information about most relevant aspects in a most relevant way. Our system uses an adaptive help module based on every specific user’s model that help users in a personalized way avoid navigation mistakes due to his/her age.

2.3 E-learning

The field of e-learning was the first to implement adaptivity and user modelling techniques in order to create successful tutoring systems. These techniques were later passed to the e-commerce field. In these next paragraphs we present some successful intelligent e-learning systems that their adaptivity techniques constituted as the basis of adaptive e-commerce systems also.

There are many examples of intelligent tutoring systems in e-learning, like the work of Frasson et al, 1997. Their work uses pedagogical agents to in a multi strategic tutoring system. Their paper describes the use of actors for implementing pedagogical strategies and more generally for detecting which strategy is more suited to a given learner. Their approach leads to the definition of a multi-strategic ITS, based on pedagogical actors, that is able to switch among various strategies. In this framework pedagogical actors are used to model the expertise of the various pedagogical strategies. Another important work on the same field has been done by Heffernan and Koedinger, 2002. Their work presents present Ms. Lindquist, an Intelligent Tutoring System (ITS) designed to carry on a tutorial dialog about symbolization. Ms. Lindquist has a separate tutorial model encoding pedagogical content knowledge in the form of different tutorial strategies, which were partially developed by observing an experienced human tutor.

A very important work has also been done by Suraweera and Mitrovic, 2004. Their paper presents KERMIT, a Knowledge-based Entity Relationship Modelling Intelligent Tutor. KERMIT is a

problem-solving environment for the university-level students, in which they can practice conceptual database design using the Entity-Relationship data model. KERMIT uses Constraint-Based Modelling (CBM) to model the domain knowledge and generate student models. We have used CBM previously in tutors that teach SQL and English punctuation rules.

Significant work has also been done by Brusilovsky et al., 1996 with ELM-ART. Their research discusses the problems of developing WWW-available ITS and, in particular, the problem of porting existing ITS to a WWW platform. They present the system ELMART which is a WWW-based ITS to support learning programming in Lisp. ELM-ART demonstrates how several known ITS technologies can be implemented in WWW context. Another important work in web ITS has also been done by Tsiriga and Virvou, 2004. In their work they describe a framework for the initialization of student models in Web-based educational applications. The framework is called ISM. The basic idea of ISMis to set initial values for all aspects of student models using an innovative combination of stereotypes and the distance weighted k-nearest neighbour algorithm. In particular, a student is first assigned to a stereotype category concerning her/his knowledge level of the domain being taught. Then, the model of the new student is initialized by applying the distance weighted k-nearest neighbour algorithm among the students that belong to the same stereotype category with the new student.

Last but not least we have two different approaches on the field of e-learning ITS. The first research has been done by Graesser et al., 2005. Their work presents AutoTutor that simulates a human tutor by holding a conversation with the learner in natural language. The dialogue is augmented by an animated conversational agent and three-dimensional (3-D) interactive simulations in order to enhance the learner's engagement and the

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

depth of the learning. And the second work has been done by Baker et al., 2006 and introduces a system which gives a gaming student supplementary exercises focused on exactly the material the student bypassed by gaming, and which also expresses negative emotion to gaming students through an animated agent. Students using this system engage in less gaming, and students who receive many supplemental exercises have considerably better learning than is associated with gaming in the control condition or prior studies.

2.4 Recommendation Systems

E-commerce applications have become very popular since they provide easy access to all kinds of products. Nowadays tv-shopping has also become very popular due to the fact that relies on the technology of interactive TV and thus has the potential to be a user-friendlier medium than a desktop computer since customers are more likely to be more familiar with TV than with computers. Both of the applications mentioned above belong to the well known field of recommending systems. However, most of existing applications are generic and do not address specific needs, preferences and attributes of individual customers. A remedy to this problem can be achieved by web personalization techniques.

As De Pessemier and his colleagues (Pessemier et.al. 2008) suggest, new technologies such as Internet, iDTV, and mobile applications create the possibility to advertise in a different, more attractive manner than the traditional commercial breaks. This leads to a rise of interactive commercials on iDTV, banners on the internet, and commercials on mobile devices. This leads the market to more converged architectures corresponding to different types of mediums such as mobile phones, internet. Despite the fact that there are many techniques in order to achieve personalization, there is a lack in the effort to produce frameworks that can be applied to any recommending application without concerning the product that sells or the medium that this application uses. Also as Veruska Aragao suggests there is no widely-available mechanism to allow users to personalize their interaction with web data and services (Aragao et.al. 2001) meaning that as years pass the need for general personalization architectures is becoming more and more imperative. The difficulty of making such a framework is high and its reinforced by the fact that product brokering requires assisting users in finding information in a complex multidimensional space (Pu & Faltings, 2002).

There are many remote shopping applications that try to make recommendations using many techniques. These techniques usually involve the construction of user models that are either based on explicit user information or on data about the user behavior that is collected implicitly by the system. User model can be constructed using many techniques like stereotypes, adaptive hypermedia or individual user models.

For example, WindOwls (Kazienko and Kolodziejski, 2005) is a recommending system that uses user modeling techniques to propose products to individual users. WindOwls uses association rules to calculate weights in order to group acquired tastes together. These weights are used as tastes inferences. In contrast, in our methodology we use a clustering algorithm to group users according to their tastes. More specifically, our system extracts representatives from a clustering technique of users' tastes and then uses these representatives to recommend its products.

Another interesting approach has been made by Choi and his colleagues (Choi at al., 2006). They chose a multi-attribute decision making method to find similar products. In their system, the customer must first order a product in order for the system to propose a similar one. The procedure of finding a similar product is based on a weighted attribute theory that can fill incomplete specifications of a similar product if these specifications do not exist. Our system follows a very different approach. It observes users' navigational behaviour and tries to infer their tastes through their visits and navigation on product pages. On the other hand in our system a product can be proposed even if a customer has not bought anything. Moreover our system uses a clustering technique in order to acquire representing vectors of taste percentages. It uses these vectors to propose a product instead of the weighting product attributes technique, used in Choi system.

Another recommending system that uses clustering techniques in order to group products is the system proposed by Guan (Guan et al., 2005). Their system uses ranking as the major way to acquire generic attributes from products. The following step of their procedure includes clustering new attributes into the different groups of the above attributes. Guan used the k-NN algorithm in order to create these clusters of similar attributes and in this way find similar customer tastes. On the other hand in our system instead of asking customers to rank products we observe their behavior and interaction with the system. In our system we do not cluster similar product attributes but similar taste percentages in product attributes. In this way instead of creating a system that is based on product attributes we create a system that is based on customer taste percentages which gives us a product independent system. Moreover, instead of asking the customer to define what attribute s/he likes more, our system learns it from his/her high taste percentage.

Many systems (Schwarzopf 2001, Ardissono and Torasso 2000) have used clustering algorithms for the revision of initial user models that had been created based on hand-crafted stereotypes. In constrast to those systems, our methodology was developed in two phases. First, an adaptive version of the system was created which did not incorporate any stereotypes. This version of this system was used by real users and all their actions within their interaction with the system were collected and analyzed by the artificial immune network. As a result of this process we constructed double stereotypes that were than incorporated into the system for improving the initialization and accuracy of the user modeling component of Vision.Com. The fact that we have constructed Vision.Com in two phases has provided the advantage of the construction of user stereotypes based on the clustering algorithm.

Another interesting work in the same field is developed by Cayzer and Aickelin (2002) and Morrison and Aickelin (2002). That system utilized an Artificial Immune System (A.I.S) in order to tackle the task of film and web site recommendation by collaborative filtering. In that system there was a movie database called EachMovie that was used as a source of votes concerning movies that had been seen. These votes were explicitly supplied by each user and were used in order to build a recommender system that provided estimation votes for unseen movies. In contrast, in our work we obtained the movie preference related data in an implicit manner. In our approach we aim at recommending movies to a user based on his/her inferred preferences. Individual user preferences are inferred based on users' interaction with the electronic video store.

In the Cayzer and Aickelin system there is neither a user model nor stereotype-based reasoning. Thus, their system cannot recommend movies that have not been seen by any user. In our case the fact that we construct user models allows the system to predict user preferences in all kinds of movies whether these have been seen by other users or not. Our aim was to develop a regulated network of antibodies that corresponded to multidimensional representative vectors of the user preferences.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

User preferences refer to movie attributes such as who the director of the film is, who the leading actor is, what kind of film it is etc. More specifically, our work has focused on how a significant amount of redundancy within the customer profile dataset can be revealed and reduced, how many clusters intrinsic to the customer dataset can be identified and what the spatial structure of the data within the identified clusters is.

Another approach that goes beyond attributes exploitation like the work mentioned previously is done by Alspector and his colleagues (Alspector et.al. 1997). This approach builds a CART network with the help of three main adaptive techniques: feature-based, clique-based and linear model. The feature-based approach rests on the notion that the features of the movies can be useful in recommending movies. Some of the features that can be used to recommend movies include MPAA ratings (e.g., parental guidance), expert critic ratings, movie category (e.g., drama), name of the director, leading actors/actresses, and awards received. For example, a particular user may have a strong inclination to see only movies that are rated G (general admittance), acted by Ben Kingsley, and belonging to the category Comedy or Drama. Thus, the feature-based approach exploits the bias of a user towards a set of important features of the movies. Their study showed that an effective movie-recommendation system should combine all these approaches in order to maximize performance.

On the other hand, our system takes another approach on the matter. We are using a clustering algorithm in order to create groups of similar users in interest. The degrees of interests of users are calculated based partly on their stated preferences and partly on the inferred interests from the users’ actions. Furthermore, our system does not base its reasoning system in movie ratings but in user behaviour. For example, if a user puts a movie in his/her cart then this means that the user is interested in this movie. Then if the

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

user reaches the exit and does not buy the product, this means that s/he is not interested enough to pay the money to buy it whereas if s/he buys it then it means that the product is of the highest interest to the user at that moment. Moreover, our system is a general framework for recommendations that can be applied on any product not only movies.

A very interesting technique also, based on a rating system has been conducted by Li and Kim (Li & Kim, 2004). The system acquires rates and then calculates fuzzy inferences and extracts similarities between users. Their method proved very successful according to the evaluation presented.

In the field of tourism, Ardissono (Ardissono et al., 2003) has conducted significant research. The system created for personalized recommendations about tourist attractions is called Intrigue. This system targets both desktop and handheld devices. Personal recommendations are extracted by combining users' stereotypes and presented in the form of lists relevant to this user's taste. Simiraly our architecture uses stereotypes but in an entirely different approach. Our architecture instead of combining users' stereotypes it constructs hierarchies of stereotypes based on user behaviour. These hierarchies follow the strategy of more general to more complex stereotypes. In this way users are classified to more complex stereotypes as they continue to use the ecommerce system more and more. Moreover, the recommendations are not presented in the form of personalized lists based on relevancy but they are presented in two ways. First, with the help of adaptive hypermedia and more specifically adaptive symbols that correspond to different levels of taste and secondly, with the help of a lifelike agent that conducts general proposals to the users.

In the field of user plan recognition Ardissono and Sestero have conducted a very interesting research (Ardissono and Sestero, 1995). They have used dynamic user models to recognize user

plans through a dialogue interpretation framework. Their user modelling creation is based on stereotypical information and acquisition rules. Their research shows how to embed a user model in plan recognition. Simiraly our fremawork is used to embed user modelling and adaptivity but in a different domain, the domain of ecommerce systems. Instead acquisition rules our system relies on machine learning algorithms to infer user needs and tastes.

Another very interesting approach has been conducted by ardissono and her colleagues. (Ardissono et al., 2001). Their goal is to create an adaptive system for personalised access to news. Their system uses user modelling techniques based on stereotypes in order to achieve personalization and it assumes that news data as a hierarchical structure. Despite the fact our system follows architecture similar to this system, based on modules, it differs a lot on many aspects. Our system does not rely only on stereotypes for user model creation and the presentation of recommendations is based on adaptive hypermedia techniques.

The field of customer relationship manager (CRM) was also greatly researched on hot to incorporate user modelling techniques into CRM systems. Significant work has been conducted by Piller and Schaller (Piller and Schaller, 2002). In their work they present four models of collaboration for CRM. Their work proves that individualization on customer needs can only be achieved by collaborating different partners together and not relying only in one firm. Our framework with the generalization of the user modelling components and the reusability that is offered through the separation and generalization of these components can make collaboration easier ass data from different stores can be stored in the same user model server.

Work has also been done in incorporating machine learning techniques into systems. Such a work has been done by Castro and his colleagues (Castro et al., 2001). They constructed a fuzzy

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

machine learning technique in order to obtain evident knowledge of an existing set of training examples. This technique can be used to help expert systems obtain knowledge about users and thus provide better recommendations. Their research showed that knowledge acquisition process as a process for extending, updating and improving an incomplete knowledge base, in which machine learning is useful.

Another common technique for achieving adaptivity is clustering. Such a work based on clustering has been done by Kim and Ahn (2008). They constructed an alternant of the GA K-means in order to create a recommender system in an online shopping market. They applied this algorithm to a real-world case for market segmentation in electronic commerce.

Another work using clustering techniques concerning software segmentation, recovery and restructuring was done by Lung and his colleagues (Lung et al., 2004). They used bottom–up method and clustering to group similar components together to form clusters or subsystems. Those clusters or subsystems are partitions which constitute a system. According to Lung applications of clustering analysis can be found in many disciplines but all comprise three common key steps: (1) Obtain the data set, (2) Compute the resemblance coefficients for the data set and (3) Execute the clustering method.

Such an approach has been made by Miao and his colleagues (Miao et al., 2007). Their paper proposes personalized recommendation agents called fuzzy cognitive agents. Fuzzy cognitive agents are designed to give personalized suggestions based on the user's current personal preferences, other user's common preferences, and expert's domain knowledge. Fuzzy cognitive agents are able to represent knowledge via extended fuzzy cognitive maps, to learn users' preferences from most recent cases and to help customers make inferences and decisions through

numeric computation instead of symbolic and logic deduction. In contrast, in our system we use a clustering algorithm to group users according to the information that itvmobi gets through their behavior. In particular, our system extracts representatives from a clustering technique of users' tastes and then uses these representatives to recommend its products.

There are other web-based recommendation applications that have also used clustering algorithms (e.g. (Jin et al. 2004, Adil et al. 2000, Menczer et al. 2002, Jina et al. 2004, Ajith 2004, Wang et al. 2004)). All of these applications were primarily concerned with the acquisition of user behaviour-related data. In contrast, our work has focused on incorporating an clustering algorithm in order to personalize the developed eshop system through grouping similar users' behaviour. This means that we have used a very different algorithm that has not been used by these systems.

2.4.1 Intelligent Sales Assistants

The concept of the intelligent sales assistant is wide spread in the research domain of personalised product recommendations. The general idea of intelligent sales assistant has been presented by Boy and Gruber (Boy and Gruber 1990). Nowadays the intelligent sales assistant and variants has been used by many researchers such as Molina (Molina 2001). Molina proposes an advanced type of software application that simulates how a sales assistant dialogues with a consumer to dynamically configure a product according to particular needs. He presents a general knowledge model that uses artificial intelligence and knowledge-based techniques to simulate the configuration process.

Another significant work in the field of sales assistant has been done by Shergil et al (Shergil et al., 2004). In their paper they

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

describe a computerized intelligent sales assistant computer package that gives sales personnel the ability to allocate their time where it will produce the best results, both for the customer, and for the business. As the customer enters the shop, this customer's features are scanned and analyzed and the customer is categorized as a browser, future customer, potential customer or buyer. The customer's facial data are also used to retrieve their details, if available, from the shop's database, and the data are used to determine whether a human sales assistant is required. The software package's expression recognition feature would also tell the sales personnel whether or not the customer requires or desires assistance in the first place.

Important work in the field has also been done by Schneider (Schneider, 2003).In his research an adaptive shopping assistant system utilising plan recognition is described. Radio Frequency Identification (RFID) sensory is used to observe a shopper's actions, from which the plan recogniser tries to infer the goals of the user. Using this information, an automated assistant provides help tailored to the shopper's concrete needs. We discuss why it is crucial to make the plan recognition process itself user adaptive and present ideas how to realise this through modification of existing plan recognition approaches.

Another work in the field has also been researched by Stahl et al. (Stahl et al., 2004).They give a survey of the research project REAL, where they investigate how a system can proactively assist its user in solving different tasks in an instrumented environment by sensing implicit interaction and utilising distributed presentation media. The architecture presented uses a blackboard to coordinate the components of the environment, such as the sensing and positioning services and interaction devices. A ubiquitous user model provides contextual information on the users' characteristics, actions and locations. The user may access and control their profile

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

via a web interface. They also present two mobile applications to employ the environmental support for situated dialogues, a shopping assistant and a pedestrian navigation system. Both applications allow for multi-modal interaction through a combination of speech, gesture and sensed actions such as motion.

2.5 Mobile recommendation

In the field of mobile-shopping little work has been done. An interesting approach has been made by Billsus et al (Billsus et al 2002). In their work they have created and adaptive interface for mobile devices that makes personalised suggestions based on search questionnaires. Despite that fact that their work can be very efficient, it mainly focuses on filtered lists of suggestions on various areas such as restaurants, presented to the user's mobile device. Their adaptive user interface lacks the generality of a user modelling server that can be applied in both internet and mobile shopping applications. On the other hand, our e-shop architecture does not lack generality and it can be incorporated to every mobile e-shop system.

A very important research in the field of mobile recommendation has been made by Miller and his colleagues (Miller et al., 2003). In their paper they have examined the challenges associated with building a recommender system for four wireless interfaces: A PDA using the AvantGo service, a web enabled cell phone browser, a voice interface over phone, and a wireless PDA. They have presented their solution to these challenges called MovieLens Unplugged. MovieLens is a recommendation system, a companion service to the MovieLens website. MovieLens Unplugged provides recommendations to customers based on a five star ratings system, a wishlist and search criteria. On the other hand our

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

system basis its movie recommendation system on customer navigational behaviour. It measures customers' interests based on pages' visits, movie moved to cart and bought movies. Furthermore, our system uses adaptive hypermedia to create an adaptive user interface. Moreover, our system evolves an intelligent incremental initialization based on dynamic double stereotypes of both products and users.

In the field of mobile shopping also, we have the work of Yang and his colleagues (Yang et al., 2008). In their study they investigated location-aware personal recommendations by constructing a system that amalgamates the information abundance of the Internet with the tangible richness of physical shopping, in terms of location. They discuss how the proposed approach is implemented to exploit the functionality afforded by a powerful location-aware architecture, and have evaluated its performance using both synthetic and empirical data. In order to improve the performance of their system they used a technique baed on R+ trees. Our system on the other hand, is a generic mobile architecture that focuses on mobile e-shop. Our architecture uses recommendation techniques with the help of clustering, stereotypes and adaptive hypermedia to provide personal suggestions to customers according to their behaviour.

In the same field we also have the work of Kurkovsky and Harihar (2006). They present SMMART, a contextaware, adaptive and personalized m-commerce application designed to deliver targeted promotions to the users of mobile devices about the products they like while guarding the users' identity and protecting them from any unsolicited messages. Promotions distributed by SMMART are personalized by performing intelligent matching of the user's shopping interests to current promotions available at a retail site. SMMART can adapt to changing preferences of its user by inconspicuously monitoring his or her shopping habits. Our system

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

also monitors users' behaviour but uses three different techniques to propose to customers' products that are interested in. These techniques are clustering algorithm, dynamic double stereotypes and adaptive hypermedia.

A vey interesting approach has also been done by Setten and his colleagues (Setten et al., 2004). Their work describes a context-aware mobile tourist application COMPASS that adapts its services to the user's needs based on both the user's interests and his current context. In order to provide context-aware recommendations, a recommender system has been integrated with a context-aware application platform. The recommender system uses the following prediction methods: social filtering, case-based reasoning (CBR), item-item filtering and category learning. On the other hand, our system uses entirely different prediction methods which include clustering similar users, double stereotypes and adaptive hypermedia. Furthermore, our system focuses on a different domain which is product recommendation.

A significant work in the field of mobile recommendation systems has also been made by Andronico and his colleagues (Andronico et al., 2003). In their work they present a framework of a Learning Management System (LMS). In this framework they integrated InLinx, a multi-agent Web-based hybrid recommender system that provides an on-line bookmarking service in their m-learning architecture. Their work showed that InLink's hybrid approach is able to effectively filter relevant resources from a wide heterogeneous environment like the Web, taking advantages of the shared interests among users without losing the benefits provided by content analysis. Our system on the other hand, is a generic mobile architecture that focuses on mobile e-shop. Our architecture uses recommendation techniques with the help of clustering, stereotypes and adaptive hypermedia to provide personal suggestions to customers according to their behaviour.

Last but not least, in the field of mobile gaming significant research has been made by Bell and his colleagues (Bell et al., 2006). In their work they describe an architecture that allows the recommendation of new system components from systems with similar histories of use. Software components and usage histories are exchanged between mobile users who are in proximity with each other. They applied this architecture in a mobile strategy game in which players adapt and upgrade their game using components from other players, progressing through the game through sharing tools and history. Our system focuses on entirely different field and instead letting users' share opinions; it monitors all users' behaviour and then extracts conclusions from these behaviours

2.6 Interactive TV and TV Shopping

Interactive television (Jensen, 2005) allows viewers to interact with television content as they view. In iTV the viewer must be able to alter the viewing experience or return information to the broadcaster. This "return path" or "back channel" can be realized by telephone or cable. Cable viewers receive their programs via a cable, and in the integrated cable return path enabled platforms, they use the same cable as a return path. Satellite viewers (mostly) return information to the broadcaster via their regular telephone lines. Increasingly the return path is becoming a broadband IP connection, and some hybrid receivers are now capable of displaying video from either the IP connection or from traditional tuners. Some devices are now dedicated to displaying video only from the IP channel, which has given rise to IPTV - Internet Protocol Television. The rise of the "broadband return path" has given new relevance to Interactive TV, as it opens up the need to interact with Video on Demand servers, advertisers, and web site operators.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

T-commerce or television commerce is e-commerce undertaken using digital television. It has yet to become as widespread as e-commerce on the Internet (Jensen, 2005). Several technologies can be used to achieve t-commerce. These technologies generally work like ordering a cable pay-per-view or video-on-demand (VOD) movie. Shoppers press a button on the remote control to select an item and toggle through such choices as size, shipping address and payment. In most cases, shoppers must register in advance such information as addresses and payment preferences. T-commerce technology requires satellite TV or digital cable service. The advantage of t-commerce is the ability to target markets that are almost impossible to reach through traditional channels.

2.6.1 TV Recommendation

An important field in recommendation that has a large affinity with interactive TV is TV programs recommendation. Important steps have been made in this field too, e.g. (O' Sullivan et al. 2004, Marbury et al., 2004). Their aim is to help users find a program of interest to them. Marbury's work applies a user model based on keywords and named entities in every user's query for a tv program and in this way provides the user with personalized results based on this user interests. This system uses an individual user model technique in order to propose clips of programs that a user might be interested into. Despite the fact that this technique provides the system with good results for a user after several queries for programs, it lacks on the topic of user modelling initialization. If a new user enters the system the program cannot recommend an appropriate set of clips according to his/her tastes.

On the other hand, our system makes recommendations through grouping users with similar tastes by the use of k-means

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

clustering. Every user belongs to a group and every group has a representative. In this way, if a new user enters the system and interacts only a little with it, the system can extract his/her tastes with the help of his/her representative. Moreover, by doing so, the system provides the user with immediate product recommendations and solves the problem of leaving a new user with the feeling of "no results according to his/her tastes", that is usually witnessed on systems that take long to create the new user's profile.

On the other hand O'Sullivan created the personalized Electronic Programme Guide (pEPG), which they believe that is one solution to the problem of locating the right programmme information at the right time. pEPG leverages artificial intelligence and user profiling techniques to learn about the viewing preferences of individual users in order to compile personalized viewing guides that fit their individual preferences. Their research proposes the use of data mining techniques as a way of supplementing meagre ratings-based profile knowledge with additional item-similarity knowledge that can be automatically discovered by mining user profiles. They argue that this new similarity knowledge can significantly enhance the performance of a recommender system in even the sparsest of profile spaces.

This system is a personalized program guide that recommends TV programs after recording them into mpeg format in the hard drive. A collaborative filtering technique is used to acquire users' preferences. This technique uses data mining in order to acquire the meta-data needed for making that ranking of the TV programs through the use of a similarity matrix. In our system instead of finding similarities between the products, we cluster user interest percentages using the k-means clustering algorithm. After the clustering, we calculate representative vectors of users' preferences and then use these representatives to compare them through association rules with the products database in order to

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

achieve product recommendations, thus finding similar users and not products.

Another application in the fields of interactive TV and recommendation is the application of Chorianopoulos and Spinellis (2004). In this application a Virtual Channel Api is created that allows the user to skip videos. The application uses meta-data of music video clips in order to make recommendation through the use of an animated agent. On the other hand our approach does not rely only on information about a product, in Chorianopoulos and Spinellis case the music clip, but also measures the customers' behaviour throughout the system and clusters similar tastes calculated from customers' behaviours.

An interesting approach has been made by Goren-Bar and his colleagues (Goren-Bar et al., 2004), which discusses a recommendation system with an emphasis on the support of the family group. Questions of interface design and interaction metaphors are evaluated in "User Interface Development for Interactive Television" and discussed along with user tests. Lessons from iTV concept design are presented within "Integrating a Game with a Story".

This field has also been greatly researched by Ardissono and her colleagues (Ardissono et al., 2003). In their work they created a personal program guide that tailors recommendations of TV programs to the viewer's interests. Their work similarly to PERCOM follows the idea of three types of user models, explicit, stereotypic and dynamic. The difference is that PERCOM takes these three types of information, explicit, from user behaviour and stereotypic and combines them into unified representatives of the user. Then it extracts information about a user's inference through these representatives. Then PERCOM instead of constructing a ranking system like the personal program guide it uses different adaptive hypermedia techniques to personalise the user's interface according

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

to his/her needs and inferences. Furthermore, PERCOM with the use of separate modules architectures succeeds in generalizing the purpose and architecture of these modules thus making possible to be incorporated to any product and medium.

Another interesting research in the field that takes up with the problem of stereotype creation is conducted by Gena (Gena 2001). Her work shows how a user modeling knowledge base for personalized TV servers can be generated starting from an analysis of lifestyles surveys. Her research shows that stereotypical knowledge is meaningful for users clearly fitting a lifestyle, but does not make good predictions in the cases where users match different lifestyles in different aspects of their behavior. PERCOM similarly uses stereotypic for personalization but in order to avoid different lifestyle tastes takes into consideration two more types of information, explicit and user behaviour. The explicit information contains social and demographic data and in this way different life styles can be extracted through this data. Furthermore, user models based on user behaviour reinforce PERCOM's decision about personalised recommendations.

2.7 Intelligent Help Systems

Adaptive help systems are systems that try to help users perform tasks by exploiting information that acquire implicitly through users' actions. There many adaptive help systems in various fields of science like hydrological time series (Zounemat-Kermani et al. , 2008), medical support and patient life quality (Tokmakcj et al., 2008) and exonomics (García-Barrios et al., 2008) but very few in fields of computer support, web support and recommendation. Such a system is AdaptHelp by Iglezakis (2004). She presents an approach that uses the techniques of plan

recognition not only to infer short-term plans and goals, but also to infer the long-term procedural knowledge of a user in a non-binary way. The information about the procedural knowledge in terms of activation builds the user model of AdaptHelp, an adaptive help system for web-based systems. AdaptHelp creates user models based on web logfiles and by using xml tries to help users. On the other hand, iTVMobi does not use xml data to create user profiles. Instead it combines client-server architecture and databases to exploit user information extracted from users' behavior throughout the system.

Another system that helps the elderly is made by Zhao and Tyugu (1998). Their system has a personalized web browser that helps elderly people browse the web. The browser adapts its presentation according to the users' behavior. Similarly with their system, our methodology also acquires information from the users' navigational behavior. However, differs from that system in the kind of information that it acquires and the kind of adaptive presentation techniques that it uses. For example, on top of visual adaptive navigation and link annotation techniques there is also speech output through an animated agent.

In Savidis' work (Savidis et al., 2005), a system called Unified User Interface is presented. That system is a framework that can adapt to users depending on their age and kind of incapability by creating polymorphic user interfaces. In their work they apply the Unified User Interface on a health application scenario, namely the MediBridge C-Care web-based EHR system. The polymorphic interfaces are produced through rules of the "tasks" of the user performances. On the other hand, iTVMObi instead of having specific tasks to categorize the users it creates dynamic groups of users based on their navigational mistakes while they interact with the system. Our system creates these groups through the use of a

user model that uses clustering techniques to dynamically produce groups of similar users.

In Muller's work (Muller et al., 2002), a multimodal navigational system is presented that learns from the cognitive load of users and then categorizes them into two different stereotypes: elderly and average aged adults. Despite the fact that their application has many modes of functioning, including speech, that can help elderly people, it does not focus on the topic of group categorization. This means that a person may be an elderly but s/he can have different problems from another elderly person. Our system uses the user model that is based on a clustering technique in order to categorize elderly people according to their own specific problems concerning the navigation, thus solving the problem of specific needs of an elder user against another elder one.

Another system that tries to help visually impaired and elder people is the VISTA Project by Carmichael and his colleagues (Carmichael et. al., 2003). This project for the digital tries to help visually impaired users find the best TV program for them through a conversation with an animated avatar that has the ability to synthesize speech. This avatar works as bridge between the electronic program guide of the digital TV and the user. On the other hand our approach called iTVMobi, is not an interstitial between a program of the digital TV and the user but provides the user all the information and tries to help people with special by learning from their usage of the system. In the VISTA project the avatar provides the impaired user recommendations of programs through the conversation and with the help of the of the electronic program guide. In iTVMobi the system learns from user behavior and by this provides recommendations and changes its interface according to users' mistakes. ITVMobi also has an animated avatar with speech output that uses to help the visually impaired. This

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

animated avatar can move anywhere in the screen and provide help whenever the system identifies a mistake made by the user.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

2.8 Animated Agents

In the past years the animated agent technology was mainly used by e-learning applications. The animated agent technology proved successful in helping students use the e-learning systems and enhance their learning experience (Baylor et al., 2003). The success of the animated agent technology lead to the creation of a new research domain and researchers called the animated agents pedagogical agents.

An interesting research in the field of e-learning has been conducted by Amy Baylor (Baylor et al., 2003). Her research involved the creation of three pedagogical agents with three different roles. The first had the role of the motivator, the second of the expert and the third the role of the mentor. The experimental results showed that all agents fulfilled their roles and supported students during learning. On the other hand, we used the animated agent technology on an entirely different field. In our field, sales support, we used the animated agent in order help users buy products they want more effectively. In both cases presented here the animated agent communicates with the user model of every user provided by the main mechanism of the application. Then the animated agent helps the user in a personalized way based on this user model.

Another approach on the field of pedagogical agent has been made by Veletsianos and his colleagues. (Veletsianos et al., 2005). In their research, they created two pedagogical agents, one male and one female, and employed them in two teaching courses. The findings of their research suggest that learners found male agents to be more outgoing and agreeable than female agents. In our applications, we used male animated agents in order to help users. Our animated agents were not only created to help and motivate users buy products, but to provide product recommendations based on users' models in a novel way.

An important work on the same field of pedagogical agents has also been made by Johnson and his colleagues (Johnson et al., 2003). In their work they created Adele, a pedagogical agent that assists students as they assess and diagnose medical and dental patients in clinical settings. The experimental results showed that Adele can be applied to future educational applications of interface agent technology. On the other, we used the animated agent technology to support customer buy products and use commerce applications. Our agents interacted with users and provided them with useful suggestions and information based on their user models.

Despite the fact, that the main research on animated agents has been done on the field of e-learning, there are other fields that incorporated animated agents. An example is the work done by Andre and his colleagues (Andre et al., 1996). In their work they created multipurpose animated presentation agent that was used in different applications in order to point gesture at points of interest in the user interface. On the other hand, our animated agent was not only used for pointing out suggested products but helped users in an adaptive way. Our agents interacted with every user in a personalized way according to their needs and interest extracted from their user model.

Another example of interesting work with animated agents is the work of Kirschning and Rueda (2005). They incorporated microsoft agents into html documents and used them to motivate children to interact with digital encyclopedias. Their results showed that children were helped by the use of the animated agent. In our work shown here, we incorporated an animated agent not only in an internet based application but also in an interactive TV application. We also used the animated agent not only for helping users interact

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

with application but also for suggesting products in a personalized way.

Lastly, on the field of e-commerce and product recommendation much work has been done by Kießling and his colleagues (Kießling et al., 2001). They created the COSIMA project, which a smart e-sales assistant. The COSIMA is java based server that controls the animated agent. The agent incorporates chatting with customers and smart sales advices. Despite the fact, that our technology was incorporated on the same field and had the same purpose of selling products more efficiently, we followed a different approach. Instead, of relying on sql based search engines of products in order to extract customer preferences, used in the COSIMA project, we connected our animated agent architectures with the user models created from the main reasoning mechanism of the application we incorporated the animated agent. In this way, our technology is not affected by the product being sold. We have also incorporated the animated agent in two different mediums, the web and the interactive TV, using very similar architectures. This shows that our technology is not affected by the medium that incorporates it.

2.9 Generic Architectures

Despite the fact that all the above systems provide users with recommendations, they are so domain and problem depended that lose the ability to be applied with easy in different fields or products. In this way every time a new application is built a new architecture must be constructed from scratch in order to address the specific application's problems.

As we mentioned above the sales assistant concept has two major aims, the first is product recommendations and the second

help support for customers. Researchers from the e-shopping field mainly focused on product recommendations and left the help support for customers at a second fate. On the other hand, generic architectures for user support have been greatly researched in an entirely different field, which is e-learning. Tutoring and supporting the users in e-learning has always been the main aims of the system of this particular field.

Significant work in the field of user support has been done Baldoni and his colleagues (Baldoni et al., 2004). In their work they describe an approach to the construction of adaptive tutoring systems, based on techniques from the research area of Reasoning about Actions and Change. Their approach led to the implementation of a prototype system, having a multi-agent architecture, whose kernel is a set of rational agents, programmed in the logic programming language DyLOG. In the prototype the reasoning capabilities of the agents are exploited both to dynamically build study plans and to verify the correctness of user-given study plans with respect to the competence that the user wants to acquire.

Another work in the field has been done by Dolog and Schäfer (Dolog and Schäfer, 2005). Their work presents the conceptualization and implementation of a framework which provides a common base for the exchange of learner profiles between several sources. The exchange representation of learner profiles is based on standards. An API is designed and implemented to create/export and manipulate such learner profiles. The API is implemented for two cases, as a Java API and as web services with synchronized model exchange between multiple sources. Application cases of the API are discussed shortly as well.

Significant work has also been done by Oviatt and his colleagues (Oviatt et al., 2008). In order to minimize users' cognitive load they developed an interface design based on implicit

engagement techniques so that users can remain focused on their tasks. In their research, data were collected with 12 pairs of students who solved complex math problems using a tutorial system that they engaged over 100 times per session entirely implicitly via speech amplitude or pen pressure cues. Results revealed that users spontaneously, reliably, and substantially adapted these forms of communicative energy to designate and repair an intended interlocutor in a computer-mediated group setting. The continuous using of these implicit engagement techniques, showed that students maintained their performance level at solving complex mathematics problems throughout a one-hour session. All the applications mentioned above provide successful user support to their users but the domain is entirely different from e-shopping. This means that despite the fact that user support is successful; the same successful principles cannot be incorporated to the field of product recommendations. On the other hand our user modelling server is built exactly for applications of product recommendation and in this way it can be incorporated very easily to these applications.

Another very important research on this field has been made by Brusilovsky and his colleagues (Brusilovsky et al., 2005). Their research is focused on user modeling and adaptation in distributed E-Learning systems. They describe CUMULATE, a generic student modeling server developed for a distributed E-Learning architecture, KnowledgeTree. They also introduce a specific, topic-based knowledge modelling approach which has been implemented as an inference agent in CUMULATE and used in QuizGuide, an adaptive system that helps students select the most relevant self-assessment quizzes. We also discuss their attempts to evaluate this multi-level student modeling. On the other hand our user modelling server applies to entirely different domain. Its role is not distribute student

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

models but create adaptive customer models in real time as the customer navigates through the specific application.

Important work in the field has also been conducted by Razmerita (Razmerita et al., 2003). They present a generic ontology-based user modeling architecture, (OntobUM), applied in the context of a Knowledge Management System (KMS). The main contribution of their research consists in identifying aspects of user modeling relevant to KMSs and integrating them in a generic framework based on ontologies. The user ontology is implemented using Semantic Web technology and it is structured on extended IMS LIP specifications.

A very significant work has also been done by De Bra (De Bra et al., 2003). They created AHA! ("Adaptive Hypermedia Architecture") to support an on-line course with some user guidance through conditional (extra) explanations and conditional link hiding. AHA! is a versatile adaptive hypermedia platform and can be used to add different adaptive "features" to applications such as on-line courses, museum sites, encyclopedia, etc.

One of the very few systems that try to combine recommendation and helping elder is the system constructed by Fink and Kobsa (1998). Their system is called AVANTI and its aim is to evaluate distributed information and provide recommendations through adaptive hypermedia information about a metropolitan area (e.g., about public services, transportation, buildings) for a variety of users (e.g., tourists, citizens, travel agency clerks, elderly people, blind persons, etc.). On the other hand, our system is an iTV commerce application that recommends products. Our system can be accessed from all users that possess an interactive TV.

A later work by Fink and Kobsa (Fink and Kobsa, 2006) also describes a user modelling server (UMS) that offers services to personalized systems with regard to the analysis of user actions, the representation of assumptions about the user, and the inference

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

of additional assumptions based on domain knowledge and characteristics of similar users. The system is open and compliant with major standards, allowing it to be easily accessed by clients that need personalization services and is based on the Lightweight Directory Access Protocol (LDAP). External clients such as user-adaptive applications can submit and retrieve information about users. The results showed that the performance of UMS meets the requirements of current small and medium websites already on very modest hardware platforms, and those of very large websites in an entry-level business server configuration.

Despite the fact that all research works mentioned above resulted in providing successful product recommendations to customers, they lack providing support to customers concerning the customer's interaction with the application. On the other hand, our user modelling aims in providing support while the customer interacts with the application. This support is provided intelligently and adaptively to every customer that uses the application that incorporates our user modelling server for adaptive help. Our server observes customer behaviour and through this observation extracts customer needs concerning the usage of the system. Our server responds in customers' mistakes with adaptive help actions according to their previous behaviour.

Another significant work towards convergence has been done ns Personis by Kay (Kay et al., 2002). Personis is a server for user models that every user can control his/her user model and modify it. Their user model server gives every application its monitoring a different view of the user model database. This structure of this database is based on objects. Our system has a user model server but goes even further and creates user models dynamically and also provides components that help systems create recommendations and effectively classify new customers without any prior knowledge of their tastes. Our framework modifies user model dynamically and

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

adaptively to every specific customer's behaviour in real time following the customer's moves as s/he navigates and interacts with the application that is incorporated into our framework.

2.10 The Rational Unified Process

Many adaptive applications are innovative and gave users results concerning adaptive help to users but none of them addressed software life-cycle issues. In adaptive applications like the ones mentioned above it's very difficult to apply the traditional software life cycle techniques. A very useful tool in software life-cycle is the Rational Unified Process (RUP). RUP is an object-oriented process that advocates multiple iterations of the software development process. It divides the development cycle in four consecutive phases: the inception, the elaboration, the construction, and the transition phase. Each phase is divided into four procedural steps, namely, requirements capture, analysis and design, implementation, and testing. The phases are sequential in time but the procedural steps are not. Moreover, one important advantage of RUP is the highly iterative nature of the development process. For the above reasons, RUP can be selected as the basis for presenting adaptive systems too.

