User Modeling and Recommendation Strategies for Tourism

Departamento de Engenharia Informática

Bruno Ernesto da Silva Coelho

Dissertação para obtenção do Grau de Mestre em Engenharia Informática

Área de Especialização em Tecnologias do Conhecimento e Decisão

Orientadora: Prof. Doutora Ana Maria Neves de Almeida Baptista Figueiredo
Co-orientador: Mestre António Constantino Lopes Martins

Júri

Presidente: Prof. Doutora Maria de Fátima Coutinho Rodrigues, Professora Coordenadora, Instituto Superior de Engenharia do Porto, Departamento de Engenharia Informática

Vogais: Professor Doutor Nuno Cavalheiro Marques, Professor Auxiliar, Universidade Nova de Lisboa, Faculdade de Ciências e Tecnologia, Departamento de Informática; Prof. Doutora Ana Maria Neves de Almeida Baptista Figueiredo, Professora Adjunta, Instituto Superior de Engenharia do Porto, Departamento de Engenharia Informática; Professor Doutor Luiz Felipe Rocha de Faria, Doutor, Equiparado a Professor Adjunto, Instituto Superior de Engenharia do Porto, Departamento de Engenharia Informática; Mestre António Constantino Lopes Martins, Mestre Equiparado a Assistente do 2º Triénio, Instituto Superior de Engenharia do Porto, Departamento de Engenharia Informática

Porto, Junho de 2009

To all people who wish to visit Porto…

Acknowledgements

I would like to take this opportunity to thank all people that, more or less directly, helped me getting throughout the end of this work. Developing such an important work for a master degree, as well as within the scope of a nation-wide research project, brings responsibility and a substantial amount of pressure and stress when, uncontrollably, things do not go as expected. Furthermore, even when things are going fine, there is still the need for support and assurance from out closest ones, either by admiring our work or directing our further way.

First and foremost, I would like to thank two individuals which have accompanied my work since day one. Although they serve as the most important people within the project and command all work done, they were always very understanding and opened with me, allowing me to creatively develop my work. Differences were settled with a remarkable amount of efficiency, and the project was never in a halted state. Thank you, Ana Almeida and Constantino Martins.

I would also like to thank my family. Although none of them understands a single line of programming code, they were always very supportive of my professional and academic life, early figuring out that this is what I really wanted to do in life. A very particular person regarding this part of my life is my girlfriend, which, mainly in the most negative times, has always been there for me, remembering me of my abilities and capabilities, while at the same time being the most hopeful and beautiful beacon of happiness in my life.

Lastly, I would like to thank my colleagues and friends, both at GECAD and DEI in general. They represent a mixture of the last kinds of acknowledgements already given, as they provide more content in their feedback but at the same time mean a more intimate and friendly kind of support, necessary between the last two kinds of individuals.

Resumo em Português

O turismo é uma área claramente elegível para a aplicação das tecnologias provenientes da área da Inteligência Artificial, nomeadamente dos Sistemas de Apoio à Decisão e Sistemas de Recomendação. O panorama actual no que diz respeito à importância do utilizador não é animador. A Modelação de Utilizador é normalmente pobre e rudimentar, quando existe. A generalidade dos sistemas não aproveita toda a informação que pedem aos seus utilizadores, por vezes de forma morosa e cansativa. A recomendação, na perspectiva de apoio à decisão com base nas características do utilizador é normalmente incipiente, não fazendo uso optimizado das referidas características. Este trabalho tem como principais objectivos a elaboração de uma plataforma de Modelação do Utilizador que funcione como base de um Sistema de Recomendação para a área do turismo. A plataforma de Modelação do Utilizador criada representa uma miríade de técnicas de representação e raciocínio de conhecimento que se complementam entre si de forma a criar uma imagem do utilizador coerente, complexa e diversa. O Sistema de Recomendação é, depois, o culminar do esforço colaborativo do qual fazem parte todos os constituintes dessa mesma modelação. Além da aplicação de componentes tradicionais, o sistema é também inovador no que respeita à introdução de técnicas inexploradas na área do turismo, pelo menos no que concerne ao conhecimento adquirido sobre tais sistemas.

Palavras-chave: Modelação do Utilizador, Turismo, Sistemas de Recomendação, Estereótipos, Sistemas Adaptativos

English Abstract

Tourism is a privileged area for the application of Artificial Intelligence technologies, namely Decision Support Systems and particularly Recommender Systems. The current state of the art concerning the user role and importance within systems is not cheering. User Modeling is used in a poor and rudimentary fashion, when even present; systems do not make use of all information given by the users, sometimes in a tedious and boring manner. As opposed to community and social-based suggestions, recommendations based on the user itself are far from being perfect and optimally used given its potential. This work has the main purposes of developing a User Modeling platform which acts as the basis for a Recommender System for the tourism area. The developed architecture features a myriad of knowledge representation and reasoning techniques which complement themselves in order to assimilate a coherent, complex and diverse user image. The Recommender System is then the culmination of the collaborative effort performed by all User Modeling building blocks. Besides making use of traditional techniques, such as user interests / preferences, the system also innovates in what concern to misused features or unexplored techniques in the area of tourism, such as stereotypes.

Keywords: User Modeling, Tourism, Recommender Systems, Stereotypes, Adaptive System

Resumo Alargado em Português

O turismo é uma área claramente elegível para a aplicação das tecnologias provenientes da área da Inteligência Artificial, nomeadamente dos Sistemas de Apoio à Decisão e particularmente dos Sistemas de Recomendação. O infindável lote de opções existentes, aliado à diversidade e heterogeneidade de locais para visitar, bem como as especificidades inerentes ao destino de férias escolhido, aumentam a necessidade da existência de mecanismos de decisão apoiados por computador. Para além disso, é necessário ainda, contar com a componente social, extremamente importante, do turismo, que eleva a categoria desta questão para um problema multi-utilizador / social ao contrário de outras áreas em que basta uma análise simples de utilizador. Apesar da resolução deste problema envolver, sem sombra de dúvida, mecanismos de Modelação do Utilizador elaborados, a verdade é que o panorama actual no que diz respeito à importância do utilizador não é animador. A Modelação de Utilizador é pobre e rudimentar, quando existente; os sistemas não aproveitam toda a informação que pedem aos seus utilizadores, por vezes de forma morosa e cansativa. Para terminar, a recomendação com base no utilizador em si está longe de ser perfeita e optimamente aproveitada, ao contrário da recomendação colaborativa, apara a qual têm sido direccionadas mais esforços nos últimos tempos.

Este trabalho tem como principais objectivos a elaboração de uma plataforma de Modelação do Utilizador que funcione como base de um Sistema de Recomendação para a área do turismo. A plataforma de Modelação do Utilizador criada representa uma miríade de técnicas de representação e raciocínio de conhecimento que se complementam entre si de forma a criar uma imagem do utilizador coerente, complexa e diversa. Esses componentes representam um equilíbrio histórico-evolutivo entre técnicas tradicionais, tais como preferências / interesses, e técnicas inexploradas na área do turismo, tais como estereótipos. Segue-se uma lista de mecanismos de Modelação de Utilizador com componente de raciocínio:

Mapas Auto-Organizados: actuando de forma similar às regras de associação da área de descoberta de conhecimento (embora servindo ainda mais propósitos), são o único componente orientado à comunidade em geral e não ao utilizador em particular.

Estereótipos: classificam os utilizadores num ou mais estereótipos, gerando dessa forma nova informação abstraída conduzindo à obtenção de mais resultados.

Modelo Psicológico: modelando a componente psicológica do utilizador, é possível obter resultados relacionados com o estilo de vida e personalidade do mesmo.

Matriz de Atractividade: modela os interesses do utilizador de forma clássica, através do relacionamento do mesmo com diversos conceitos de pontos de interesse existentes no sistema.

Palavra-chave: modela os interesses do utilizador de uma forma pura, livre e evolucionária, representando assim, uma forma de conhecimento orgânica.

Obtenção de Informação de Forma Explícita: as assunções do sistema são completamente visíveis para o utilizador, conseguindo-se que o mesmo se sinta na disposição de melhorar essas informações e consequentemente as respostas do mesmo.

O principal *modus operandi* do sistema é a reunião, intersecção e colaboração de todos estes mecanismos que se auto complementam, diminuindo desvantagens pontuais, e aumentando a confiança dos dados a que irão dar origem quando forem usados como base do Sistema de Recomendação. Uma outra tarefa específica, essencial para o funcionamento de todo o sistema foi a construção de uma elaborada taxonomia de pontos de interesse que serviria de ponte entre o utilizador e o conteúdo tácito do turismo (os pontos de interesse), de forma a permitir a representação da relação entre os dois.

A Modelação do Utilizador existente no sistema segue uma filosofia criada no seio desta tese denominada de "Processo de Modelação de Utilizador", que divide essa mesma tecnologia em três passos bem distintos. O primeiro passo consiste na **representação do utilizador**, ou modelo do utilizador, que geralmente é constituído por um conjunto de diferentes componentes. Depois, essa informação é usada para **inferir ou gerar conhecimento** relevante que pode ser adicionado ao modelo do utilizador ou então usado pela última fase desse processo, a própria **adaptação do sistema**, cujo objectivo é melhorar a eficiência e a adequabilidade do sistema em relação ao utilizador. O âmbito deste trabalho absorve o primeiro e o segundo passos, ao mesmo tempo que fornece todos os *inputs* necessários à aplicação eficaz do terceiro, embora se tenha desenvolvido um protótipo que caminha nessa direcção.

O Sistema de Recomendação é depois o culminar do esforço colaborativo do qual fazem parte todos os anteriores constituintes da modelação. O Sistema de Recomendação, para além de utilizar técnicas tradicionais de filtragem, tais como a filtragem colaborativa e filtragem de conhecimento, introduz ainda uma nova técnica de filtragem, à qual foi dado o nome de filtragem comportamental, através da utilização de estereótipos e modelos psicológicos. Os pressupostos em que este trabalho assenta foram comprovados com a construção de um protótipo dotado de uma base de dados real que diz respeito à grande área do Porto.

Em jeito de conclusão, seguem-se alguns pontos importantes que ajudam à constituição deste trabalho como um importante estudo no que diz respeito à Modelação do Utilizador e Sistemas de Recomendação:

Qualidade de Arranque: usando mecanismos de abstracção de informação, consegue obter-se informação rica e complexa sobre o utilizador de uma forma intuitiva e rápida.

Sistema de Recomendação Poderoso: utilizando várias técnicas de filtragem de conteúdo, incluindo a inovadora filtragem comportamental, consegue-se um Sistema de Recomendação variado, construtivo, compensador e que pode até ajudar a melhorar os interesses do utilizador.

Evolução Instantânea do Perfil: quase todas as partes constituintes da Modelação do Utilizador são actualizadas em tempo real, resultando numa resposta sempre adequada, precisa e instantânea do sistema.

Conhecimento Variado: utilizando conhecimento com diversos graus de controlo e liberdade, consegue atingir-se um equilíbrio necessário no que diz respeito à evolução do mesmo.

Palavras-chave: Modelação do Utilizador, Turismo, Sistemas de Recomendação, Estereótipos, Sistemas Adaptativo

Index

Picture Index

Table Index

1 Introduction

In a world where stress is increasingly making part of our daily vocabulary, due to the constantly agitated life that we live in, our holidays and travels gain importance each day that passes. Choosing our holiday destination is a very complex task, not only because of the endless number of options and the diversity of places to visit, but also due to motives internally related to the place in itself, like the type of Points of Interest (POIs) that it possesses, the events that it hosts, etc. Another reason which adds complexity to that choice is that, beyond our own interests and preferences, we also take into consideration those of other people, even when the actual trip is performed alone; tourism is, therefore, a highly social sector.

Given the complexity of this theme, tourism is a privileged area for the application of artificial intelligence, and, in particular, Decision Support Systems (DSSs) (Felfernig, et al., 2007). DSSs devise the use of computational means to calculate a great number of decision components that the human brain can't integrally assimilate (it has been proven that the human brain can only retain, in average, 5 to 9 components / factors of decision (Tartaglione, Antonio, et al.,1991)), giving users the result that they expect. This way, DSSs, in a very succinct manner, as described in (Marreiros, 2002), (Ramos, 2007) and (Coelho, 2007), present the following advantages:

- Greatly decrease the time spent in decision support process, because the process is computational, and therefore, automatic and almost instantaneous; although the tourism area might not be the best example of time criticality (at least when it comes to important parameters like human lives at stake, etc), we all know how long before we start planning our own holidays, and how we'd like that spent time to decrease;
- Make the decision process more exact, precise and normalized, because it is made by computers; that factor also makes the process immune to judgment or mental errors, generally associated with the human being;
- Allow the user to control all decision processes, from the input parameters (its number, characterization, etc), to the refinement of found solutions, by loosening constraints, re-adjusting heuristics, etc.

Within DSSs, it can still be highlighted a particular kind of systems, in which this thesis will greatly focus on: Recommender Systems (RSs), which are a special kind of DSSs with, amongst others, the following two particularities:

- They are used in areas in which information sources are very extent, which makes impracticable the total item presentation to users; this problem leads to questions like the reduction of the number of items shown to users, and therefore, the selection of the best ones (Berka, et al., 2003);
- They give a special importance (much more than regular DSS's) to user preferences and interests, because results will be highly user-related and user-dependent.

RSs are, therefore, generally applied to leisure commerce areas like, for instance, games, movies, music, amongst others, which deal with the complexity of thousands of items. Tourism is a privileged area for the application of such systems: in a first phase, the user has to choose the place where he will travel to, and, in a second phase, choose what to do in that place (generally a given city) to fully occupy his holidays and make a good use of them. This second phase is particularly complex, because the following factors have all to be attended to in the prosecution of the RS tasks:

1. User interests and preferences; these can be positive, as well as negative, and explicitly defined by the user or calculated, i.e., implicitly encountered (the latter ones sometimes being the most valuable);
2. The transport type to use in tours (car, for example), or the choice of the best ones (rail, bus, for instance); choosing the best itinerary for a certain visiting plan may require different types of transportation, the indication of transport lines and stops, and so on;
3. Several other user restrictions, like, for instance, time, money, distance or handicaps;
4. Restrictions associated with POIs such as schedules, opening and closing times and accessibility;
5. Other constraints, like weather, traffic, etc.

As may be implied, all required user information for all these processes must come from a comprehensive user profile which represent all significant user information or abstractions required for the system to deal with, i.e., a User Modeling (UM) process. This component will serve as the basis for all other upward systems and therefore will be the center of this project.

This work, influenced by all motivations and domain context that will be explained next, will therefore try to create a new reference within tourism systems, by making use of powerful techniques in the areas of UM and RSs.

1.1 Context and Objectives

This project has the important motivation to increase research collaboration between different research centers throughout the country, and it represents the result of a partnership made between the Knowledge Engineering and Decision Support Research Center (GECAD) and the Artificial Intelligence Center (CENTRIA) to achieve such goal. In order to state the thesis objectives, it was found more comprehensive to insert those within the overall project objectives, presented next:

1. Make use of an extensive experience in text mining technologies in order to create a much more capable and sustained component in relation with the one already created (more information in 3.3.7); this objective was created when the proposed and initial text mining algorithm was found surprisingly interesting and effective when compared with more classical and formal approaches.
2. Create a rich and complex user model which represents the numerous and various aspects relating tourism users such as preferences, interests, personalities, dislikes, etc.