A work towards implement RUP into systems has been done by Jaferian and his colleagues (Jaferian et al., 2005). In their paper they present extensions on Business modeling and Requirement discipline of RUP. They use RUP in order to present extensions that concern possible security threats and attacks. Their study showed that RUP helped presenting these security threats and that it should be extended for developing security-critical systems.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

A very novel work in incorporating the RUP technique in the life cycle of adaptive systems has been done by M. Virvou and K. Kabbassi. Their works show the development of a Graphical User Interface that incorporates intelligence and deals with files and folders in a similar way as the Windows 98 Explorer. The system is called IFM. In (Virvou and Kabassi 2000) they present an object - oriented approach in knowledge based software engineering of an intelligent GUI. In (Virvou and Kabassi 2003) they conduct experimental studies within the software engineering process for the IFM. In the (Kabassi and Virvou 2006) they extend their work and present a knowledge-based software life-cycle framework for the incorporation of multicriteria analysis in intelligent user interfaces. In all their works they use RUP as the tool of designing and developing their system. The usage of RUP proved to be very efficient on designing an adaptive system and enhanced the software life –cycle process of these systems.

In our work we presented an RUP based software life cycle on how to incorporate a clustering algorithm on a prototype system. The process has four major steps. First, designing and building the prototype system that does not include any clustering techniques. Second, evaluating the system and through this process obtaining data for the clustering algorithms. Third, comparing several clustering algorithms with the above data as input and choosing the most efficient algorithm. Fourth, incorporating the clustering algorithm into the system and building stereotypes based on this algorithm. In this research we present an RUP based software life cycle on how to incorporate a clustering algorithm on a prototype system based on the four major steps mentioned above. In our case we used an adaptive e-shop application as a test bed in order to apply these methods called Vision.Com and an interactive tv-shop called iTVMobi.

2.11 The drawbacks of the above systems and our approach

Product-sale applications are an easy way for every shop to promote its products to a very large customer pool. Nowadays, this fact is greatly reinforced by the extensive usage of broadband internet and recently through the interactive TV. Indeed, recent studies on customer behaviour show that computer adoption and internet connections in households are constantly growing (Goy et al., 2007). However, most of the e-commerce applications are developed in a generic way that does not take into account personalised needs and preferences of individual customers. Furthermore, the lack of generalized tools supporting the analysis of customers' browsing behaviour (e.g., shopping cart abandonment) does not enable vendors to collect feedback useful to redesign and optimize their Web sites (Hall 2001).

All of the above problems in e-commerce applications may be addressed by the incorporation of user modelling components that are based on intelligent techniques. The determination of the users' intention is generally complex because they do not always state their intentions explicitly (Ardissono and Sestero, 1995). A solution to this situation can be achieved by personalization techniques that can be extracted from the user modelling theory (Rich 1983). Indeed as the systems mentioned in the above sections, a lot of research effort has been put into the creation of intelligent e-commence systems that provide recommendations to their clients (users). (e.g. from PERCOM (Dias et al., 2008; Ntawanga et al., 2008; Cao and Li,2006; Alahakoon 2006; Koutsabasis and Darzentas, 2008).

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

But a lot of these systems face several drawbacks. The most of them are difficult to build and they require a lot of analysis that involves knowledge engineers and product sales analysts. Furthermore, a large part of the reasoning mechanisms that are needed for one medium may be the same for another medium. Moreover, many of them do not combine recommendation with user support, especially support for special user groups.

In order to solve these problems we present here an architecture that can be applied in any product recommending application in order to transform it into an intelligent application. This architecture was the result of a thorough research and investigation of the problem of intelligent e-commerce and recommendation systems in general. This framework is product and media independent. The main advantage is that it can be applied on any product recommending application without consideration of the products used such as personal computers, mobile phones or even cars. The architecture of the user modelling server includes user modelling techniques, intelligent user interfaces and clustering algorithms in order to produce recommendations. Furthermore, this architecture applies its performance into any group of users according to the group's specific needs, following similar principles to Savidis (Savidis et al., 2005).

This architecture was tested in three different cases relying on entirely different media. Also two of these cases sold entirely different products, thus applying to entirely different domains. These three cases were empirically evaluated through evaluation studies participating real users and experts. The results were very promising leading us into the conclusion that the generic e-commerce architecture can be very helpful in the construction of intelligent e-commerce systems. We also propose the basic steps on how to build the user modelling server and present two case studies in which we incorporated our server. The first one is an e-shop

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

applications and the second a mobile shop application. The incorporation of our user modelling server on both cases proves that this server can be easily applied on any product recommending system.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

CHAPTER 3
MEDIA AND MACHINE LEARNING FOR REMOTE SHOPPING APPLICATIONS

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

3. Media and Machine Learning for Remote Shopping

The use of an intelligent sales assistant in an electronic shop application can prove to be a great help for customers while they buying products. As Kobsa points out, the utilization of personalization and the underlying `one-to-one' marketing paradigm is of paramount importance for businesses in order to be successful in today's short-lived, complex, and highly competitive markets (Fink and Kobsa, 2000). Moreover, the one-to-one marketing follows the basic principles of knowing, remembering a customer and serving him as an individual. The intelligent sales assistant is a common utilization of this personalization and one-to-one marketing. The intelligent sales assistant can be used to help the shop improve selling rates of products and attract more customers.

Personalization has been reported to provide benefits throughout the customer life cycle, including drawing new visitors, turning visitors into buyers, increasing revenues, increasing advertising efficiency, and improving customer retention rate and brand loyalty (Hof et al., 1998; Bachem, 1999; Cooperstein et al., 1999; Hagen et al., 1999; Schafer et al., 1999). Personalization can be achieved through techniques, but one of the most common effective techniques in shopping applications is the use of machine learning algorithms. These algorithms can be used to extract user information, group them into similar groups according to criteria given and thus help systems make assumptions about users needs and inferences.

However, the concept of intelligent sales assistant in most e-shops is confined in recommending products that are similar to

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

customer's inferences. These products suggestions are very helpful, but product suggestions are only a part of the solution to the large problem of assisting a customer. Customers can also face problems concerning their interaction with the application. These problems can be caused by various reasons, such as the inability to comprehend interface elements, low computer familiarity, or even other problems, such as sight and hearing disabilities. Many of these problems lay in the medium used by the application. For example, mobile shopping applications have to compete with two kinds of problems, hardware and resources limitations and also users with unfamiliarity in the usage of mobile phones. These problems can spoil the customer's experience with the shopping application, confuse the customer and push him/her to make mistakes.

This chapter focuses on two major topics, media and machine learning. We present here how these two topics affect the shopping applications field, our research on the problems of using different media and machine learning algorithms and what were the conclusions that we extracted from the results of our research.

3.1 The Use and Role of Different Media in Shopping Applications

Nowadays shopping applications have expanded in large of products and domains. However, this expansion is not only bound in the domain but exceeds it and extends also into different media. Companies with shopping applications that work through internet want to be effective in mobile phones also. Furthermore, because TV is still a more widespread medium than computers, despite the fact that computers are now used by more people than ever before, companies want to extend their services in this field too. Thus

interactive TV can be a user-friendlier medium for novice users and prospective e-shoppers.

However, passing an application through different media is a difficult and the difficulty raises even more when this application is an intelligent one with adaptive and personalization functions. For this reason, many interactive TV applications are generic and addressed to all kinds of users without taking into account different needs and problems of different users or groups of users. This is also the case with users belonging to different age groups. For example teenagers have very different needs from senior users on internet applications.

A study showing such differences on online newsgroups has been conducted by Zaphiris and Sarwar (2006) that clearly shows many differences in tastes between teenagers and seniors. Teenagers may prefer mobile phones instead of TV sets. Another difficult age group is that of elderly people. Some systems that address problems with groups of people of different age are made by Muller and Wasinger (2002), Zhao and Tyugu (1998) and Savidis (Savidis et. al. 2005). However, all of the above researches do not take take into consideration the tastes of these users and base their user classification only on age criteria and not the actual problems of the elderly. Indeed, the elderly may have more problems than other users when interacting with computers and especially mobile phones. These users frequently have problems with impaired sight or hearing ability that may affect their interaction with these devices.

Moreover, people of older ages may be less familiar with computers than younger users or in many cases they may be computer illiterate. This fact is indicated by surveys concerning ageing population. For example, an Oxford Internet Survey, undertaken in Great Britain in 2002 and again in 2005, investigated Internet use by life stage and showed that of those who were

retired, 22 per cent used the internet in 2002 and 30 per cent in 2005. This suggests that numbers of computer literate people among the elderly are increasing rapidly, particularly those who are just below or within the younger old age grouping 65 to 74 (Capel et. al 2007). Yet, the surveys also show that the vast majority of the elderly people are still unfamiliar with the internet. This may be so, because the culture of computers and mobile phones was not as widespread in the past decades as it is now. To remedy all of the above problems there is a very imposing need for treating older people in a special way while designing computer programs so that these may use internet facilities.

In view of the above, a researcher must take into consideration that different media can affect the applications towards different groups of customers thus resulting in poor productivity. In order to find out the problems of different media we conducted a thorough research through three different media: internet, interactive TV and mobile. We created three different test bed applications, everyone belonging to a different medium, in order to extract customer needs through the use of these applications. The results provided us with a generalized picture of the problems a real user can throughout the interaction with different media. With this generalized picture of problems we were able to address these problems easier and in a more abstract way.

3.2 Machine Learning Algorithms in Remote Shopping Applications

The large quantity of information that exists in an electronic shop, as well as the lack of human salesmen to assist customers impose a need for adaptivity in remote shopping applications.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Adaptivity provides individualised assistance to users, which is dynamically generated. Adaptive responses to users are usually created using the technology of adaptive hypermedia (Brusilovsky 2001). To create such adaptive behaviour of the e-shop to individual users, the system needs information about them, so that it can generate hypotheses about what they need and how they can be assisted in the most appropriate way. This means that an adaptive shopping needs user modelling components, which monitor the user's actions and generate hypotheses about his/her preferences based on his/her behaviour.

According to Rich (1979) an effective technique researchers use to build models of applications' users very quickly is the evocation of stereotypes, or clusters of characteristics. Given the fact that grouping people can provide quick default assumptions about their preferences and needs (Rich 1979), the clustering of users' interests has drawn a lot of research energy for purposes of personalization of user interfaces.

One solution to the problem of grouping users' behaviour can be provided by machine learning algorithms that may group users dynamically based on their behaviour while they use a system on-line. The main advantage of such an approach is that the categorization of user behaviour can be conducted automatically. Machine learning algorithms help adaptive systems to categorize users according to their behaviour. More specifically adaptive commerce systems need not only to acquire information about users' interests in products but also to have the ability to group users with similar interests.

By grouping users together systems can understand more clearly the user's intention and generate more efficient hypotheses about what a user might need. In this part, machine learning algorithms undertake the role of grouping users in an efficient way, thus creating the bone structure of the user model. In this chapter,

we present how and we used five different machine learning algorithms in the structure of user models for both needs and inferences. These algorithms are an immune network-based clustering approach, an implementation of k-means algorithm, fuzzy c-means algorithms, a spectral clustering algorithm and the well known hierarchical clustering algorithm. This chapter also presents the methodologies on how we incorporated these algorithms into test bed applications of three different that we created for testing purposes. Examples from these test bed applications are also presented in order for the reader to understand how these algorithms affect the system's behaviour.

3.3 Overview of the process

There are a lot of web-based recommendation applications that have used clustering algorithms (e.g. Jin, X., et al., (SWP'04) 2004; Adil, C.S., et al., 2000; Menczer, et al., 2002; Jin (KDD'04) 2004; Ajith, 2004; Wang et al., 2004). All of these applications are recommendation systems, but are primarily concerned with the acquisition of user behaviour-related data. In contrast, our work has focused on incorporating a clustering algorithm into a remote shopping application. As a first step we chose to use the medium of internet as the field of our experiments. We chose to create a test bed application that sold movies through the internet called Vision.Com. Vision.Com incorporated an animated agent framework in order to help users in an adaptive way through the help of animated agent (figure 1).

We implement four different clustering algorithms in order test the effectiveness of these algorithms into our test bed applications and find out what problems a researcher may face in

the process of incorporating a machine learning algorithm into an e-shop application.

The algorithms incorporated were:

1. **Hierarchical Clustering (Day and Edelsbrunner 1984),**
2. **Fuzzy C-means clustering (Bezdek et al. 1984),**
3. **Spectral clustering (Jordan and Weiss 2002),**
4. **AIN clustering (Cayzer and Aickelin 2002), (Morrison and Aickelin 2002)**

The last algorithm was based on the construction of an Artificial Immune System (AIS), in order to personalize the developed e-shop system through grouping similar users' behaviour. This means that we have used a very different algorithm that has not been used by these systems. Similarly to our approach, Cayzer and Aickelin (2002) and Morrison and Aickelin (2002) have also utilized AIS in order to tackle the task of film and web site recommendation respectively by identifying a neighbourhood of user preferences. Although, their systems have used a similar algorithm to the one presented in this paper, the algorithm that we have used is significantly more sophisticated.

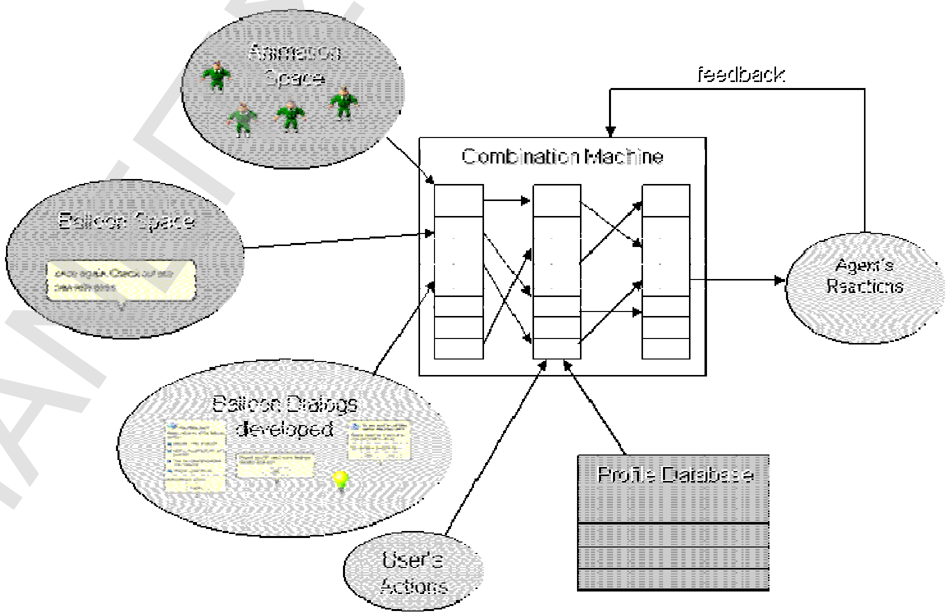


Figure 1 Animated Agent Framework in Vision.Com

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

In particular, we have based our research on the construction of an Artificial Immune Network (AIN) which incorporates a mutation process that applies to the customer profile feature vectors. More specifically, our work has focused on how a significant amount of redundancy within the customer profile dataset can be revealed and reduced, how many clusters intrinsic to the customer dataset can be identified and what the spatial structure of the data within the identified clusters is.

Specifically, we develop an AIN for the problem of clustering a set of unlabelled multidimensional customer profile feature vectors generated in an electronic video store application in which customer preferences are quantified and groups of customer profiles are maintained. We have examined thoroughly the effectiveness of this algorithm by comparing it with three other clustering algorithms, namely agglomerative hierarchical clustering, fuzzy c-means clustering and spectral clustering. For this purpose, we have collected data through a video e-shop application and fed them into each of the four clustering algorithms and compare the corresponding results.

As a second step we chose an entirely different medium to test our methodologies that we discovered from the incorporation of the previous four algorithms into the e-shop application. Interactive TV was used as the appropriate medium for our testing purposes. Our purpose was to incorporate a clustering algorithm into a novel application for interactive TV that could help the elderly to use an e-commerce application. As a test bed for our research we have created an iTV-shop that sells mobile phones. In an e-shopping application the elderly people may have problems both with the interaction with the e-shopping application itself as well as with the products to be sold to them because for both they need to have familiarity with the recent technological advances (Internet, mobile phones etc.)

The personalized application that has been developed for these purposes is called iTVMobi. ITVMobi architecture diagram is presented in figure 2. In iTVMobi, the elderly are able to "go shopping" in electronic stores at the comfort of their homes and be assisted for as long as they wish by virtual sales agents through an interactive TV. The application provides personalized help and recommendation that assists these people in searching/navigating through the interactive TV in a guided way that prevents them from being disoriented and helps them find easily what they seek. Moreover, there is an on-line community system that encourages users of similar tastes to communicate with each other and provide help to each other.

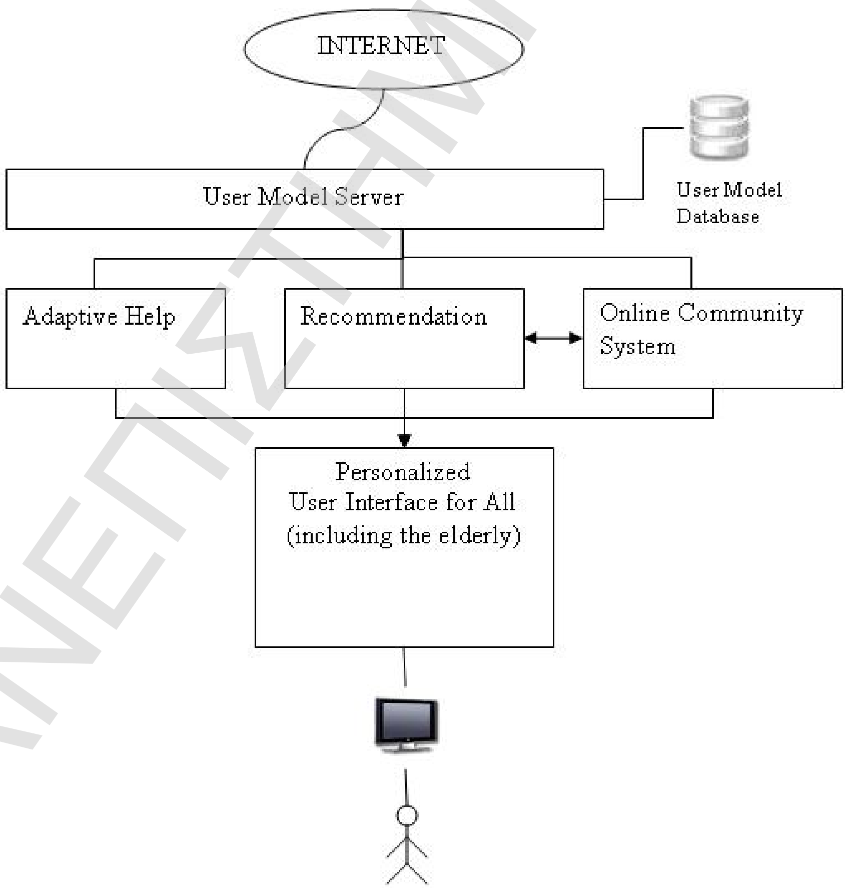


Figure 2 ITVMobi architecture

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

In this step we developed a methodology of incorporating only one algorithm for our personalization purposes instead of four as in the previous medium used. However, this time the incorporation of the clustering algorithm used would allow to be used for many different purposes. In the interactive TV test bed application the clustering algorithm was used in three different ways. Firstly for developing groups of users with similar products tastes, secondly for creating groups of users with similar behaviours concerning mistakes through their interaction with the applications and thirdly creating groups of users with similar opinions in topics through the use of an on-line community system developed for and embedded into the interactive TV shopping application.

We chose to create an implementation of the k-means clustering algorithm in order to incorporate it to our test bed application. The exact same algorithm was used for these three entirely different operations. This procedure was created in order to prove the generality of our methodology and give an example that the purpose and kind of recommendation given through the application does not affect the algorithm's procedures used in the way that we propose. The k-means algorithm was chosen for two reasons. First is a simple algorithm and requires minimum resources and secondly runs fast thus providing quick results. The real-time challenges of interactive TV applications made this algorithm suitable for our testing purposes.

As a third and final step we chose the mobile shopping as medium for our research purposes. We chose this field due to hardware and resources limitations and to test how the incorporation of a clustering algorithm can affect a mobile shopping application. We, again, used three clustering algorithms in order to be incorporated to our mobile shop application. The purpose of the algorithms was to create groups of users with similar mistakes while they interacted with the application. For testing purposes we

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

created a mobile shopping application that sold movies. We tested these three algorithms in the use of the mobile shopping applications and compared their results in order to identify their effectiveness.

This three step process led to various conclusions about the usage of different media and machine learning in commerce related applications. Furthermore, it leads us in the creation of a general methodology based on the rational unified process that can help researchers incorporate such algorithms into their applications in order to achieve adaptivity.

3.4 Incorporating a Clustering Algorithm into an E-shopping application

Vision.Com (Figure 3) is an adaptive e-commerce video store that learns from customers preferences that was built for our testing purposes. Its aim is to provide help to customers choosing the best movie for them. For every user the system creates a different record at the database. In Vision.Com every customer can visit a large amount of movies by navigating through four movie categories: social, action, thriller and comedy movies. Every customer has a personal shopping cart. If a customer intends to buy a movie she/he must simply move the movie into her/his cart by pressing the specific button. S/He also has the ability to remove one or more movies from his/her cart by choosing to delete them. After concluding which movies to buy a customer can easily purchase them by pressing the button buy.

All navigational moves of a customer are recorded by the system in the statistics database. In this way the Vision.Com saves statistics considering the visits in the different categories of movies

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

and movies individually. The same type of statistics was saved for every customer and every movie that was moved to the buyers' cart. The same task is conducted for the movies that are eventually bought by every customer. All of these statistical results are scaled to the unit interval [0,1].

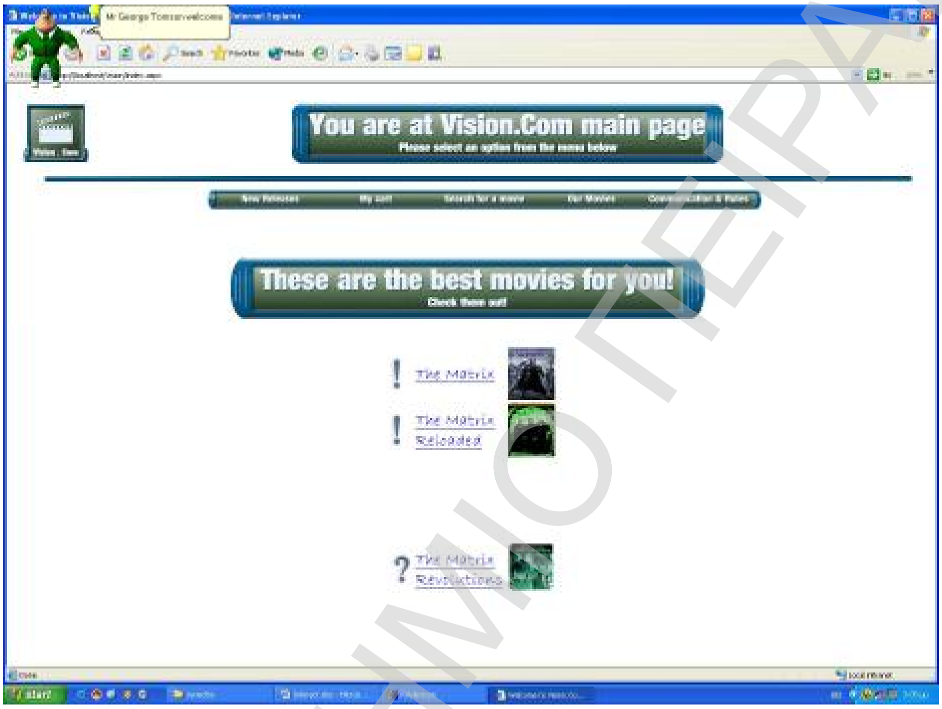


Figure 3. Screenshot of Vision.Com User Interface.

The four clustering algorithms incorporated in turn into Vision.Com reasoning system are: Hierarchical Clustering, Fuzzy C-means clustering, Spectral clustering and AIN clustering. In particular, Vision.Com interprets users' actions in a way that results in the calculation of users interests in individual movies and movie categories. Each user's action contributes to the individual user model by showing degrees of interest into one or another movie category or individual movie. For example, the visit of a user into a movie shows interest of this user to the particular movie and its category. If the user puts this movie into the shopping cart this shows more interest in the particular movie and its category. If a user buys this movie then this shows even more interest whereas if

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

the user takes it out of the shopping cart before payment then there is not any increase in the interest counter.

Apart from movie categories that are already presented, other movies features that are taken into consideration are the following: price range, leading actor and director. The price of every movie belongs to one of the five price ranges in euro: 20 to 25, 26 to 30, 31 to 35, 36 to 40 and over 41. As a consequence, every customer's interest degree in one of the above features is recorded as a percentage of his/her visits in movie-pages. For example, interest of the customer at a particular movie category is calculated by the equation (1). Every result can also be considered as a probability of one customer intention on buying a movie.

$$
\begin{aligned}
Interest\_in\_movie\_category &= \frac{Visitis\_in\_Specific\_Category}{Visits\_in\_All\_Categories} + \\
&+ \frac{Movies\_moved\_to\_cart\_from\_this\_category}{All\_movies\_moved\_to\_cart} + \\
&+ \frac{Bought\_Movies\_from\_this\_category}{All\_bought\_movies}
\end{aligned} \tag{1}
$$

Vision.Com was used by 150 users that bought movies using this particular system. The system collected data about the user's behaviour implicitly through the use of Observing Behaviour Agent component. The data collected, consisted of three parts. Every part is similar to the others. The first one contains statistical data of the visits that every user made to specific movies. The second part contains data of the cart moves (i.e. which movies the user moved into his/her cart). The last part consists of statistical data concerning the preferences on the movies bought by every user.

Every record in every part is a vector of the same 80 features that were extracted of the movies characteristics and represents the references of one user. The 80 features of these vectors are the

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

movie features we described above. Every 80 featured vector is consisted of the four movie categories, the five price ranges, all the leading actors and all the directors. The value of each feature is the percentage of interest of every individual customer in this particular feature (equation (1)).

3.4.1 AIS-based Customer Data Clustering Algorithms

AIS-based clustering relies on a computational imitation of the biological process of self/non-self discrimination, that is the capability of the adaptive biological immune system to classify a cell as "self" or "non-self" cell. Any cell or even individual molecule recognized and classified by the self/non-self discrimination process is called an antigen. A non-self antigen is called a pathogen and, when identified, an immune response (specific to that kind of antigen) is elicited by the adaptive immune system in the form of antibody secretion. The essence of the antigen recognition process is the affinity (molecular complementarity level) between the antigen and antibody molecules. The strength of the antigen-antibody interaction is measured by the complementarity of their match and, thus, pathogens are not fully recognized, which makes the adaptive immune system tolerant to molecular noise. In the present e-shop application, the antigenic population consists of the initial set of customer profile feature vectors. The produced set of memory antibodies provide an alternative, more compact way of representing the original dataset while conserving their original spatial distribution.

Learning in the immune system is established by the clonal selection principle (De Castro and Timmis, 2002) which suggests that only those antibodies exhibiting the highest level of affinity with a given antigen be selected to proliferate and grow in concentration.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Moreover, the selected antibodies also suffer a somatic hypermutation process (De Castro and Timmis, 2002), that is a genetic modification of their molecular receptors which allows them learn to recognize then given antigen more efficiently. This hyper-mutation process is termed affinity maturation (De Castro and Timmis, 2002). To develop an AIN for data clustering, we generate a minimal set of representative points that capture the properties of the original dataset and can be interpreted as the centers of the initial feature vector distribution. In AIS terminology, this representative point set in a multidimensional feature space constitute a set of memory antibodies that recognize, in the sense of Euclidean distance proximity, the antigenic population that corresponds to the original dataset.

The AIN that we developed for clustering the customer profile feature vectors in the present application was an edge-weighted graph composed of a set of memory antibodies (i.e., nodes) and edges (i.e., sets of node pairs with an assigned weight or connection strength to reflect the affinity of their match). To quantify immune recognition, we consider all immune events as taking place in a shape-space S, constituting a multi-dimensional metric space in which each axis stands for a physico-chemical measure characterizing molecular shape (Cayzer and Aickelin 2002).

Specifically, we utilized a real-valued shape-space in which each element of the AIN is represented by a real-valued vector of 80 elements, thus, S = R^{80}. The affinity/complementarity level of the interaction between two elements of the AIN was computed on the basis of the Euclidean distance between the corresponding vectors in R^{80}. The antigenic pattern set to be recognized by the AIN is composed of the set of 150 80-dimensional feature vectors. The memory antibodies produced by the AIN can be considered as an

alternative compact representation of the original customer profile feature vectors set.

The AIN learning algorithm was as follows:

1. Initialization: Create a random initial population of network antibodies.

2. Antigenic presentation: For each antigenic pattern do:

(a) Clonal selection and expansion: For each network element, determine its affinity with the antigen presented. Select a number of high affinity elements and reproduce (clone) them proportionally to their affinity.

(b) Affinity maturation: Mutate each clone inversely proportionally to affinity. Reselect a number of highest affinity clones and place them into a clonal memory set.

(c) Metadynamics: Eliminate all memory clones whose affinity with the current antigen is lower than a predefined threshold.

(d) Clonal interactions: Determine the network interactions (affinity) among all the elements of the clonal memory set.

(e) Clonal suppression: Eliminate those memory clones whose mutual affinity is lower than a predefined threshold.

***(f) Network construction: Incorporate the remaining clones of the clonal memory in all network antibodies*.**

3. Network interactions: Determine the similarities among all network antibody pairs.

4. Network suppression: Eliminate all network antibodies with affinity below a prespecified threshold.

5. Diversity: Introduce a number of new randomly generated antibodies into the network.

6. Cycle: Repeat steps 2 to 5 for a prespecified number of iterations.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

3.4.2 Comparison of Customer Data Clustering Algorithms and Conclusions

We tested and compared three widely used clustering techniques, namely a) agglomerative hierarchical clustering, b) fuzzy c-means clustering and c) spectral clustering, against the less well-known AIS-based clustering. Specifically, we applied these four clustering methodologies on the 150 customer profile feature vectors collected. In Figure 4, we show and compare the dendrograms corresponding to the feature vectors produced by hierarchical (top left), spectral (top right), and AIN-based (center left) clustering, respectively. We observe that spectral clustering does not provide a clearer revelation of the intrinsic similarities in the dataset over hierarchical clustering.

On the other hand, the leaves in the AIN-based dendrogram are significantly fewer than the leaves in either the hierarchical or the spectral dendrograms in Figure 4, which stems from the fact that the former corresponds to clustering only 22 representative points in the 80-dimensional feature space, while the latter two correspond to clustering the complete set of 150 data points. Thus, the AIN-based dendrogram demonstrates the intrinsic data point clusters significantly more clearly and compactly than the corresponding hierarchical and spectral dendrograms.

Also, in Figure 4, we show the partitioning of the complete dataset into six clusters by the spectral (center right), fuzzy c-means (bottom left), and AIN-based (bottom right) clustering algorithms, respectively. We observe that spectral clustering does not result in cluster homogeneity, while fuzzy c-means clustering results in higher cluster homogeneity, but in only four clusters rather than six required. Specifically, we observed that fuzzy c-means clustering assigned the same degree of cluster membership

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

to all the data points, which implies that certain intrinsic data dissimilarities were not captured by the fuzzy c-means clustering algorithm and this makes the clustering result less useful. On the contrary, AIN-based clustering returned significantly higher cluster homogeneity. Moreover, the degree of intra-cluster consistency is clearly significantly higher in the AIN-based rather than the hierarchical and spectral clusters, which is of course a direct consequence of a data redundancy reduction achieved by the AIN.

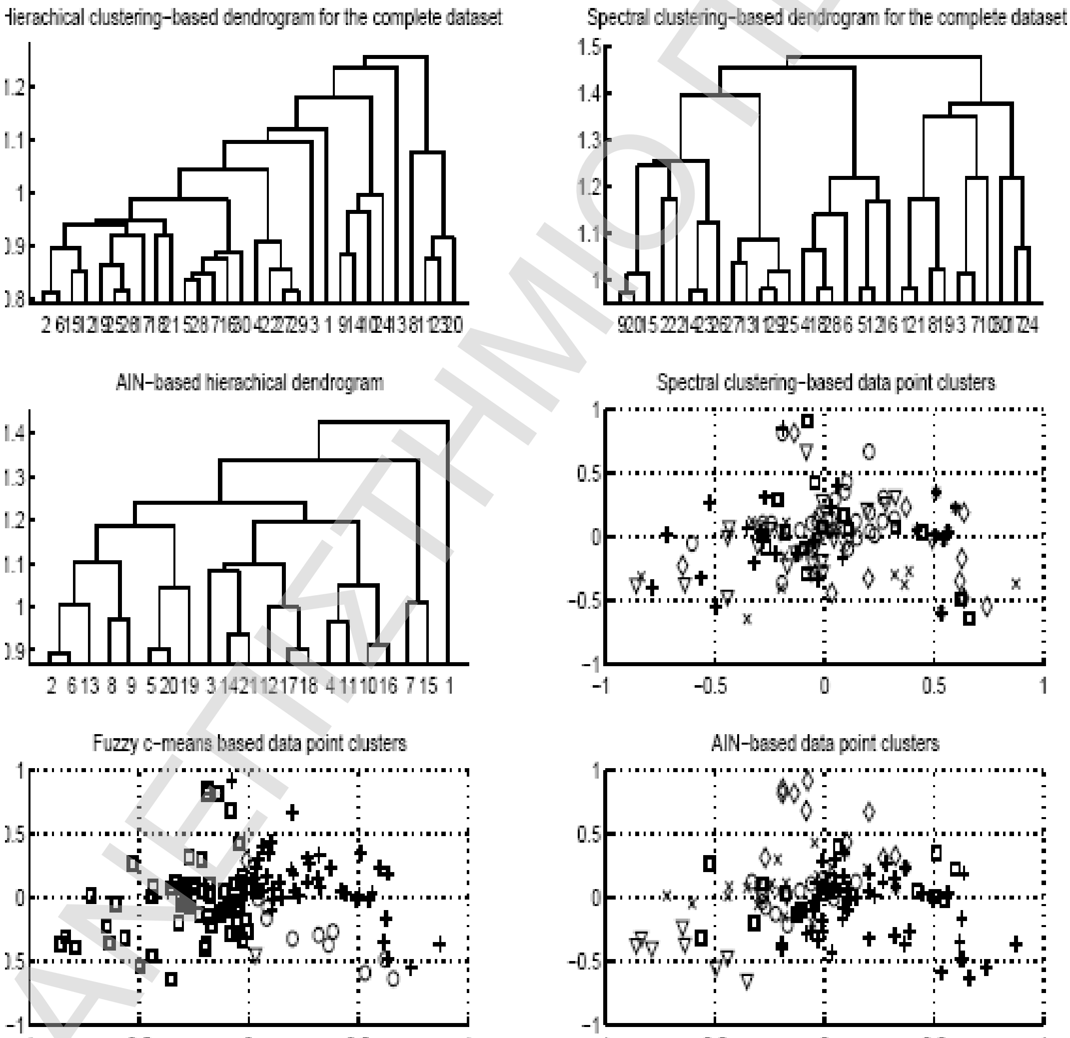


Figure 4. Dendrograms and clusters of customer profile feature vectors returned by four clustering methodologies.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

In Figure 5, we show a plot of the spatial distribution of 22 representative points obtained by reducing the original 80-dimensional antibodies to their 2dimensional projections with a principal component analysis algorithm. Clearly, the set of representative antibodies in Figure 5 maintain the spatial structure of the complete dataset in Figure 4 (center right, bottom), but, at the same time, form a minimum representation of 150 feature vectors with only 22 antibodies. This indicates significant data compression, combined with clear revelation and visualization of the intrinsic data classes.

Figures 4 and 5, lead to the conclusion that Vision.Com customers exhibit certain patterns of behaviour when shopping and tend to group themselves into six clusters. The 22 antibodies that arose via the AIN-based clustering algorithm correspond to customer behaviour representatives and, thus, can be seen as important customer profiles, which eventually correspond to stereotypes in user models. This process is promising because it can provide recommendations based on the users' interests and the characteristics of a movie irrespective of whether this movie has ever been selected by a user before. Thus, the recommendation system can recommend new movies or movies newly acquired by the e-shop as efficiently as previously stored movies. This approach forms the basis of work which is currently in progress and will be reported on a future occasion.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

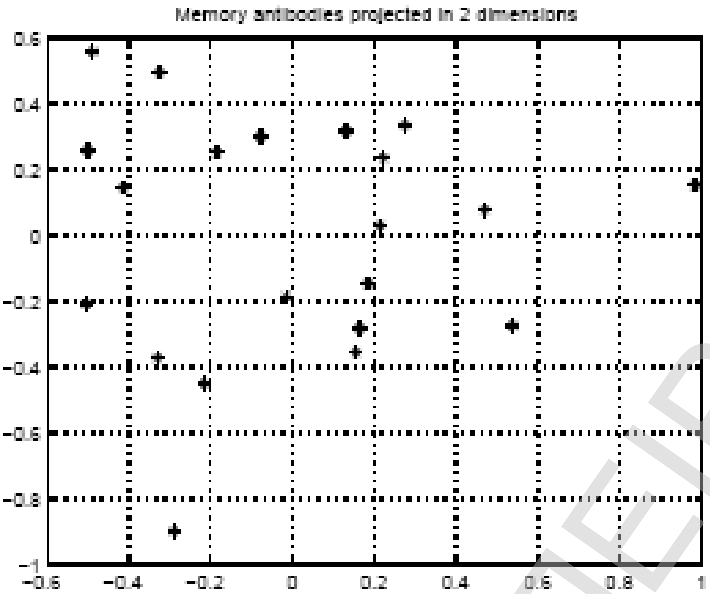


Figure 5. Plot of the spatial distribution of the projection of the 22 AIN-produced memory antibodies onto two dimensions.

These antibodies are significant vectors of the same set of features as with the mean vectors. The antibodies present the propagation in the 150 mean vectors that were used as an input to the algorithm. Because every mean vector corresponds to a user, we can also see every antibody as a user that represents preferences of a group of users from the original set. In order to see clearer results we clustered these 22 antibodies into 2, 3, 4, 5 and 6 classes accordingly. By clustering these antibodies into the above classes we observed more clearly the differences between the different preferences in the same feature. The above classes can be seen as stereotypes that can be used for the initialization of a user.

3.4.3 Constructing double stereotypes based on the immune system

After constructing the classes, we used them to build stereotypes. The double classification (users' interests – movies) was performed in a hierarchical way that resulted in several levels

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

of user stereotypes: At first, there was a coarse classification of two stereotypes which was next refined several times to produce a final classification of six stereotypes as can be seen in Figure 6. Figure 6 illustrates a brief diagram of the hierarchical classification of stereotypes. These stereotypes that have been constructed in the first phase are then used dynamically by the e-commerce application to infer users' interests in movies based on a small set of observed users' actions. In fact, for a new system user, modeling is performed based on the first classification of users. Then, incrementally and while the user interacts with the system, inferences about his/her preferences are drawn from more refined user stereotypes of subsequent levels.