This user model must also be encompassed with a significantly degree of intelligence and innovation comparing with current methodologies;

3. Create a reasonably detailed and dynamic domain model to encompass all content within the application, regarding POIs. The model was thought, from the first moment, to be some kind of low-level ontology or a medium-sized taxonomy. Specific data, like feature and attribute systems, had also to be developed. This objective is thought of being the launch platform for the next one;
4. Create a more complex and intelligent ontology platform to use in future project phases. This system will replace the previously stated objective and allow for a much more dynamic, user-based and free domain content evolution. Other project's first phase components, like keywords (3.3.5) might also be obsolete or evolved when this component comes online;
5. Create a rich user stereotype system that will characterize users and help in the UM process. Stereotypes, which were the main initial focus and building blocks of the overall project, were soon put at the same level of all other knowledge retrieval mechanisms that the ultimate UM process would be gifted with;
6. Create refined filtering algorithms to be used in RS. These algorithms will use existing techniques and also attempt to come up with new methodologies, by making use of the new approaches already applied within the UM technology. The RS is supposed to be the profiting result from a well-developed UM architecture;
7. Create optimized route generation algorithms which account for user, POI, contextual, environmental and transportation means constraints in order to compute the best tour plans. As the prototype offers a wide variety of planning options, these algorithms must deal with different constraint combinations and variations;
8. Create a Data Access Layer (DAL) that will provide the presentation area with all the necessary functionality. This layer will be responsible for directly accessing the database, making the necessary data refinements, and redirecting the information upwards in the application chain. This layer will also be used by other system platforms, such as mobile ones (see bullet 10);
9. Create the final web-based application which will enable its users to be efficiently, intelligently and effortlessly recommended about all the visiting possibilities inside the great metropolitan Porto area. The application will only serve as a means of providing visual output of the UM process and will be enriched and perfected as the time provides possible, as the website itself is not an important requisite for the work presented in this thesis; a significant back-office system must also be elaborated in order to integrate all researchers and teams currently working within the project;
10. Allow the application to be ubiquitous, by creating other kinds of system platforms, such as Smartphone, Windows Mobile, Android, and so on; these platforms will more thoroughly explore other system features like context-awareness, real-time adjustments, and so on.

The work presented in this thesis attends to objectives 2, 3, 5, 6, 8 and 9, while other members of the project team will attend to the other objectives. Following is a general task plan for all of this thesis work (less detailed than the one just provided, but introducing the time element and some research formalities):

Task Description		Months	Results
State of the art analysis 1. Research and analysis of user modeling techniques 2. Research and analysis of machine-learning techniques 3. Research and analysis of recommender filtering techniques 4. Research and analysis of study cases 5. State of the art complete analysis			State of the art report
	Subtotal	2	1 state of the art report
Methodology definition 6. Objective specification 7. Choice, application and analysis of the techniques described in the objectives 8. Methodology definition			Methodology report
	Subtotal	1	1 methodology report
Implementation and developing 9. Lower-level implementation 10. Back-office implementation 11. Prototype elaboration 12. System implementation and development			Technical documentation Paper in conference, Paper in magazine
	Subtotal	7	1 technical documentation 1 paper in conference 1 paper in magazine
Evaluation 13. Conclusions 14. Future work 15. Summary			Evaluation report
	Subtotal	2	1 evaluation report
Dissertation writing 16. Dissertation writing			Paper in magazine or conference, Dissertation, Thesis resumed paper
	Subtotal	12	1 paper in magazine or conference 1 dissertation 1 thesis resumed paper
	Total	12	

Table 1 - Thesis task schedule

The same information is resumed and displayed next, now in the form of scheduling Gantt diagrams; crossed tasks could not be executed in due time:

Task	1	2	3	4	5	6	7	8	9	10	11	12
State of the art analysis												
Methodology definition												
Implementation and development												
~~Evaluation~~												
Dissertation writing												

Table 2 - Thesis task chronogram

Outcomes	1	2	3	4	5	6	7	8	9	10	11	12
State of the art report												
Methodology report												
Technical documentation												
Paper in conference												
Paper in magazine												
~~Evaluation report~~												
Paper in magazine or conference												
Dissertation												
Thesis resumed paper												

Table 3 - Thesis task deadlines

1.2 Motivations

The motivations behind this project are as follows:

I. An important issue is that tourism websites in Portugal are, in a general manner, very poor. Even by forgetting the fact that they do not show any means of intelligent interaction with the user, they seem to be very sparse, i.e., websites tend to specialize in one type of POIs (like accommodations, generally). Admitting that they can be efficient and complete sites when focusing on one area, there's still the lack of a great, complete, rich and intelligent portal that effectively unites all the components of holidays and travels in one single place, also eliminating the user need to visit various kinds of websites in order to plan a complete trip;

II. By going deeper into tourism RSs in particular, it can be seen that Portugal is light-years behind other systems. First of all, general RSs in Portugal are few, and, in fact, with the exception of some applications that are branches of major multinational companies (like Google, Hi5, Netlog, etc) or few national projects (mainly in the area of socialization like NetJovens) they do not contain any intelligent component at all. One of the main reasons behind this, is the fact that the user is still not largely taken into consideration within the application;

III. Last, but not least, and since we're going to develop a project for a particular geographical area, this work has a final motivation of being an entry door to the city of Porto (we'll specifically not only recommend Porto's POIs, but also those of Porto's biggest peripheral cities like Matosinhos, Maia, Gaia, etc). Porto is a very important cultural center, with ancient traditions, and has a growing rate by far greater than Lisbon, for example. Here are some of the most important Porto's cultural features: (1) it's the second biggest city in the country, and a clear leader city for the north region of the country; (2) it produces one of the most well-known flavored wines of the world, Porto's Wine; (3) it has a large number of world heritage protected areas like it's historical downtown, Clérigos Tower, etc; (4) it was the European Capital of Culture in 2001, which resulted in the construction of "Casa da Música". That building, along with, for example, the Serralves Museum Foundation, are probably the city's most important cultural centers, which are also nationally and internationally known; (5) it hosts one of Europe's most important football clubs, "Futebol Clube do Porto"; this actually represents the only reason why many people even know the city (and that fact brings additional thousands of tourists every year), which makes it one of its most important symbols. The club site actually appears before anything else if we search for 'Porto' in Google.

1.3 Contributions

As it was said before, Portugal's state of the art in web-based tourism RSs is very weak. Apart from, obviously, trying to compensate for all the problems of the current national systems, we also aim at developing components and functionalities currently very rare or unavailable at all, such as:

- The inclusion and use of geo-referenced (geographical) data about users POIs, including the presentation of dynamic, interactive and intuitive maps to help with the solution interface. We will most probably recur to an external source for this kind of purposes (such as Google Maps or Microsoft Virtual Earth);
- The inclusion of negative user preferences and interests. Current systems (also the major ones) do not seem to give this component the respect it deserves. In fact, a user dislike for a certain type of POIs, for example, may be actually stronger and more interesting than a positive interest, which is very important in the recommendation phase;
- The inclusion of limited in time attractions generally left aside in this kind of systems, like: movie theaters, theatrical plays, expositions, fairs, etc. This particular kind of functionality seems very rarely present, not only in nationwide applications, but also in greater worldwide projects;
- The generation of user visit plans, containing itineraries with POIs for the user to follow. The user will be able to summon its computation, refine the results, define constraints, etc; the system must always display the best visiting course taking into account different constraints like distance, traffic and time. It is most likely that we'll make use of an external transportation database in order to help with this functionality; it might be better to explain

that Porto's transportation system is constituted by trains and a very large, complete and interconnected network of buses and subways (the latter one very recent, but with an extremely high social penetration rate);

- The use of a very extensive, dynamic and detailed stereotype taxonomy for defining users and infer new knowledge about them; stereotypes will be very useful when information about the user is unavailable or does not exist in a suitable quantity and will thus, also help the recommendation component;
- The creation of a social network (Web 2.0) of users with several features such as: rating POIs, commenting on various objects, accessing different kinds of rankings, having travel buddies, etc. It must be said that these functionalities are already full implemented and very common in several worldwide tourism RSs, but, as said before, they are not present in nationwide tourism projects.

1.4 Organization

This dissertation will be organized in the following manner: this first chapter, Introduction, introduced the project, presented its origins, its objectives and its motivations. Also, it was explained how this work integrates with the wider project scope. The second chapter will describe the state of the art, in which every related work subject (namely UM and RSs techniques) is analyzed in respect with their history, current use cases, evaluation, and so on. The third and fourth chapters will focus on the work developed by the author, first through a conceptual point of view and then through a technical report, and will be divided by the different created components. The final chapter will end this work by presenting its conclusions, strengths, weaknesses and future improvements.

This document is yet composed by two technical attachments related to chapter 4.

2 State of the Art

As it was previously referred, this work will focus on UM computation that will be used by a RS which in turn is part of a greater tourism application, as shown in the following “Used by” diagram, acting as a very broad architecture diagram:

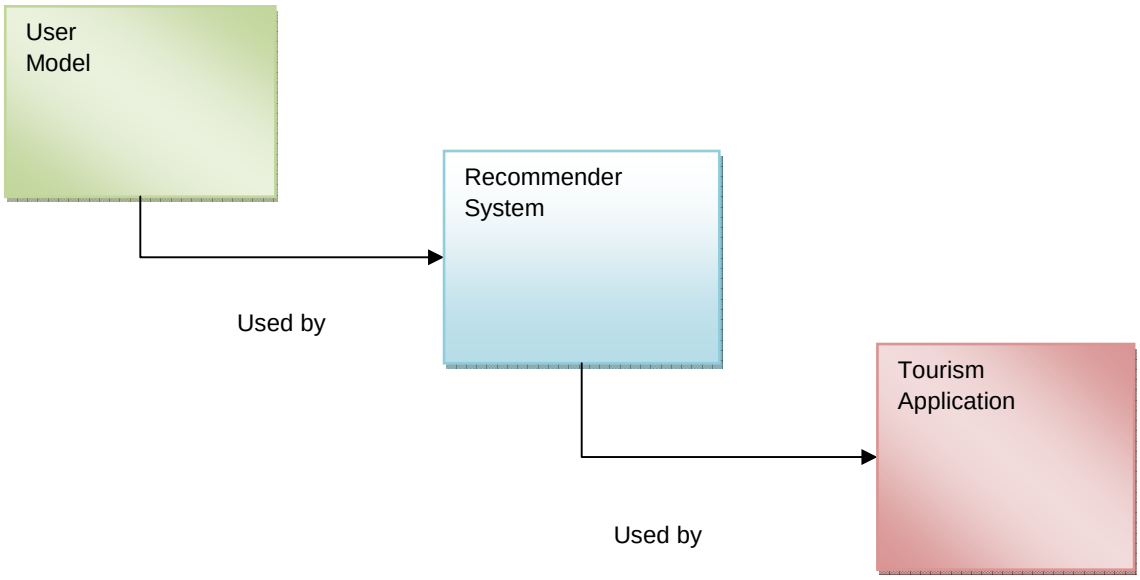

Figure 1 - Work Structure

In this state of the art chapter we’ll start by analyzing UM in an historical perspective, which includes its definition, the need for user models to exist and how that need has evolved throughout the time, as well as the more particular case of RSs. After that, we’ll exemplify several different application domains where UM and recommending computation are currently being used and then describe the different techniques used behind such systems. A comprehensive comparison and evaluation of all different methodologies will then follow, providing the user with an intuitive means to summon up the chapter content. After that, the state of the art will be presented regarding some use cases of practical implementations regarding the previously stated techniques. The chapter will then end by presenting an analysis showing the different types of user information generally stored in the user model, which will serve as the basis for the different presented techniques to infer knowledge about users.

2.1 Historical Perspective

In subjects that refer to computer software history, it is generally explained that many software domains moved from a machine-perspective methodology, where the software was the main system’s component and the user would have to adapt itself and learn how to work with it, to a user-focused design, where the software is designed and works to match user needs, objectives and desires. A vaguely similar reaction has also occurred in marketing evolution. This industry also evolved from a product-oriented approach (companies decided what to develop and sell) to a customer-oriented

paradigm, where all products are developed after costumer surveys and field studies to ensure that all their demands are satisfied. In computer applications, user needs are just as important (Tedlow, 2000). Users must have the feeling that the system is working for his benefit, improving people's everyday tasks' accuracy and efficiency. Users must also find the system perfectly modeled within his image, being adapted to him in every single and possible way. All these issues, if correctly managed, will cause the user to work with the system much more willingly and effortlessly, making him more application-loyal. This matter is very important, for example in web environment, where different applications battle for having the "possession" of regular users. So, if a system will be developed for the user, it must be kept in mind what users wants and intend to achieve; that's where UM comes into play.

The first traces related to UM research date back from the late 70's in several works done by Allen, Choen, Perrault and Elaine Rich (Kobsa, 2001). In fact, during our research experience, Elaine Rich (Rich, 1979) and more recently Alfred Kobsa (Kobsa, 1994), were found to be two of our most important references within this subject. In the following decade, numerous systems were developed with the purpose of storing different kinds of user information, in order to perform several adaptation techniques. Some of those applications were analyzed and reviewed in works done by Morik, Kobsa, Wahlster and McTear in 2001. In those first systems, UM was performed by the application itself and there wasn't a clear distinction between system's components and UM processes, just like happened in several other computational areas, before the explosion of the advantages of software encapsulation and modularization. Throughout the years, despite the technology evolution that has taken place within UM (it has become more complex and intelligent, by making use of recent technological breakthroughs), the concepts and ideas that formed the basis for the appearance of this research topic are still the same: the identification of user needs, desires, personalities and, most importantly, objectives.

Despite that, the last years have witnessed an enormous shift in human-computer system's dynamics; this kind of applications have evolved from a static existence to the point in which the same system represents a completely new and different experience depending on the current user, by making use of distinct adaptations in several system's components such as features and interfaces, amongst others. The birth and boom of the Web certainly had a direct influence in this evolution: on one hand, products and applications are now viewed and used by a worldwide audience, whose inherent heterogeneity demands for concerns relating the modeling of different kinds of users; on the other hand, businesses themselves started to be made online, which has intensified the need for the enhancement of several aspects like interface, usability, speed, customization and service precision, just like decades before happened with products and businesses in their physical form (Gay, et al., 2004).

For UM implementation, two sets of techniques are generally referred: knowledge-based and behavioral (Kobsa, 2001). Knowledge-based adaptation is typically the outcome of information gathered using forms, queries and other user studies, with the purpose to produce a set of heuristics. Behavioral adaptation results mainly from user monitoring during his tasks or activities. The system described here, uses mainly the behavioral approach since one of the objectives of the final

application is to let the user free to do whatever he wants in the system, rather than spend time answering questions.

One of our most important acknowledgements and work methodologies was to see UM as a process. Our definition of UM is as follows:

User Modeling: it's the means by which a system keeps information about his users and uses that information in a variety of ways with the ultimate purpose of improving and customizing user experience within that system (Kobsa, 1994). It pertains to a process that begins with a suitable **representation of the user**, or user model, which can be the sum of a wide variety of different components. Then, that information is used **to infer and generate possibly new knowledge** that is used by the last component of the process, the **system adaptation**, which exists to enhance system's efficiency and suitability, as perceived by the user. By making use of that system adaptation, the **user model is adapted** and the cycle once again begins. Not too few times we see user model wrongly mistaken with UM. As we said, user model is, in our perspective, only a part of the UM process, described in Figure 2.

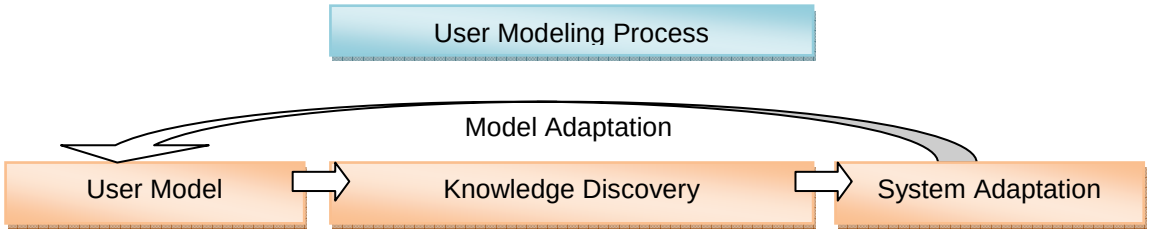

Figure 2 - User Modeling Process

RSs are a little bit more recent than UM because they mainly advent from the internet boom, in the 90's (Porter, 2006). Although RSs may be used in other environments than the web, it's in that perspective that we'll analyze them, because the entire work developed in this thesis was done, having a web application in mind. RSs are the response to the ever increasing information availability supported by the web, which causes some domain areas to contain item spaces too large to be cognitively acknowledged by a human being, or, in simpler terms, containing too many items to be shown to the user simultaneously. In our point of view, the definition of RS is as follows:

Recommender System: is the collection of whatever techniques a system uses to filter its items in order to select either the best ones or the most suitable ones for presentation, according to the user (Porter, 2006). Although the most common situation is when the system has to choose the best items from a certain group which otherwise (without the filtering) would only be randomly selected, there are other more important cases where certain items or types of items just can't be shown to the user at a given moment, for example, due to handicap issues. Complete RSs must therefore be prepared to handle both types of situations. RSs's mode of operation is to use some kind of knowledge base (the user model) as the basis for a series of calculations in order to infer which are going to be, amongst all the items available, the ones that will better please the user, according to a wide variety of theories or approaches, which will be detailed later ahead in this chapter.