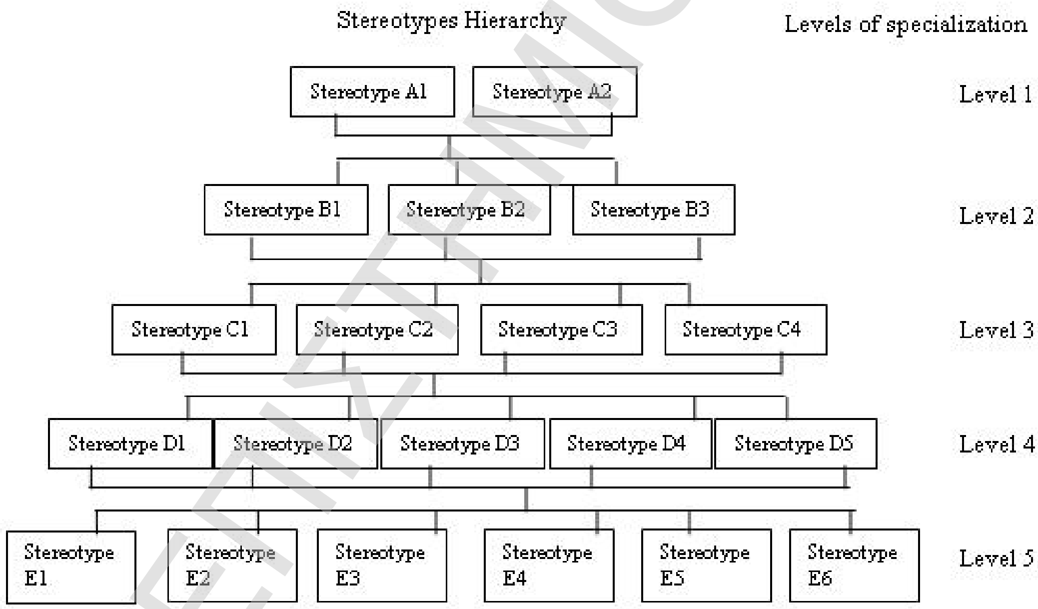


Figure 6 Hierarchical Diagram of Double Stereotypes

More specifically, at first, on level 1, we created two general stereotypes (Stereotype A1 and Stereotype A2) based on the two general classes. The main differences between these two classes can be observed mainly in the interest in movie categories. In the particular classification, the main difference is in the thriller category. Stereotype A1 shows interests divided into the three categories (action, comedy and social). On the other hand, Stereotype A2 shows an assembly of users' interest into the thriller movie category. In this way the two stereotypes based on these classes focus their differences of preferences in a category of a movie. As we go further down in the top-down presentation the differences in the stereotypes extend to more features of interest. More specifically, in the stereotypes of the second level of the immune system, differences can be seen not only on the movie categories but on the interest in medium and high priced movies feature as well.

As we proceed to higher levels, stereotypes become more complex and the differences between them extend to more features of interest. At level three of specialization, differences in stereotypes are extended to leading actors as well. At level four, stereotype differences include directors such as Steven Spielberg, Brian de Palma and others. These directors are the ones that attracted the highest interest of users in the previous levels. The leaves of this hierarchical stereotype tree, at level 5, consist of six stereotypes. The differences between them extend to all interest features.

In view of the above results, we have built the hierarchy of stereotypes not only for customers but for movies as well. Whether a customer or a movie belongs or not to a stereotype is measured by the Euclidean distance. The measurement process is opposite to the top-down presentation that we showed just above. If a customer's distance is very far from the leaves then we proceed to the next level above. If in this level the distance is also far we move up to the next. This process is continued until we reach the top level or a threshold of distance that is considered close enough. When this distance is reached the system follows the same process for movies. The movies that are close enough to the customer's stereotype just measured are recommended to the customer through an adaptive hypermedia (Brusilovsky, 2001) procedure.

3.4.4 Incremental Initialization of user model based on double stereotypes

Stereotypes have been widely considered as a very effective technique for initializing user models of new users. The answers to the questions concerning which stereotype a user belongs to and which recommendation is best for a particular user, are based on vector subtraction. The decision process is quite complex but here we will present a simpler one because our aim is not to fully explain mathematically this process but to present the system recommendation system. When a new user becomes a member to the system, the e-commerce application creates a profile, sets all interest values into zero (the system assumes that initially the user has no interest in any movie) and starts to monitor his/her actions.

After the interaction of the new user with the system (visiting few movies) the system classifies the new user into a stereotype of the first level of specialization. The first level generally considers users' interest concerning the four movie categories. For example, if the new user shows a preference towards thriller movies them s/he classified to Stereotype A2 of the first level, as the main difference between the two stereotypes lies in this movie category. Otherwise if the new user chooses movies from one or more of the other three categories the system classifies the new user into the group that belongs to Stereotype A1 of the first level. In the same way, the system chooses from movie stereotypes the movies that belong to the corresponding movie stereotype. If the user belongs to the first stereotype the system chooses to propose movies from any of the three categories except thrillers in a presentation percentage similar to the interest in every category. The movies that are most close to

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

this movie stereotype are those recommended by the system up to this point of interaction.

As the user continues with moving movies into his/her cart and buying some of them the system moves to the next level of classification that extends stereotypical information to the price features. This is so because in this level stereotypes differ greatly in the price ranges along with movie categories. For example, if the new user selects movies with medium prices the system chooses to classify him/her to the stereotype that has the greatest interest in this price range concerning always the interest in movie categories. On the other hand, if the new user chooses many movies with high price then the system chooses to classify him/her in a different stereotype always taking in mind the movie categories interest.

In this way if two users have similar interests in movie categories and have little differences in price ranges the system will propose similar movie titles to them. Otherwise, if the difference in price ranges is high then the system will most certainly change the most recommended movies. Level four and five of specialization extend the features of interests into interests in leading actors and directors accordingly. In this way, as users show with their actions which actor or director they prefer the system easily classifies them into the respective stereotypes and selects the right movie stereotypes in order to make recommendations. Again if their differences in interest in the actor and directors are low the system chooses to group users in same stereotypes thus emulating the grouping process into the previous level of specialization. On the other hand if these differences are high the users are grouped into different stereotypes of these levels.

The initialization process is conducted until the user reaches level six of specialization. This level represents the leaves in the hierarchical tree of stereotypes and extends the differences in all the features of interests. When a user reaches this amount of

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

interaction with the system s/he has given a lot of information to the system and s/he has been classified in a way that the system knows almost every interest about the user. When the system reaches this level of specialization, not only recommends movies in the appropriate web page but with the use of adaptive hypermedia predicts the next moves of the user and changes the interface dynamically. The specialization in these level is very high and even the smallest difference in user's interest can classify him/her to a different stereotype of this specific level.

3.5 Incorporating K-means into Interactive TV-shopping

As we mentioned in previous sections the incorporation of k-means has three different purposes: groups of users with similar tastes, groups of users with similar mistakes and groups of users with similar opinions. In this paragraph we will give a brief of the test bed application used for the incorporation of the algorithm and next we will present the three different types of user models that k-means was used to create.

3.5.1 The Experimental Personalized Interactive Tv System – Itvmobi

iTVMobi (figure 7) is an adaptive mobile shop created for the interactive television that learns from customer preferences. ITVMobi was built on Microsoft TV (MSTV) technology. Its aim is to provide help to customers with hearing and sight problems by suggesting the best mobile phone for them. The recommender system that makes suggestions concerning mobile phones and accessories is based on user modeling. The system can learn about

the users' preferences and provide more helpful responses. User models are created using clustering algorithms. These techniques will be explained more thoroughly in the next section.

For every user, iTVMobi creates a different record at the user model database. In iTVMobi every customer can visit several mobile phones. For the purposes of our research we have implemented the system for five popular mobile brands. Every customer has her/his own personal shopping cart. If customers intend to buy a phone they must simply move the phone into their cart by pressing the specific button or they can press the buy button at their remote control at the time that the specific product is shown on their TV screen. They also have the ability to remove one or more phones from their cart by choosing to delete them. After deciding which phones to buy, a customer can easily purchase them by pressing the button "buy" at their shopping cart.

Figure 7 iTVMobi user interface environment

In particular, ITVMobi interprets users' actions in a way that

results into three different functions. The first is the calculation of users' interests in individual phones and production companies, the second is the interpretation of users' actions concerning possible navigation mistakes and the third dominant opinions on topics concerning products discussed in the online-community system. Each user's action contributes to the individual user profile by showing degrees of interest into one or another company or individual phone or by showing likelihood on a specific mistake.

For example, the visit of a user into a phone-icon shows interest of this user to the particular phone and its brand. If the user puts this phone into the shopping cart this shows more interest in the particular phone and its brand. If a user buys this phone then this shows even more interest whereas if the user takes it out of the shopping cart before payment then there is not any increase in the interest counter. On the other hand if a user follows a different pattern of navigational moves, like repeated clicks on the same brand-name, the system interprets this action but as a confusion navigational mistake rather than as a high degree of interest in this brand. Thus, in this case, the system decides to intervene with an adaptive help action. Suggested phones are presented in the suggestions window through the help of adaptive hypermedia (Brusilovsky 2001).

3.5.2 The recommender

The recommender system is based on user modeling that is constructed using the k-means algorithm and involves the participation of the Adaptive User Interface components of PERCOM. The recommender function is based on the principle that many customers tend to have similar interests. Every customer's interest in one of the phone features described above is recorded as a

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

percentage of his/her visits in the respective phone-pages. An interest degree of the customer at a particular phone brand is calculated by the equation 2. The interest of a customer at phones' different features (Size, Display, Memory, Connectivity, Features, and Battery) is calculated by equations 3 to 8.

$$InterestInCompany_1 = \frac{VisitsInPhonesBelongToCompany}{VisitsInAllPhones} \quad \textbf{(2)}$$

$$InterestInCompany_2 = \frac{PhonesPlacedInBasketBelongToCompany}{AllPhonesPlacedBasket} \quad \textbf{(3)}$$

$$InterestInCompany_3 = \frac{PhonesBoughtBelongToCompany}{AllBoughtPhones} \quad \textbf{(4)}$$

$$InterestInCompany = W_{C1} * InterestInCompany_1 + W_{C2} * InterestInCompany_2 + W_{C3} * InterestInCompany_3 \quad \textbf{(5)}$$

$$InterestInPhoneSize_1 = \frac{VisitsInPhonesWithSpecificSize}{VisitsInAllPhones} \quad \textbf{(6)}$$

$$InterestInPhoneSize_2 = \frac{PhonesPlacedInBasketWithSpecificSize}{AllPhonesPlacedBasket} \quad \textbf{(7)}$$

$$InterestInPhoneSize_3 = \frac{PhonesBoughtWithSpecificSize}{AllBoughtPhones} \quad \textbf{(8)}$$

$$InterestInPhoneSize = W_{S1} * InterestInPhoneSize_1 + W_{S2} * InterestInPhoneSize_2 + W_{S3} * InterestPhoneSize_3 \quad \textbf{(9)}$$

$$InterestInPhoneDisplay_1 = \frac{VisitsInPhonesWithSpecificDisplay}{Visitsin\,AllPhones} \quad \textbf{(10)}$$

$$InterestInPhoneDisplay_2 = \frac{PhonesPlacedInBasketWithSpecificDisplay}{AllPhonesPlacedBasket} \quad \textbf{(11)}$$

$$InterestInPhoneDisplay_3 = \frac{PhonesBoughtWithSpecificDisplay}{AllBoughtPhones} \quad \textbf{(12)}$$

$$InterestInPhoneDisplay = W_{D1} * InterestInPhoneDisplay_1 + W_{D2} * InterestInPhoneDisplay_2 + W_{D3} * InterestPhoneDisplay_3 \quad \textbf{(13)}$$

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

$$InterestInPhoneMemory = \frac{VisitsInPhonesWithSpecificMemory}{VisitsInAllPhones} \quad \textbf{(14)}$$

$$InterestInPhoneMemory_2 = \frac{PhonesPlacedInBasketWithSpecificMemory}{AllPhonesPlacedBasket} \quad \textbf{(15)}$$

$$InterestInPhoneMemory_3 = \frac{PhonesBoughtWithSpecificMemory}{AllBoughtPhones} \quad \textbf{(16)}$$

$$InterestInPhoneMemory = W_{M1} * InterestInPhoneMemory_1 + W_{M2} * InterestInPhoneMemory_2 + W_{M3} * InterestPhoneMemory_3 \quad \textbf{(17)}$$

$$InterestInPhoneConnectivity = \frac{VisitsInPhonesWithSpecificConnectivity}{VisitsInAllPhones} \quad \textbf{(18)}$$

$$InterestInPhoneConnectivity_2 = \frac{PhonesPlacedInBasketWithSpecificConnectivity}{AllPhonesPlacedBasket} \quad \textbf{(19)}$$

$$InterestInPhoneConnectivity_3 = \frac{PhonesBoughtWithSpecificConnectivity}{AllBoughtPhones} \quad \textbf{(20)}$$

$$InterestInPhoneConnectivity = W_{PC1} * InterestInPhonemConnectivity_1 + W_{PC2} * InterestInPhoneConnectivity_2 + W_{PC3} * InterestPhoneConnectivity_3 \quad \textbf{(21)}$$

$$InterestInPhoneFeatures = \frac{VisitsInPhonesWithSpecificFeatures}{VisitsInAllPhones} \quad \textbf{(22)}$$

$$InterestInPhoneFeatures_2 = \frac{PhonesPlacedInBasketWithSpecificFeatures}{AllPhonesPlacedBasket} \quad \textbf{(23)}$$

$$InterestInPhoneFeatures_3 = \frac{PhonesBoughtWithSpecificFeatures}{AllBoughtPhones} \quad \textbf{(24)}$$

$$InterestInPhoneFeatures = W_{F1} * InterestInPhonemFeatures_1 + W_{F2} * InterestInPhoneFeatures_2 + W_{F3} * InterestPhoneFeatures_3 \quad \textbf{(25)}$$

$$InterestInPhoneBattery = \frac{VisitsInPhonesWithSpecificBattery}{VisitsInAllPhones} \quad \textbf{(26)}$$

$$InterestInPhoneBattery_2 = \frac{PhonesPlacedInBasketWithSpecificBattery}{AllPhonesPlacedBasket} \quad \textbf{(27)}$$

$$InterestInPhoneBattery_3 = \frac{PhonesBoughtWithSpecificBattery}{AllBoughtPhones} \quad \textbf{(28)}$$

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

$$InterestInPhoneBattery = W_{B1} * InterestInPhoneBattery_1 + W_{B2} * InterestInPhoneBattery_2 + W_{B3} * InterestPhoneBattery_3 \quad (29)$$

As the previous equations show the degree of interest in a phone feature (for example: the company that the phone belongs to in the case above) is measured in three ways. Then in order for the full degree of interest to be acquired the system calculates a weighted sum of the three different degrees of interest, the degree of interest that corresponds to the visits of the user in the phone pages, the degree of interest that corresponds to the phones placed by the user to his/her basket and the interest that corresponds to the phones bought by the user. The weights used by the system are different for every phone feature and were extracted through the experience from the evaluation process of the system. For example a user chooses to visit a phone through the phone icon and does not have the ability to know the display abilities of this phone before opening the specific phone page. As such, the opening of a specific phone page through its icon may not mean that the user is necessarily interested in the phone but that s/he is just browsing several phones.

On the other hand every company's name is displayed from the very beginning to every user and in this way the user is aware for the company that he/she selects to visit thus making his/her selection more accountable. As a result the W_{C1} **weight used to measure the Interest in Company from the user visits is bigger than the weight** W_{D1} **used to measure the Interest in Phone Display from user visits in different phones. The recommender module uses the k-means clustering algorithm in order to create representatives of customer groups that the system uses to make buying suggestions. The recommender takes as input the statistical data, described above, of the navigational moves of every customer and feeds them to the clustering algorithm. The clustering algorithm**

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

provides the recommender with clusters-groups of customer that have similar tastes. The recommender module takes these results and calculates the representatives of every group.

Every time a customer uses the system the recommender module finds his/her representative and proposes phones based on the representative taste percentages through the use of adaptive hypermedia. After creating the proposing phones list, the recommender system considers the mistakes statistics database and chooses a corresponding list of accessories. These accessories are combined with proposed phones list in order to provide the customers with hearing and sight problems with a more complete solution for their needs. For example if the recommender finds a phone that is very close to the representative's tastes than this phone is noted as "recommended" product and is given a different type of indicator from the Adaptive Hypermedia component than a phone that is more far, considering the tastes of the representative. Then the system finds an accessory corresponding to this phone and to the mistakes made from this user. If a new user enters the system the recommender classifies him/her to the group that has the largest number of members. This is based on the idea that if many users have similar tastes then a new user is probably going to have similar tastes and mistakes with the majority of them.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Figure 8 Screenshot from the phone and accessories recommendations that the system produces.

The degree of recommendation is presented through adaptive hypermedia. For example, the product that has the highest degree of interest for this user is noted as a "recommended" product and the one with a lower degree is noted as "check this too" product. Similar degrees of annotation are used for the corresponding accessories. Sample screenshot of the recommendation page is illustrated in figure 8.

3.5.3 The Adaptive Help System

The adaptive help module concerns the adaptive help responses. This module tries to identify mistakes in the navigational moves of every user. This module is based on the principle that many users with sight and hearing problems tend to have similar navigational mistakes. Again, the k-means algorithm is used to group users but in this case a different set of input data is used. The input data consists of the mistake degrees that were introduced

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

in the above section.

Mistakes are considered as different "wrong" navigational patterns. For example, a user can make "confusion navigation" like the continuous visiting of two neighboring production company buttons. This action raises the possibility of vision problem. Another example is "navigation without a purpose" witch can be achieved by a pattern of pressed buttons and clicked areas that leads to no purpose. This action raises the confusion problem. Degrees are calculated as a percentage of specific mistakes committed in a specific phones' page.

For example the disability to see companies' buttons degree is calculated by equation (9) and disability to recognize phone icons is calculated by equation 10. For every disability to see a phone feature, like wifi or screen resolution, an implementation of equation 11 is used. The general mistake degree is calqulated by equation 12.

$$HardtoSeeCompany_i = \frac{MistakeInCompany_i}{TimesInCompanyPage} \quad (9)$$

$$HardtoSeePhoneIcons = \frac{MistakeInPhoneIcon}{TimesInPhonePage} \quad (10)$$

$$HardtoSeePhoneFeature_i = \frac{MistakesInPhoneFeature_i}{MistakesInPhoneFeatures} \quad (11)$$

$$HardtoSee = \sum W_i HardtoSeeCompany_i + W_2 HardtoPhoneIcones + \sum_{i}^{1} W_i HardtoSeePhoneFeature_i / i \quad (12)$$

If a user has many navigational mistakes then the system responds and tries to help this user with help actions customized to his/her mistakes. These actions can vary a lot. For example, the "confusion navigation" results in actions such as the automatic changing of the size of brand names buttons or phone links. This

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

can result in a clearer presentation of the user interface. Another action taken by the system is changing the location of brand names' buttons in order to avoid confusion. Other actions involve speech synthesizers and agents pointing on the screen in order to help customers understand the locations of the user's interface components.

The "navigation without a purpose" can result in actions like showing an options message and asking the user directly what he/she wants to do. An example of a wrong navigational pattern can be the following: a user chooses to click on company button, then click the adjacent company button, then click the previous company again and then click the same adjacent company again. These four moves are interpreted by the system as a possible mistake of confusion between company buttons.

Every time a customer uses the system the adaptive help system finds his/her representative and responds with adaptive help actions. An example can be seen in Figures 9 to 12. In this particular example the system observes the user's navigation moves between two neighboring mobile phones and counts his/her mistakes.

Figure 9 First stage of the phone user interface. Showing small pictures of mobile phones.

If a user has made a lot mistakes in this section, like

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

browsing two neighboring phones repeatedly without putting any of phones in his cart at the meantime, then the system identifies that the user cannot view the phone pictures clearly and chooses to enlarge them.

Figure 10 Second stage of the phone user interface. The user has made mistakes. Bigger phone pictures and a next button showing that phones are split in two pages.

If the mistakes between the two neighboring phones continue then the system identifies that the user has confused only these two phones. The action taken by the system is to change the location of these two phones and move the one away from the other, while bringing a different phone close in order not to destroy the whole arrangement in the screen of the phones.

Figure 11 Third stage of the phone user interface. The user has confused the first two phones on the bottom. The system has changed their locations and brought the silver phone near the first phone on the bottom.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

If the user continues to make the same kind of mistakes then the system uses the animated agent in order point the phones by moving next to them, showing them with its “hand” and then telling with its “voice” the model of the phone. The system also increases the sound volume in order to help people with hearing problems understand more clearly the point out function of the animated agent. If the user finds annoying the changes of the user interface than he can disable them from his profile page.

Figure 12 Fourth stage of the phone user interface. The user continues to confuse the phones. The system enables the animated agent in order to point the phones and increases the sound volume the agent.

3.5.4 The Online Community System

iTVMobi incorporates an online community system that allows its special users to exchange and ideas about the electronic store and phones that are being sold (figure 13). In this system users can rate each other according to the expertise that a user may have on a specific field. The online community system uses user groups in order to define the members of each conversation. These groups are manufactured through the help of the user models of the system. In order for the community groups to be created the recommender module is used. Groups with similar tastes extracted from the recommender are form together in the sense that they will

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

have similar topics to discuss about products.

Figure 13. The online community system interface

The online community system uses the representative agents from the recommender and the adaptive help system in order to categorize users into different groups. In this way, users with similar mistakes, problems and tastes belong to the same group and are prompted to enter their corresponding chat rather than another one that doesn't much their profile. With this technique the system ensures that users will be able to find the conversations more interesting and keep the chat messages on topic. Users in this way can be helped more easily by others with similar experiences.

In every group there is an "expert" user. A user can acquire this title through the rating system. Every user can rate other user with a scale of one to five stars, with the one being the inexperienced and the five star rates being the expert on the matter. The user with the highest votes becomes the expert on the matter. This is very important to the system because iTVMobi uses expert users in order to annotate phones and accessories as top selections of these users. More specifically iTVMobi processes every expert users profile and suggests phones and accessories through

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

these profiles as top selections of these users, according to their tastes. These selections are presented through the use of adaptive hypermedia while a user's is navigating through the system. The annotation to the phone that is top selection from an expert user contains the name of the user and the topic of expertise s/he has. A screenshot of the online community system is illustrated in the figure below.

Figure 14. The annotation of the top selection by an expert user.

On the left side of the system there is the user's name, the conversation topic and the user's rating on this topic. In the top right we have the users' opinions and messages about this topic and below the user can write his/her personal message and send it to all users. In the figure illustrated below we can see the annotation of products through the use of adaptive hypermedia based on the expert users buys.

3.6 Incorporating Three Clustering Algorithms in Mobile Shopping

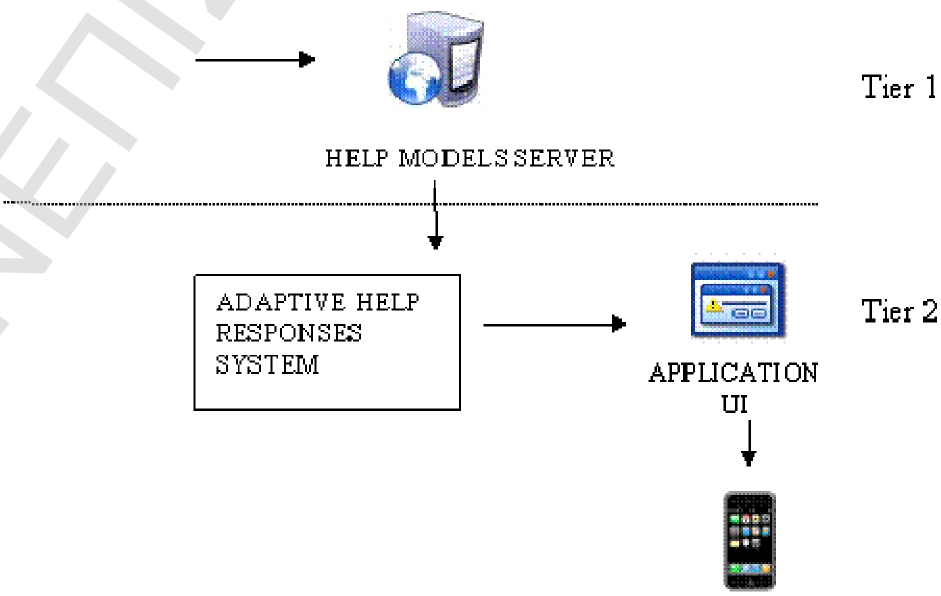
The challenges of creating a personalized mobile shopping application can be much greater than classic personalized e-shopping applications due to hardware and resource limitations. In this way, in order to create an effective m-shopping application we

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

first had to create an appropriate architecture in order to support our test bed application and combined it with our clustering algorithm methodology. This paragraph explains the fundamentals of our architecture, presets in brief the test bed mobile application created, demonstrates the usage of the clustering algorithms in our mobile shopping application and shows results coming results between the three algorithms used.

3.6.1 The Architecture combing mobile shopping and machine learning

The aim for developing an adaptive help user modelling server is to assist users while they are interacting with the application. This user model server can provide users with suggestions and perform help actions in order to solve navigation, interaction and other problems related with the user interface, usage and performing tasks in general, throughout the application. The general diagram for combining mobile shopping and help actions is presented below (figure 15). In order to do so the user modelling server measures classifies user moves in two types, right interaction and mistakes.



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Figure 15 General diagram generating adaptive help in a mobile environment

Mistakes are considered as different "wrong" navigational patterns. A navigational pattern is a series of moves made by a user in order to achieve a result. For example, a user can make "confusion navigation" like the continuous visiting of two neighboring production company buttons. This action raises the possibility of vision problem. Another example is "navigation without a purpose" witch can be achieved by a pattern of pressed buttons and clicked areas that leads to no purpose. This action raises the confusion problem. Degrees are calculated as a percentage of specific mistakes committed in a specific phones' page. If a user has many navigational mistakes then the system responds and tries to help this user with help actions customized to his/her mistakes. These actions can vary a lot. For example, the "confusion navigation" results in actions such as the automatic changing of the size of brand names buttons or phone links. This can result in a clearer presentation of the user interface. Another action taken by the system is changing the location of brand names' buttons in order to avoid confusion. Other actions involve speech synthesizers and agents pointing on the screen in order to help customers understand the locations of the user's interface components.

The "navigation without a purpose" can result in actions like showing an options message and asking the user directly what he/she wants to do. An example of a wrong navigational pattern can be the following: a user chooses to click on company button, then click the adjacent company button, then click the previous company again and then click the same adjacent company again. These four moves are interpreted by the system as a possible mistake of confusion between company buttons.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

The adaptive help user model server can be incorporated in any mobile recommending application, thus creating a personalised environment adapted to the users' needs. The user modelling server architecture is based on two tiers (figure 15, 16). The first tier consists of components that perform reasoning about users' actions and users' needs, databases with information concerning the user data collected and the help models database. The first tier is the server side of the architecture and does communicate with users directly. The second tier consists of user interface components. More specifically, the second tier contains the Adaptive Help Response System and the Application UI. The second tier is responsible for user interaction and contains components that communicate directly with users. The second tier can also give feedback to the users. This feedback information returns to the Help Models Server for further processing.

The Server Side of the architecture has three main databases and one running process. This architecture tier is responsible for collecting data from users, saving users' statistics through behaviour observation and managing this data in order to create stereotypes. More specifically the three main databases are explicit data, implicit data and stereotypic data. The Explicit Data database contains all the information in a database that users have provided to the system in an explicit way, either, by answering interview questions or by rating products. Every time a new user is registered in the application this database collects this information from demographic data, educational data and computer education data that the user provides through the registration process. The computer education data contains question related with basic computer knowledge and usage in order for the architecture to be able to determine the familiarity level related with computers technology. The demographic data involves questions about age, family and place of origin. This data is collected in order for the architecture to determine a probability level of possible disabilities concerning comprehension, sight or even hearing. The educational data contains questions related to the level of education.

These questions can provide the architecture with some first assumptions on how easily a new user can understand new concepts and how quickly can process these concepts. The data collected from registration process are combined and elaborated and then are saved in the Explicit Data database.

The Implicit Data database contains all the information about the user's interactions with the system. The data collected involve action moves related with categories selections, page selections and views, actions that users perform (e.g. moving products in a shopping cart) and users' interaction with specific user interface elements. For example, this database collects information about the categories of products that a user has visited, specific products that s/he visited, products that s/he moved in or out of his/her cart and products that s/he bought. The Implicit Data contains a statistics database of all users' actions and features collected by the systems.

In order for the Implicit Data database to be able to collect this kind of information, we created the Passive Observer process. This process conducts a user behaviour observation in the background, while the user freely interacts with the application. Every time a user performs a task, even the simplest one of clicking on a product icon, the Passive Observer records this action. All actions recorded by the Passive Observer are collected, elaborated, classified and saved into the Implicit Data database. The classification is performed according to the two error categories, interaction and purpose.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

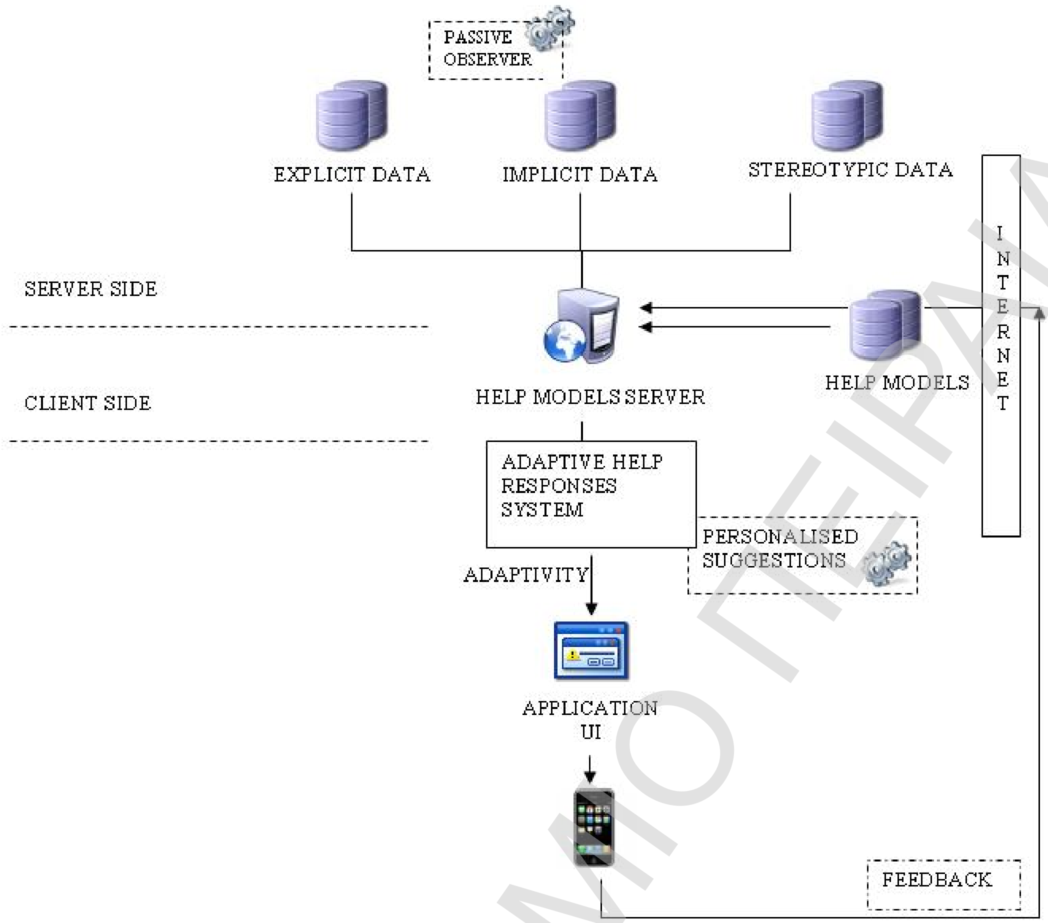


Figure 16 User Modelling Server architecture for mobile applications

The interaction error category, involves problems related with sight and hearing. For example, if a user cannot distinguish a button from the button next to it, this can cause an interaction problem. This interaction problem may be materialised in the form of repeatedly clicking these two buttons without clicking any other button during this process. The purpose errors are problems related to aims of users' tasks. For example, a series of simple actions that are stopped abruptly and an entirely irrelevant action is performed can be considered as a purpose error. Purposes errors are often related to comprehension disabilities. Computer familiarity can also play a big role in the incidence of these errors. These two main error categories are further divided into subcategories according to the domain and medium used. The Passive Observer is

not visible to the user and does not try to interact with users directly in any way.

The Stereotypic Data database contains information of stereotypes built with combined explicit and implicit data. These stereotypes are created dynamically in the Help User Model Server through a clustering process. The stereotypes refer mostly to new users or users that the server has little knowledge about. The stereotypes contain general assumptions based on levels of errors of the two main categories. With these stereotypes a new user can be classified easily and the system can give quick first suggestions that can improve his/her interaction with the application.

The Help Models Server sits between the server and client side of the architecture and it is responsible for the communication between the server side and the client. The Help Models Server interacts with the Help Models database and its main purpose is to create, modify and manage the user models taken from the Help Models database. The communication between the two sides (client and server) follows a very specific order. The Help Models Server takes as input the data from the three databases on the first tier of the architecture. The next step is to create user models through a clustering process. Then these user models are saved in the Help Models database.

Next, the Adaptive Help Responses System requests information from the Help Models Server, information that the last acquires from the Help Models of the Help Models database. The Adaptive Help Responses System uses this information to adaptively change the Application UI (user interface) according to the user's needs. Lastly, users with their actions and responses, while interacting with the application, can provide the Help Models Server with feedback information.

The Help Models Database, as mentioned earlier, contains all the user models of the users. These Help Models are user models

calculated in the Help Models Server and contain attributes concerning users' mistakes. The calculation of these user models involves the usage of the clustering algorithm. The data input for the algorithm, from the three databases (explicit, implicit and stereotypic) are converted by the algorithm into similar groups. The input data are inserted in the algorithm as degrees of mistakes.

An example of the equations that construct the HardToSee mistake degree can be seen below. Weights change, depending on the application that this framework is incorporated. However, general rules on weights apply on all applications that use this framework. For example, the weight concerning clicks on a specific product category is always smaller then the corresponding weight of the miss clicked products. The first weight is smaller, because clicking on a product category may not mean that the user is necessarily interested in this product but that s/he is just browsing several products.

(1) $$HardtoSeeCategory_i = \frac{MistakesInCategoryElement_i}{TimesInCategories}$$

(2) $$HardtoSeeProductIcon_i = \frac{MistakesInProductIcon_i}{TimesInProductPages}$$

(3) $$HardtoSeeProductAttribute = \frac{MistakesInClickingAttribute}{ClickedAttributes}$$

(4) $$HardtoSee = W_1 * HardtoSeeCategory + W_2 * HardtoSeeProductIcon + \\ + W_3 HardtoSeeProductAttribute$$

Another example equation the follows bellow measures the confusion of the user while interacting with the application. As these two equations show a user can make a mistake pattern concerning

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

product features or a basic move pattern. For example, a mistake in product features can be the following. Let's assume that users can collaspe or expand features of product that are grouped under a small number of titles. The features can be viewed by pressing the well plus symbol, thus exapnd the group into full view. If a specifc user insists in collapsing and exanding many times two or three neighbooring features categories without doing anything in between the system can assume that this user has confused the operation of the collapse and expand view. This is measured as a mistake pattern in these two features.

A mistake in a basic move pattern can be, for example, the confusion the the move to cart button with another user interface button. For example, the reapeted click of this specific button that is followed by an imidiate removal of these products from the shopping cart can be comprehented by the system as a confusion of the move to cart procedure. Similary, weights are used in the last equation in order to incoorporate different importance. For example, confusing product features has a lower wight than confusing basic interactions like moving products to the shopping cart.

$$ConfusedFeature_i = \frac{\sum MistakePatternsInFeature_i}{TimesSeeingFeature} \quad (5)$$

$$ConfusedBasicMove_i = \frac{MistakeInBasicMovePattern_i}{BasicMovePatterns} \quad (6)$$

$$ConfusionDegree = \sum_1^n W_i * ConfusedFeature_i + \sum_1^k W_i * ConfusedBasicMove_i \quad (7)$$

The clustering algorithm processes these degrees and provides the system with groups based on similarity. From these groups representative feature vectors are extracted. The representatives' vectors are the centres of every cluster provided by the clustering algorithm, work as group leaders and show the groups tendency to specific product features. The groups of users produced by the previous procedure are also used to create stereotypes. These stereotypes follow a general to specific hierarchy, meaning that the system constructs a low number of generic stereotypes at first and then continues to construct more specific stereotypes until it reaches a certain point of complexity.

The client side of our architecture consists of components that interact with users directly. The Adaptive Help Responses System component acquires information from the user model and tries to provide the best help suggestion to users. The Adaptive Help Responses System component contains all system personal suggestions about mistakes in a database. This component can use two techniques in order to make suggestions: adaptive hypermedia and an animated agent. The adaptive hypermedia is used to change user interface components according to the users' needs or preferences.

Adaptive hypermedia has the ability to change the user interface according to the help suggestions provided by the Adaptive Help Responses System. This component can annotate elements that are mistaken and change the position of these products in order to be seen first. It can also, change the symbols of products depending on the degrees of mistakes. Adaptive hypermedia can also change the font of product names or features in order to get user's attention. For example, if a user has made many mistakes between two products features, then adaptive hypermedia will change the font of one of the features in the product page, thus making this feature clearer to the user.

The animated agent can help users throughout the navigation of the system. This agent can provide useful information about the usage of the system and provide recommendations about products by acquiring information from the user model. The

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

animated agent cannot be incorporated to the mobile applications due to hardware limitations. Personalised help suggestions in the Adaptive Help Responses System are provided by the Personalised Suggestion process. This process is responsible for creating, modifying and controlling help suggestions extracted from the users' help models. This process runs in the background, in real time and intervenes when needed. This process provides help suggestions to the Adaptive Help Responses System. These suggestions are materialized in the Application UI in the forms described above (adaptive hypermedia and animated agent).

The Application UI component is a dynamic user interface that not only adjusts to the medium used automatically, but also changes according to the users' interests. The Application UI component can change the whole user interface appearance dynamically and personalise the user interface according to the specific users' mistakes. These changes are acquired from the Adaptive Help Responses System that contains all the user help suggestions and interacts with the Help User Model to get information from the Help Models. Because the Application UI is an entirely separate component it can be adapted to any medium, thus making the user modelling server medium independent. In this way the Application UI component can create a unique personalised experience for every specific user, resulting in a friendlier and more efficient user interface. The Help Models Server receives feedback through the reactions of users in the system help suggestions.

As the diagram of architecture (figure 12) shows, users interact through their device and in this way they respond to the architecture's personalised suggestions. The communication between users, client side components and server side components is performed through the internet TCP-IP protocol. The diagram below shows a schematic representation of the architecture and how the above components are connected together.

3.6.2 The Mobile Shopping Application

The application itself is a mobile shop that sells movies. The shop is called mVision monitors customer's actions and makes movies suggestions (figure 17). MVision is built on ASP.Net Mobile and uses the mobile device resources of every customer to create an effective user interface for every customer. For every customer mVision creates a different record at the database. There are two types of information saved for every user, the explicit and implicit. The explicit information is saved on the Explicit Data database and the implicit is saved in the Implicit Data database.

In mVision every customer can visit a large number of movies, by navigating through four movie categories. These four movie categories are: social, action, thriller and comedy movies. All customers have their own personal shopping cart. If a customer intends to buy a movie, she/he must simply move the movie into her/his cart by pressing the specific button. Users, also have the ability to remove one or more movies from their cart by choosing to delete them. All navigational moves of a customer are recorded by the system and are stored in the statistics database by the Passive Observer process. In this way mVision saves statistics, taking into account the mistakes of customers during their usage of mVision.

All of these statistical results are moderated from one to zero and saved in the statistics database. Apart from movie categories that are already presented, other movie features that are taken into consideration by mVision are the following: price range, leading actor and director. The price of every movie belongs to one of the five price ranges: 20 to 25 €, 26 to 30 €, 31 to 35 €, 36 to 40 € and over 41 €.

3.6.3 Incorporating Clustering For Adaptive Help

The adaptive help system in mVision incorporates the architecture presented in the previous section. The Help Models Server takes as input all data collected as statistics from mVision which are saved in the three appropriate databases of the server side of the architecture. Then the Adaptive Help Responses system acquires information about customers' mistakes from the Server. The information is based on the Help Models database.

Every time a customer performs a mistake the Adaptive Help Responses System changes mVision user interface dynamically. The statistics gathered implicitly by the Passive Observer follow a similar equation form as the one mentioned in the architecture. In order for a mistake in categories, to be recovered, the Passive Observer looks for mistake a pattern in the navigational behaviour of the customer. An example of a mistake pattern in categories can be repeatedly visiting two neighbouring categories, without visiting specific movies in between. Neighbouring categories are the ones placed in near positions in the screen of the mVision interface.

Another example of confusing interface elements can be confusing movie icons. A customer may visit movies, and then revisit them many times without moving any of them in his/her shopping cart. This raises a confusion mistake concerning the confusion between movie icons. We must take in mind that percentage degrees do not measures actual mistake number but a percentage of mistakes throughout the visits of this customer in the corresponding page.

All measured mistakes are multiplied by the appropriate weights caclulated in the Help Models Server in order to be converted to mistake degrees. For example the categories have

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

smaller weights than movie icons because the movie icons size is larger in the mobile than the category element. Furthermore, a movie category can be clicked many times just out of curiosity by several users. The illustrations that follow show two different examples on how the Adaptive Help Responses System responds to high degrees of mistakes in these particular areas. Figures 17 and 18 refer to the first example and figures 19, 20 to the second example.

The first example describes a situation where the customer has a high degree of mistakes in the movie categories area. This degree shows the confusion and inability of the customer to comprehend the difference between the four movie categories. Let's assume that this particular customer has made many mistakes in the two neighbouring categories of "action-adventure" and "thriller-horror". The Adaptive Help Responses System (AHRS) requires from the Help Models Server the degree of mistakes for these categories and discovers that this degree is above normal.

The first help action taken from the AHRS is to move the categories positions far away in order to be presented in a clearer way (figure 17).

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Figure 17 The top screen shows the initial state of mVision categories page. The bottom screen shows the first attempt of the Adaptive Help Responses System to change the UI in order to present categories in a clearer way. The categories have moved apart in screen position.

If the user continues to make mistakes between these two categories then the AHRS increases the size of these two categories and decreases the size of the other two (figure 18). If the user continuous to have this mistake behaviour then the AHRS changes

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

the font of these two categories in order to be very clear that they are two distinctive categories (figure 18).

Figure 18 The top screen the next step of Adaptive Help Responses System to change the UI. Categories are moved more apart and the two more mistaken categories have increased font size. The bottom screen shows the final step of this process. The two most mistaken categories have altered fonts in order to be more distinctive.

The second example refers to a customer that has mistaken two movie icons in the movies page. The AHRS checks the appropriate degree of mistakes from the Help Models Server. The first step to correct the customer's behaviour is to increase the size of these two icons and decrease the size of the other two. The AHRS moves the decreased movie icons to the left side of the screen. If the customer continues to make mistake patters concerning these two icons then the AHRS changes the Application UI again in two ways.

Figure 19 The left screen shows the initial state of the UI. The middle screen shows movie icons placed more apart and the most mistaken ones have larger size than the others.

At first splits the movies' icons in pages and inserts a next (⊙) and previous (⊙) button in order to show the customer that movies are split in pages. Secondly the AHRS increases the size of the movies' icons and places them in the centre of the mobile application UI. Lastly, the AHRS places two coloured borders around these movies' icons making the difference even more distinctive.

Figure 20 The third state shows movie icons size increased and frames of different colours placed in order to be more distinctive. Also movie icons are split in pages.

3.6.4 Comparing the Results of Three Algorithms

After implementing all algorithms and embedding them into our mobile application, we executed the clustering algorithms and evaluated the results of this method. Figure four below shows comparison results between hierarchical algorithm and k-means, and spectral and k-means. The results below clearly show that k-means produces better results concerning the data input. We also observe similarities between the vectors that were in fact users that had used our system. These vectors tended to aggregate into three main centers. This shows us that users tend to have three different behaviors.

These behaviors were more evident in the degrees concerning the mistakes of a user. They led to us to the conclusion that every

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

user grouped phone features into three main categories: style (company name, size, screen size etc), technical features (camera, connectivity etc) and autonomy (battery, talk time etc). The users also tended to make mistakes in three main actions: choosing a phone company, choosing a specific phone and manage their cart.

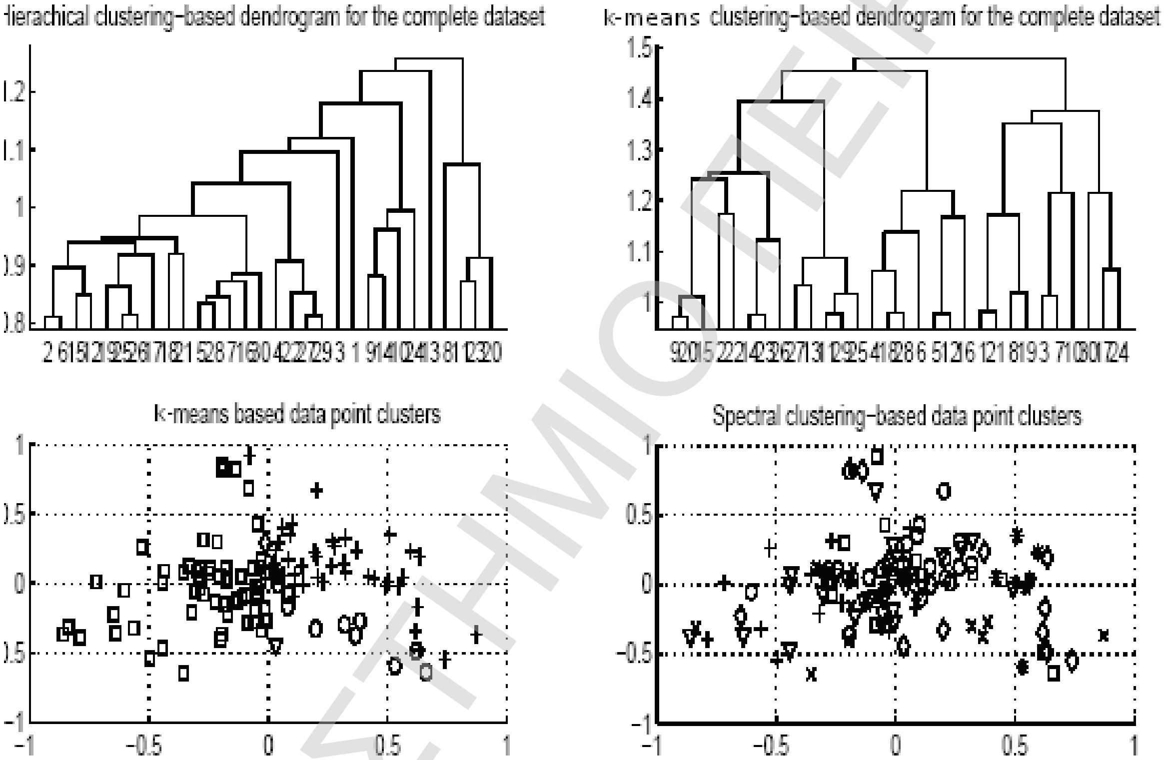


Figure 21 Comparison Diagrams between k-means, spectral and hierarchical clustering

3.7 Conclusions on the usage of different media and machine learning in remote shopping applications

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Our research in three different media and the usage of five different machine learning algorithms leads us to several conclusions. At first, we have developed an adaptive e-commerce application and the recorded behaviour of these 150 users provided data to the AIN algorithm which in turn provided 22 representative antibodies. This process leads to the conclusion that Vision.Com customers exhibit certain patterns of behaviour when shopping and tend to group themselves into six clusters. The 22 antibodies that arose via the AIN-based clustering algorithm correspond to customer behaviour representatives and, thus, can be seen as important customer profiles, which eventually correspond to stereotypes in user models.

This process is promising because it can provide recommendations based on the users' interests and the characteristics of a movie irrespective of whether this movie has ever been selected by a user before. Thus, the recommendation system can recommend new movies or movies newly acquired by the e-shop as efficiently as previously stored movies. At the second stage these representative antibodies were classified using a hierarchical algorithm which produced user clusters. The user clusters led to the construction of user stereotypes for user modelling.

At the end we expanded this system further and made these stereotypes dynamic. Dynamic stereotypes are able to refine themselves as more users interact with the system. In this way the stereotypes can become more accurate as the system acquires more expertise concerning the users' interests.

As a result, the benefit that we have gained is that our system can provide recommendation to all kinds of users (even new ones) on all kinds of movies (even those that have not been seen by any user yet). Moreover, the construction of a dynamic stereotypes hierarchy help us to solve a problem that a lot of intelligent e-shop

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

systems face which is incremental initialization. In this way Vision.Com was able to make successful predictions for new users with little information about.

Our research in the second medium showed that buying a mobile phone through interactive TV is not an easy task especially for the elderly users because they often have to face two different problems. The first is little knowledge about mobile phones, their features and their technology, and the second problem is the difficulty of using a computer application to buy these products. Also many studies show very different tastes between teenagers and older people (Zaphiris and Sarwar 2006). Many applications have tried to incorporate recommendation methods in order to resolve taste differences (Choi et al. 2006; Guan et al. 2005; Kazienko and Kolodziejski 2005). Also many applications have tried to help special groups of users (Muller et al., 2002; Zhao and Tyugu 1998). But despite the fact that all the above applications had very promising results concerning their aims, they didn't provide a universal approach to the problem of adaptivity.

Very few applications researched the problem in an integrated way (Fink and Kobsa 1998) but again on very different domains from our application. Our novel application researches the problem of combined adaptivity for both recommendations and adaptive help. ITVMobi has three major functions, a recommender, a help system and an on-line community system. All functions work in parallel to provide customers with the best navigation and product recommendations. These functions work under the same user modeling mechanism but take into account different features. The personalized recommendations take into account features of mobile phones based to the elderly tastes. On the other hand the adaptive help measures features concerning navigation mistakes. Despite the fact that we chose to emphasize in these features due to the fact that we chose this age group, iTVMobi monitors the whole user

behavior throughout the system. In this way, despite the fact that iTVMobi can monitor a user's whole behavior; depending on the criteria selected the reasoning mechanisms can focus on different features.

The group of the elderly people we chose as a test bed for our research is a special group of users with specific tastes and needs. Thus, the criteria we have chosen to include to our system had a great deal to do with the age of these people. We chose three criteria to include to our reasoning mechanism that are commonly found in the elderly. These criteria where shopping aim of this group, sight and hearing problems and little computer knowledge. Based on the above criteria we adapted our user model and chose to focus on different features in each one of the three functions.

However, our research revealed that if someone changes these criteria iTVMobi can be easily focused on a different group of people. In this way iTVMobi can be easily adapted to every group of people. Changing this criteria with the ones of another group iTVMobi recommendations, user interface and help system are personalized to this new group. For example, if someone chooses to sell phones on professional users, s/he could change take into account criteria such as extended connectivity features or mobile office abilities. In this way our system would focus on these features and change its behaviour entirely. As a result we can say that iTVMobi is a dynamic application for all people and specific groups of people co instantaneously

Lastly, our research on the field of mobile-shopping revealed many problems but also led to many conclusions. Despite, many interesting approaches in the field by Billsus (Billsus et al 2002), Miller (Miller et al. 2003), Yang (Yang et al. 2008), Kurkovsky and Harihar (2006), Setten (Setten et al. 2004), Andronico (Andronico et al. 2003) and Bell (Bell at al. 2006), all of these applications aimed at certain domains or products. Our approach was to create a

novel architecture general enough to be incorporated into any mobile e-shop application. Our purpose was to support the user throughout the interaction with the application in a personalised way. We did not focused o creating successful product recommendation as this problem has been thoroughly researched by many successful researchers. In this way we managed to create a generic user modelling server that provides personalised help for customer of mobile e-shop applications.

The aim of developing a user modelling server with an adaptive help function has two enforcements. The result was to create a generic architecture capable of delivering personalised help to its users and secondly with ability to be incorporated to a vast number of mobile commerce applications. Our user modelling architecture can successfully fulfil both actions. The ability to change every application's UI dynamically based on customers' mistake behaviour shows that an e-shop application can change its UI in order to create a friendlier environment for every user of every age. Secondly, we presented a case study that we incorporated this user modelling server. The case study was on a mobile device e-shop application. The case study shown here proves that our architecture can be easily incorporated into any product. This fact gives the ability to mobile e-shop applications to worry only about their customers and not about the technical demands of building such functions.

Summarizing, our conclusions were the following: all algorithms were incorporated in similar ways in all three applications, changing the criteria or the purpose of the recommendation does not affect the basic functions of the clustering algorithms, clustering algorithms can help in making successful assumptions about new users or users with little knowledge about, different media play a big role only to the interaction with the user and not in the incorporation of the

algorithms and lastly a generalized architecture is needed if someone wants to address problems of transferring intelligent and personalization techniques between different media and using different machine learning algorithms. These conclusions have led our research in two different fields. Firstly, the creation of a general methodology on how a developer can incorporate a clustering algorithm into a remote shopping application and secondly, the creation of a generic architecture, that can incorporate different clustering algorithms and manage different user interfaces based on different media.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

CHAPTER 4
INCORPORATION OF MACHINE LEARNING ALGORITHMS IN ADAPTIVE E-COMMERCE APPLICATIONS USING THE RUP

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

4 Incorporation of Machine Learning Algorithms in Adaptive E-Commerce Applications using the Rational Unified Process

Adaptive applications on various media have become very popular in latest years. Many researchers have proposed techniques in achieving adaptivity in this applications trying to help users interact with these applications in more direct and effortless ways. These techniques include many adaptive technologies such as user modelling, stereotypes and clustering algorithms and have provided these applications with promising results. However, the drawback of incorporating these technologies is reusability and mobility. This means that in order for these adaptive techniques to be used in different application or be altered in the same application is very difficult due to large degree of complexity of these technologies.

In this chapter we present an approach called RESCA-RUP that a researcher or a programmer can use to incorporate such a technology to a recommending system that can be used every time s/he builds such a system. This approach is based on the software life cycle techniques used in software engineering for systems lacking adaptive technologies but expands its usage to software that incorporates clustering algorithms. In this approach we use the rational unified process, with the help of UML language to incorporate a clustering algorithm to a recommending system regardless the medium and product used. This approach takes as basis the step of the RUP process of software life cycle and alters the steps in order to confine the incorporations of a clustering algorithm and double stereotypes to the recommending system.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

In this chapter we also present two prototype systems that we used RUP in order to incorporate two different algorithms and create appropriate dynamic stereotypes based on these algorithms.

4.1 Related work on software life cycle and rational unified approaches

The field of recommendation has been researched by many researchers (Kazienko and Kolodziejski, 2005; Choi et al., 2006; Guan et al., 2005; Li and Kim, 2004; Castro et al., 2001; Alspector et al., 1997). Other interesting media have also been researched like interactive TV (Maybury et al., 2004; O' Sullivan et al., 2004). On the field of systems that try to help elderly users or users physically impaired, there is significant on-going research work too. (Muller and Wasinger, 2002; Savidis et al., 2005; Zhao and Tyugu, 1998). A very common technique for achieving adaptivity is clustering. Such a work based on clustering has been done by many researchers (Lung et al., 2004; Kim and Ahn, 2008).

All of the above researches were very innovative concerning adaptive help to every specific user. The drawback of all of the above researches is that none of them addresses software life-cycle issues. Adaptive applications present a greater difficulty in applying e traditional software life cycle techniques. A very useful tool in software life-cycle is the Rational Unified Process (RUP). RUP is an object-oriented process that advocates multiple iterations of the software development process. It divides the development cycle in four consecutive phases: the inception, the elaboration, the construction, and the transition phase. Each phase is divided into four procedural steps, namely, requirements capture, analysis and design,

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

implementation, and testing. The phases are sequential in time but the procedural steps are not. Moreover, one important advantage of RUP is the highly iterative nature of the development process.

For the above reasons, RUP can be selected as the basis for presenting adaptive systems too. During our research we presented an RUP based software life cycle on how to incorporate a clustering algorithm on a prototype e-shopping system. The process has four major steps.

- Firstly, designing and building the prototype system that does not include any clustering techniques.
- Secondly, evaluating the system and through this process obtaining data for the clustering algorithms.
- Thirdly, comparing several clustering algorithms with the above data as input and choosing the most efficient algorithm.
- Fourthly, incorporating the clustering algorithm into the system and building stereotypes based on this algorithm.

In this chapter we show the generality of the RUP built in our research. We show that the same steps can be followed as before but this time the clustering algorithms are different. We will also use an entirely different medium for our built system witch is interactive TV and at last our test bed system will not only try to make recommendations about the product that sells but also, try to personalize its behaviour in order to help people with special needs use the system more effectively

Despite the fact that all the above applications of adaptive methods are very innovative and gave great results concerning the help of their users, none of them addressed software life-cycle issues. In adaptive applications like the ones mentioned above it's very difficult to apply a general life cycle. A very useful tool in software life-cycle is the rational unified process (RUP). RUP is an object-oriented process that advocates multiple iterations of the software development process. It divides the development cycle in four consecutive phases:

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

1. inception phase
2. elaboration phase
3. construction phase
4. transition phase.

Each phase is divided into four procedural steps, namely, requirements capture, analysis and design, implementation, and testing. The phases are sequential in time but the procedural steps are not. Additionally, RUP is an object-oriented process; thus, it is appropriate for the development of graphical user interfaces such as the one described in our research. Moreover, one important advantage of RUP is the highly iterative nature of the development process. For the above reasons, RUP can be selected as the basis for presenting adaptive systems too. An implementation of RUP life –cycle into systems has been researched by Jaferian (Jaferian et al., 2005), which presented extensions on Business modelling and Requirement discipline of RUP. RUP has been used to present extensions that concern possible security threats and attacks.

Significant in the field has also been done by Virvou and Kabbassi. Their work showed that developing a Graphical User Interface incorporating intelligence concerning in files and folders managing and modification like Windows Explorer. The system called IFM (Kabassi and Virvou 2000) presents an object - oriented approach in knowledge based software engineering of an intelligent GUI. In their second work (Virvou and Kabassi 2003) experimental studies were conducted for the IFM. Lastly, their third work (Kabassi and Virvou 2006) they extended the knowledge-based software life-cycle framework and they incorporated a multicriteria analysis. The common use of a modified RUP life –cycle as a tool for the design and development of IFM proved to be very efficient enhanced the software life –cycle process of IFM.

In this thesis we will present an RUP based software life cycle on how to incorporate clustering algorithms on two systems called RESCA-RUP. As test bed we have used two prototype systems, an adaptive e-

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

shop application called Vision.Com and a smart tv-shopping application called iTVMobi.

4.2 The RESCA-RUP Software Life Cycle

The RESCA-RUP life – cycle framework is presented in the table below (table 1). RUP is based on iterations but does not specify what sort of requirements analysis has to be conducted for adaptive systems and what kind of prototype has to be produced during each phase or procedural step. This table follows the phases and procedural steps of RUP but the difference is that we specify what prototype has to be constructed in each phase and what kind of experiment has to be conducted. Consequently, here is presented a modified solution to the problem of clustering incorporation and how this procedure can be generalized and be applied in an entirely different medium or product.

Table 1 RESCA-RUP Life Cycle Process

Procedural Steps/Phases	Inception	Elaboration	Construction	Transition
Requirements Capture	Requirements of a prototype adaptive recommender system without clustering.		The most efficient clustering algorithm.	
Analysis & Design	Analysis and Design of the prototype adaptive recommender system without clustering.	Computing the resemblance coefficients for the data set and developing the clustering algorithm.	Designing double stereotypes resulted from the selected clustering algorithm.(of users and	

			products)	
Implementation	Building the prototype adaptive recommender system without clustering.	Execute the clustering method for the prototype.	Building the user modelling component based on the stereotypes and incorporating them into the system.	Dynamically improving system performance while used by real users.
Testing	Evaluating the system and obtaining the data set for the clustering techniques.	Evaluating the Results of the clustering algorithm used in the prototype.	Comparing the results provided with those of the prototype system.	
Iterations	Iter #1	Iter #2 / Iter #3 / ... / Iter #n	Iter #n+1	Iter #n+2

4.3 RESCA-RUP Inception

4.3.1 Defining Requirements for the prototype system and Analysis and Design of the prototype adaptive recommender system

As with our previous research (iTVMobi prototype) we used the UML technique in order visualize our system initial requirements. UML helped to understand what where the true requirements of our system.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

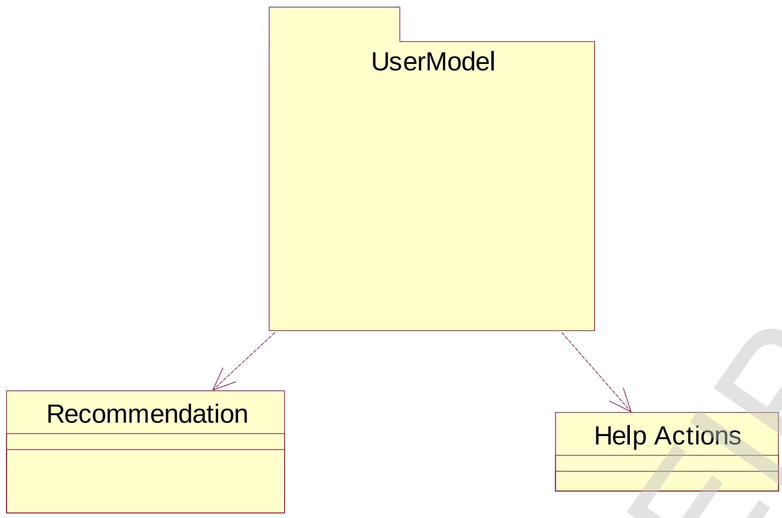


Figure 1 Class diagram of the user model and personalised classes

A class diagram is presented in figure 1 showing the connection between user models, recommendations and adaptive help actions. The second class diagram (figure 2) presents the classes that the user model package is consisted. The user model is consisted of four main classes. These classes include user behaviour mainly based on statistics of user's interaction with the system, interests that include interest degrees on product features, mistakes done by this user while interacting with the system.

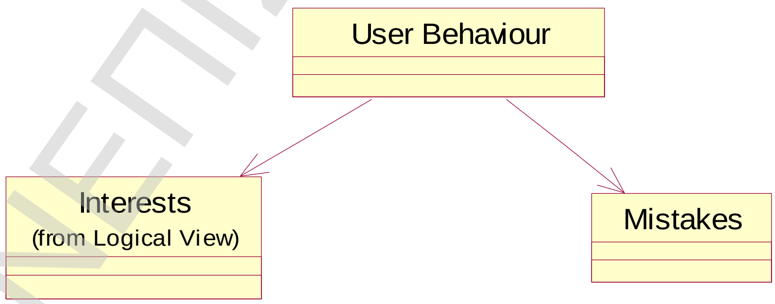


Figure 2 Class diagram of user model package

Similarly to the first prototype, our second system (Vision.Com) was designed according to the initial requirements we used UML technology in every step of the RUP to help us define classes, objects and

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

relationships between them. In this section we present a class diagram of the prototype system classes. In the diagram below we see the recommendations class that communicates with the interests class to acquire interest degrees. The recommendations and stereotype class give the appropriate information to the animated agent class and user interface in order to personalise their behaviour to customer's interests and needs.

4.3.2 Building and evaluating the prototype adaptive recommender system

The first prototype system was an interactive tv-shop that sells mobile phones. The users had their own unique shopping cart that they could use it in order to buy phones. They could also visit phone pages and watch live videos of phones being sold by the system. The prototype system called iTVMobi. The evaluation of iTVMobi included 50 elderly men and women to use our system and then answer a 16-question questionnaire in order to make comparisons between system results and real users' answers. The first results revealed significant weaknesses in achieving successful recommendations and take effective help actions.

The second prototype built, Vision.Com, an adaptive e-commerce video aimed to provide help to customers in movie selection. The rule-based personalisation built on the system was based on what a customer usually considers when choosing a movie. The experiment for evaluating Vision.Com included 150 real users that interacted with the system. The users divided into three groups conducted the evaluation into three different time zones. More specifically the evaluation conducted three different days and hours and consisted of three sections. In the end of the evaluation every participant was

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

asked questions about the systems visual appearance, usage, effectiveness and adaptivity in his tastes. Again results revealed several problems concerning the rules used to provide personalised recommendations.

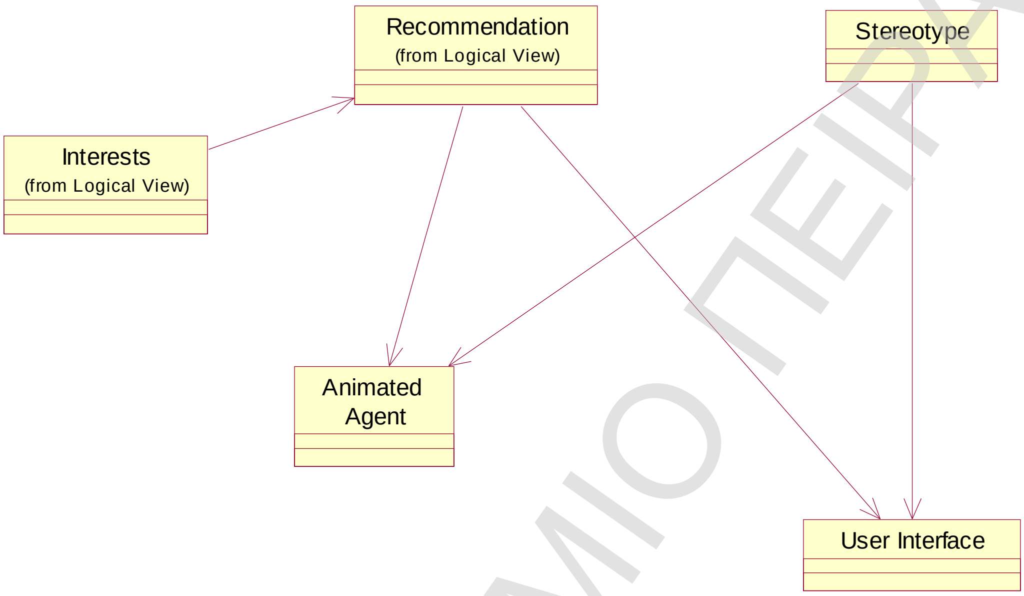


Figure 3 Class diagram showing main system classes

4.4 RESCA-RUP Elaboration

4.4.1 Computing the resemblance coefficients for the data set and developing the clustering algorithm

iTVMobi Prototype

In order to address requirements and solve the first prototype's problems we decided to incorporate a clustering algorithm. This incorporation would provide our system with a strengthen user modelling module. A comparison between several clustering algorithms was performed in order to select the most appropriate one. The input

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

data was the experimental results of the evaluation that we converted to vectors of measuring degrees in order to be fed as input to all clustering algorithms. The vector consisted of all the phone features measured by our system and all the navigational mistakes that users made throughout the usage of iTVMobi. The input data propagation revealed that the simple k-means (MacQueen, 1967) clustering algorithm best suited our purposes. We then developed a modified version of k-means algorithm from scratch using the visual basic programming language that was able to run in real time.

Vision.Com Prototype

The problems revealed after a careful study of the evaluation results were many. More specifically, stereotypes showed lack in effectiveness concerning users' categorisation. The main reason was that stereotypes are based on strict categories of users and cannot change dynamically as the customer used the system. Furthermore, the rule-based system of Vision.Com provided customers many times with inadequate product recommendations, especially when interest degrees showed small differences. Changes were to be made if we wanted to tend to these requirements. In this case we decided to incorporate again a clustering algorithm but we made a comparison between different four clustering algorithms. The input data for the algorithms were again taken from experimental results of the evaluation. In this work, we developed four clustering techniques, namely

a) agglomerative hierarchical clustering ,

b) fuzzy c-means clustering and

c) spectral clustering

against AIS-based clustering.

4.4.2 Execute the clustering method for the prototype and Evaluating the Results of the clustering algorithm used in the prototype

iTVMobe Prototype

This section presents the clustering algorithm comparison results. The selection of k-means was made after executing and the two other mentioned above. The k-means showed many similarities between vectors from users that had used our system. These vectors tended to aggregate into three main centres showing that users can be categorised in three different behaviours. These behaviours were even clearer in the degrees concerning the mistakes of a user.

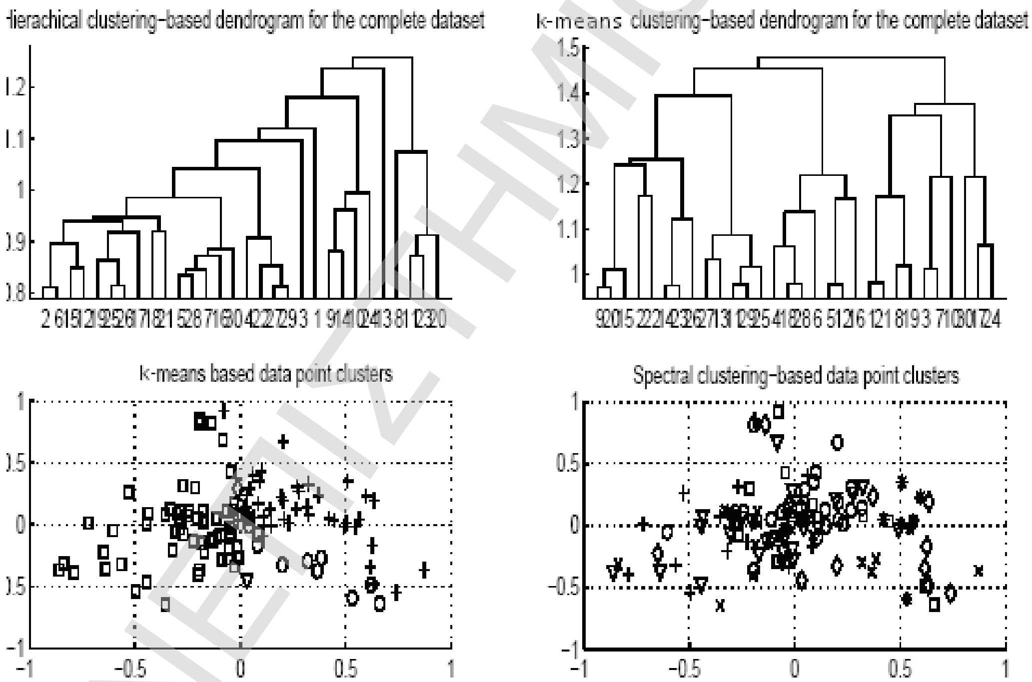


Figure 4 Comparison Diagrams between k-means, spectral and hierarchical clustering

After further research between the clusters created, their distances and the intra-cluster distances the conclusion was that every user could be grouped according to phone features. Three main

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

categories were revealed, style (company name, size, screen size etc), technical features (camera, connectivity etc) and autonomy (battery, talk time etc). Furthermore, three mistake degrees were discovered, choosing a phone company, choosing a specific phone and managing their cart. The UML diagram below shows a state chart diagram concerning the class of user model. At first the user is clustered with the help of k-means. Secondly, the user model is updated with this information. Next, a representative is selected to represent the user. Lastly, two stereotypes are selected one concerning interests and one for the mistakes.

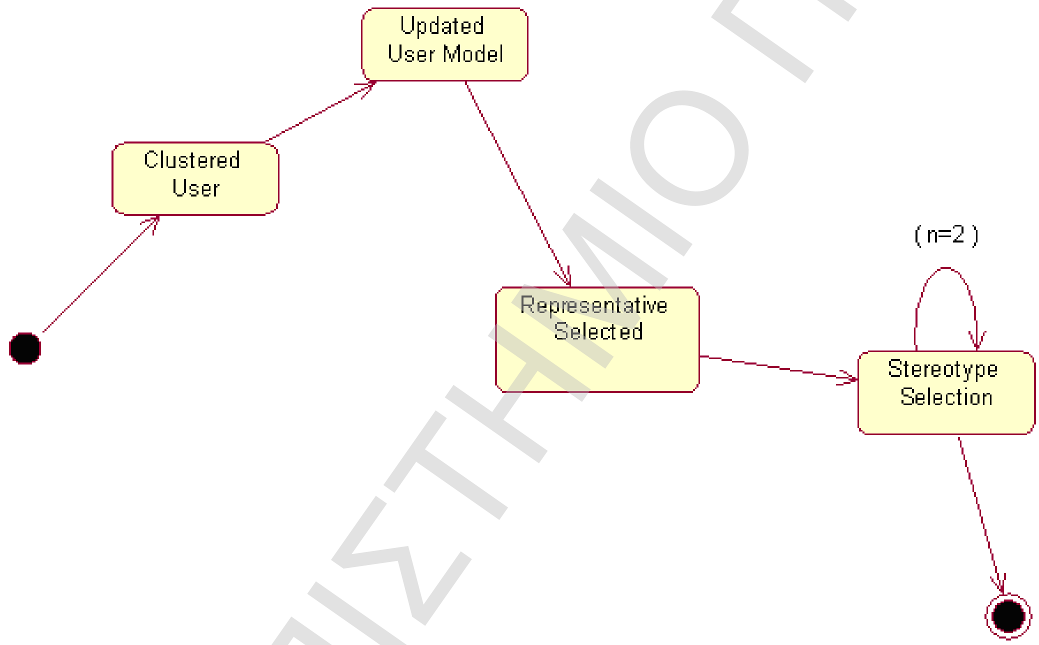


Figure 5 UML state chart diagram of user model class and different states.

Vision.Com Prototype

For Vision.Com we collected data from the experiment which consisted of three parts, visits that every user made to specific movies, movies the users moved into their cart and movies bought by every user. This data formed a vector of 80 features of the movies' characteristics. The features include all the categories, all the sets of prices, all the actors and all the directors that are known to the system. These 80 feature vectors were in the representation of users' interests and movie descriptions. Finally, 80 - dimensional vectors were inputted to the four different clustering algorithms mentioned. This was done in order compare the results of the first three with the last and to prove that AIN excels against the others.

Figure 6 shows dendrograms corresponding to the feature vectors produced by hierarchical (top left), spectral (top right), and AIN-based (centre left) clustering. Spectral clustering lacks to provide a clear revelation of the intrinsic similarities in the dataset over hierarchical one. Moreover, AIN-based dendrogram leaves are fewer than the leaves of other dendrograms, which is a result from the fact that AIN clusters only 22 representative points in the 80-dimensional feature space, while the other two algorithms clustered the complete data set. The AIN-based dendrogram demonstrates the intrinsic data point clusters clearly in contrast to the hierarchical and spectral dendrograms. Furthermore, the partitioning of the complete dataset into six clusters by the spectral (centre right), fuzzy c-means (bottom left), and AIN-based (bottom right) revealed that the spectral clustering does not result in cluster homogeneity, the fuzzy c-means clustering results in higher cluster homogeneity, but in only four clusters rather than six required.

The AIN-based clustering on the other hand returned significantly higher cluster homogeneity. Moreover, the degree of intra-cluster consistency is higher in AIN-based rather than the other two. Figure 6 clearly shows that Vision.Com customers exhibit certain tendencies when shopping and can be grouped into six clusters. The 22 representatives that arose via the AIN-based clustering algorithm correspond to customer behaviour representatives and, thus, can be

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

seen as important customer profiles, which eventually correspond to stereotypes in user models.

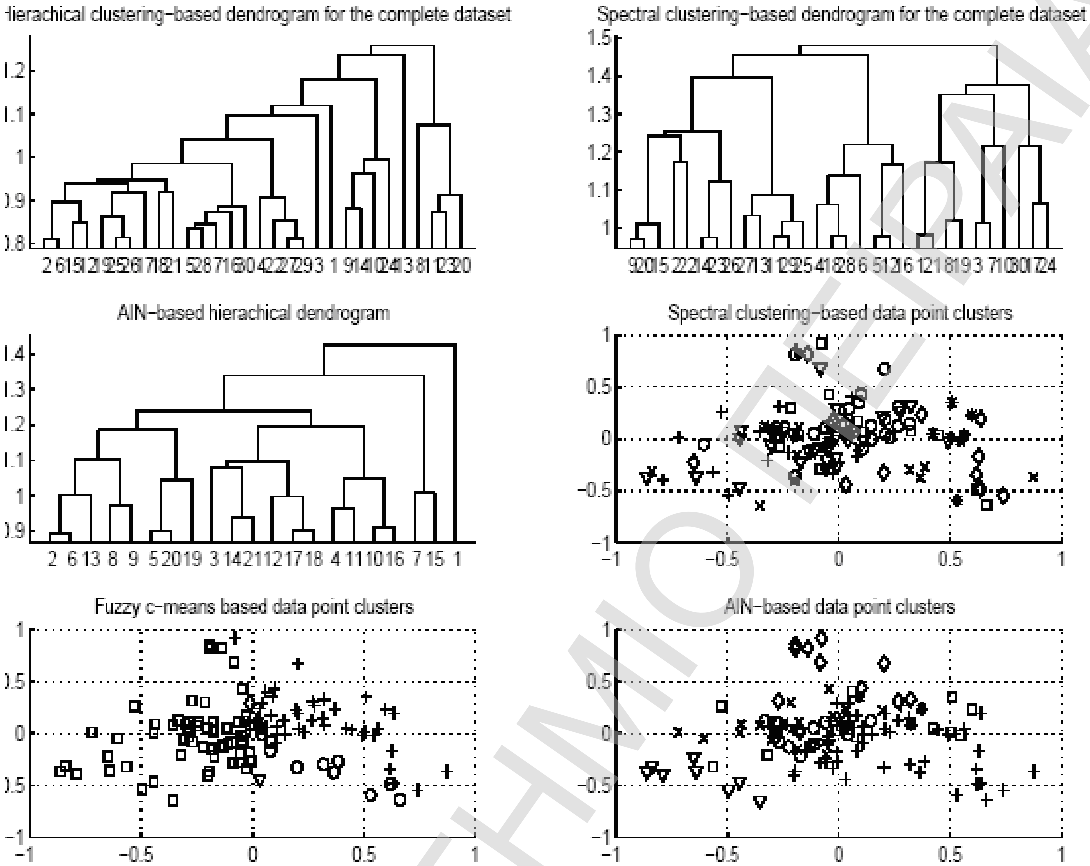


Figure 6 Result diagrams of the clustering methods

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

4.5 RESCA-RUP Construction

4.5.1 The most efficient algorithm and designing stereotypes based on this algorithm

iTVMobi Prototype

For iTVMObi prototype we used the k-means algorithm chosen through the procedure mentioned before we extracted the three main vectors for the product recommendation and the three main vectors for the mistakes made by users. With these vectors as guide a stereotypes hierarchy was created starting from general stereotypes till this user concludes to have an individual user model. Below we see a activity diagram of the “four level” procedure in order for a user to reach the individual user model. The user starts belong to a general stereotype and as he uses the system, the user model becomes more specific and ignores general stereotype information.

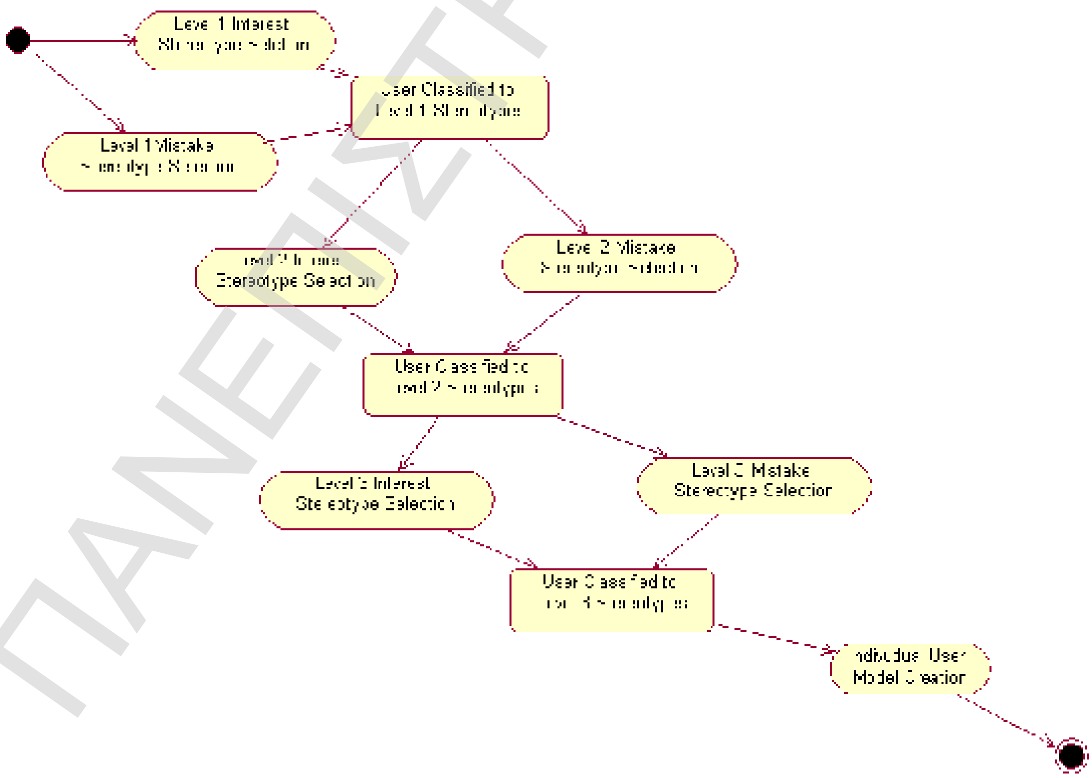


Figure 7 activity Diagram of User Classification

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Vision.Com Prototype

The decision of AIN based clustering as best suiting our system led to the redesign of Vision.Com. The new model works as follows: At the initial state the system includes all user model data of all users. This data is peddled to the AIN algorithm that produces clusters of similar users. Every cluster has a center or more that is the representative of this cluster. The centers are called antibodies. In order to connect a user with an antibody the system calculates the minimum Euclidian distance between the antibodies and the user. Then with the help of the antibody finds the minimum distances between movies vectors and antibody vector. The movies with the minimum distance are rated as very similar and presented to this specific user as the recommended ones with the help of adaptive hypermedia.