It can easily be established that a RS is dependent of, and needs, a user model working in the background in order for recommendations to be as customized and unique as possible. These two technologies, UM and RSs, form the main body of this thesis' work, and, in the context of this state of the art chapter, will be further analyzed and evaluated.

2.2 User Modeling

This section of the state of the art will focus on UM only. The first section of the UM state of the art is to demonstrate first and foremost how this technology is useful.

2.2.1 Application Domains

Next is a brief list of some domains or areas that may require UM, the reason for that need and how that technology fits those domains. Of course that if we were to use a single expression for systems that use UM, it would be something like "adaptive systems", which can be defined as any system that changes and adapts itself to enhance user experience, whichever would it be. Despite that, it is felt that the enumeration of some more precise examples better reveals UM usefulness and interdisciplinary nature in practical terms.

Educational applications: this kind of systems started out to be very general, with very little dependence on the students. Such course of action soon was found to be very poor, because all students learn in a different manner, have different learning speeds, etc. By modeling user information such as tests taken, difficulties revealed and subjects correctly learned, the system has better chances of delivering the learning contents most suitable for each user. Also, it must be kept in mind that a student model is not necessarily exponential such as in many other domains. Students can decay their performance in certain subjects, therefore back-tracking their knowledge, which leads to the use of some further special knowledge mechanisms such as truth maintenance modules. In (Martins, et al., 2008), a User Modeling Framework for an Adaptive Learning Tool was presented, having very successful results.

Critical applications: applications like these master very important processes (such as industry, mechanics, electronics, and so on) and are very failure-sensible: generally a single error can jeopardize operations and bring disastrous consequences. Users of such systems might not always have the same level of knowledge (they might be new operators, for example), and therefore, a model of different kind of users must be present, along with particular users 'modus operandi'. For example, for beginners, actions might have lots of confirmations before they are actually committed into the system, in opposition with experienced users with given proof that they know what they are doing and therefore whose actions will be taken immediately.

User interface: getting applications to look the way we want is a need that has arisen in recent years due to technological evolutions in visual interfaces, computer design and web design. UM can be used, for example, to adapt the interface so that the most important areas for the user are the most viewable and accessible ones. Another feature that can be achieved with UM is altering the interface

in respect to eventual handicaps that users might have, for example, by changing colors. In (Coelho, et al., 2008), a visual interface system designed for product commercialization that changes itself according to the type of user is presented. The system includes panel adaptations, different types of help systems, a help agent, etc.

Commercial applications: this domain is where this project inserts itself into. Product applications are any kind of systems that sell or offer (they don't necessarily need to have a commercial background) products that pertain to one or more domains. As it's easy to infer, any kind of such systems has databases with lots of products that they work with, which makes them impossible to be viewable by the user all at once. Of course that the user could eventually navigate the system until he found the pretended products, but that would just not be admissible. Therefore, user models have to be created which attend to user preferences and interests, serving as the means by which RSs infer what would be the most suitable products for that user. These systems might range from simple applications that only rely on explicit information given by the users to more advanced systems that attempt to infer new knowledge using more complex techniques, which is the case of the system scoped within this work.

2.2.2 Use Cases

Regarding UM use cases, it is extremely difficult (much more than RSs) to collect specific data examples of the UM mechanisms technically put into practice, including methodologies, theories employed, etc. Since these technologies are embedded into the inner mechanics of applications and ownership generally prohibits access to them due to privacy and competition reasons, the majority of use cases analyzed are also academic Even by analyzing academic applications, UM examples are not very technical in their explanations, so actual implementations are not easy to encounter. The best source of such systems was (Martins, et al., 2008), which presents a handful of applications in a student-targeted environment, therefore with very different requirements from those appliable in tourism.

UMT (Martins, et al., 2008): this system contains an hierarchical definition of user stereotypes and their respespective rule set. It also allows the detection of contradictions in the stored information. Given that, received information about the user of the system might be classified as being invariable or suppositions.

BGP-MS (Martins, et al., 2008): this system also allows the use of suppositions about the user stereotype or groups of users. This suppositions are represented with logical predicates and their subsets are kept using logical terms. Inferations are made on top of different kinds of suppositions in order to define user knowledge. It can also function as a UM-server, supporting multi-user enrivonments.

DOPPELGÄNGER (Martins, et al., 2008): this applications acts as a server which accepts user information through a series of hardware and software sensors. One of the user data gathering techniques in such sensors are Markov Models (similar to Bayesian Networks, explained next). Users have the possibility to visualize and edit their user models.

TAGUS (Martins, et al., 2008): this is yet another system that allows UM through stereotype definition, as well as an inference mechanism.

UM (Martins, et al., 2008): this UM toolkit tries to represent user knowlede using suppositions and preferences, amongst others. Each information is accompannied by a value that represents its level of confidence, therefore adopting a small part of the Belief, Desire and Intention (BDI) methodology.

Since the data available is not very extent, it was not chosen to perform a formal comparison between the different systems, such as in RSs. However, some more informal considerations can be still be derived, namely:

- The referred systems follow a knowledge-approach, by making use of several kinds of **knowledge management techniques** (suppositions, beliefs, etc), which, although dealing with more certainty in inferred data, is much more computational intensive. These techniques are, in our opinion, too strict when we consider the final natural task of the UM: the RS. It is felt that such techniques are indeed necessary when dealing with domains slightly more delicate, such as education. In the tourism domain, importance needs to be given to the ultimate feature, which is the RS, and the user model must also be designed with that in mind. The RS needs an extremely well balanced relation between fast and reliable data, which is not achieved when using those complex knowledge-based techniques;
- The use of **stereotypes** was positively detected in these applications, and will be an integral part of the work to be described. However, while in those applications knowledge-based techniques were used (see last bullet), our system has instead adopted semi-automatic evolutionary techniques (which might be done offline) to ensure a more proper stereotype growth (see 3.3.3);
- DOPPELGÄNGER allows the user to **see and edit his profile**, by following a transparency point of view. Based on this example and on a deeper approach to such theory (see (Cramer, 2008)), the work described in this thesis will also adopt such methodology, as explained in 3.3.4.

Due to the limited size of the UM use case analysis, is cannot be said that it has influenced much of the work done in this thesis, with the exception of the situations just provided. However, that was not the case with RSs, which will now be presented.

2.2.3 Techniques

After the presentation of some application domains where UM can be applied to, it will be presented some of the different techniques and methodologies that have been used in the recent years to represent user knowledge and allow for inference mechanisms regarding user information, which means that some of them might be more oriented to data representation and others to data inference. Not all of the techniques will be presented, much less all of the variants they have. The

analysis will contain techniques found to be the most important and most recognized, as well as some heuristics or variations that we feel worth mentioning. It's also important to remind the reader that the following models are not mutually exclusive. Their range of applicability does not always overlay, which means that some models might be used together, and generally are. The techniques to be presented actually account for low-level computational theories that do exist in the core of broader upper-level techniques. For example, the use of stereotypes is a known UM theory, but, at a lower level basis, they are first clustering techniques. The following techniques are all forms of predictive statistical models, since they are applied in areas with thousands or millions of items (from products, clients, actions, etc) and can also beneficiate from recent machine learning evolution (Zukerman, et al., 2000). Finally, not all of them might actually be applied in some domains, such as tourism, due to their nature; however, for completeness reasons, they will also be presented.

Linear models: this is one of the most used techniques, and it can probably even be said that every system uses linear models, one way or another, although there are systems that entirely rely on linear models and explore all their possibilities. These models are easy to build and understand; they are efficient and assume probabilistic data as believable effects, which has been a successfully employed theory so far (Zukerman, et al., 2000). They generally use weighted sums or means of frequently accessed items to conclude user interests, in the case of the product applications described previously, and, therefore, infer the likelihood for new unknown items. An example of a linear model might be inferring that a user might like a recently released horror movie with the fairly correct assumption that if 90% of the movies the user saw were horror-based, then the user might be interested in the new one.