The results of AIN based Vision.Com were good concerning product recommendations and general recommendations throughout the navigation of the system for every individual customer. The more a customer uses the system the better personalised recommendations are produced. The main problem of Vision.Com in this state was the lack in recommendation production for new customers and customers with little knowledge about. This problem was solved by creating dynamic double stereotypes. Stereotypes for customers and stereotypes of movies based on the AIN clustering that varied in complexity and accuracy.

The sequence diagram (figure 8) shows how the system constructs stereotypes starting with a new user until the system classifies him as old user. This is the initialization process of Vision.Com in order to achieve the point of adequate information about this specific user. Every level of stereotype that the user passes to is more complex and involves more movie features. This procedure is continued until level 6 is reached. Six level were selected

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

corresponding to the six clusters revealed from the evaluation process and algorithm comparison process. The last level includes all movie features and the user is categorized as old.

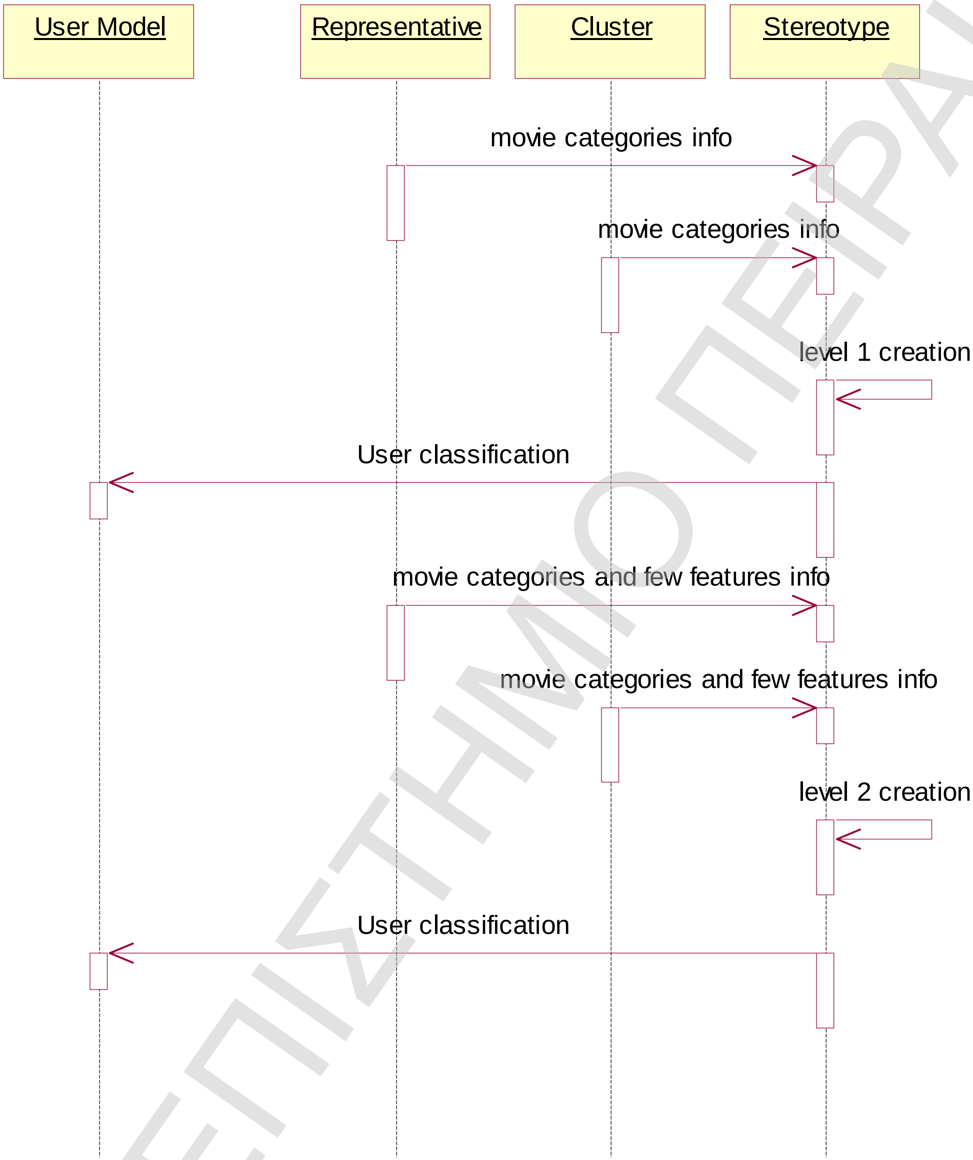


Figure 8 sequence diagram of the initialization process

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

4.5.2 Building the user modeling component based on the stereotypes and incorporating them into the system

iTVmobi Case Study

A hierarchy system of double stereotypes for both interest and mistakes degrees was created through the method described above. In this way, selling phones in the tv-shop could be performed more efficiently due to the fact that the system divided interests and mistakes. This system of double stereotypes provided us with a quick incremental initialization of the user model meaning that iTVMobi responded to customer needs directly. The double classification performed resulted in three levels of user stereotypes and a fourth level of individual user model. These levels corresponded to clusters of similar users created be k-means.

The first level of stereotypes is a coarse classification of three stereotypes. Their complexity is refined several times to provide users with their specific individual user models. These dynamic stereotypes are then created and used by iTVMobi to extract users' interests in movies based on a small set of observed old users' actions. More specifically, for a new user, this extraction is based on the first level of stereotypes classification. Then, incrementally new inferences are constructed about his/her preferences from more refined user stereotypes of subsequent levels. The procedure ends when this new user reaches the level of individual user model.

The users of iTVMobi showed an aptitude to group into three main sections for both interests and mistakes. This fact has influenced the stereotypes' creation and led to three divisions inside every

stereotype. Every division concerns a section of interest or mistakes. For example, the interest stereotype consists of the following three divisions. The first contains stereotypical information about style (company name, size etc), the second stereotypical interest degrees about technical features (camera, connectivity, operating system etc) and the third interest degrees about autonomy.

So in every subsequent level the system removes a section from the stereotype and affiliates this section into the information taken by the individual user model. For example in the first and more abstract level of classification the user model is mainly based on a general stereotype that may have all of its three divisions but it is built with general stereotypical information because the system has little knowledge about this user. As the user uses continues to use the system the general stereotypical information is lessened and replaced with individual personalised information.

Vision.Com Case Study

Similarly in Vision.Com the construction of dynamic stereotypes led to a double classification of both users and movies. More specifically, every customer was provided with individual suggestions that were produced very quickly. Moreover, the quick incremental initialization achieved led to enhanced responsiveness to customer needs in the minimum amount of time. The hierarchical double classification (users' interests – movies) constructed resulted in several levels of user stereotypes. Every level corresponded to clusters of the AIN based clustering.

At first, there was a very general classification of two stereotypes which was next refined several times to produce a more complex classification. The final level classification led to six stereotypes. These stereotypes are then used dynamically by the e-commerce application to discover interests in movies based on a small set of observed users' actions. A new user is mainly classified from the first level of stereotypes and then, incrementally, while the user interacts with the system, inferences about his/her preferences are discovered and user stereotypes of subsequent levels are redefined.

The actual construction of stereotypes mainly involves defining the triggering conditions, and the inferences (what can be assumed for users belonging in the triggered stereotype). Whether a customer or a movie belongs or not to a stereotype is measured by the Euclidean distance. If a customer's distance is high then we proceed to the above level of stereotypes. This process of moving to above level and measuring distance again stops when a certain distance is reached or the first level of complexity is reached. If in this level the distance is also far we move up to the next. The same process for movies is followed by Vision.Com in order to extract similar movies of the same level of stereotypes. This movies stereotype selection stops when the distance between the user stereotype and movie stereotype reaches a certain threshold.

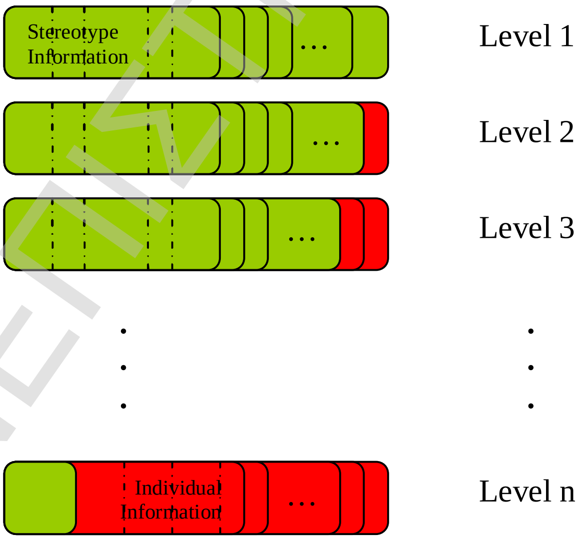


Figure 9 Diagram showing stereotype increasing complexity through levels

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

The diagram above shows how the general idea of stereotypes increasing complexity is achieved. Also in this diagram every stereotype is consisted of several sections corresponding to users' tendency to group in similar behaviour according to their interests. For example, iTVMobi as mentioned above tend to show three different behaviours concerning mobile phones, style, technical features and autonomy.

4.6 RESCA-RUP Transition

4.6.1 Dynamically improving system performance while used by real users.

iTVMobi with real users

When a new user becomes a member to the system the application creates a user model, sets all interest values into zero (the system assumes that at start the user has no interest for any phone) and starts to monitor his/her actions. After few interactions with the system (visiting few phones), the e-shop classifies the new user in a stereotype of the first level of specialization. The first level has very general information about the three main sections of interest and mistakes.

For example, if the new user shows a tendency towards big screen size and phone size then s/he classified in the second stereotype, as the main difference between the two stereotypes can be seen in this feature. If the user belongs to the first stereotype the system chooses to propose phones with smaller sizes. The phones that are most close to the user's interest degree are those recommended by the system up to this point of interaction. As the user continues with moving phones into his/her cart and buying some of them the system moves to the next level of classification that removes a section of stereotypical information and replaces it by individual user model information. This means that in this level stereotypes differ in one section but group users in the other two. For example, if this new user selects phones with small size but big screen then the system chooses to classify him/her to the stereotype that has general information about interests in technical features and autonomy but not fore style that is greatly influenced by size.

Level three and four of specialization extend the features of interests and constrain stereotypes in favour of individual information. In this way, as users show with their actions which more features or mistakes the system easily classifies them into the according stereotypes and selects the right interest or mistake stereotypes in order to make recommendations and help actions. The initialization process is conducted until the user reaches level four of specialization. This level represents the leaves in the hierarchical tree of stereotypes and extends the differences in all features of interests. The level of complexity here is very high and the smallest difference in user's interest or mistake degree can alter his/her user model.

Vision.Com with real users

When a new user becomes a member to the system the e-commerce application creates a profile, sets all interest values into zero (the system assumes that at start the user has no interest for any movie) and starts to monitor his/her actions. After few interactions with the system (visiting few movies), vision.com classifies the new user in a stereotype of the first level of specialization. The first level generally checks interest concerning the four movie categories. For example, if the new user shows a tendency towards thriller movies them s/he classified in the second stereotype, as the main difference between the two stereotypes can be seen in this movie category.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

If the user belongs to the first stereotype the system chooses to propose movies from any of the three categories except thrillers in a presentation percentage similar to the interest in every category. The movies that are most close to this movie stereotypes are those recommended by the system up to this point of interaction. As the user continues with moving movies into his/her cart and buying some of them the system moves to the next level of classification that extends stereotypical information to the price features. This means that in this level stereotypes differ greatly in the price ranges along with movie categories. For example, if the new user selects movies with medium prices the system chooses to classify him/her to the stereotype that has the greatest interest in this price range concerning always the interest in movie categories.

Level four and five of specialization extend the features of interests into interests in leading actors and directors accordingly. In this way, as users show with their actions which actor or director prefer the system easily classifies them into the according stereo-types and selects the right movie stereotypes in order to make recommendations. If their differences in interest in the actor and directors are low the system chooses to group users in same stereotypes thus emulating the grouping process into the previous level of specialization. On the other hand if these differences are high the users are grouped into different stereotypes of these levels.

The initialization process is conducted until the user reaches level six of specialization. This level represents the leaves in the hierarchical tree of stereotypes and extends the differences in all features of interests. The specialization in these level is very high and even the smallest difference in user's interest can classify him/her to a different stereotype of this specific level.

4.7 Comparing the RESCA-RUP life cycle created for both prototypes

The resulted systems of this life cycle process were very effective systems that could help its customers choose the right product. The confusion of a new customer while navigating through the user interface of the prototype systems was decreased. This study shows that RUP life cycle process can also be applied in other domains with roughly the same steps.

The table below (table 2) shows the differences in the steps of procedure for the two prototype systems and the general RUP life cycle presented at the start of this paper. From the table observe it is observed that the steps have remained the same and those altered followed similar guides as the previous ones. Consequently, the RESCA-RUP process was not altered in its basic form and can be easily applied to different domains. The different media used in the two prototypes did not compromise the procedure at all. The different products of two stereotypes also have not influenced the basic procedure of RUP and only changed the point of view of some steps without changing that real procedure conducted in the background.

The tendency of helping users in an adaptive way in the iTVMobi prototype only changed the procedure steps concerning stereotypes on a technical level as their aim remains the same.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Table 2 Comparison Table between the two prototype RESCA-RUP was incorporated.

Procedural Steps/ Phases	Inception	Elaboration	Construction	Transition
Requirements Capture	Same as before		The most efficient clustering algorithm. iTVmobi | Vision.Com k-means | AIN	
Analysis & Design	Same as before	Same as before. (conducted twice)	Designing double stereotypes. iTVMobi | Vision.Com Users, mistakes | Users, products	
Implementation	Same as before	Same as before. (conducted twice)	Building the user modelling component. iTVMobi | Vision.Com Dynamic Stereotypes Individual User Models | Dynamic Complexity Stereotypes	Sane as before
Testing	Same as before	Same as before. (conducted twice)	Same as before.	
Iterations	Iter #2	Iter #3...n	Iter #n+1	Iter #n+2

4.8 Conclusions about the RESCA-RUP process

The two test bed applications used to test the proposed RUP process revealed very interesting results about the process itself and the similarities in engineering such intelligent remote shopping systems. Firstly, the RUP process is very difficult to be incorporated into

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

adaptive systems that include many intelligent technologies. Furthermore, difficulties can also be found in testing such systems and what kind of evaluations must be conducted in order to extract useful information about their performance. As it can be seen from the whole process UML methodology can help in the incorporation and better presentation of RUP states during the software life cycle process. With the help of UML a researcher can grasp the concepts of the processes much easier.

Secondly, the creation of the RUP table can help future researcher by providing a guide on what steps should be followed in order to incorporate a clustering algorithm, use this algorithm to provide successful results and test both built system and results for their efficiency. Furthermore, the small differences in some steps of the RUP process reveal significant details that researcher must taking into account in order to avoid mistakes during the software life cycle.

Lastly, because RUP does not define the algorithms which will be used or the actual types of stereotypes used in the appropriate steps, gives the researcher to incorporate any algorithm in the process and use for any reason. Here, in our approach we have incorporated two kinds of all algorithms, kmeans and AIN, and used them to provide both product recommendations and help support through the appropriate stereotypes. Both algorithms and kinds of stereotypes were incorporated in different test bed applications without making any changes to the actual steps of the RUP process.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

CHAPTER 5

GENERIC ARCHITECTURE OF ADAPTIVE REMOTE SHOPPING APPLICATIONS

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

5 Generic Architecture of Adaptive Remote Shopping Applications

E-commerce applications have become very popular since they provide easy access to all kinds of products. However, most of the e-commerce applications are developed in a generic way that does not take into account personalised needs and preferences of individual customers. In conventional shops, human sales assistants can easily find out customer's specific needs. On the other hand product-sale applications do not address this problem and the customer ends up searching in endless product databases or even using other search sites. Furthermore, retailers deal with concrete difficulties in merging available information about customers and keeping it updated.

In this way many negative effects follow (Goy et al., 2007): The lack of data integration and synchronization prevents vendors from effectively exploiting customer information to promote their products and services. Furthermore, the lack of tools supporting the analysis of customers' browsing behaviour (e.g., shopping cart abandonment) does not enable vendors to collect feedback useful to redesign and optimize their Web sites (Hall 2001).

All of the above problems in e-commerce applications may be addressed by the incorporation of user modelling components that are based on intelligent techniques. These techniques allow the system to draw inferences about specific user's needs and preferences based on information collected about these users. A solution to this situation can be achieved by personalization techniques that can be extracted from the user modelling theory (Rich 1983). A system that incorporates user modelling can be adaptive to every specific customer's interests and needs and is

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

called product recommendation system. More specifically, in order to achieve personalization the software agent providing the services must have the capabilities of maintaining a user model containing data about this user's needs, interests and preferences and the capability of using strategies for adapting its behaviour to each specific user. The choice of such strategies depends on many factors such as the application task and domain, the goals of the system, the target users and the context (Ardissono et al., 2001).

However, these systems are difficult to build and they require a lot of analysis that involves knowledge engineers and product sales analysts. Moreover, all of this developing effort is dedicated to specific e-commerce systems or research prototypes that are built at a time for a particular kind of e-commerce product (e.g. books, dvds, etc). This means that each time a new personalised e-commerce application has to be built, knowledge engineers and researchers have to start from scratch. This is also the case for e-commerce applications that are developed for different media such as computers or interactive TV sets. With the recent expansion of TV content, digital TV networks and broadband, smarter TV is needed as well (Ardissono et al., 2003).

A lot of research prototypes made are dedicated to desktop e-commerce applications, or the interactive TV or mobile devices. However, a large part of the reasoning mechanisms that are needed for one medium may be the same for another medium. But other challenges may rise here, concerning the way that these reasoning mechanisms can be incorporated into different media or products. For example, a large number of recommending systems use clustering to create groups of users with similar tastes (Tipnoe et al., 2005; Lu et al., 2006), but this reasoning mechanism is dedicated to measure characteristics only related to the specific domain of the personalised proposals. Furthermore, these proposals are developed specifically for the medium that the system uses

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

without consideration of system's resources and capabilities. Consequently, a researcher must consider two different factors here which are the characteristics that the algorithm should use in order to create the groups needed and how the personalised proposals will be developed in order to be medium independent.

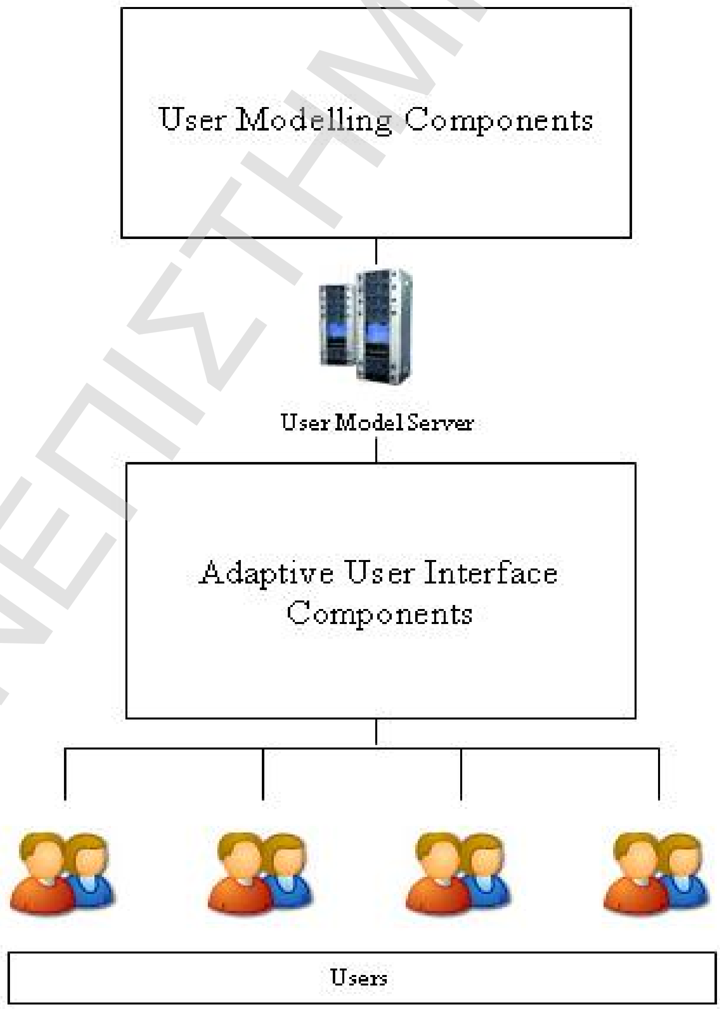
A large amount of work has been devoted by researchers and companies to address, at least partially the challenges that e-commerce systems pose, with the ultimate goal of improving customer services. The end result was two types of systems, research prototypes and commercial systems but both types were proved to have a major drawback. The former offered advanced interaction and personalization features, but they are usually developed as "closed" systems, which embed proprietary customer and product databases. Thus, they fail to support a seamless integration with the applications broadly used by vendors to manage their business activities. The latter are more complete systems and they offer tools to analyse the customer behaviour.

However, they usually support rather simple personalization features to assist the users during the navigation of the product catalogue and to tailor the interaction style to individual needs. In view of the above a recommendation system must be more generalised and be able to deal with both problems of the two types of systems presented above.

Currently, there is need to move towards to more converged architectures that can correspond to different types of media such as mobile phones and the internet in order to address the above problems. These converged architectures may be able to incorporate intelligent techniques in order to achieve personalization, but until now there is not much effort towards binding generic e-shop architectures and product recommendation techniques. Product brokering that requires assisting users in finding information in a complex multidimensional space (Pu and

Faltings, 2002) also adds to the difficulty of constructing such frameworks. A very important comment has also been noted by Aragao (Aragao et al., 2001) that there is no widely available mechanism to allow users to personalize their interaction with web data and services. This means that such generic architectures with adaptive abilities are becoming more and more necessary for this kind of applications.

In view of the above we have developed an architecture called PERCOM (PErsonalised Recommendations in e-COMmerce) (figure 1) for remote shopping applications that may be used for all kinds of products that are being sold through different kinds of media (iTV, internet applications, mobile phones etc.). In order to address the two main challenges of e-commerce systems which are product and medium independent, we developed two intelligent techniques.



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Figure 1 General PERCOM Framework Architecture

Firstly, for the reasoning mechanism based on clustering, we chose an entirely different approach. Instead of choosing product characteristics to model our users, we chose a set of characteristics based on user behaviour that were common to all e-commerce systems. These characteristics were the number of visits in a product category page, the number of visits in a specific product page, the number of a product moved to the cart and the number of a product bought by the user. Every time a customer makes one of the above actions the appropriate product characteristics rise, based of course on which product visited, moved to cart or bought. Furthermore, every one of these four main actions has also a weight which is used to measure the action significance. More specifically, the visiting actions have smaller weights than the ones referring to moving products to cart or buying a product. This is so because a customer's visit to a product page does not always mean direct interest for the product visited.

In this way there is a two step process in order to extract the level of interest in a product characteristic. Thus, the architecture can generalise users' actions into the four main actions mentioned above and create a first level of customer behaviour observation that can be used in any e-commerce application. Furthermore, with this process the resulted level of possible interest is a percentage. This results in a vector of percentages that can be easily manipulated by any clustering algorithm.

Secondly, in order to surpass the challenge of different media we chose to incorporate two different technologies. The first is a dynamic user interface and secondly adaptive hypermedia. Furthermore, we developed these two technologies as separate modules giving the architecture the ability to generalise and manage these modules in an independent way. The dynamic user

interface module is responsible for changing the user interface according to the resources of the system used whether this are a mobile phone, an interactive TV or a desktop computer. This is achieved by applying the following rule. Every e-commerce system has five major elements, a shopping cart, a categories product page, a specific product page, a search page and a recommended products page.

Our architecture in the user interface module stores the basic frame for these elements. Then the system can apply its specific interface add-ons to these frames thus creating the interface needed. These frames can change dynamically according to the medium used by the system. Moreover, the adaptive hypermedia module is responsible for creating the sets of recommended products. The visualization of the recommended products is managed by this module in three ways, low resources, medium resources and high resources. More specifically, low resources offer fewer ways to propose products and are mainly used in mobile devices. Medium resources are used on the interactive TV due to lower specifications, than desktop computers and the simpler interaction through the remote control. Higher resources are used to desktop computers and laptops. Higher resources can also use the animated agent in order to achieve a friendlier human computer interaction. The above means that the architecture is both domain and medium independent.

PERCOM incorporates a clustering algorithm which is used to create groups of similar users in terms of their needs and interests. The architecture of PERCOM framework consists of two main modules. The first module consists of components that perform reasoning about users’ actions and users’ preferences. The second module consists of user interface components.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

More specifically, the first module consists of the following components:

Explicit Information User Profiles,

Monitoring Agent,

Clustering Algorithm Process,

Double Stereotypes and

User Model Server

The second module consists of user interface components:

The Incremental Initialization,

Recommendations,

User Interface,

Animated Agent and

Adaptive hypermedia

The PERCOM architecture creates user models based on the k-means clustering algorithm and then uses these user models to create personalised recommendations. PERCOM is sufficiently generalised so that it may be used for any kind of e-commerce application. In fact we have evaluated the generality by incorporating it into two different kinds of applications. The first application is an e-commerce video store where e-shoppers receive personalised recommendations on what movie to buy, called Vision.Com. The second application is an interactive TV-commerce store that sells mobile phones to users called iTVMobi. Again, iTV-shoppers receive personalised recommendations of what products to buy. From these case studies and their empirical evaluations emerges the fact that PERCOM can be incorporated in any e-shop application making it adaptive to its customers' needs.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

5.1 Overview of The Recommendations Architecture

First of all, the definition of the problem, that PERCOM architecture solves, is required in order to define where PERCOM architecture applies to. PERCOM framework addresses the problem of generic e-shop applications and unsatisfied customers due to the lack of understanding customers' needs and inferences. These needs and inferences are usually hidden behind users' behaviour. In this way, for an e-shop application to exploit customer data and extract valuable information from it is imperative.

Exploiting and extracting this information can be done in two ways, either explicitly or implicitly (Rich 1979). The first way involves direct contact with customers-users and the second is a passive observation of customers' behaviour. Taking in mind all the above, PERCOM aims to solve the above problem and give the ability to e-shop applications to be adaptive according to customers' needs or inferences. PERCOM applies to every e-shop application regardless of product or medium and to every user group, regardless age. This fact is also reinforced by the three case studies presented in the next chapters, which involve two different products and three different media. In order for PERCOM architecture to be able to react effectively to the problem mentioned above it incorporates several functionalities and technologies. This means the PERCOM is mainly divided into two groups (functionalities and technologies) concerning the general structure of the architecture.

The functionalities that PERCOM includes are: 1. Adaptive Presentation, 2. Personal Recommendations and 3. Personal Shopping Cart.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Adaptive Presentation functionality helps the system present recommendations, products, product attributes or user interface elements in an adaptive way according to the users' needs. This functionality can annotate products or attributes with special symbols, change the fonts of product names or attributes, change the size and position of user interface and completely hide user interface elements that are considered of less importance. For example, if user has great interest in products that have a certain number of features then the system can propose similar products with different suggestion degrees. The degrees are visualised in the form of different annotation symbols (figure 2).

Figure 2 Product annotation and recommendation example

On the other hand, if a user faces problems navigating through different products categories PERCOM can change position and size of the categories elements (figure 3).

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

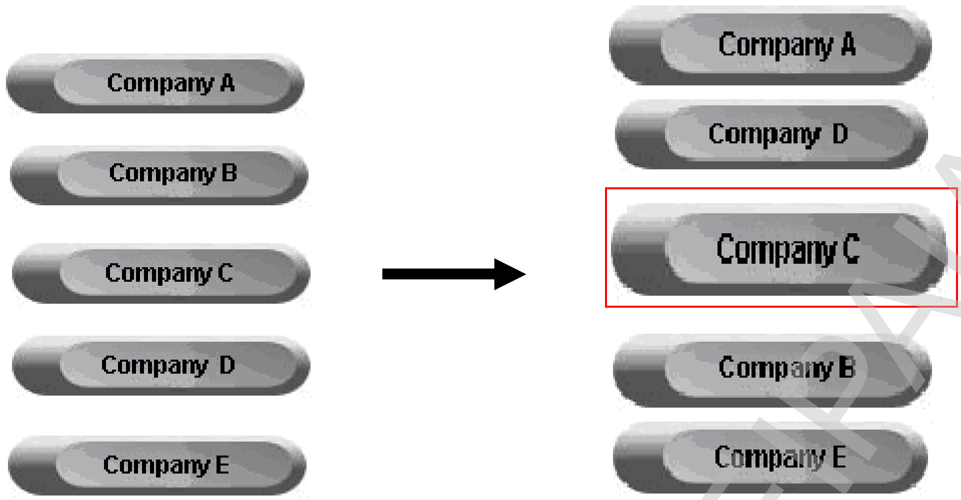


Figure 3 On the left original size and position of categories button and on the right changed size and position of the mistaken button.

Personalised Recommendations functionality is responsible for creating and modifying personalised recommendations according to customers' tastes (figure 1). This functionality is based on clustering similar groups of users with the help of kmeans algorithm. There are two types of groups created, one with the basis of similar tastes and one with the basis of similar mistakes. These groups of similar users are used by PERCOM as the basis creating representatives. These representatives are then used by PERCOM to measure products and how close these products are from users' interests.

The main difference between Adaptive Presentation and Personal Recommendations is that the first functionality controls the User Interface and gives feedback to second. The User Interface is also responsible of changing dynamically the presentation of the application, according to users' mistakes (figure 2). On the other hand, Personalised Recommendations provide with interest information that Adaptive Presentation, in order for the last to be able to change the user interface in an adaptive way. The procedure of clustering similar groups and extracting representatives will be

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

explained in more detail in the next section. The Shopping Cart functionality involves a personalised shopping cart for every customer, which can be used for moving and buying products.

The Shopping Cart communicates with the PERCOM architecture functionalities and provides data extracted implicitly from the customer's behaviour. For example, products that are inserted into the Shopping Cart rise interest degrees in these specific attributes that moved products have. Bought products from the Shopping Cart rise the specific customer interest degrees even more. If a product is removed from the Shopping Cart the degree is lowered by PERCOM. The quantity of the product that is moved in the Shopping Cart by the customer plays a big role in the interest degree also.

The adaptive technologies used by PERCOM are: 1.User Modelling, 2.Machine Learning and 3.Stereotypes. The User Modelling technology is used by PERCOM in order to create user models based on customer behaviour, both on products and mistakes. The data considered by PERCOM for this behaviour are both explicit and implicit, but PERCOM gives emphasis on implicit data. Explicit information from customers is also considered as valuable information but mainly for new users. As the customer interacts more and more with the application PERCOM discards explicit information given by this specific customer and focuses more in implicit taking by his/her behaviour. The explicit information taken by PERCOM includes demographic, educational and general interest information. The implicit information considered by PERCOM is all users' interaction moves with the application. These moves include visiting pages, clicking on products or categories, clicking on product attributes, moving products to cart and removing products from cart, buying products and searching for products.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

In order for user models and double stereotypes to be create PERCOM needs to incorporate procedures that can extract and exploit customer data from the user models and through certain processes these must result in meaningful information about customer tastes, mistakes and similarities.

The Machine Learning technology used in PERCOM takes this role exactly. This functionality takes as input data from the user models, then processes these data and results in groups of similar users. The machine learning technology used by PERCOM is clustering. The clustering algorithms used in PERCOM were five; k-means, hierarchical, fuzzy c-means, spectral and AIN. The procedure is created in an abstract way and in this way it can generalised in order to be used by any clustering algorithm. The results of the Machine Learning processes can be used for two reasons. Firstly, in the Recommendations functionality as we previously mentioned and secondly, in the creation of double stereotypes.

The third adaptive technology that PERCOM uses is Stereotypes. In PERCOM Stereotypes concern two application domains, products and user needs. In this way PERCOM combines stereotypes from both products and users. The main drawback of Stereotypes is that they are static, because they contain static information. In order to surpass this problem, we created a dynamic hierarchy of stereotypes. This means that Stereotypes and the stereotype hierarchy in PERCOM are created dynamically when the user enters the system. The hierarchy of the Stereotypes follows an increasing complexity philosophy, meaning that top level stereotypes are more general then the ones of the lower levels. The main usage of stereotypes in PERCOM is for new users or users with little knowledge about their needs and tastes. Double stereotypes used in PERCOM will be explained more thoroughly in the next section.

In the next section we will present in more detail, all the components used by PERCOM and how these components follow the main structure presented here.

5.2 The PERCOM architecture

The architecture of PERCOM architecture consists of two main modules (figure 3). These modules are in fact two separate categories of modules that are divided by their interaction with the user. The first module (figure 4) entitled User Modelling Components interacts in a more passive way with the user. This does module does not intervene with how the customer sees and uses the application, instead it mainly focuses on gathering, saving and processing information about customers in an implicit way.

On the other hand the second module entitled Adaptive User Interface Components plays the role of interaction with the user. This module is more aggressive and it can interfere with the customer interaction in order to suggest or help the customer. The first module consists of components that perform reasoning about users' actions and users' preferences and mistakes and includes four components. More specifically the first module consists of the following components:

1. Explicit Information User Profiles,
2. Monitoring Agent,
3. Clustering Algorithm Process
4. Double Stereotypes.

The second module consists of user interface components and includes five components, entitled:

1. Incremental Initialization Process,
2. Recommendations,
3. User Interface,

4. Animated Agent
5. Adaptive Hypermedia.

Both modules can communicate through the User Model Server component, which is mainly used as a service between these two modules.

The User Model Server (figure 4) component contains all the information, implicit and explicit, about a specific user, the stereotype that s/he belongs to and his/her more similar representatives and manages all the above components. The User Model also contains all users' models and manages them in order to give the right information to the components of the second module. The modules can acquire user and product information from the User Model Server and also give appropriate feedback to the Sever. For example, the User Modelling Components module can acquire information containing previous groups' representatives from the User Model Server but also give feedback to the User Model Server if a user has become member to a different stereotype.

The Adaptive User Interface components can acquire information about tastes in order alter the personalised recommendations for a specific user but also give feedback about user's reaction to the products suggested. The Adaptive User Interface can also give feedback about user's mistakes in navigation or interaction with the system. In this way, the User Model acts as a manager of both modules and as controller of the information flow. Users can interact with PERCOM from the Adaptive User Interface components as figure 1 illustrates. Users can interact using any medium (mobile phone, pc, tv) as the data flow is controlled by the User Model Server.

The Adaptive User Interface Components' behaviour can be altered by the User Model Server according to the medium being used. Due to the fact the User Model Server controls the data flow and all the information about every user's model, this component

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

can modify and distribute information and user model data in a cross-platform way. For example, a user can use an e-shop incorporating PERCOM and then ask for product suggestion is his/her mobile phone. The User Model Server can use the acquired information from the User Modelling Components of the e-shop and send the requested feedback the Adaptive User Interface Components on the mobile phone.

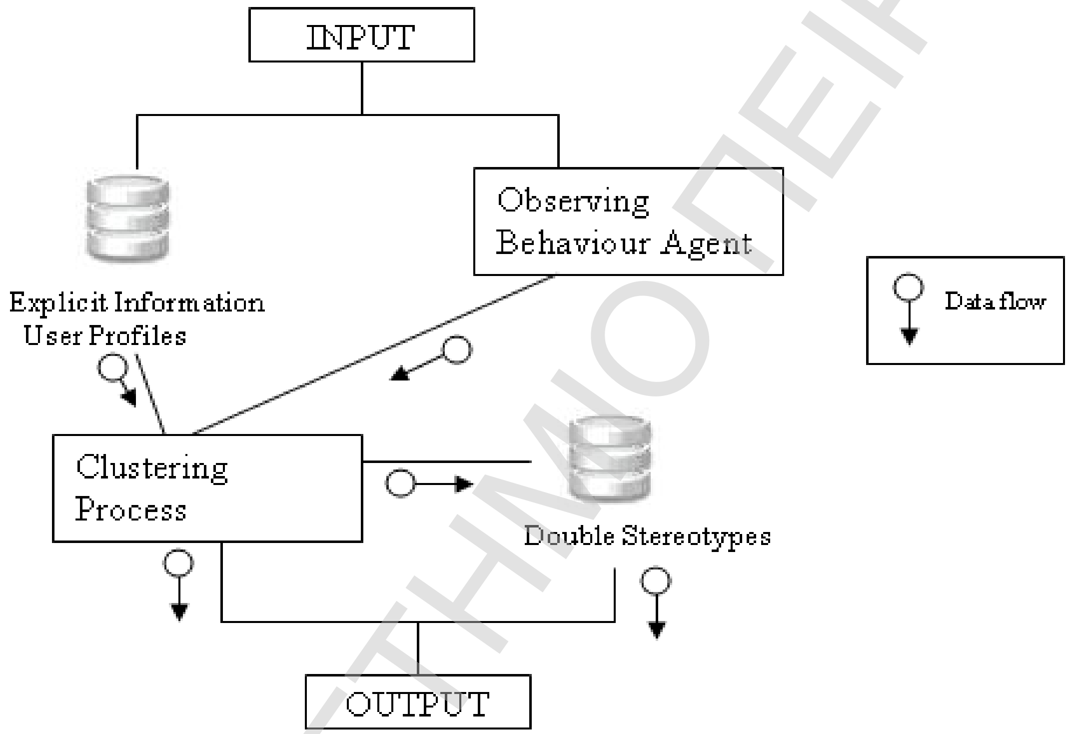


Figure 4 User Modelling Components module diagram

The User Modelling Components play a very significant role in the creation and manipulation of user models in PERCOM. As previously mentioned PERCOM's User Modelling Components module includes four components (figure 4). The diagram in figure 4 shows the components that participate in this module, and also how these components interact with each other. The diagram also illustrates how that data flows through these components and where exactly are the inputs and outputs for this module. The input data in this module come from two separate ways. The explicit input

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

data are taken in a form answering questions about demographic data, educational data, product ratings concerning certain product attributes and not specific products and user interface preferences questions. PERCOM architecture includes demographic data to the explicit information because users' birth place, living place and conditions and family condition can play a big role in the user's opinion about product purchase (reference) and their interaction with the application. The implicit input data are taken by observing user interaction.

The system measures every user's action while s/he interacts with the application. The Explicit Information User Profiles component contains all the information in a database that users have provided to the system in an explicit way, either, by answering interview questions or rating products. Every time a new user is registered in the application that incorporates PERCOM, this component collects this information from demographic data, educational data and interest data that the user provides through the registration process. The user's answers in the registration process are combined and take the form of statistical degrees in order to be processed by the Explicit Information User Profiles. Then this explicit data is processed in the component and the process results in valuable information concerning interest and mistake degrees. For example, users with higher education that live in big cities, tend to buy expensive products with top technology functionalities. Furthermore, users with scientific studies tend to interact more easily with computers than users with literature studies. In this way this component can exploit this information and fill this customer profile with degrees concerning certain attributes concerning high end technology.

The next component is Monitoring Agent. This component plays a key role in the construction of the User Model and contains all the information about the user interactions with the system.

Moreover, monitors users’ actions throughout the usage of the system. For example, this component contains information about the categories of products that a user has visited, specific products that s/he visited, products that s/he moved in or out of his cart, products that s/he bought, mistakes done and navigation patterns. In this way this component, is product independent because it monitors user behaviour and not products characteristics or specific GUI (graphical user interface) system elements. The Monitoring Agent component also contains a statistics database of all users’ actions and features collected by the systems.