Decision trees: decision trees are also a very easy to use technique, and probably the most visually easy one to understand. They consist of trees with nodes that represent the different values or choices amongst an attribute, all the way until a solution, or inference, is found (Zukerman, et al., 2000). Generally, decision trees have the common disadvantage of needing expert knowledge to be created and to be evolved. They represent limited in time knowledge and don't support new situations, which makes them high maintenance. That is the reason why they are also used in expert systems, which have high levels of expert dependence. It doesn't mean that there aren't autonomous algorithms that infer new tree nodes and therefore new knowledge, but these are always less trustworthy than the initial ones, besides requiring more computational effort. A stereotype representative decision-tree is used in (Rich, 1979) in order to, as the knowledge about the user changes, relate him to a pre-defined stereotype representation, which will, in turn, result in different system mechanics.

Neural networks: neural networks are one of the most recent techniques used in UM, in relation with the other models. Their idea comes from the human brain which is composed by neurons that work singularly but do exist in order to help a much greater entity to work, the actual brain (Zukerman, et al., 2000). Therefore, a neural network is composed by nodes (neurons) and relations between them, which represent the power between two certain nodes. Nodes have activation functions that calculate their value or power. The network is activated by input data that will in turn activate the neurons. Calculations will propagate results out of the neural network, giving the required

knowledge. Nodes themselves will learn (update) each time data is propagated, so that the network actually represents a photogram of the current system knowledge. In dynamic neural networks, nodes are not fixed and can be deleted or created in order to support more intelligent and flexible knowledge forms. Neural networks do exist in a wide variety of variants: static and dynamic networks, in which node quantity or node activation functions, for example, are not static, forward-only versus bi-directional networks, single-network versus multi-network networks, in which several networks work together to achieve a common goal or solution, fuzzy networks, Self-Organizing Maps (SOMs) (Kohonen, 2001), etc. A successfully applied UM neural network-related model can be found in (Jennings, et al., 1991), in which network nodes represented several document keywords and the relations between them represented the strength of the co-occurrence of two of those keywords, all relating to the current user. Since technically that model is neither a neural network nor a SOM, but this thesis will later refer to this work, it will be entitled just “Jennings Model”.

Text mining: text mining is a special branch of data mining discovery processes, and, although it works with data, that information is unstructured, in the form of documents or textual descriptions, being, therefore, a process apart from the data mining processes (Pazienza, 2005). The objective behind text mining is to extract meaningful information from text, generally in the form of keywords. More complex algorithms can try to extract important complete sentences, resume whole documents or even break down or structure unorganized texts. One of the biggest challenges of text mining algorithms is to correctly deal with all the nuances and irregularities of vocabularies, such as different meaning words spelled the same way, and so on. Text mining can be used when information about a certain domain is not structured, either as a long structuring effort whose results will repopulate the information base, or within a RS, where the results of that structuring process will be the basis for recommending items to the user. Domains where text mining may be fruitful are research papers analysis, news services, web content analysis, etc. Text mining is also the most important technique for the content-based RSs.

Bayesian networks: this technique has been the “hype” in UM in recent years, due to its good performance and autonomous capabilities. Bayesian networks consist of inter-connected nodes that represent the probability of an event or a user attribute value being true. Just like neural networks, they are also self-propagated, meaning that a change in a super-node probability triggers changes on all child-nodes, and therefore evolve with new information. Furthermore, they can contain time-changing information and utility functions, featured in dynamic Bayesian networks and utility diagrams, respectively. In (Zhang, et al., 2006), a Bayesian-based UM component backing up a RS is successfully applied, having great advantages such as (1) achieving a better performance in relation with traditional models and (2) having an increased speed by which the Bayesian network converges to optimal recommendations.

Data mining: there are several knowledge discovery processes that pertain to data mining that are worth being described singularly (Zukerman, et al., 2000):

I. Classification: this kind of techniques tries to classify new items according to the classification of previous items. It analyses attributes and finds the ones that will better contribute for creating the knowledge associated with the classification process.

Generally, the output of classification algorithms is a decision tree, but neural networks may also be used, although they are not as visually user friendly as the latter ones. Certain heuristics can enhance the performance of classification algorithms, like error-based pruning, which tries to limit the size of the resulting decision tree, due to readability and performance reasons;

II. Clustering: clustering attempts to detect natural groups or clusters of items within the item space, based on their similarity. The number of clusters can be either pre-defined or automatically inferred, depending on the used algorithm. The more we let these algorithms try to infer knowledge by themselves, the more we increase the probability of finding unexpected, confused or even bad results (unsupervised clustering methods). Supervised clustering techniques seem more appropriate as they allow more control over all the variables that come into play in this kind of algorithms, such as similarity functions, different attribute weights, etc. One of the important challenges of such algorithms is to deal accordingly with isolated cases and therefore avoid complex cluster networks. K-Means is the most recognized clustering algorithm that attempts to build an initially known number of clusters by iteratively relating each item with its closer cluster, using the K-nearest neighbor heuristic and then re-defining each cluster center. Within UM, the classic application method for clustering is the use of stereotypes, which is going to be used within the described work;

III. Association rules: algorithms like these were created to find seemingly invisible patterns and relations between items or groups of items, being generally applied to supermarket shopping carts, also named basket analysis algorithms. Their mode of operation is actually very simple. They compute several item combinations and check for their occurrence within the system in relation with the overall data. The most important item combinations will be the output of the algorithm, and with carefully chosen parameters, they will represent valuable new knowledge. One of the disadvantages of such algorithms is that the resulting knowledge may be too logical or irrelevant (whoever buys baby bottles also buys milk), or even incomprehensible (whoever buys chicken also buys car tires) to be practically applied. The Apriori algorithm is one of the most used techniques and uses a heuristic which allows it to avoid the combinatory explosion issue, by discarding rules whose items don't have enough case support (Agrawal, Rakesh, et al., 1994). Another related technique is the SOM, which, although coming from the area of neural networks, outputs similar data.

In the next section these techniques will be evaluated and compared amongst themselves, and, in the development chapter, based on that comparison, several techniques that were elected to be used in our system will be further explained in the context of the conceptual model designed, as well as the developed prototype.

2.2.4 Evaluation

Within this UM evaluation, we'll analyze and compare all UM formalisms and techniques that have been discussed before, by using the simple and intuitive approach of specifying their advantages and disadvantages, therefore summing up everything. After that, comparisons will be made using some more specific and precise feature evaluations, in where technique vs. technique differences will be better perceived. Although project decisions also include informal, emotional and common sense choices, this comparison is also important for choosing suitable techniques. Following is a comprehensive advantages / disadvantages table that will, in a simpler way, state the most important aspects of each technique.

Technique	Advantages	Disadvantages
Linear models	✓ Simple to use and understand ✓ Efficient ✓ Lots of application domains ✓ Easy to modify	× Not enough for complex knowledge representations
Decision trees	✓ Extremely easy to read ✓ Good performance, mainly in the case of binary trees ✓ Can tackle cold-start issues because doesn't need initial knowledge	× Require expert knowledge × Hard to maintain and change
Neural networks	✓ Good operating performance ✓ Can evolve over time autonomously	× Optimal results take some time to be achieved
Classification	✓ Can result in intuitive and useful decision trees ✓ Can assist in decision-making process	× Needs a substantial amount of data to be efficient
Clustering	✓ Can discover invisible data groups ✓ Can detect isolated cases, if that's an objective	× Challenge dealing with isolated cases × Challenge dealing with the ideal number of clusters
Association rules	✓ Can detect invisible item associations ✓ Can assist in decision-making process	× Can result in unimportant, illogical or useless associations
Text Mining	✓ Only way to extract knowledge from text ✓ The way to cope with content-based filtering	× Textual information means a totally complex domain to correctly explore
Bayesian networks	✓ Good operating performance ✓ Represents both initial and future facts ✓ Evolves autonomously	× Needs expert knowledge for the initial assumptions

Table 4 - User Modeling Technique's Advantages and Disadvantages

The following table describes in a more specific manner some of the presented techniques, by putting them to comparison with some characteristics that were found most useful and important, followed by a brief explanation of each of the selected features. An effort was made into classifying each feature with 'Yes' or 'No' for faster information readability, which can in reality not be as accurate as it would be preferable, since there are only two alternatives to chose amongst. Still, the idea for the table is clearly to be the quickest one, since more detailed information is available in the other sections.