The Clustering Process Algorithm component contains the clustering algorithm that the system uses to group similar users and extract representative users of these groups (figure 5). The clustering algorithm takes as input the statistical data of all the explicit and implicit information that the system infers and collects about users. This data is saved on the statistics database of the Monitoring Agent. This data include two types of information. Data concerning the visits in products specific pages, products general categories, search queries made by the user, products moved to the shopping cart and products eventually bought. Data concerning mistakes, repeated clicks, navigational patterns and accidently made moves.

The Clustering Process in PERCOM follows three major steps:

At first the algorithm takes as input the data from the User Data Space. This data is converted to vectors of characteristics based on the behaviour of every customer mentioned above. Every characteristic in the input vector corresponds to an interest percentage in a product attribute or to a mistake of a user.

The next step for the clustering algorithm is to create groups of similar vectors based on the Euclidian distance. These are created in real time as the customer navigates through the system. These groups are saved in the database for later use in the Double Stereotypes. Inside every group the algorithm creates mean vectors that are centres of the group (figure 6). These centres do not represent existing customers but are newly created vectors of characteristics. The number of group centres on every group is not static and varies depending on the group's member number and scattering.

At the last step of the Clustering Algorithm process representatives of groups are calculated. The representatives are vectors extracted from the groups centres created by the clustering algorithm. In order to extract a group representative the algorithm combines the group centres, measures their relative distances and then based on this data calculates an average vector. The three components mentioned above along with the Double Stereotypes component contribute to the construction of the User Model Server component.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

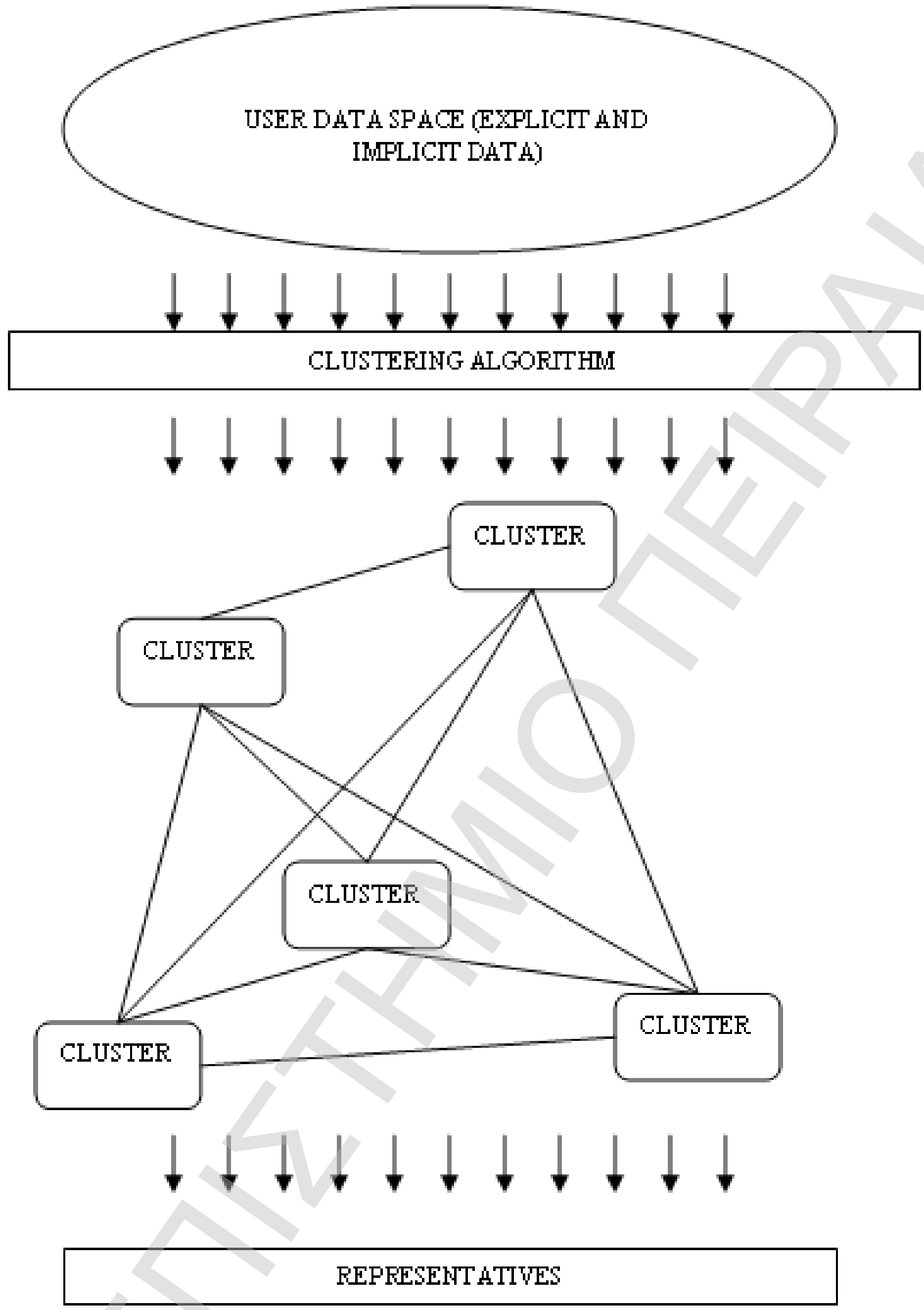


Figure 5 Diagram of the Clustering Process in PERCOM

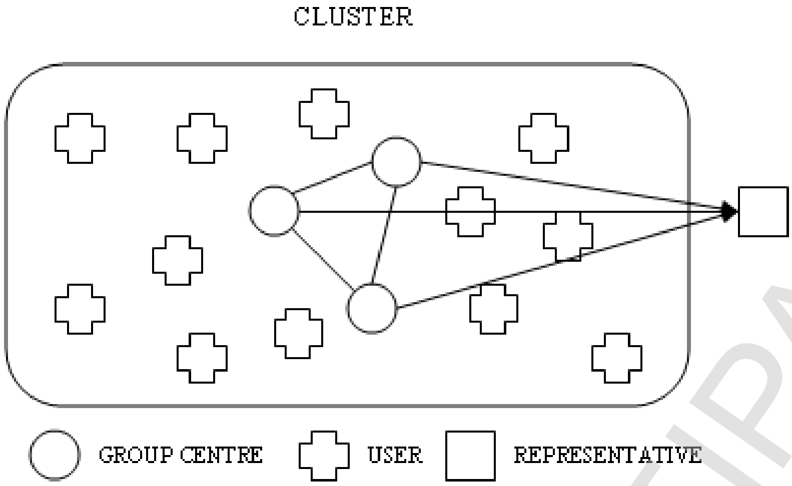


Figure 6 Inner cluster example diagram of users and group centres.

5.3 PERCOM Interest Degrees and User Models

Every category and product characteristic is a calculated by the algorithm. However, this calculation is not based on degrees from products classification; instead the calculation is based on user behaviour while s/he interacts with the system. Also, every mistake and navigation pattern is calculated by the algorithm. The data input for the algorithm are calculated based on degrees from the features measured.

An example equation for the interest degrees can be seen below. This equation measures interest degree in product category. Weights change, depending on the application that this framework is incorporated. However, general rules on weights apply on all applications that use this framework. For example, the weight concerning visits on a specific product category is always smaller then the corresponding weight of products moved to the cart. The first weight is smaller, because visiting a product may not mean that the user is necessarily interested in this product but that s/he is just browsing several products. The second set of the example equations are referring to mistake degrees.

Interest Degrees Concerning Product Categories

$DegreeOfInterestInCategory_a = \frac{VisitsInProductBelongToCategory}{VisitsInAllProducts}$	(1)
$DegreeOfInterestInCategory_b = \frac{ProductsPlacedInBasketBelongToCategory}{All\ ProductsPlacedBasket}$	(2)
$DegreeOfInterestInCategory_c = \frac{ProductsBoughtBelongToCategory}{AllBought\ Products}$	(3)
$DegreeOfInterestInCompany = W_a * DegreeOfInterestInCategory_a + W_b * DegreeOfInterestInCategory_b + W_C * DegreeOfInterestInCategory_c$	(4)

Mistake Degrees Concerning Sight Difficulties

$HardtoSeeCateory = \frac{MistakesInCatecoeryElement}{TimesInCompaniesPage}$	(5)
$HardtoSee\ ProductIcons = \frac{MistakesIn\ ProductIcon}{TimesInPhonePages}$	(6)
$HardToSee = \frac{w_a * HardToSeeCategoryElement + w_b * HardToSeeProductIcons}{AverageHardToSeeFromAllUsers}$	(7)

The equations above can be specialised to every product, or to different kinds of mistakes with the application that incorporates PERCOM. In this way PERCOM framework remains product independent. The clustering algorithm processes these degrees and provides the system with groups based on similarity. From these groups representative feature vectors are extracted. The representative vectors work as group leaders and show the groups tendency to specific product features. These vectors are then compared with the vectors of products' characteristics and the closest vectors are extracted and saved to the Recommendations Database component. In this way the Recommendation Database

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

component is updated dynamically as users navigate through the application.

After this procedure the Double Stereotypes component calculates dynamic double stereotypes from these representatives. These stereotypes follow a general to specific hierarchy, meaning that the system constructs a low number of generic stereotypes at first and then continues to construct more specific stereotypes until it reaches a certain point of complexity. The construction of Double Stereotypes refers to both products and users (figure 6). For the construction of products, product attributes are taken as input data and converted to vectors. A simple rule in these vectors is applied; if a products has an attribute then the percentage for this attribute takes the value of one, otherwise takes the value of zero. In this way for every product a vector of zeros and ones is created. Then these vectors are fed into clustering algorithm and groups of similar products are created.

In this step we have two types of groups, users and products. Customer groups and product groups are connected through Euclidian distance measures. In order for the Double Stereotypes to be created we first choose a small number of characteristics from the vectors. This number refers to main categories of the product sold by the application that incorporates PERCOM. Based in this small number of characteristics and the groups of similar vectors we create centres of groups. The previous centres create the first level Double Stereotypes that refer only to the major product categories or major mistake categories. This process continues by adding more characteristics to each group centre creation, thus making the group centre vector larger and more complex. In this way every stereotypes of next levels are more complex than the previous level stereotypes.

The process ends when the distance between two centre groups is lower then a certain value, which is defined by PERCOM.

Connecting products and customers through distance measures in the stereotypes process gives PERCOM the ability to directly connect products with users and perform more general recommendations quicker.

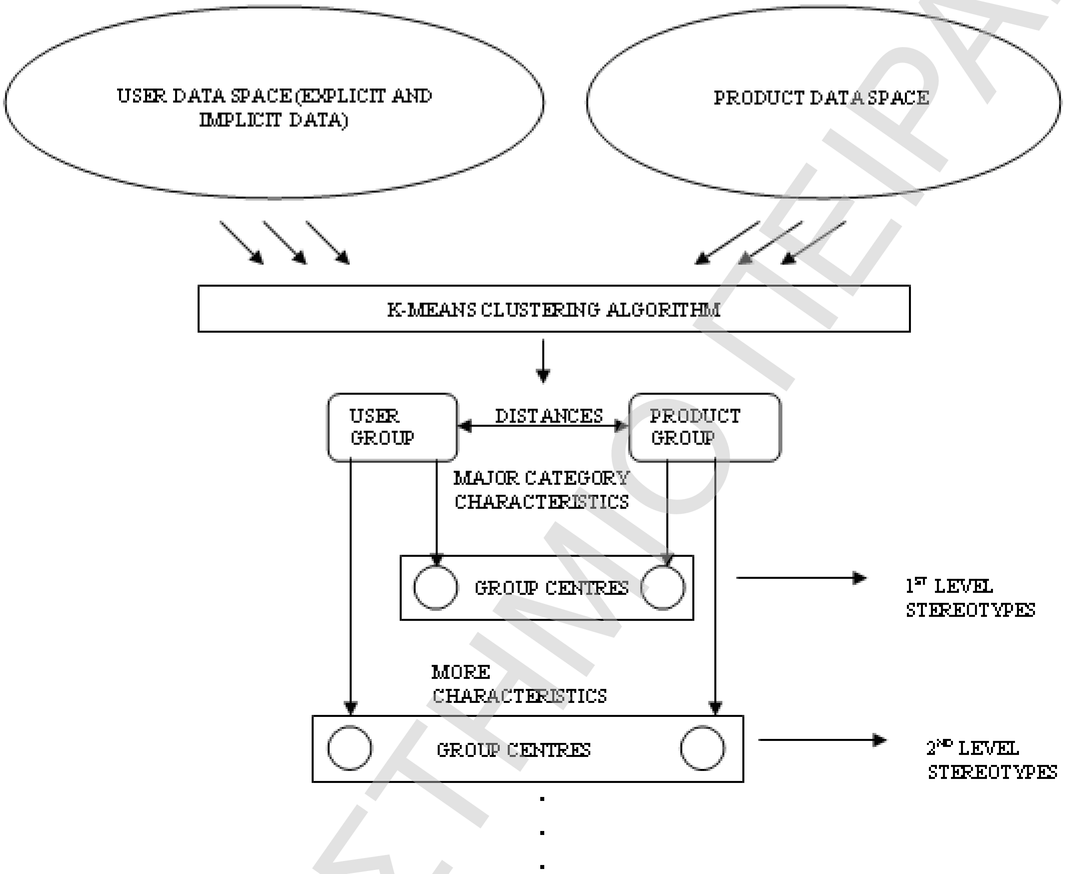


Figure 6 Double Stereotypes creation and hierarchy

5.4 Personalising the interaction based on the user models

The second module (figure 7) of PERCOM is responsible of personalising the user interaction with the application. The Incremental Initialization Process component acquires information from the user model and tries to provide the best recommendations

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

and help to new users or users that the system has little information about. This component uses mostly stereotypic information from the Double Stereotypes component and chooses where the user should belong according to the moves that s/he has made so far. The Recommendation component communicates with the User Model component and provides the users with recommendation about products and system usage.

The Recommendation component contains all system recommendations about products, mistakes or other recommendations in a database. This component can use many techniques in order to make recommendations such as adaptive hypermedia, dynamic annotations and animated agent. The Recommendation component also takes feedback from the users and provides the User Model with more useful information. The next component is the Animated Agent. It is a system component that manages an animated agent that can help users throughout the navigation of the system. This agent can provide useful information about the usage of the system and provide recommendations about products by acquiring information from the user model.

Next, we have the User Interface component. We use a dynamic user interface that not only adjusts to the medium used automatically, but also changes according to the users' interests and needs. The User Interface component can change the whole user interface appearance dynamically and personalise the user interface according to the specific users' preferences and needs. These changes are acquired from the User Model Server that contains all the user models. Because the User Interface is an entirely separate component it can adapt on any medium thus making PERCOM medium independent. These two functions of the User Interface component create a unique personalised experience for every specific user, resulting in a friendlier and more efficient user interface. Also, the Recommendation and Incremental

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Initialization component can change the User Interface according to the User Model of every user. We must note here that despite the fact that user interface experience is different for every medium, it contains its basic characteristics in order to avoid user confusion between media.

Last but not least, we have the Adaptive Hypermedia component that is used to change user interface elements according to the users' needs or preferences. This component has the ability to change the user interface according to the recommendations extracted from the user model. This component can annotate products that are recommended and change the position of these products in order to be seen first. It can also, change the symbols of recommended products depending on the degree of the recommendation. Furthermore, it can change position and size of products or other interface elements that confuse the user according to the shown behaviour. For example, if a product is highly recommended by the framework then this component places a different symbol next to this product's name. On the other hand, if this product is recommended on a low degree then a different symbol is placed by this component. Moreover, if a user tends to confuse two neighbouring products the system swap their position with other products thus creating a clearer user interface.

Adaptive hypermedia can also change the font of product names or features in order to get user's attention. For example, if a user is greatly interest in a specific product feature then in adaptive hypermedia will change the font of this feature in the product page, thus making this feature more clearly to the user. On the other hand, if a user confuses products' features it will change the font between these two features resulting in an easier discrimination of these features. All these components communicate with each other using the User Model Server as passage from one to another. In this way these components can be independent and easily changeable.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Meaning that if a developer wants to change a component of the architecture can easily do it without changing the whole architecture from scratch. The only limitation is that the developer must preserve the input and output forms of the architecture components.

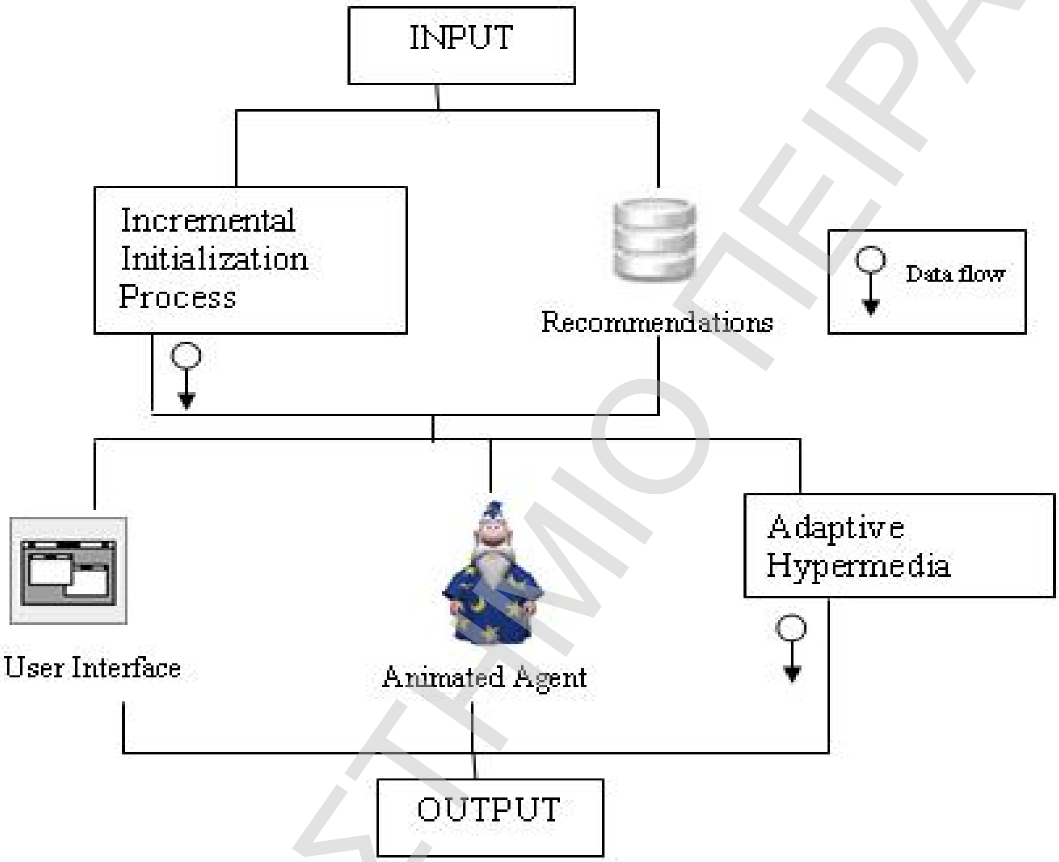


Figure 7 Diagram of the second module of PERCOM

5.5 Conclusions about the generic architecture

Over the last decade the market's demand for customer-individual, configurable products at costs of mass-production has been constantly increasing and a mass-customization (Pine et al.,1993; Piller and Schaller, 2002) business strategy has been adopted in many industrial sectors (Ardissono et al., 2003). User modelling techniques can be a solution to today's market demand but as the complexity of products and the variety of domain and

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

media raises research prototype systems and closed commercial systems can not address all these problems. Moreover, nowadays an adaptive system must target several factors in order to achieve successful personalization. These factors can be classified in to three major sections, information about the user, information about the device used to interact with the system and information about the context of use (Goy et al., 2003).

Information about the user involves user's knowledge about the domain, user's interest and needs and of course user's goals. Information about the device used involves the actual hardware used that a user may use to interact with the system, e.g. pda, pc. The information about the context contains physical context and social context. Physical context refers to the environment of the user such as location, noise, temperature and the social refers to features like social community or groups. Many systems have integrated user modelling and adaptivity in their reasoning mechanisms (Kazienko P., Kolodziejski P. 2005; Muller et al., 2002; Guan et al., 2005; Maybury et al., 2004).

The above systems have shown significant results in achieveing personalization but failed to address many of the factors mentioned above. PERCOM's approach based on different modules can be a more generalized solution that can address a lot these factors. As adrissono points out in her work (ardissono et al 2007) a modular approach has several advantages. First of all, the creation stereotypes and user modelling can be simplified. Secondly, modularity allows the re-use of used modelling knowledge bases through the use of a single user model server. More specifically, in PERCOM the different generalized modules can be used in different commerce systems and re-used later in others giving in this way the ability to the later systems to use knowledge bases from the previous ones. Furthermore, the information about the user factor can be easily targeted by PERCOM three different modules, the

explicit profile, the dynamic stereotypes and the user behaviour agent.

On the factor of information about the device PERCOM presents a novel approach of the entirely separate user interface and adaptive hypermedia modules. In this way PERCOM can adapt its user interface and interaction to any device used by the user. On the context factor PERCOM target the social content and not environment data. On the social context PERCOM intergrates two ways of receiving information. Firstly, the new user in the registration process provides social information about education, life style and demographic data. Through this information a system can extract information about user's tendencies towards products. Secondly, PERCOM specializes its' user interaction into special groups like the elderly. In this way the interaction is dynamic and can be altered in order to help people with special needs. PERCOM's module and generalization approach can be a very effective solution in creating adaptive commerce systems that share user model knowledge bases but differ in domain and media.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

CHAPTER 6
EVALUATING THE METHODOLOGIES IN REMOTE SHOPPING SYSTEMS

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

6 Evaluating the Methodologies in Remote Shopping Systems

The techniques that many intelligent programs use in order to export information include analysis of protocols, observations, interviews, case analysis and questionnaires (Bell & Hardiman, 1989). However the result often is a "pile of useless data" that leads to knowledge poorly structured, inflexible and semifinished (Thurman, Braun, and Mitchell, 1997). Consequently, conducting empiric studies and presenting their results can, undeniably, facilitate and promote the future inquiring efforts in the field of human interaction systems. According to Chin (2001), the key to good empirical evaluation is the proper design and execution of the experiments so that the independent variables can be easily separated from other depended ones.

Empirical study is a process that follows the construction of a system and includes the conduction of an experiment that real users participate in order to evaluate the results that an e-commerce system gives to them. The empirical study helps scientists to find out whether their system gives meaningful and helpful results about users' behavior.

Relative studies have been conducted also by Jerrams-Smith (2000) and Horvitz (Horvitz et al. 1998). Horvitz held studies "Wizard of Oz" type, at which experts watched real users interacting with Microsoft Excel and provided them direct help. The users however did not know were the help was emanated from. This study revealed certain categories of actions that can be used for reaching conclusions concerning problems that users face and when they need help. A lot of empirical studies have been conducted in the field of adaptive e-commerce systems. One example is the work of Micarelli and Sciarrone (2004) about the anatomy and empirical evaluation of an adaptive web-based information filtering system.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Another work is the exploration/exploitation in adaptive recommender systems (Ten Hagen et al., 2004) that uses reinforcement learning for recommender systems and conducts a simulation experiment in order to prove the need of exploration in recommended systems. Moreover, Branting (2001) shows that web-based applications can improve their performance by customizing their performance to customer preferences.

In view of the above, we have conducted three different empirical studies in order to study and compare our results with real users' behaviour through the use of three different media. We followed similar experiments in all three media in order to have a more concrete opinion in the end about the results.

6.1 The Experiment

In accordance to Chin's suggestions (2001) about the settings of an empirical study we have conducted the following experiment: The application was used by approximately 150 users. These users were divided into three groups and the experiment was conducted into three different time zones. Every user was isolated at a high percentage and the computer area was pre prepared for the experiment. We used a local network and avoided high load times. The instructions given by the experimenters were uniform and consisted of simple phrases to help the participant understand thoroughly what they were going to do. Every user used Vision.Com for about fifteen minutes. This duration gave them sufficient time to understand and use the system effectively.

The experimentation for each user was consisted of three phases. First phase: The subject was asked to visit product-pages and select the ones to put into his/her cart. Second phase: The subject was asked to re-evaluate his/her selection. At this phase the subject would take products of his/her cart or add new ones. Third phase: The subject was asked to buy the products s/he wanted. After having completed the experiment users were asked to share their opinion about the system, the recommendations that it had made. Moreover, users were asked to give their suggestions about improving the system.

Many found the web system very easy to handle with and about the animated agent that acted as a salesman they had a very positive opinion. As for the recommendations a significant amount of users found them satisfactory.

6.2 Evaluation in E-Shopping

In this section we present results of the empirical study in the field of e-shopping. The results correspond to the three phases of the experiment. The experiment was conducted twice. The first time we conducted the experiment without the incorporation of a clustering algorithm and the second with the clustering algorithm incorporated.

6.2.1 Results before the Incorporation of a Clustering Algorithm

Five different bar-diagrams were produced in order show the differences in customer interest between the five different ranges of prices. Here we present on one them for economy reasons but we discuss them all. The bar-diagram (Figure 1) presented shows the interest of customers in high priced movies.

In the diagram (figure 1) we witness that users tend to have low statistics in this range of prices. This means that customers tend to buy cheap movies. In contrast to the above diagram, the very low

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

and low prices interest percentage of movies is higher than in movies with medium and high prices.

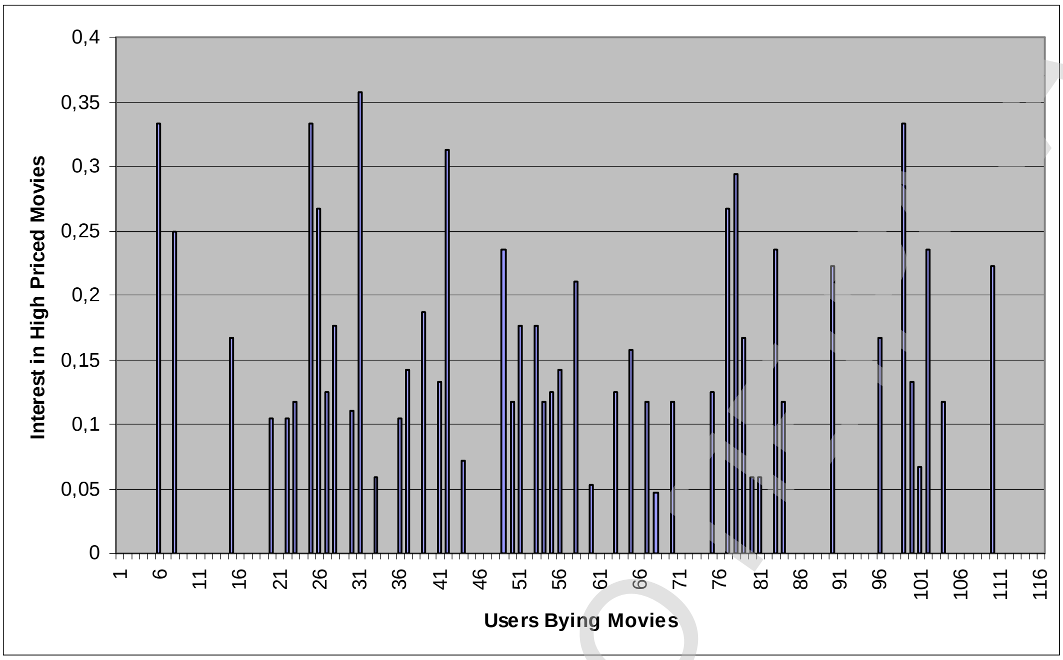


Figure 1. Individual interest of users in movies that belong to high price range.

A high degree of interest can be also witnessed in the very high priced movies. More specifically the participation of users in the diagrams of very low, low priced movies is very high instead of high priced movies. This can be easily seen in the above diagrams as the number of bars is very small. This leads us to the conclusion that almost every user bought at least one low priced movie of every category. Moreover the interest of these users about these movies is mane times close to 70 %. A logical explanation might be that every time a customer goes to a video store, he prefers to buy a cheap movie than a medium priced one, regardless the quality between them. As for the high degree of interest in very high priced movies we can come to the following conclusions. We chose to price some movies very high because of their quality and success at theaters.

In this particular diagram we can monitor that when it comes to a movie that has made a success or it's a classic one, users tend to

buy it because of its quality. This result came from the fact, we earlier reported, that we chose to put high price in the movies that were classics or had a big success in the cinemas. Moreover, the participation of the users rises against the high and medium priced movies and the interest degree many times rises up to 50 %. The above explanations are also reinforced by the fact that both medium and high priced movies have low participation numbers by users and the interest degree is as low as 40%. Furthermore, users that bought high priced movies are in such a low participation, as shown in Figure 1, they can be consider as non significant to the experiment.

We will also present two pie diagrams (Figure 2 and 3) that show the amount of interest in every category and in every price range. At the first diagram (Figure 2) we used the statistical equation of average deviance in order to get our results. We chose this function because it gave more clear results about users' behaviour. At the second diagram (Figure 3) we used the same function for the same reason.

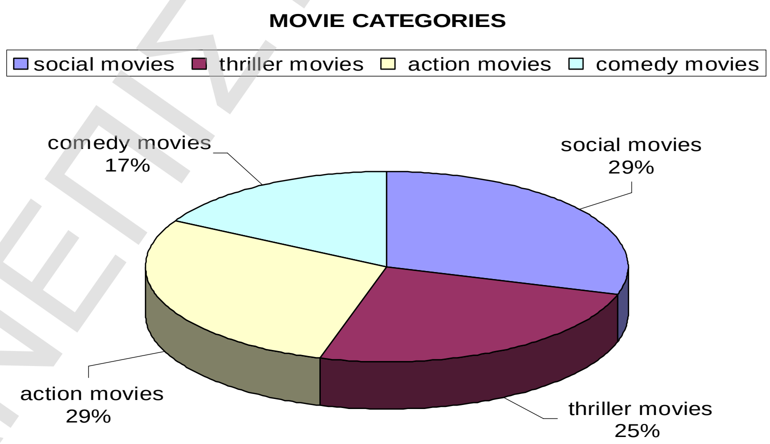


Figure 2. Pie diagram of users' interest in movie categories

In Figure 2 we see that users tend have a lot of interest in social and action movies instead of thriller movies and comedy movies. The diagram clearly presents that only taking action and social movies we get more than 55% of the whole interest of the customers. As shown in Figure 3 users prefer cheaper movies and movies with success in theaters (explained just above) instead of movies with a higher price but not so much quality. The percentage in very low priced movies is the largest one and combined with the one of low priced movies fills almost the 50% of the users' interests. If we add the percentage of the highest priced movies (movies with great success in theaters) we can see that the amount exceeds 65% of the whole interest.

By the collaboration of these two different results, we come to the conclusion that a great amount of customers in a video store chooses to watch cheap action and social movies. As far as it is concerned about the low amount in interest in comedy movies we must take in mind that a large amount of these movies targets in a group of users in smaller age that may not have the amount of money that users from other groups do.

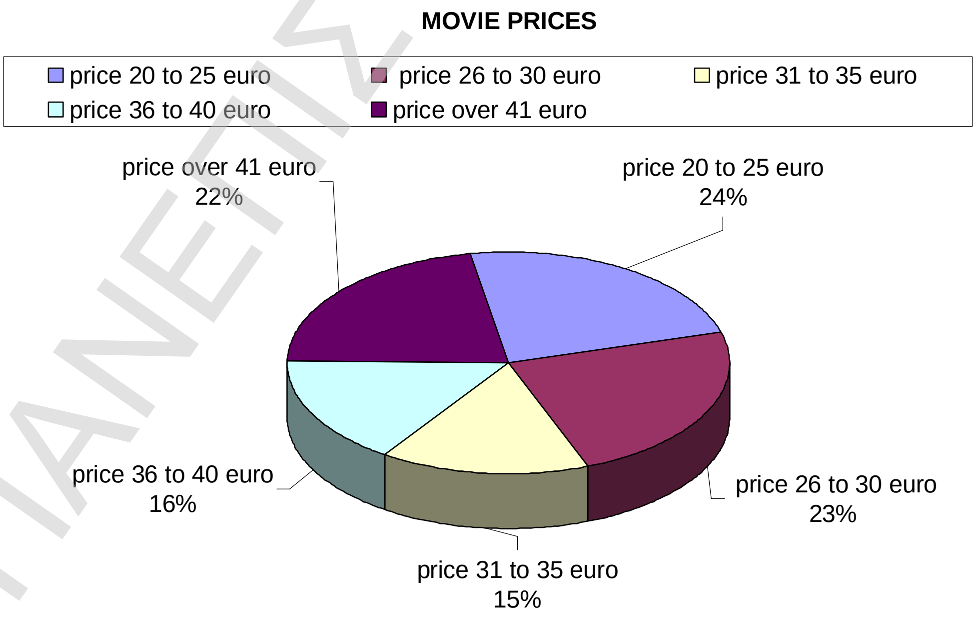


Figure 3 Pie diagram of users' interest in movies ranges of prices

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

6.2.2 Results After The Clustering Algorithm Incorporation

A diagram comparing users' opinion, vision.com original recommendations and vision.com recommendation with the help of AIN clustering algorithm can be seen in figure 4. This diagram (figure 4) clearly shows that Vision.Com recommending ability has improved after using AIN clustering. The recommendation stats are closer than those before using AIN. Diagram in figure 5 shows a comparison between users' opinion and vision.com's predictions in four main categories. Again we witness that vision.com predictions a similar pattern to those of real customer opinions.

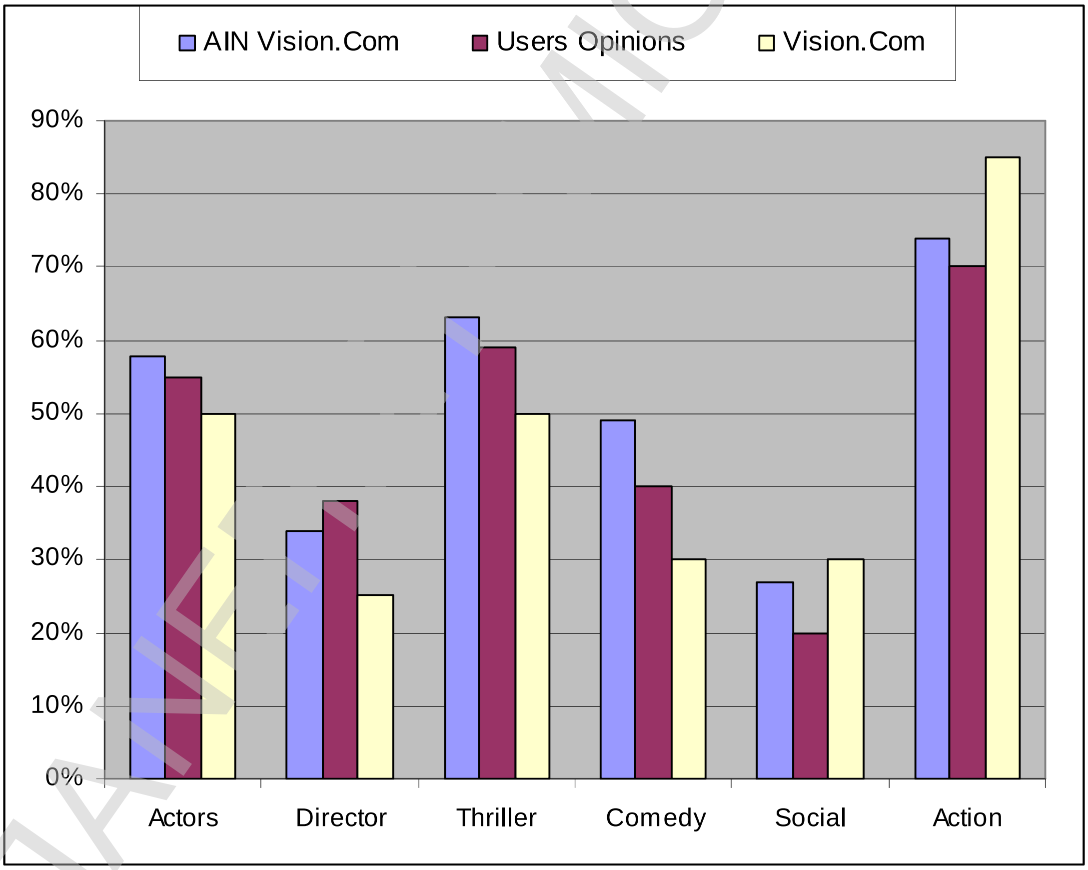


Figure 4 Diagram Shows Comparison between Evaluation Results

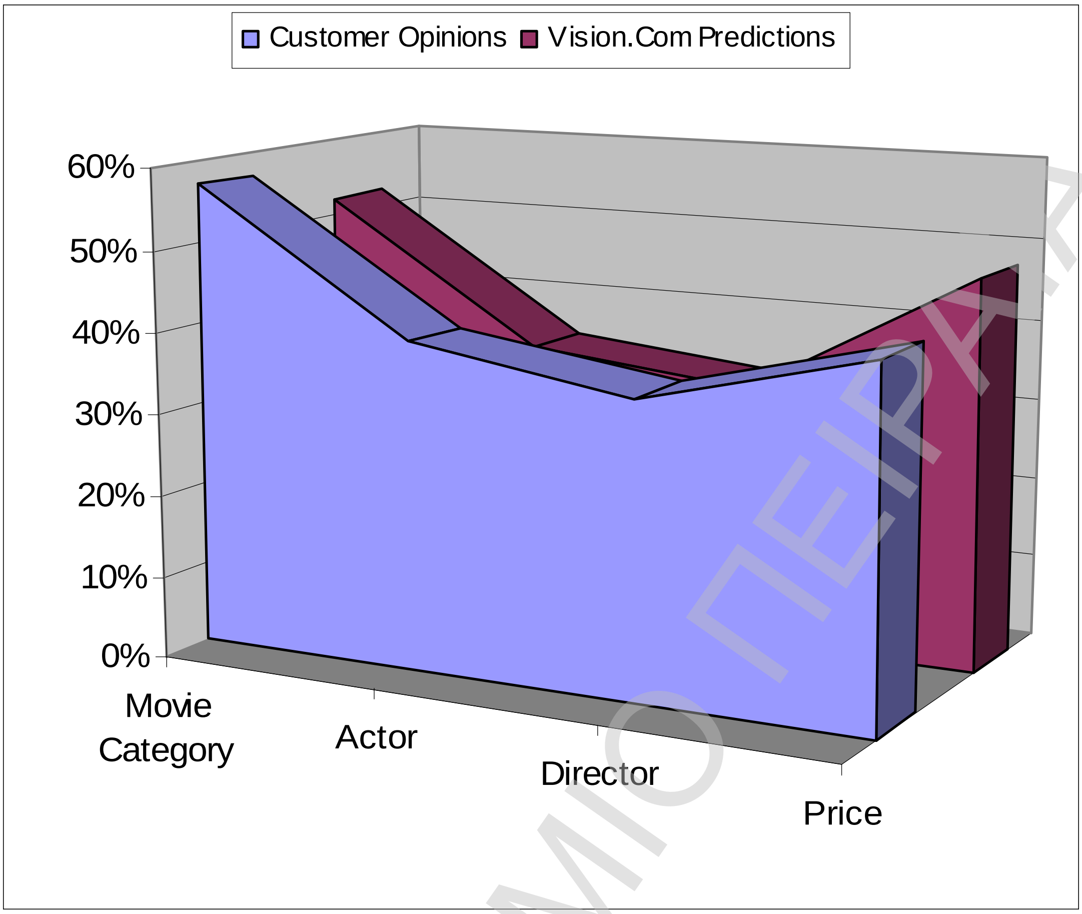


Figure 5 Diagram Shows Comparison between Evaluation Results in four mesaured categories

6.3 Evaluation In TV-Shopping

In order to evaluate iTVMobi we asked 50 users, men and women, to use the system similarly to e-shopping experiment and then answer a 16-question questionnaire in order to compare the system results with the answers given to the questionnaire. The group of 50 users consisted of 25 men and 25 women. 20 people of the group had considerable knowledge in computer use and the rest had very little or no knowledge on computers. The questionnaire had three sections of questions. The first section consisted of three questions:

Question 1. Are you familiar with computers?

Question 2. On which technology would you prefer to use an

application that will help you buy mobile phones?

Question 3. Did you find easy to use iTVMobi?

The results of the second question showed that 80% of the participants found it easy to use the system and 10% found it very easy to use it in a scale of 1 to 5, with 1 to mean “not easy at all” and 5 “very easy”.

The second section of questions concerned tastes in phones. More specifically tastes in phone technical features, battery, memory, display etc. These questions correspond to the phones’ features measured by the system by observing the users’ navigational moves. We compared the answers of every user in the questionnaire with the percentages of interest of the representative of the group that this user belonged to. The results were satisfactory as the system was able to reflect 81% of the users’ tastes with the use of their representative of the group that the user model created.

In Figure 6 we show interest percentages of users of iTVMobi in some major phone features like connectivity and size. The diagram in Figure 6 is a comparison diagram between users’ interests extracted from the questionnaire and those generated automatically by iTVMobi. This diagram shows that iTVMobi is very successful in predicting users’ preferences. More specifically in mobile phone features, display and battery, we can observe that the successful prediction level is very high.

On the feature of “phone size” there is a higher difference between user opinions and system predictions. This can be explained by the fact that users may respond in a different to questions about the size of a mobile phone without their having seen the actual mobile phones (as in the case of the questionnaire). On the other hand, when the user sees the phone through a picture (as in the case of iTVMobi) then phone size becomes clearer to the user who responds differently to this phone feature. This is why the

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

percentage of interest in size is quite higher than the one extracted from the questionnaire.

Percentages also differ in connectivity features. This is probably due to the fact that the average user had little knowledge on differences of connection abilities of mobile phones. This is reinforced by the fact that connectivity technologies such as wi-fi, are hard to comprehend by the average user. These results in users buying phones without their knowing what connectivity abilities these phones have.

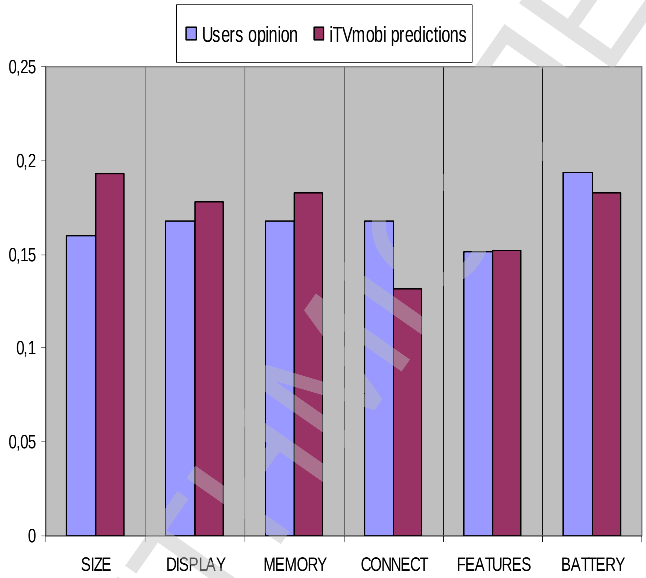


Figure 6. A diagram showing user's interest percentages as they are expressed by users themselves through the questionnaire versus percentages inferred by the iTVMobi recommendation system.

Figures 7 and 8 illustrate two pie diagrams of the interest percentages on phone features. At the first diagram (Figure 7) these percentages are provided by iTVMobi and at the second (Figure 8) are provided directly from the customers directly. As it can be seen these two diagrams are very much alike. This means that iTVMobi had successfully predicted most of customers' choices.

The next diagrams (Figure 9, 10) show interest percentages on mobile phone companies according to user opinions and iTVMobi. The results are also very similar hear. The differences in the percentages are due to the fact that the questionnaire answers where restricted to a scale from 1 to 5, with 1 to mean “not easy at all” and 5 “very easy”.

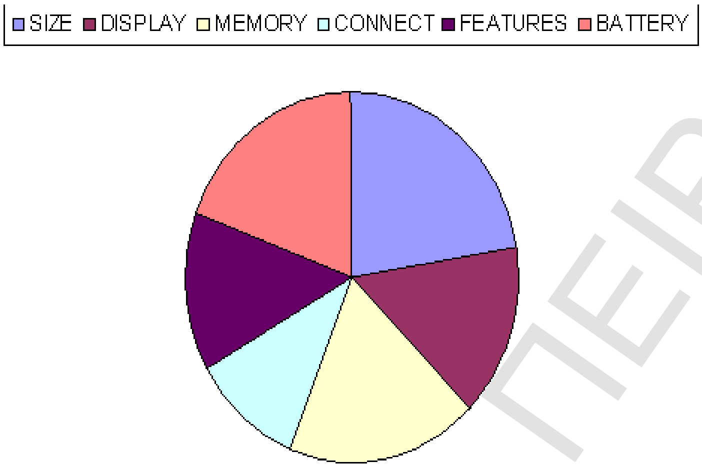


Figure 7. Pie diagram of interest percentages of phone features according to iTVMobi.

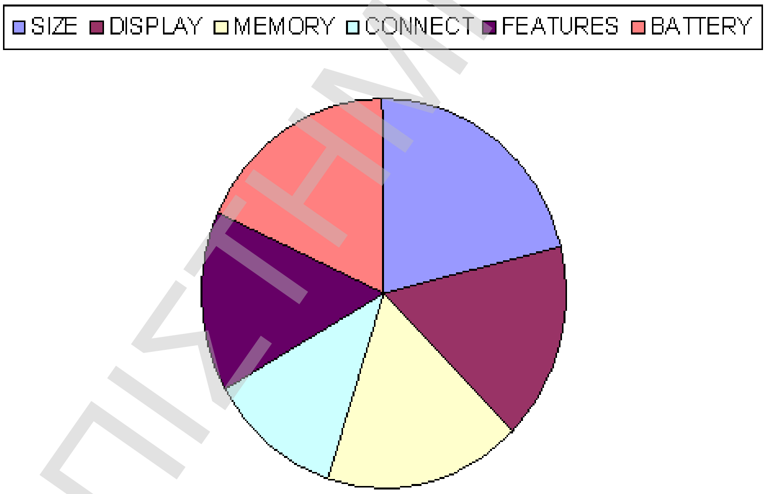


Figure 8. Pie diagram of interest percentages of phone features according to user opinions.

Diagram 9 shows user stats about how the system helped in navigation and the user interface according to confusion and mistakes in general. We can observe that all stats are above 70% which means that the majority of users found the system easy to use and they were helped by iTVMobi effectively. The last diagram (figure 10) show mistake percentages made by users recorded by

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

iTVMobi. As we can see the majority of mistakes have been done at the company menu icons. This was expected as these icons were of similar color and shape. On the other hand phone icons are more colorful and thus easier to spot and discriminate.

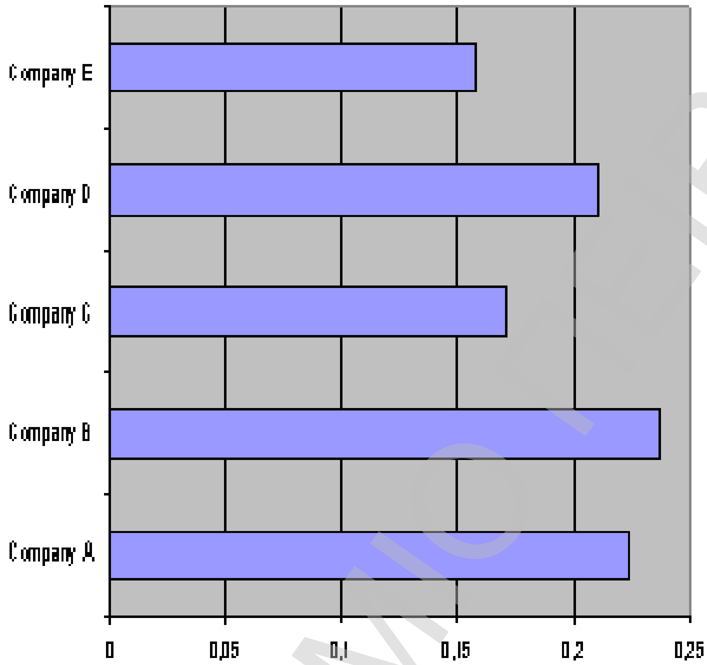


Figure 9. Diagram showing interest percentages of phone companies according user opinions.

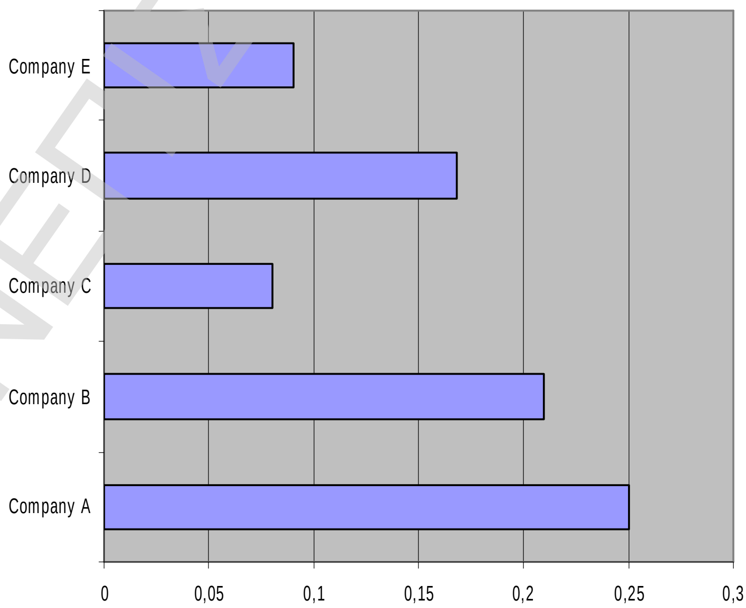


Figure 10. Diagram showing interest percentages of phone companies according

to iTVMobi predictions.

We can also compare the results between the last two diagrams. On the first diagram (figure 11) we see that users tend to give more positive answers about phone icons and how easy was to discriminate them. This results on a lower degree of mistakes at this section recorded by iTVMobi on the second diagram (figure 12). The opposite, between these two diagrams, can be observed on company menu icons. Combining these two last diagrams shows us that iTVMobi really helped people with special needs buy phones more easily.

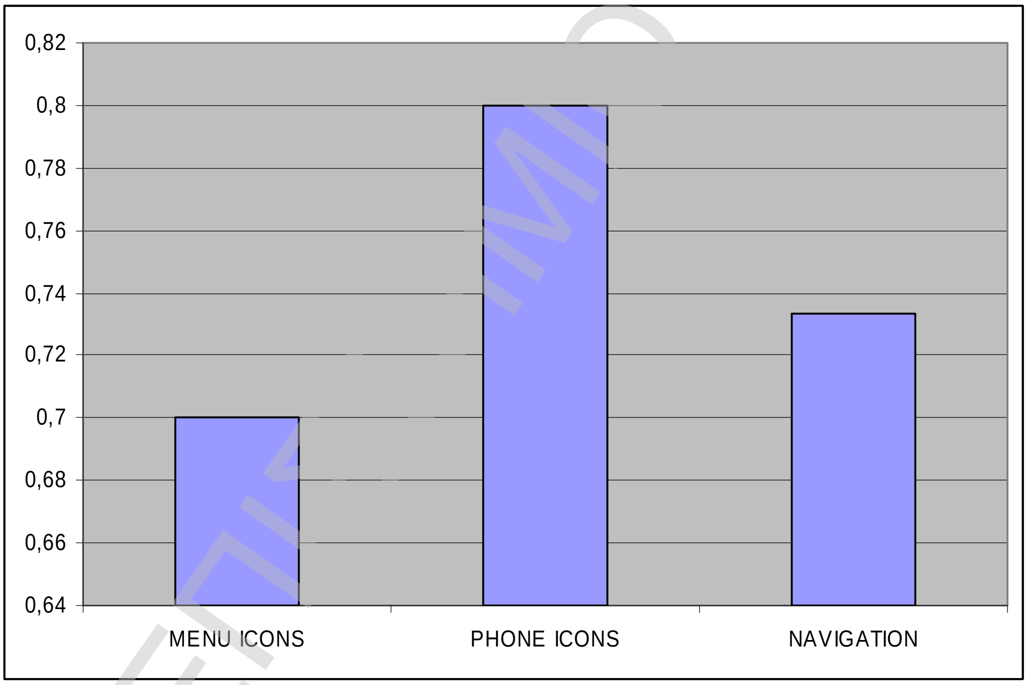


Figure 11. Diagram showing stats about usefulness and simplicity of menu icons, phone icons and navigation in general.

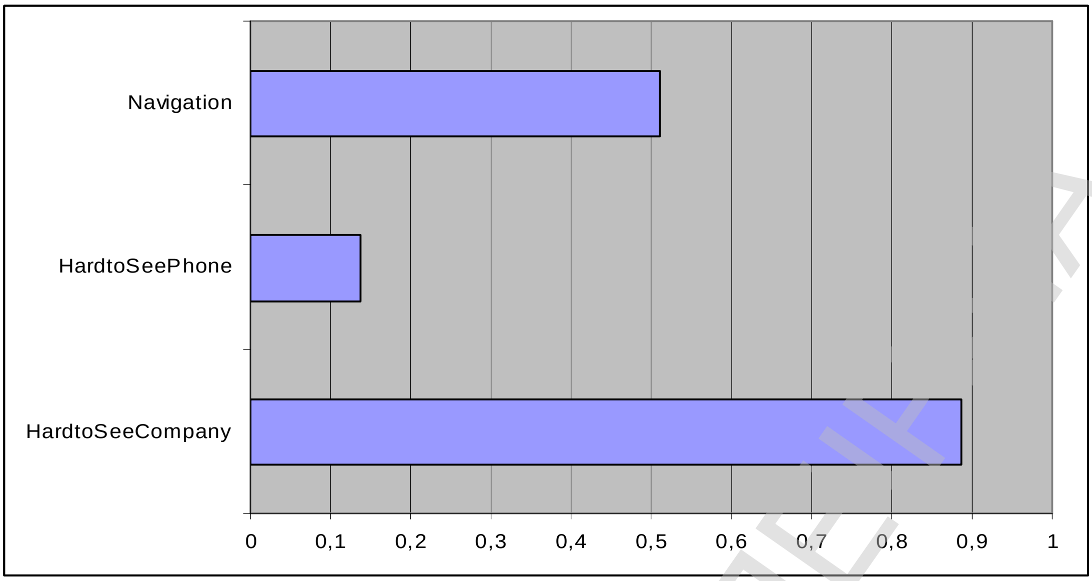


Figure 12. Statistical results of mistakes in company menu icons, phone icons and navigation in general.

6.4 Evaluation in Mobile Shopping

In order to evaluate our methodology in mobile shopping we used our test bed mobile application and conducted a similar experiment with the other two media. Again here at the end of the experiment real users were asked to fill an appropriate questionnaire regarding their opinions about products and interaction with the application. In this section we will present the results in both product predictions according to assumptions about customer tastes and mistake predictions according to assumptions about users' mistakes.

The first bar diagram (figure 13) presents a comparison between the test bed application's predictions and real users' opinions. We observe that the application's predictions are very close to real users' tastes. More specifically, thriller, social and actors categories seem to have predicted most successfully by the mobile application. Furthermore, we can observe that the three most significant categories in real users' opinions are actors, thriller and action movies. This fact can be also witnessed in the mobile shop predictions. Moreover, social movies percentages are low on both real users' opinions and mobile shop predictions. This diagram can also show us a large tendency of the customer pool towards action movies.

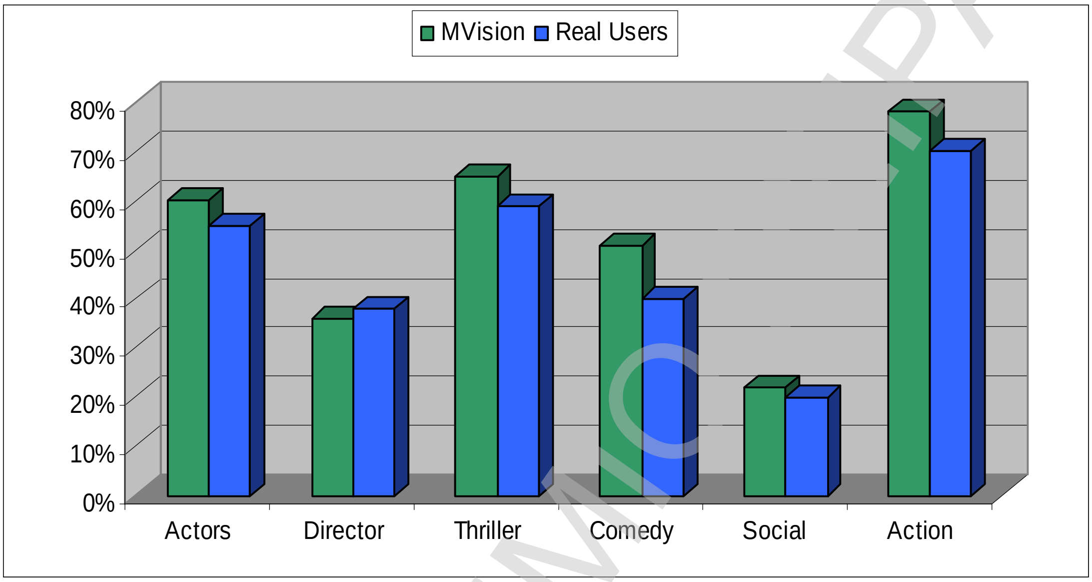


Figure 13 Diagram of comparison results between MVision predictions and real users interests

The second diagram (figure 14) presents a comparison diagram on how movie categories and the price of the movies affected real users' opinions and mobile shop predictions. These two categories played a significant role in our mobile shopping evaluation as they affect a large number of customers' opinions about products. The presented predictions of the mobile shop application are again close to real users' opinions. More specifically, the difference between real users' opinions and application's predictions in the price percentages is smaller then 8%. Furthermore, we observe that these categories affect customer in more than 45% in their possible product buys. This is also reinforced by the fact that users think that movie categories affect their buying behaviour more than 50%.

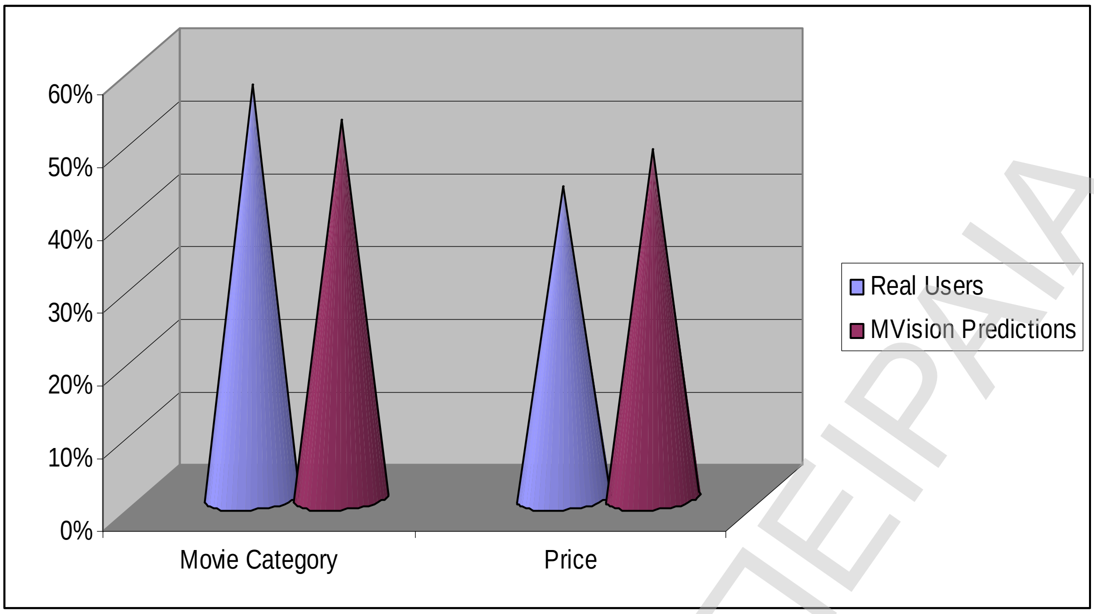


Figure 14 Comparison in two main categories between real users and mvision predictions

The third diagram presented shows the results from evaluating the interactivity of the mobile shop application. Mobile interactivity can be very difficult for a large group of users and its' evaluation can prove to be even more difficult for researchers to conduct and produce significant results. This diagram shows a comparison between mobile shop predictions and real users' mistakes.

Mistakes for real users were registered by the human experts observing every user that interacted with the application during the experiment. Here, we can observe that our methodology has produced some significant results close to the real mistakes percentages done by the real users. Especially, the menu links category was predicted very successfully as the smallest affecting factor in a user's interaction with the mobile application.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

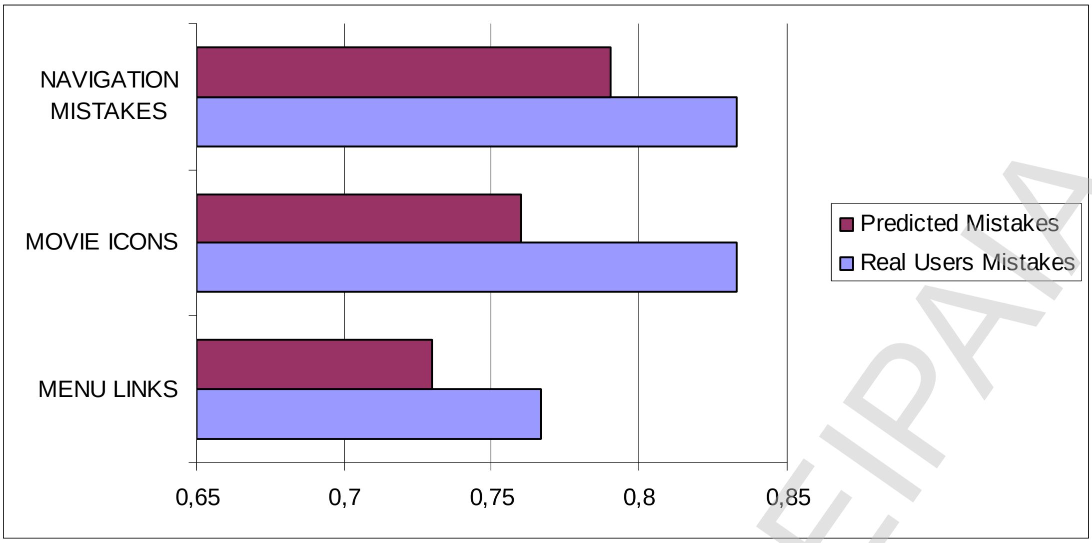


Figure 15 Comparison in users mistakes through the usage of the system in three categories

The last diagram shows a comparison between two different media, TV shopping and mobile shopping. As anyone suspects the mistake degrees in the mobile field are greater than the ones of the TV field. Many reasons have affected the mobile results, starting from smaller screen to mobile phone unfamiliarity. Moreover, the difference between navigation mistakes in mobile shopping and navigation mistakes in TV shopping is also very high. The roots of this great difference can be found in many mobile phones drawbacks, such us unresponsive touch screens or inadequate keypads.

Despite their differences in degrees, both media share an important aspect. This aspect that is also shown by the diagram is the relative percentages of mistakes in both media are similar. This means that in both media, menu choices share the smallest mistake degrees and product icon related mistakes share the largest degrees. This fact means that users tend to have mistake tendencies in shopping applications. This information can be very useful for shopping applications as it can show where developers can aim their efforts in order a more efficient user interface.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

6.5 Discussion the Results of the Evaluation

The three case studies used to incorporate and test PERCOM are entirely different. The first case study sells movies and it is built for web usage, the second sells mobile phones and it is built for TV usage and the third is built for a mobile phone and it's a movie shopping application. In this way the three case studies sell entirely different products meaning that they apply to entirely different domains. Secondly, the applications are built for entirely different media meaning that technical solution of the first case study do not apply to the second and vice versa. These two facts, the different domain and medium, give a PERCOM the ability to be generic. This ability allows PERCOM to be incorporated to different applications applying to various domains.

The numbers of users that evaluated both applications may be small but the members in both evaluations were from various backgrounds and ages, resulting in a very representative sample. Moreover, both applications used to test PERCOM apply to very large audiences, due to their medium, and in this way can be evaluated by hundred of users. Despite the small numbers of users used for evaluation, the evaluation process of the two case studies led to many conclusions. The usage of a machine learning algorithm showed significant improvement in product suggestions and improved applications' conclusions about new users due to the creation of dynamic stereotypes based on the clustering algorithm.

Evaluation results also showed that the usage of clustering users into similar groups can reveal tendencies to specific product features. From a user modelling point of view, the evaluation of the two case studies showed that user models based on user behaviour and not product features where able to describe users' interests to a large extend. This leads to the conclusion that these user models

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

can be used in many applications selling different products without changing the features measured. Lastly, adaptivity in applications, whether this adaptivity is achieved by adaptive hypermedia or adaptive user interfaces, can result in an easiest way of delivering competitive products to users. This is fact can be explained by the comparison diagrams, if we observe the pattern of the interest percentages. The interest percentages follow similar highs and lows in both applications predictions and users' opinions, leading to the conclusion that PERCOM is in the right road and in the future will provide significantly better results.

Despite the fact both applications used as case studies were very different, both used similar technologies. These technologies led us to the generic architecture of PERCOM. More specifically, the usage of clustering in both applications proved to be an important function of both applications. Furthermore both applications used user models in order to save users preferences. The two adaptive technologies; stereotypes and adaptive hypermedia, used in both case studies showed that these technologies can be incorporated into a single generic architecture aiming at providing product recommendations.

Having the above facts in mind we developed a generic application that was based in the technologies used in the two case studies. The basis of this generic architecture follows the logic of the personal recommendations used in the two case studies. We decided that PERCOM should have six major elements; three functionalities and three adaptive technologies. The three functionalities for PERCOM to incorporate would be an adaptive shopping cart, personalised recommendations and adaptive presentation. These functionalities used in both case studies proved to be essential to both applications so PERCOM should incorporate them. We incorporated these functionalities into the second module of PERCOM. The three technologies used in both case studies were

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

user models, clustering and stereotypes. We incorporated these technologies in the first module of PERCOM.

Moreover, we used explicit and implicit data for user models and created an entirely separate component for clustering, the Clustering Process. By dividing these six PERCOM into direct and indirect modules concerning the user interaction we achieved a medium and domain independent behaviour in PERCOM. These two modules can function separately, thus giving PERCOM the ability to manage components in different locations. For example, the indirect components can be incorporated to a server machine and direct components to application machine.

Furthermore, the creation of indirect components can give the ability to PERCOM to change the attributes these components. For example, the k-means clustering can change in the Clustering Process component and another clustering algorithm can be used instead. The Double Stereotypes component can change the type of stereotypes created. For example, stereotypes for user mistakes during interaction with the application can be created in this component. With these three advantages in mind, the easily generalised components, the two modules architecture and the medium and domain independence, PERCOM is advancement over a single personalised e-shop application.

6.6 Conclusions on the evaluation

By applying and evaluating this framework into three entirely different cases, an e-shop application that sells movies and an interactive tv-shop that sells mobile phones, we showed that our generic framework responded well in both these cases. The evaluation results also showed group tendencies towards specific

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

product features that could not be extracted with other ways such as direct questions to customers.

Our work shows that by applying a general recommendations' framework to a shopping application has two major advantages. The first is that the framework can be easily transferred through different mediums such as internet, interactive tv or mobile phones and the second is that the same mechanisms for a specific medium and product, can be easily expanded into assumptions about these users to an entirely different product or medium without any prior knowledge about the connection of these users and the product itself.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

CHAPTER 7

CONTRIBUTIONS AND CONCLUSIONS

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

7 Contributions and Conclusions

This chapter presents the contributions made from this research in the three research fields that relate to this phd thesis. Moreover this chapter discusses general conclusions from this research. Future work is also proposed in order to enhance our methodologies and expand them into new research fields.

7.1 Contributions in the field of remote shopping

The field of remote shopping applications has been greatly researched during the last decade. Many advances have been conducted by many researchers (Choi et al. 2006, Guan et al. 2005). However, the main drawback of these works has always been the medium of the application. The applications created for research purposes have been targeted to one medium at a time. As a result trying to transfer the techniques of recommendation used in this application to another are using a different medium can be a very difficult task that requires developing the system from scratch.

This problem can seriously affect the difficulty of creating intelligent applications for shopping and increase the time needed for their creation. Another serious problem of the applications used by researchers is the fact that despite significant results in product recommendations, they do not address issues of user familiarity with the medium used and comprehension abilities of their users. For example, a user may be unfamiliar with the medium used by the application due to age or other demographic features (place of living). Moreover, elderly users or users with special needs may not be able to comprehend fundamental elements of the application's user interface due to sight problems, other disabilities or simply to unfamiliarity with computers.

Many intelligent remote shopping applications lack intelligent user support. More specifically, many of intelligent applications tend to target only the part of product recommendations and gathering automatic collection of user preferences concerning products. In this way other problems like human computer interaction problems are left behind as second to the main task that is selling products. However, these secondary problems may affect a customer's behaviour and discourage him/her from buying a product through this particular application. For example, a customer that finds an application's user interface very difficult to use and has no effective help in order to overcome his/her problems may abandon the shopping task.

In view of the above problems that relate to remote shopping applications can be summarized to four major problems:

- The first problem relates to the domain of a shopping application.
- The second is dealing with media problems (e.g. the platform that the application is built on), which means managing resources and user interface difficulties.
- The third is targeting a wider pool of users with different ages, knowledge and demographic differences.
- The fourth is an extended user support that will aid users of all kinds to use this application.

Some of the researchers in the last years in the field of remote shopping have dealt with few of these problems but very few tried to create generic methodologies in order to surpass larger difficulties. In order to address these problems, one has to deal with all of these problems in a more generic way. A more general methodology can provide a wider solution to the above problems

thus providing the basis for further research to create more effective techniques for dealing with these problems.

Our approach to deal with the above problems lies in four major features. Firstly, we used user modeling for personalising the human-computer interaction of remote shopping applications. The field of e-shopping applications has used user models to manage user behaviour and extract personalised information. However, our approach differs in many ways from the way that was used by older researchers. We constructed user models based on two different kinds of information, users' needs in products and users' needs in interaction. Using the same user model for both product-related information and interaction information has provided us with the ability of combining product and interaction information. Combining these two kinds of information we managed to create more effective product recommendation and more effective intelligent help for users.

Many applications that use only one kind of user model are faced with the problem that many users' actions may not necessarily mean interest in a product or on the other hand may not necessarily mean a wrong interaction. For example, a repeated visit to a specific product page may not necessarily mean that this specific user has a specific interest for this product. A different meaning of these user actions may mean that this user is confused and does not know what to do next. By combining user models for product recommendation and wrong interaction we managed to successfully separate wrong interaction moves from interest tendencies, thus creating effective product recommendations and help actions.

The second main feature of our methodology is creating an entirely separate component for managing the user interface of our application. The result was an adaptive user interface component that manages the resources and the presentation of the

application's user interface. The adaptive user interface component has given us the ability to effectively use and transfer applications through different media, thus resulting into a media independent methodology. The adaptive user interface component has the ability to change the presentation of the user interface according to two centers. The first is changing the presentation according to the medium used. For example, if an application mobile devices the user interface creates the appropriate user interface elements dynamically for the medium used taking into consideration the screen size, interaction abilities (e.g. touchscreen) and cpu capabilities. The second problem taken into consideration by the user interface ability is changing the user interface according to users' behaviour concerning the interaction difficulties. For example, if a user has sight difficulties, then the user interface can extract information by the monitored user's behaviour and change the presentation accordingly (e.g. change size and position of small user interface elements).

The third main feature of our methodology is the combination of user models concerning user support, adaptive user interface and adaptive hypermedia in order to fully support users' interaction with the application. The combination of these three components has given us the ability to have a complete picture of every specific user's behaviour. With this picture, our methodology could react to users' actions in a very effective way. More specifically, our methodology could support users in three different ways. Firstly, by providing help suggestions, secondly by changing the user interface according to specific user's needs and thirdly by annotating the appearance of specific user elements. These three helpful actions have created a friendlier environment for groups of users with interaction difficulties as evaluation studies prove.

The fourth and most important novelty of our approach is the architecture created through our methodology. We managed to

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

combine all three of the above features into an architecture that every researcher can use in order to create intelligent remote shopping applications. The architecture that we developed was built in such a general level that any researcher can easily change its components and satisfy the needs of any remote shopping application. The whole body of our architecture is built in entirely separate components that interact through specific input and output forms of data. In this way if the forms of data for input and output are kept intact, components can be changed very easily. Furthermore, because the philosophy of our architecture is based on separate components, every component used by our architecture can follow the client-server approach, meaning that our architecture does not need to know the actual process of the component, but only for the input and output data provided.

Furthermore, our architecture is built in two different tiers that are tied through a user modeling server that saves all user models from all shopping applications that use our architecture. This user modeling server, can provide researchers data from users' behaviour through different kinds of shopping applications. In this way data from different applications can be combined to extract general behaviour patterns about users in new domains based on data from previous domains. Moreover, the two tier approach helps in minimizing the resources used by the application that uses the architecture, as the user model server and the user modeling components can be located in a server and in the same machine that uses the application.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

7.2 Contributions in the field of shopping applications that use machine learning algorithms

Many researchers have used machine learning algorithms in order to create successful recommendation systems in shopping applications (Schwarzopf 2001, Ardissono and Torasso 2000, Cayzer and Aickelin). Their aim was to use machine learning algorithms in order to extract groups of users with similar behaviour and in this way find interest similarities. Then they used their results to give personalised recommendations about products sold by the shopping application.

However, these applications face a variety of problems. First of all, the actual construction of an intelligent application that bases its reasoning system on a machine learning algorithm can be a very difficult task. Furthermore, frequently the algorithm that researchers choose may be successful for the initial pool of users but as the application is used by a larger amount of users, the algorithm may prove inadequate. For example, the algorithm chosen may provide good results but the amount of time taken may be affected greatly by the number of users in the applications database. In this way as the number of users in the database increases the amount time taken to provide results may increase to unacceptable limits. These problems may require a change of the algorithm used which may require large changes in an application's software meaning loss in time and results.

Another of using machine learning algorithms in shopping recommendations is making assumptions about users that the system has little knowledge of. This problem has two aspects. The first is that the majority of machine learning algorithms lack in providing effective results with a small set of data, thus resulting in

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

inefficient recommendations from the applications. Due to this problem applications have to be used by many users and for a large amount of time before the algorithm can provide the application with an effective set of results. This can result in lack in trust between users and the shopping application due to inadequate and possibly inaccurate recommendations. The second aspect of this problem is for new members of an e-shopping application. New users are users that the system knows very little about thus again we may have the problem of low quality of recommendations. However, new users are a special kind of users for shopping applications because a trust must be built between the e-shop and them. In this way the shopping application must have a quick and effective way of providing successful recommendations to them in order to earn their trust.

The last problem of machine learning-based shopping applications is the difficulty of combing existing results from the machine learning reasoning system with other adaptive techniques related with users' behaviour such as user stereotypes and other techniques widely used for managing and learning from user behaviour.

Many remedies for the above problems have been researched for the above problems but always targeted to a few of these problems. A more general approach must be followed in order to address these problems all together. In our approach, in order to deal with these problems we incorporated the following features. First of all we created a set of entirely different components concerning the observation of user behaviour, creating the groups of similar users and using these results in order to create successful recommendations and help new users.

We created an entirely separate component for incorporating a clustering algorithm into a shopping application. This component can incorporate any clustering algorithm due to a simple rule: the input and output data constitute a vector of numeric values. For this reason our architecture can provide the researcher with algorithmic independence.

The process of clustering algorithms has three main steps. At first the data from the observing behaviour component and the explicit information profiles are inserted in the component in the form of vector containing numeric values. The second step is the calculation of groups with similarities. In our approach two kinds of similarities where investigated:

- similarities concerning products interests and
- similarities concerning interaction problems.

The third step is the extraction of a small set of vectors from every group. This set of vector is not necessarily a set of existing vectors already inserted as input in the algorithm. The small set of vectors is recognized by the architecture as a set of representatives for the corresponding group. With this representative our architecture manages the recommendations and help actions.

For new users that the system has little knowledge about our architecture goes even further. The same task is conducted for products sold by the application and in this way we have two kinds of groups, users and products. Based on these groups as basis our architecture creates levels of stereotypes with each level raising the complexity level until the intra cluster similarity level reaches a certain threshold. In this way we have more general and more complex stereotypes that can be used for users with little knowledge. Furthermore, we have general and complex stereotypes for products that can be tagged to corresponding users.

Our methodology for new users goes a step further creating an incremental initialization process for new users that helps the application create a successful set of effective recommendations. The process of incremental initialization has the following steps. When a new user becomes a member to the system the application

creates a user model, sets all interest values to zero (the system assumes that at start the user has no interest for any product) and starts to monitor his/her actions.

After few interactions with the system, the architecture classifies the new user in a stereotype of the first level of specialization. The first level has very general information. As the user continues interacting with the system moves to the next level of classification that removes a section of stereotypical information and replaces it by individual user model information. This means that in this level stereotypes differ in one section but group users in the other two levels. Next levels of specialization extend the features of interests and constrain stereotypes in favor of individual information. In this way, as users reveal their preferences the system easily classifies them into the respective stereotypes and selects the right interest or mistake stereotypes in order to make recommendations and help actions. The initialization process is conducted until the user reaches the final level of specialization. This level represents the leaves in the hierarchical tree of stereotypes and extends the differences in all features of interests. The level of complexity here is very high and the smallest difference in user's interests or mistakes can alter his/her user model.

Finally, in order to combine all of these processes and thereby create in this way a robust system that would successfully react to old and new users' behaviour we followed the approach of separate components. We implemented the observing behaviour component responsible for collecting data implicitly from users' behaviour. We also implemented the explicit data profiles component responsible for collecting all explicit data from users, either from the registration process or other surveys. These two data are inserted as mentioned before in the clustering process component.

Furthermore, we constructed a stereotype database responsible for storing stereotypes from both users and products

and taking results from the clustering process. Because these components belong to the user modeling components tier of our architecture they interact separately from the actual application and in this way collect, manage and extract results separately from the application. As a result, these components can be used by many applications in the same time as these components are not a part of the actual application.

Last but not least, the level of generality in our architecture can provide researchers with a tool of incorporating different kinds of algorithms by only changing the clustering process component and not the entire architecture.

7.3 Contributions in the field of software engineering for intelligent remote shopping applications

Despite the fact that many researchers have found effective ways providing successful results through the use of machine learning algorithms in the field of shopping applications very few of them addressed software life cycle issues. The creation of an intelligent shopping application that provides accurate recommendations and uses a very advanced algorithm for managing users' needs and inferences may be a very significant contribution to the field of shopping applications but another significant problem may rise.

New researchers in the future may want to reuse these techniques and maybe enhance them in different ways. The lack of a process explaining in detail the steps of creating such an intelligent shopping application based on these adaptive techniques may result in having the difficulty of constructing the application again in a similar level to the one of the first construction.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

In adaptive applications it's very difficult to apply a software life cycle. A very useful process in software life-cycle is the rational unified process (RUP). RUP is an object-oriented process that advocates multiple iterations of the software development process. It divides the development cycle in four consecutive phases: the inception, the elaboration, the construction, and the transition phase. Each phase is divided into four procedural steps, namely, requirements capture, analysis and design, implementation, and testing. The phases are sequential in time but the procedural steps are not. Additionally, RUP is an object-oriented process; thus, it is appropriate for the development of graphical user interfaces such as the one described in our research. Moreover, one important advantage of RUP is the highly iterative nature of the development process. For the above reasons, RUP can be selected as the basis for presenting adaptive systems too. An implementation of RUP life –cycle into systems has been researched by Jaferian (Jaferian et al., 2005), which presented extensions on Business modelling and Requirement discipline of RUP. RUP has been used to present extensions that concern possible security threats and attacks.