Technique	Rapid Optimum Threshold	Easy-building	Performance independent of system use	Control over results
Linear models	Yes	Yes	Yes	Yes
Decision trees	Yes	No	Yes	Yes
Neural networks	No	Yes	Yes	Yes
Classification	No	Yes	No	No
Clustering	No	Yes	No	No
Association rules	No	Yes	No	No
Text mining	No	Yes	No	No
Bayesian networks	Yes	No	Yes	Yes

Table 5 - User Modeling Technique's Feature Comparison

Rapid Optimum Threshold: if a technique's accuracy against reality and adequate representation towards a domain achieves an optimum threshold, a level at which the results are considered optimal, at a rapid speed or not. For instance, techniques which require initial knowledge instantly achieve optimum threshold, because that knowledge is considered valid.

Easy-building: deals with a techniques' building effort, namely the starting one. It is taken into consideration not just the developing difficulty itself, but also the eventual boredom or tedium degree of that job. A technique's need for initial knowledge is also involved.

Performance independent of system use: if the technique's overall performance differentiates based on the size of the database, system use, etc. This feature is related to the expected complexity change along with system changes. For example, linear models generally work with means, sums and such kinds of calculations, which are very optimized within database management systems, therefore not being much dependent on the size of the system itself, while the execution speed of a clustering algorithm is highly influenced by the number of rows and attributes available.

Control over results: the degree at which the technique can be controlled and customized by the human being working with it. It also means if the results provided by the technique are reasonably expected.

2.3 Recommender Systems

This section of the state of the art will focus on RSs only. The first section of this part is to present a group of particular and interesting application scenarios.

2.3.1 Application Domains

As was already explained, RSs are used in areas with large item spaces, demanding for suitable filtering in the user point of view. If immediate applicable areas emerge to our memories when thinking about this subject, this section is more interested in stating how really interesting, profitable and fast RSs make our life. The following enumeration is only a brief selection of how RSs are being used.

Music: music is an important area for using RSs. Apart from the situations where general-purpose RSs use music data in the same way as books, such as in Amazon (Amazon, 2009), there are systems which use true music RSs. Such examples are constituted by general music websites, online radios, amongst others. We take this time to introduce an interesting question: how useful would it be to enjoy recommended music from radio stations while driving in a car? Another question that must also be stated is that music corresponds to a highly emotional / psychological human reaction, which makes it difficult for collaborative techniques (explained next) to be successfully applied.

Books: books were one of the first resources made available on the web, and profited from the first RSs to ever appear. Amazon's first type of product to be commercialized were books, and providing this system's size, it is expected that it may be, in fact, the biggest book RS to currently exist. Much like music, books are once again an area internally related with each human being in particular, which suggests the choice for knowledge-based filtering rather than collaborative filtering. Plus, when digitalized, books may yet provide other interesting manners of retrieving information and providing suggestions (such as domain-filtering), by making use of text-mining algorithms for the data extraction phase. Just imagine how useful it would be to find all books which contain, in their actual content, a certain Latin plant name.

People: recommending people is a relatively new application domain. However, it is rapidly gaining importance as social networks' popularity also increases. It can be said that social network's commercialized products are persons instead of trade items, as in the previous examples. Suggesting persons is slightly different from recommending items, because information must be previously given by the user in trustful environments. Plus, the relation between humans is incomparably more complex than the relation between a user and an item, and that kind of information can also be used within the RS in order to adopt emotional theories.

Tourism: and finally, tourism. Getting to know the world means the selection of both places and specific POIs in order for the time-limited vacations (or otherwise) to be efficient and customized concerning the respective travelers. Tourism suggestions are more social-exploitable than music and movies, for example, as it pertains to an activity which is most of the time executed in large groups.

This domain area is also one that demands more efficiency and intelligence, due to the significantly higher costs at stake, namely time and money: for example, performing a bad suggestion regarding a trip is much more disastrous than one regarding a book.

2.3.2 Use Cases

RSs are more recent than UM, and a vital part of the current phenomena happening on the internet relating social networks, Web 2.0, amongst others. With that said, it is much more difficult to hide how recommendations are indeed being made, and the analysis is made easier to those eager to know more. However, before, listing RSs use cases, a special note on mobile RSs must be made, as they were found to be important within the RS universe. Mobile applications will mostly be left apart from this list, first and foremost because they're not within the exact scope of this thesis and also because their current main objectives is to deliver context-aware recommendations. In the case where user-targeted suggestions are made, they are mainly executed after explicit user interest retrieval using poor domain models. Plus, many of these systems require specific environments to be run optimally, some of them only even functioning in specific buildings, or more generally indoors. However, the ones found more interesting and capable will be referenced. The following list of applications makes up for the current state of the art in the area of RSs (Porter, 2006) and (Almeida, 2008).

Tourist Guide (Tourist Guide, 2009): a location-based tourist guide application for the outdoor environment, Tourist Guide was developed for visitors of the Mawson Lakes campus of the University of South Australia and of the North Terrace precinct in the Adelaide city center. The user interacts with the system using a PDA that displays his current position, showing detailed information about specific features linked to the current position (a self guided tour of a specific area) like a building view, attractions and nearby equipment, such as public telephones and toilets. This system can be operated in three different modes: Map Mode shows user's current position on the map and the attractions nearby; Guide Mode, which supplies the user with a map showing a tour of related attractions, and Attraction Mode, which provides textual information as well images and sounds about a particular sight.

TripAdvisor (TripAdvisor, 2009): this tourism website advises trips, locations and activities for each user, and also contains a highly social component which allows for lots of elements to be reviewed, commented and rated by others users to assist in the complex decision-making process that pertains to the tourism domain. Although this system is probably one of the most important tourism RSs, profiting from a long existence and referenced by many studies, the fact is that UM reasoning does not seem to make a great part of the system's philosophy. The actual recommendation based on the user is not very well developed. Instead, results seem to be much more social-dependent of the overall community and little importance is put on the main user itself, much less on his interests. Therefore, no matter how frequently a user changes its profile, no significant changes are revealed in results. This system also benefits from an excellent and detailed data source, which, only in Porto, addresses more than 60 restaurants, for example.

DieToRecs (DieToRecs, 2009): this system has the particularity of using case-based reasoning within its clear hybrid RS that merges collaborative-filtering and knowledge-filtering. It has a limited product range with only five types of items, as well as a complex analysis of every user interaction session which forms the basis for each case used in the former case-based reasoning. The traditional shopping cart is called a travel bag, consisting of many items that can also be added and analyzed independently.

Tourism Information Provider (TIP) (TIP, A Mobile Tourism Information Provider, 2009): this system takes the concept of hybrid filtering very seriously and unites all three techniques. This application claims that all problems generally related to RSs, such as the cold start problem, gray sheep individuals (much like clustering isolated cases, gray sheep individuals mean a very enclosed niche of users or singular users who cannot be compared with any others) or over specialization are successfully tackled by using the solidarity nature of hybrid systems. This is one of the few working examples which, just like the project at hand, understands and concurs with the existence of over specialization and tries to diminish it, unlike all other applications. Therefore, this state of the art sees TIP as a very coherent, aware and sensed system.

Heracles (Heracles - Constraint Integration, 2009): this system represents the use of content-based filtering, by using information that was extracted throughout various online data sources and search engines. Since the extracted data is used instantly, the space for inaccuracy is great, due to the unexpected nature of text mining algorithms. Therefore, Heracles also presents users with a supervised machine learning method which increases the amount of input data needed so that results may be more accurate, coherent and sustained. The system also possesses agents for controlling and evolving that tradeoff between input data and inaccuracy, as well as agents which monitor changes in the underlying data sources content, such as airfare rates and restaurants.

WAYN (WAYN, 2009): this application is an evident proof of a Web 2.0 social endeavor regarding tourism. It is very powerful regarding design, multimedia and social aspects such as the portal appearance, maps exploitation, community reasoning, social networks, buddy tasks, multimedia diversity and travel history. On the other hand, it lacks the use of the tourist profile in system results (it does not offer any kind of recommendations whatsoever), has a limited and sometimes weird taxonomy of POIs and does not have an appropriately extensive database. On the contrary with TripAdvisor, for example, it only retains a dozen of events in whole Portugal. In summary, it is much more user-targeted than domain-targeted.