In order to overcome software life cycle issues we followed the approach of rational unified process (RUP) as a basis. Our approach RESCA-RUP follows the basic rules of RUP but extends it to incorporate clustering algorithms. RESCA-RUP covers the entire cycle of a systems development. It describes the steps for incorporating the clustering algorithm and explains how the clustering algorithm can be used to create stereotypes and address issues with new users. The creation of such a process can help a researcher incorporate a clustering algorithm into a remote shopping application and effectively test and use the created application.

The RESCA-RUP process was followed in two different paradigm applications belonging to different domains, built in

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

different media and selling different products. The generality RESCA-RUP has been tested by the development of these different paradigm applications. As a result RESCA-RUP has been shown general and independent of the actual application that is constructed and thus can be used virtually be any remote shopping application that is aimed to incorporate a clustering algorithm.

By combining the PERCOM architecture created and the RESCA-RUP software life cycle a researcher has all the appropriate tools for creating an intelligent remote shopping application in any medium and selling any product from scratch.

7.4 Empirical studies and evaluations

We conducted three different empirical evaluations in three different domains: the domain of e-shopping, the domain of tv-shopping and the domain of m-shopping.

The first empirical evaluation handles selling movies via the web, whereas the second attempts to sell mobile phones thought interactive tv, and finally the last study was concerning an e-shop that sells movies through our mobile phone. The three case studies mentioned above try to sell three entirely different products, different by the fact that these products apply to three variant domains. Those three applications have been applied to three different media such as tv, the www and mobile phones. One thing that should be mentioned at this time is that the technical solution applied in the first case study does not apply to the second and vice versa. Consequently, the two features mentioned above, the different medium and domain provide our methodology a generic ability.

This ability gives our methodology the freedom to be medium and domain independent. The incorporation of our methodology into

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

an application that attempts to sell a product can be performed very easily. As a first step, the developer should install the user modelling component module of our methodology into the server machine of the shopping application. Secondly, the developer should install the second module of our methodology to the client application that is going to be used.

As a next step the developer needs to choose between media such as a mobile phone, or interactive tv and a computer. According to the medium, different functionality features are defined, concerning the adaptive user interface, adaptive hypermedia and animated agent. For example, the animated agent cannot be used on mobile phones due to hardware limitations. Furthermore, the developer must next choose the features that the developer demands our methodology to measure, by using the observing behaviour agent. After these features are chosen the developer should assign the desired weight of importance for every feature that had chosen previously, so as our methodology to be able to create the essential stereotypes hierarchy based on the weights assigned in each feature.

Next, interface elements should be assigned to the features selected. The user interface elements mentioned above will be changed and personalized in real time by our methodology according to user behaviour. At last, a small set of questions concerning interest in products should be provided by the developer, to be used through the registration process. our methodology also requires this set of questions so as to be assigned to each feature. The questions provided by the developer will be combined with the questions already stored in our methodology database in order to be used throughout the registration process so as interest related data from users to be extracted in an explicit way.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

The users that evaluated our methodology varied concerning age groups and social background, providing a really representative sample. Moreover, all applications used to test our methodology apply to very large audiences, due to the widespread medium that they use, and in this way the evaluation sample can be huge. Despite the small number of users that testes our methodology, the evaluation process of the case studies mentioned above resulted in many conclusions. In addition, the usage of a machine learning algorithm resulted in an improvement in the suggesting mechanism of our methodology and improved conclusions drawn by the application concerning new users due to the dynamically created stereotypes based on the clustering algorithm.

Evaluation results also provide evidence that using a clustering algorithm so as to group users with similar features together can provide tendencies to specific product features. From a user modelling point of view, this empirical evaluation suggested that user models based on user behaviour and not product features where able to describe users' interests in a better way. This leads to the conclusion that user models can be applied in varying applications that sell different products, without changing the features measured by our methodology.

Lastly, adaptivity in applications, no matter how this is achieved either by adaptive hypermedia or adaptive user interfaces, can result in an easier way to deliver and finally sell competitive products to users. This fact can be explained by the comparison diagrams, if the pattern of the interest percentages is observed. The interest percentages follow similar highs and lows in both applications concerning predictions and users' opinions, leading to the conclusion that our methodology is in the right direction and in the future we hope that will provide significantly better results.

Despite the fact that all three applications used as case studies were very different, they had some common features. The

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

technologies used in all applications led us to the development of generic architecture, our methodology. In addition, all three applications used user models in order to store user preferences so as to achieve personalization. Stereotypes and adaptive hypermedia, the two adaptive technologies used in the case studies showed can be applied in a single generic architecture that aims at providing efficient recommendations, of both products and support, to users taking into account product preferences and past user actions.

Taking into consideration the facts mentioned above, a generic architecture has been developed based in adaptive technologies that applied in both case studies. The basis of this generic architecture follows the logic and structure of the personal recommendation component used in both case studies. It was resolved that our methodology should include six major elements consisting of three functionalities and three adaptive technologies. The three functionalities for our methodology were adaptive shopping cart, a personalised recommendations mechanism and an adaptive presentation of products, including the dynamic change of the user interface accordingly in order to cover user preferences. The case studies showed that these functionalities are essential to both applications so our methodology should incorporate them.

The above mentioned functionalities were included in the second module of our methodology. The three technologies that were used in both case studies were user models, machine learning and stereotypes. We incorporated these technologies in the first module of our methodology. Moreover, explicit and implicit data were user in order to create user models and an entirely separate component for clustering was implemented called, the clustering process.

The division of components into direct and indirect concerning user interaction gave them the ability to function separately and to manage them from different locations. Furthermore, the creation of indirect components can provide our methodology the ability to change the attributes of these components. For example, the machine learning algorithm can be modified directly in the clustering process component and another clustering algorithm can be used instead. Moreover, the dynamic stereotypes component can change the type of stereotypes created. For example, stereotypes concerning user mistakes throughout interaction with the application can be created with help from this component. The advantages presented, the easily generalised components, the two module architecture, the medium and domain independence, give our methodology an advance over a single personalised e-shop application.

The evaluation of this framework into three entirely different cases was evident that the generic framework presented above responded really well in all cases. The evaluation results also showed group tendencies towards specific product features that could not be extracted otherwise, using methods like direct questions to customers. Our work indicates that through the application of a general framework to a shopping application have two major advantages. The framework can be easily transferred through different mediums such as the internet, interactive TV or mobile phones and at the same time the same assumptions drawn by the system concerning a specific medium and product, can be easily expanded to an entirely different product or medium without any prior knowledge about the preference of the users relatively with the product.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

7.5 General Conclusions

Summarizing the research work described in this phd thesis many novel approaches have been proposed. Firstly we proposed a more generalized architecture for creating intelligent remote shopping applications and secondly we supported the creation of such an application by providing a software life cycle process based on the Rational Unified Process. The resulted systems that use this architecture offer an advanced way of making effective suggestions to users not only based on their preferences and tastes but also supporting them in the actual use of a remote shopping system. Moreover, the way that suggestions are displayed to users was also enhanced by our architecture by incorporating dynamic user interface and adaptive hypermedia techniques, user modelling theories such as stereotypes and machine learning algorithms.

Combining the software life and architecture a tool is created for future researchers to use in future development if such remote shopping applications. Furthermore, evaluation results helped us realize the problems in creating such an intelligent remote application but most importantly extract similarities between all three remote shopping applications.

The tool created for future researchers is a very useful tool that covers many aspects of the software engineering process of a remote shopping application thus making the process of constructing an intelligent remote application easier.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

7.6 Future Work

The research that was carried out in the context of this thesis contributes considerably to the related research fields. However, it leaves certain questions open for further research. More specifically, the architecture created for three different media (interactive tv, ,mobile devices and desktop computers) and the variety of clustering algorithms, was combined with a software life cycle process to create an integrated tool for researchers that want to create intelligent shopping applications.

The application of the architecture in other recommendation systems of different domains could lead to useful conclusions on recommendation systems of electronic services, not just for products. For example, this architecture combined with the Rational Unified Process could be applied to an intelligent mobile tourist guide.

Another extension for our architecture may be the enhancement of the user models that are saved in the user model server. Our methodology has used the user models for two reasons. The first was to extract data about the interests that users show on products and product features. The second purpose of these user models was to extract information about user mistakes while users interacted with the test bed applications. These user models could be generalized to be used for a variety of domains containing demographic data, like age and educational background that could help discover larger group tendencies for domains. For example, teenagers tend to buy music-targeted mobile phones rather than business-targeted mobile phones. Another example is that women tend to watch social programs more often than men that tend to

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

watch sports programs and political talk shows. Location data could also be incorporated to these user models. This kind of data could help applications propose opinions from users close in locations about interesting places like shops, restaurants or other places of entertainment. This could create an intelligent social network of users based on location and behaviour towards product, lifestyle and demographic data. Careful statistical analysis of such generalized user models could lead to more specific knowledge about groups of users and not only for product selling.

Lastly, a very useful extension for our methodology could be the incorporation of a decision making theory. For example a multi-criteria decision making theory (Kabassi and Virvou, 2004) could incorporated and combined with the results of the machine learning algorithm in order to enhance the ability of decision making in recommendations and intelligent user support. The set of actions affecting could be the actions of the similar users through the interaction of the system and the set of n attributes could the measured interest of mistake degrees by the architecture. Of course, an extensive study of the relative criteria must be conducted in order to find out what are the effective criteria about the domain that the architecture will be incorporated. For example, an intelligent TV-guide recommendation system could be evaluated by real users and with the support of human experts in order to extract information about the criteria that real users consider as important about choosing a TV program.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

References

Adil, C.S., Banaei, F.F, Faruque, K.J. 2000. Insite: A Tool For Real-Time Knowledge Discovery From Users Web Navigation. Proceedings Of The 26th International Conference On Very Large Databases. Cairo Egypt

Adomavicius, Gediminas.Tuzhilin, Alexander. 2001. Using Data Mining Methods To Build Customer Profiles Computer. Vol. 34, No. 2, Pp. 74-81. Feb. 2001

Agichtein, E., Brill, E., Dumais, S., And Ragno, R. 2006. Learning User Interaction Models For Predicting Web Search Result Preferences. In Proceedings Of The 29th Annual International Acm Sigir Conference On Research And Development In Information Retrieval, Seattle, Washington, Usa, August 06 - 11, 2006

Ajith, A. 2004. Business Intelligence From Web Usage Mining. Journal Of Information & Knowledge Management, Vol. 2, No. 4. Ikms & World Scientific Publishing Co) 375-390.

Alahakoon,O. 2006. Use Of Buying Behavioural Characteristics As General Stereotypes In User Models Monash Univ., Caulfield; Information And Automation,. Icia 2006. Pp: 172-176

Albert T.C., Goes P.B. 2004, Gupta A. Gist: A Model For The Design And Management Of Content And Interactivity Of Customer-Centric Web Sites, Mis Quarterly 28 (2), 161–182.

Alspector J., Kołcz A., Karunanithi N. 1997 Feature-Based And Clique-Based User Models For Movie Selection: A Comparative Study, User Modelling And User-Adapted Interaction 7, 4 279-304.

André, E., Müller, J., And Rist, T. 1996. The Ppp Persona: A Multipurpose Animated Presentation Agent. In Proceedings Of The Workshop On Advanced Visual Interfaces (Gubbio, Italy, May 27 - 29, 1996). T. Catarci, M. F. Costabile, S. Levialdi, And G. Santucci, Eds. Avi '96. Acm, New York, Ny, 245-247.

Andronico A., Carbonaro A., Casadei G., Colazzo L., Molinari A., Ronchetti M. 2003: Integrating A Multi-Agent Recommendation System Into A Mobile Learning Management System. Aims

Aragao V.R., Fernandes A.A., Carole A. 2001. Goble, Towards An Architecture For Personalization And Adaptivity In The Semanticweb. In Int. Conf. On Information Integration And Web-Based Applications & Services, Linz, Austria, Sept.

Ardissono L, Torasso P., 2000. Dynamic User Modeling In A Web Store Shell. In Proceedings Of The 14th Conference Ecai. Berlin, Germany, 621-625.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Ardissono Liliana, Console Luca, Torre Ilaria. 2001: An Adaptive System For The Personalized Access To News. Ai Commun. 14(3): 129-147

Ardissono Liliana, Gena Cristina, Torasso Pietro, Bellifemine Fabio, Chiarotto A., Difino Angelo, Negro Barbara. 2003. Personalized Recommendation Of Tv Programs. Ai-Ia 2003: 474-486

Ardissono Liliana, Goy Anna, Petrone Giovanna, Segnan Marino, Torasso Pietro. 2003: Intrigue: Personalized Recommendation Of Tourist Attractions For Desktop And Hand Held Devices. Applied Artificial Intelligence 17(8-9): 687-714

Ardissono Liliana, Sestero Dario. 1995: Using Dynamic User Models In The Recognition Of The Plans Of The User. User Model. User-Adapt. Interact. 5(3): 157-190

Baylor, A. L. And Pals (Pedagogical Agent Learning Systems) Research Group 2003. The Impact Of Three Pedagogical Agent Roles. In Proceedings Of The Second International Joint Conference On Autonomous Agents And Multiagent Systems (Melbourne, Australia, July 14 - 18, 2003). Aamas '03. Acm, New York, Ny, 928-929.

Bell, M., Hall, M., Chalmers, M., Gray, P. And Brown, B. 2006. Domino: Exploring Mobile Collaborative Software Adaptation. In Proceedings Of The International Conference On Pervasive Computing. 153--168.

Bezdek, J C , Ehrlich, R , Full, W Comp. Geosci. Fcm. 1984: The Fuzzy C-Means Clustering Algorithm. Vol. 10, No. 2-3, Pp. 191-203.

Branting K.: Acquiring Customer Preferences From Return-Set Selections, Proceedings of the 4th International Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development, (2001) 59 – 73

Brusilovsky, P. 1999: “Adaptive And Intelligent Technologies For Web-Based Education.” In C. Rollinger And C. Peylo (Eds.), Kϋnstliche Intelligenz (4), Special Issue On Intelligent Systems And Teleteaching, 19-25

Brusilovsky, P. 2001. Adaptive Hypermedia. User Modeling And User-Adapted Interaction 11, 1-2 (Mar. 2001), 87-110.

Brusilovsky, P. 2004 Adaptive Help Systems. In: W. S. Bainbridge (Ed.) Berkshire Encyclopedia Of Human-Computer Interaction. Great Barrington, Ma: Berkshire Publishing Group.

Brusilovsky, P., Schwarz, E. W., And Weber, G. 1996. Elm-Art: An Intelligent Tutoring System On World Wide Web. In Proceedings Of The Third International Conference On Intelligent Tutoring Systems (June 12 - 14, 1996). C. Frasson, G. Gauthier, And A. Lesgold, Eds. Lecture Notes In Computer Science, Vol. 1086. Springer-Verlag, London, 261-269.

Carmichael, A., Petrie, H., Hamilton, F. & Freeman, J (2003) The Vista Project: Broadening Access To Digital Tv Electronic Programme Guides. Psychology 1(3) Pages 229-241

Carroll, J. And Aaronson, A. 1988. Learning By Doing With Simulated Intelligent Help. Commun. Acm 31, 9 (Sep. 1988), 1064-1079.

Castro J.L., Castro-Schez J.J., Zurita J.M. 2001 Use Of A Fuzzy Machine Learning Technique In The Knowledge Acquisition Process

Cayzer S., Aickelin U, 2002. A Recommender System Based On The Immune Network. In Proceedings Of The Congress On Evolutionary Computation

Chen, Y. And Hu, L. 2005. Study On Data Mining Application In Crm System Based On Insurance Trade. In Proceedings Of The 7th International Conference On Electronic Commerce (Xi'an, China, August 15 - 17, 2005). Icec '05, Vol. 113. Acm, New York, Ny, 839-841

Chin D. N.: Empirical Evaluation of User Models and User-Adapted Systems, User Modelling and User-Adapted Interaction Vol. 11, Kluwer Academic Publishers, (2001) 181-194.

Choi S. H., Kang S., Jeon Y. J, 2006. Personalized Recommendation System Based On Product Specification Values. Journal Of Expert Systems With Applications Vol. 31, Elsevier. 607–616

Chorianopoulos K., Spinellis D.: User Interface Development For Interactive Television: Extending A Commercial Dtv Platform To The Virtual Channel Api. Journal Of Computers & Graphics Vol. 28(2), (2004) 157–166.

Coppel, J., 2000. E-Commerce: Impacts and Policy Challenges. OECD Economics Department, Working Paper No. 252, Organisation for Economic Co-operation and Development OECD

De Castro, L.,N., Timmis, J.: Arti¯Cial Immune Systems: A New Computational Inteligence Approach.1st Edn. Springer-Verlag, London Berlin Heidelberg (2002)

Dias, M. B., Locher, D., Li, M., El-Deredy, W., And Lisboa, P. J. 2008. The Value Of Personalised Recommender Systems To E-Business: A Case Study. In Proceedings Of The 2008 Acm Conference On Recommender Systems (Lausanne, Switzerland, October 23 - 25, 2008). Recsys '08. Acm, New York, Ny, 291-294.

Ding Y., Korotkiy M., Omelayenko B. Et Al.., 2002. Goldenbullet: Automated Classification Of Product Data In E-Commerce, Proceedings Of Bis 2002, Posen, Poland.

Edelsbrunner H. Efficient Algorithms For Agglomerative Hierarchical Clustering Methods Whe Day, - Journal Of Classification, 1984 - Springer
Elliott, C., Rickel, J. & Lester, J. (1999) "Lifelike Pedagogical Agents And Affective Computing: An Exploratory Synthesis". In M.J. Wooldridge And M. Veloso (Eds.): Artificial Intelligence Today, Lncs 1600, Pp. 195-212, 1999, Springer-Verlag Berlin Heindelberg 1999.

Fensel, D., Ding, Y., Omelayenko, B., Schulten, E., Botquin, G., Brown, M., Flett, A. 2001. Product Data Integration In B2b E-Commerce. Ieee Intelligent Systems 16, 4 (Jul. 2001), 54-59.

Fink J., Alfred Kobsa A. 1998. Adaptable And Adaptive Information Provision For All Users, Including Disabled And Elderly People. Journal Of The New Review Of Adaptive Hypermedia And Multimedia Vol. 4) 163-188.

Fischer, G. 2001. User Modeling In Human–Computer Interaction. User Modeling And User-Adapted Interaction 11, 1-2 (Mar. 2001), 65-86.

García-Barrios, L.E., Speelman, E.N., Pimm, M.S.: An Educational Simulation Tool For Negotiating Sustainable Natural Resource Management Strategies Among Stakeholders With Conflicting Interests, Journal Of Ecological Modelling Vol. 210, (2008) 115-126.

Gena, C. 2001. Designing Tv Viewer Stereotypes For An Electronic Program Guide. In Proceedings Of The 8th International Conference On User Modeling 2001 (July 13 - 17, 2001). M. Bauer, P. J. Gmytrasiewicz, And J. Vassileva, Eds. Lecture Notes In Computer Science, Vol. 2109. Springer-Verlag, London, 274-276.

Goren-Bar D, Glinansky O, 2004. Fit-Recommending Tv Programs To Family Members. Journal Of Computer And Graphics Vol. 28, 149-156.

Goy Anna, Ardissono Liliana, Petrone Giovanna (2007) Personalization In E-Commerce Applications. The Adaptive Web: 485-520

Gregor, P., Newell, A. F., And Zajicek, M. 2002. Designing For Dynamic Diversity: Interfaces For Older People. In Proceedings Of The Fifth International Acm Conference On Assistive Technologies (Edinburgh, Scotland, July 08 - 10, 2002). Assets '02.

Guan S-U., Tai Kheng Chan T. K., Zhu F, 2005. Evolutionary Intelligent Agents For E-Commerce: Generic Preference Detection With Feature Analysis. Journal Of Electronic Commerce Research And Applications Vol. 4, Elsevier 377–394

Hall, C.: Business Intelligence - The Personalization Equation. Software Magazine (April) (2001)

Iglezakis D.: Adaptive Help For Webbased Applications. Lecture Notes In Computer Science Vol. 3137, Springer-Verlag, Berlin Heidelberg, (2004) 304-307

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Irizarry, C., Downing, A., Elford, C. Seniors-On-Line: Introducing Older People To Technology (1997) Australasian Physical And Engineering Sciences In Medicine, 20 (1), Pp. 39-43.

Jaferian, P., Elahi, G., Ayatollahzadeh Shirazi, M. R., And Sadeghian, B. 2005. Rupsec: Extending Business Modeling And Requirements Disciplines Of Rup For Developing Secure Systems. Proceedings Of The 31st Conference On Software Engineering And Advanced Applications (2005). Ieee Computer Society, Washington, Dc, 232-239.

Jens F. Jensen. 2005. Interactive television: new genres, new format, new content. In Proceedings of the second Australasian conference on Interactive entertainment (IE 2005). Creativity & Cognition Studios Press, Sydney, Australia, Australia, 89-96.

Jin, X., Zhou, Y., Mobasher, B.: Unified Approach To Personalization Based On Probabilistic Latent Semantic Models Of Web Usage And Content. Proceedings Of The Aaai 2004 Workshop On Semantic Web Personalization (Swp'04). San Jose (2004)

Jin, X., Zhou, Y., Mobasher, B.: Web Usage Mining Based On Probabilistic Latent Semantic Analysis. Proceedings Of Acm Sigkdd (Kdd'04) (2004) 197-205

Johnson, W. L., Shaw, E., Marshall, A., And Labore, C. 2003. Evolution Of User Interaction: The Case Of Agent Adele. In Proceedings Of The 8th International Conference On Intelligent User Interfaces (Miami, Florida, Usa, January 12 - 15, 2003). Iui '03. Acm, New York, Ny, 93-100.

Jones, A., Scanlon, E., Tosunoglu, C., Morris, E., Ross, S., Butcher, P. & Greenberg, J. (1999): Contexts For Evaluating Educational Software. Interacting With Computers, 11(5), 499-516.

Jordan M, Weiss Y. On Spectral Clustering: Analysis And An Algorithm A Ng, - Advances In Neural Information Processing Systems 14: , 2002.

Kabassi K. & Virvou M.. A Knowledge-Based Software Life-Cycle Framework For The Incorporation Of Multi-Criteria Analysis In Intelligent User Interfaces. 2006 In Ieee Transactions On Knowledge And Data Engineering, 18 (9): 1265-1277 September 2006 Ieee Publications.

Katsionis, G. Virvou, M. A Cognitive Theory For Affective User Modelling In A Virtual Reality Educational Game Systems, Man And Cybernetics, 2004 Ieee International Conference On Oct. 2004 Volume: 2, Pp 1209-1213.

Kay J: Stereotypes, Student Models And Scrutability, Its, Lncs, Springer-Verlag Berlin Heidelberg (2000) 19-30

Kazienko P. And Kolodziejski P.: Windowls – Adaptive System For The Integration Of Recommendation Methods In E-Commerce. Awic 2005,

Lecture Notes In Artificial Intelligence Vol. 3528, Springer-Verlag Berlin Heidelberg (2005) 218–224

Kellogg, Wendy A., Breen, Timothy J. Evaluating User And System Models: Applying Scaling Techniques To Problems In Human-Computer Interaction. (1987) Proceedings - Graphics Interface, Pp. 303-309.

Kevin Curran, David King. Investigating The Human Computer Interaction Problems With Automated Teller Machine Navigation Menus. Interactive Technology And Smart Education. 2008 Volume 5 Issue 1 Pp 59 – 79.

Kieβling, W., Fischer, S., Holland, S., And Ehm, T. 2001. Design And Implementation Of Cosima -A Smart And Speaking E-Sales Assistant. In Proceedings Of The Third International Workshop On Advanced Issues Of E-Commerce And Web-Based Information Systems (Wecwis '01) (June 21 - 22, 2001). Wecwis. Ieee Computer Society, Washington, Dc, 21.

Kim, K. And Ahn, H. 2008. A Recommender System Using Ga K-Means Clustering In An Online Shopping Market. Expert Syst. Appl. 34, 2 (Feb. 2008), 1200-1209.

Kirschning, I. And Rueda, J. O. 2005. Animated Agents And Tts For Html Documents. In Proceedings Of The 2005 Latin American Conference On Human-Computer Interaction (Cuernavaca, Mexico, October 23 - 26, 2005). Clihc '05, Vol. 124. Acm, New York, Ny, 148-154.

Kobsa, A., Koenemann, J., Pohl, W.: Personalized Hypermedia Presentation Techniques For Improving Online Customer Relationships. The Knowledge Engineering Review 16(2) (2001) 111–155

Koutsabasis, P. And Darzentas, J. 2008. Item-Based Filtering And Semantic Networks For Personalized Web Content Adaptation In E-Commerce. In Proceedings Of The 5th Hellenic Conference On Artificial Intelligence: Theories, Models And Applications (Syros, Greece, October 02 - 04, 2008). J. Darzentas, G. A. Vouros, S. Vosinakis, And A. Arnellos, Eds. Lecture Notes In Artificial Intelligence, Vol. 5138. Springer-Verlag, Berlin, Heidelberg, 148-159.

Krause, J., Mittermaier, E., Hirschmann, A. The Intelligent Help System Comfohelp - Towards A Solution Of The Practicality Problem For User Modeling And Adaptive Systems (1993) User Modeling And User-Adapted Interaction, 3 (3), Pp. 249-282.

Kurkovsky, S. And Harihar, K. 2006. Using Ubiquitous Computing In Interactive Mobile Marketing. Personal Ubiquitous Comput. 10, 4 (Mar. 2006), 227-240.

Kurniawan, S., Mahmud, M., And Nugroho, Y. 2006. A Study Of The Use Of Mobile Phones By Older Persons. In Chi '06 Extended Abstracts On Human Factors In Computing Systems (Montréal, Québec, Canada, April 22 - 27, 2006). Chi '06. Acm, New York, Ny, 989-994.

Lester J., Stone B. And Stelling G.:1999, Lifelike Pedagogical Agents For Mixed-Initiative Problem Solving In Constructivist Learning Environments, User Modeling And User-Adapted Interaction, 9, Pp. 1–44, Kluwer Academic Publishers.

Li Q., Kim B.M., 2004. Constructing User Profiles For Collaborative Recommender System, Lecture Notes In Computer Science, Springer-Verlag, Pp 100-110.

Lung, C., Zaman, M., And Nandi, A. 2004. Applications Of Clustering Techniques To Software Partitioning, Recovery And Restructuring. J. Syst. Softw. 73, 2 (Oct. 2004), 227-244.

Macqueen, J. B. (1967). "Some Methods For Classification And Analysis Of Multivariate Observations". Proceedings Of 5th Berkeley Symposium On Mathematical Statistics And Probability.

Mark Van Setten, Stanislav Pokraev, Johan Koolwaaij 2004. Context-Aware Recommendations In The Mobile Tourist Application Compass. In: Adaptive Hypermedia 2004. Volume 3137 Of Lncs. 235–244

Maybury M., Greiff W., Boykin S., Ponte J, Mchenry C., Ferro L, 2004. Personalcasting: Tailored Broadcast News. Journal Of User Modeling And User Adapted Interaction Vol. 14, Kluwer Academic Publishers, Netherlands 119-144.

Mazzoleni, M.C., Baiardi, P., Giorgi, I., Franchi, G., Marconi, R., Cortesi, M. Assessing Users' Satisfaction Through Perception Of Usefulness And Ease Of Use In The Daily Interaction With A Hospital Information System. (1996) Proceedings : A Conference Of The American Medical Informatics Association / ... Amia Annual Fall Symposium. Amia Fall Symposium, Pp. 752-756.

Menczer, F., Monge, A.E, Street, W.N.: Adaptive Assistants For Customized E-Shopping. Journal Of Ieee Intelligent Systems (2002) 12-195.

Miao C., Yang Q., Fangc H., Goh A. A Cognitive Approach For Agent-Based Personalized Recommendation. Journal Of Knowledge-Based Systems Vol. 20 (2007).

Micarelli A., Sciarrone F.: Anatomy and Empirical Evaluation of an Adaptive Web-Based Information Filtering System. User Modeling and User-Adapted Interaction 14, 2-3 (June 2004), 159-200.

Miller, B., Albert, I., Lam, S., Konstan, J., Riedl, J.: Movielens Unplugged: Experiences With A Recommender System On Four Mobile Devices. Acm Intelligent User Interfaces (Iui 03) (2003)

Morrison T., Aickelin U, 2002. An Artifcial Immune System As A Recommender For Web Sites. In Proceedings Of The 1st Conference On Artifcial Immune Systems (Icaris-2002). Canter-Bury Uk. 161-169

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Muller C. Rainer, Wasinger R. 2002 Adapting Multimodal Dialog For The Elderly. Abis Workshop On Personalization For The Mobile World 31-34

Murray, T. (1999): Authoring Intelligent Tutoring Systems: An Analysis Of The State Of The Art. International Journal Of Artificial Intelligence In Education. 10, 98-129.

Ntawanga, F., Calitz, A. P., And Barnard, L. 2008. Maintaining Customer Profiles In An E-Commerce Environment. In Proceedings Of The 2008 Annual Research Conference Of The South African Institute Of Computer Scientists And Information Technologists On It Research In Developing Countries: Riding The Wave Of Technology (Wilderness, South Africa, October 06 - 08, 2008). Saicsit '08, Vol. 338. Acm, New York, Ny, 169-179.

O' Sullivan D., Smyth B., Wilson D. C., Mcdonald K., Smeaton A, 2004. Improving The Quality Of The Personalized Electronic Program Guide. Journal Of User Modeling And User-Adapted Interaction Vol. 14, Kluwer Academic Publishers, 5-36.

Ono, C., Kurokawa, M., Motomura, Y., And Asoh, H. 2007. A Context-Aware Movie Preference Model Using A Bayesian Network For Recommendation And Promotion. In Proceedings Of The 11th International Conference On User Modeling (Corfu, Greece, July 25 - 29, 2007). C. Conati, K. Mccoy, And G. Paliouras, Eds. Lecture Notes In Artificial Intelligence, Vol. 4511. Springer-Verlag, Berlin, Heidelberg, 247-257

Oppermann R., 1994 Adaptive User Support: Ergonomic Design Of Manually And Automatically Adaptable Software. Lawrence Erlbaum Associates, Inc

Pazzani, M. J.: 1999, 'A Framework For Collaborative, Content-Based And Demographic Filtering'. Artiαcial Intelligence Review, 13 (5/6), 393^408.

Pessemier T.D., Deryckere T., Vanhecke K., Martens L., 2008. Proposed Architecture And Algorithm For Personalized Advertising On Idtv And Mobile Devices, In Journal Of Ieee Transactions On Consumer Electronics

Pierrakos D, Paliouras G, Papatheodorou C, Spyropoulos D. C.: Web Usage Mining As A Tool For Personalization: A Survey, Journal Of User Modeling And User-Adapted Interaction Vol. 13, Kluwer Academic Publishers, (2003) 311-372

Piller, F. And Schaller, C. (2002). Individualization-Based Collaborative Customer Relationship Management: Motives, Structures, And Modes Of Collaboration For Mass Customization And Crm. Technical Report 29, Department Of General And Industrial Management, Technische Universit¨At, M¨Unchen.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Pohl, W., 1997. Labour-Machine Learning For User Modeling. In: M. J. Smith, G. Salvendy, And R. J. Koubek (Eds.). Design Of Computing Systems: Social And Ergonomic Considerations. Proceedings Of The Seventh International Conference On Human-Computer Interaction. Amsterdam: Elsevier, Vol. B, Pp. 27-30

Prasad B.: 2005, Hyrec: A Hybrid Recommendation System For E-Commerce, Proceedings Of The 6th International Conference On Case-Based Reasoning, August 2005, Chicago, Usa

Pu P., Faltings B., 2002. Personalized Navigation Of Heterogeneous Product Spaces Using Smartclient, In Proceedings Of The International Conference On Intelligent User Interfaces.

Resnick, P. And Varian, H. R. 1997. Recommender Systems. Commun. Acm 40, 3 (Mar. 1997), 56-58.

Rich E.. Users Are Individuals: Individualizing User Models. International Journal Of Man-Machine Studies, 18:199-214, 1983.

Rich, E. 1983. Users As Individuals: Individualizing User Models' Int. J. Man-Mach. Stud. Vol 18 Pp 199-214

Rich, E. 1998. User Modeling Via Stereotypes. In Readings In Intelligent User Interfaces, M. T. Maybury And W. Wahlster, Eds. Morgan Kaufmann Publishers, San Francisco, Ca, 329-342.

Rich, E.: 1999, Users Are Individuals: Individualizing User Models, International Journal Of Human-Computer Studies, Vol. 51, Pp. 323-338

Rich, H. 1979. User Modeling Via Stereotypes, Cognitive Science 3. Morgan Kaufmann Publishers Inc. pp329-354

Savidis A., Antona M., Stephanidis C. 2005 Applying The Unified User Interface Design Method In Health Telematics. Universal Access Code Of Practice In Health Telematics, Lecture Notes In Computer Science Vol. 3041, Springer-Verlag, Berlin Heidelberg 115-140.

Savvopoulos Anastasios, Virvou Maria,. Sotiropoulos Dionisios N, Tsihrintzi George A. Clustering For User Modeling In Recommender E-Commerce Application: A Rup-Based Intelligent Software Life-Cycle. s Jckbse 2008: 295-304

Savvopoulos, M. Virvou, Dynamically Extracting And Exploiting Information About Customers For Knowledge-Based Interactive Tv-Commerce, Kes-Iimss 2008.

Schafer, J. B., Konstan, J., And Riedi, J. 1999. Recommender Systems In E-Commerce. In Proceedings Of The 1st Acm Conference On Electronic Commerce (Denver, Colorado, United States, November 03 - 05, 1999). Ec '99. Acm, New York, Ny, 158-166.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Schwarzkopf E.: 2001, An Adaptive Web Site For The Um2001 Conference, Proceedings Of The Um2001 Workshop On Machine Learning For User Modelling (2001) 77-86.

Shneiderman, B. 1997 Designing The User Interface: Strategies For Effective Human-Computer Interaction. 3rd. Addison-Wesley Longman Publishing Co., Inc.

Sotiropoulos D.N., Tsihrintzis G.A., Savvopoulos A. And Virvou M. Artificial Immune System-Based Customer Data Clustering In An E-Shopping Application, Lecture Notes In Computer Science (Including Subseries Lecture Notes In Artificial Intelligence And Lecture Notes In Bioinformatics) 4251 Lnai - I, Pp. 960-967, 2006

Sotiropoulos D.N., Tsihrintzis G.A., Savvopoulos A., Virvou M., 2006. A Comparision Of Customer Data Clustering Techniques In An E-Shopping Application, Adaptive Hypermedia And Adaptive Web-Based Systems, June 20 - 23, 2006

Ten Hagen S., Van Someren M., Hollink V.: Exploration/Exploitation in Adaptive Recommender Systems, Proceedings Of Eunite, 10 - 12 July, Oulu, Finland (2003)

Theodoridis, S, Koutroumbas, K.: Pattern Recognition.3rd Edn. Academic Press, San Diego (2006)

Tokmakçl, M., UNalan, D., Soyuer, F., OŻtuŕRk, A: The Reevaluate Statistical Results Of Quality Of Life In Patients With Cerebrovascular Disease Using Adaptive Network-Based Fuzzy Inference System. Journal Of Expert Syst. Appl. Vol. 34 (2008), 958-963.

Trappey A.J.C., Trappey C.V., 2004. Global Content Management Services For Product Providers And Purchasers, Computers In Industry 53 (1), Pp. 39–58.

Veletsianos, G., Yerasimou, T. & Doering, A. (2005). The Role Of Intelligent Agents On Learner Performance. In P. Kommers & G. Richards (Eds.), Proceedings Of World Conference On Educational Multimedia, Hypermedia And Telecommunications 2005 (Pp. 3638-3643). Chesapeake, Va: Aace.

Virvou M, Savvopoulos A, 2007. An Intelligent E-Shopping Application That Provides Recommendation In 19th Ieee International Conference On Tools With Artificial Intelligence Ictai 2007, 29-31 October, Patras, Greece.

Virvou M, Savvopoulos A, Sotiropoulos D. N, Tsihrintzis G. A, 2007. Constructing Stereotypes For An Adaptive E-Shop Using Ain-Based Clustering, Lecture Notes In Computer Science, Icannga 2007, 11-14

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Virvou M. & Kabassi K. (2001) F-Smile: An Intelligent Multi-Agent Learning Environment, Proceedings Of Ieee International Conference On Advanced Learning Technologies 2002 (Icalt'02).

Virvou M. & Kabassi K. "Experimental Studies Within The Software Engineering Process For Intelligent Assisatnce In A Gui" 2003 In Journal Of Universal Computer Science, Vol. 8, No 1, Pp. 51-85, Springer Verlag.

Virvou M. (2002) A Cognitive Theory In An Authoring Tool For Intelligent Tutoring Systems, Proceedings Of Ieee International Conference On Systems Man And Cybernetics 2002 (Smc'02), Vol 2, Pp. 410-415.

Virvou M., Savvopoulos A. 2006.Personalized Tv-Shopping Adaptive To The Elderly, 26th Sgai International Conference On Innovative Techniques And Applications Of Artificial Intelligence Ai-2006, 11-13 December

Virvou M. & Kabassi K. 2000. An Object-Oriented Approach In Knowledge Based Software Engineering Of An Intelligent Gui., Frontiers In Artificial Intelligence And Applications, Eds. T. Hruska & M. Hashimoto: "Knowledge-Based Software Engineering", Vol. 62, Ios Press,Amsterdam, 2000, Pp. 285-292.

Virvou, M. & Kabassi, K. (2004). Adapting The Human Plausible Reasoning Theory To A Graphical User Interface. Ieee Transactions On Systems, Man And Cybernetics, Part A: Systems And Humans, 34(4), Pp. 546- 563.

Wang, Q., Makaro, D.J., Edwards, H.K.: Characterizing Customer Groups For An Ecommerce Website. Proceedings Of Acm Conference On Electronic Commerce (Ec'04), New York Usa (2004)

Webb, G. I., Pazzani, M. J., And Billsus, D. 2001. Machine Learning For User Modeling. User Modeling And User-Adapted Interaction 11, 1-2 (Mar. 2001), 19-29.

Weber, G. & Brusilovski, P. (2001) Elm-Art: An Adaptive Versatile System For Web-Based Instruction. International Journal Of Artificial Intelligence In Education, Vol. 12, 351-384.

Welch M. & K. Brownell (2000) The Development And Evaluation Of A Multimedia Course On Educational Collaboration, Journal Of Educational Multimedia And Hypermedia, Vol. 9, No. 3, Pp. 169-194.

Yang, W., Cheng, H., And Dia, J. 2008. A Location-Aware Recommender System For Mobile Shopping Environments. Expert Syst. Appl. 34, 1 (Jan. 2008), 437-445.

Yukun Cao, Yunfeng Li, An Intelligent Fuzzy-Based Recommendation System For Consumer Electronic Products, Expert Systems With Applications, Volume 33, Issue 1, July 2007, Pages 230-240, Issn 0957-4174.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Zakrzewska, D. Cluster Analysis For Users' Modeling In Intelligent E-Learning Systems (2008) Lecture Notes In Computer Science (Including Subseries Lecture Notes In Artificial Intelligence And Lecture Notes In Bioinformatics), 5027 Lnai, Pp. 209-214.

Zaphiris, P., Sarwar, R. 2006 Trends, Similarities and Differences in the Usage of Teen And Senior Public Online Newsgroups. Acm Transactions on Computer-Human Interaction (Tochi), 13(3), 2006, 403-422. Acm Press.

Zhao Y., Tyugu E.1998. Towards a personalized browser for elderly users. In: Poc. of the 4th ERCIM Workshop on "User Interfaces for Everybody", Stockholm

Zimmerman J, Kurapati K, Buczak A.L., Schaffer D, Gutta S. And Martino J.: 2004, Tv Personalization System: Design Of A Tv Show Recommender Engine And Interface., In Liliana Ardissono, Alfred Kobsa, Mark Maybury (Eds) Personalized Digital Television: Targeting Programs To Individual Viewers, Kluwer: 27-52.

Zounemat-Kermani, M, Teshnehlab, M.: Using Adaptive Neuro-Fuzzy Inference System For Hydrological Time Series Prediction. Journal Of Appl. Soft Comput. 8, 2 (2008), 928-936.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