FilmTrust (FilmTrust, 2009): FilmTrust is a web-based system that explores the concept of trust in a movie related social network. With FilmTrust, users can not only express their particular opinion about a movie (by rating or writing reviews), but also define a trust degree for other users and their opinions. This follows the principle of basing predictions on reliable peers, instead of solely on similar ones. To achieve improved recommendations, FilmTrust contains a set of personal agents that help users find relevant information. While these agents retrieve the user preference profile and provide the desired content-based and collaborative recommendations, they can communicate to improve results. On the other hand, user profiles use taxonomies to hierarchically organize the topics in which users are interested in, adding knowledge-based capabilities to the RS.

WebSell (WebSell, 2009): a RS that uses case-based reasoning and decision trees. Along with collaborative filtering, it is designed to support a single business selling a range of products or services. WebSell was tested through the implementation of an agent capable of giving recommendations for apartment renting. The agent requests feedback using a web form that is later used to calculate preferences according to similar cases. The most peculiar feature of the WebSell RS might be its support for customization and configuration. Although there are many RSs that are based on simple fixed products or items (like books and movies), WebSell tries to give support for complex item recommendation. This includes holidays, insurance plans and many other recommendations that involve a large quantity of variables. To achieve customization, WebSell uses two different approaches: operator-based customization and incremental component replacement. Operator-based customization allows the user to apply a set of operations that change the provided recommendation product or service into a customized final item. Incremental component replacement assumes that products and services are structured into components. These components can be replaced for other (more similar to the user preferences) ones.

Cyberguide (Cyberguide, 2009): developed at the Georgia Institute of Technology (GIT) in Atlanta, USA, this mobile RS is based on the ubiquitous computing concept and focuses on mobile context-aware tour guides. The user interacts with the system using a mobile device. The system was designed to assist a visitor in a tour to the GIT and helps the user obtaining information about the demos in display. Knowledge of the user's current location, as well as a history of past locations, is used to provide more of the kind of services that we come to expect from a real tour guide. The system is currently only being used indoors through infrared beacon, but in the future it will be possible to use outdoors through GPS signal. On the other hand, it has very limited tourist information and recommendation capabilities.

gBDI (gBDI, 2009): this proposed RS uses BDI graded agents to deal with uncertainty and graded mental attitudes. Based on previously created tourism packages, the BDI model relies on the agent's beliefs, desires and intensions. Using this model, the system calculates a preference level used to recommend the most suitable packages. Also, the ontology used allows the system to analyze every destination point described in the package; so, although the system cannot propose a dynamically generated set of destinations, all package destinations are used to measure the tourist preference level.

Informal overview:

- In a very broad and maybe bold statement, the current main flaw regarding RSs is the **poor UM technology** backing them up. What this means is that, although knowledge-based filtering, for example, is strongly used, along with social-based techniques, the assumptions taken for granted regarding the user profile are not very sustained and accounted for. Most of them end up using the same knowledge-based limited preferences which are not enough in order to recommend truly user-targeted items;
- Regarding mobile RSs, techniques used (like context-awareness) are not within the scope of this thesis. However, they can, in a simple manner of putting things, be used along the

work described in this thesis as **another layer of filtering techniques**, adapted and adjusted to contextual factors;

- Another fashion that is dangerously surfacing in the latest years is the exaggerated (or at least unbalanced) preference for social / collaborative filtering methods. While the use of this kind of recommendations is not uninteresting (we also embrace them in this very work), we feel that the user itself is still the most important and primary source of recommending material, one whose deep analysis has not yet been performed. The proliferation of social networks and other social phenomena on the current Web is making RSs **too social-targeted**, leaving an action space in the UM architecture that can still be exploited.

As this set of use cases is much more sustained, a summary of advantages and drawbacks of all these applications is shown next.

Use Case	Advantages	Disadvantages
Tourist Guide	✓ Complete and diverse context-aware techniques	× Very limited scope × Domain preferences not accounted for
TripAdvisor	✓ Very extensive dataset ✓ Extensively developed social network	× UM platform is poor × Recommendations are mainly social
DieToRecs	✓ Case-based reasoning	× Negative information abstraction existent within case-based reasoning techniques
Tourism Information Provider (TIP)	✓ Hybrid RS ✓ Context-aware recommendations	× Poor user and domain profiles
Heracles	✓ Up-to-date data sources ✓ Pure content-based RS is unique	× Data might not be consistent × No independent UM whatsoever
WAYN	✓ Fun social network ✓ Multimedia	× No independent UM whatsoever × No RS whatsoever
FilmTrust	✓ Trust-based and hybrid RS ✓ Ontologies	× Limited knowledge-based recommendations × Trust recommendations are imposed
WebSell	✓ Case-based reasoning	× Much explicit data required
Cyberguide	✓ Context-aware recommendations	× Limited in space × Poor recommendations
gBDI	✓ BDI model	× Works at the package level, not POI level

Table 6 – Use Cases Advantages / Disadvantages

2.3.3 Techniques

While the techniques of the devised project's UM technology are complex enough to choose from and to refine, we still have to know how the actual recommendation to the final user will be made. There are basically three types of paradigms we can follow when we're trying to recommend items to a

user. Although there's a lot of information regarding RSs (see (Berka, et al., 2003), (Schafer, et al., 1999) and (Felfernig, et al., 2007)), the terminology used by authors to refer to the different techniques is not very consistent with their meaning. Also, some authors choose to only enumerate the most important ones, while others actually merge some definitions in order to analyze them in some particular point of view. In the following pages, it was chosen to present definitions that better differentiate themselves from the other ones, while at the same time using the same naming conventions as the other authors and making sense of the relation between the technique's name and what they really mean.

Content-based filtering: this technique tries to capture information from within the content of unstructured or unorganized item data elements, such as textual or descriptive attributes, generally including powerful text mining algorithms from the information retrieval area (Pazzani, et al., 2007). It can either: (1) extract important keywords from textual descriptions and compare them with the user model or other item keywords, using probabilistic calculations; (2) compute full texts into weighted vectors and compare the similarity of several of those vectors using bi-dimensional distance mathematical functions. Either way, the most similar items found will be recommended to the user. This technique does not use any domain semantics whatsoever to work with, i.e., whatever the current domain is, it just picks up unstructured data and compares it with other information data of the same type, in opposition to knowledge-based technique. For example, a description field itself says nothing about what the domain is, because almost every domain item might have a description attribute. This technique is generally used when information about domain items is only available in descriptive fields and other kinds of unstructured data representations. If structured, organized and attribute-based data is available, which is undoubtedly the best form of source data to work with, the need for this technique does not present itself, and therefore other techniques are needed. Other times organizations just don't have either the time or the financial situation to support the enormous endeavor of the structuring information process. That process, depending obviously on the business type and the organization kind, might range from a simple situation, where textual descriptions are transformed into three or four fixed attributes, to the case of a high-scale organization that generally has catalogs comprising millions of multi-category items, each with very different collections of attributes, having dozens of different values each. In this work we believe that such effort is necessary. The advantages of having structured data are so much more powerful than those of unorganized information that eventually the return of investment will present itself in time.

Collaborative filtering: this type of filtering (also called social-filtering) is one of the currently most used techniques and was greatly influenced by the Web 2.0 ("social web") phenomena. It relies on other user's information for recommending items to the current user. In recent years, websites have elevated their social contents in such a manner that it would be unwise not to use those kinds of new information (Berka, et al., 2003). In this way, similarity functions are performed between users and not items, like in the other techniques. The most similar users found will then be the source of new recommendation material, using the theory that, if a user is similar to me, then our tastes will also be similar: not as proof-safe as the previous technique, but still a reasonable one. Other types of

information that pertain to this technique are top-chosen items, user reviews and ratings on several kinds of objects, etc. An example of this technique's 'modus operandi' might be: the system looks for people with the same personality traits as the current user, like gender, age, preferred genre, etc, and then looks up for items viewed by those users that the current user hasn't yet seen. This technique has still two aspects, one positive and one negative, that need to be confronted and differ itself from the knowledge-based filtering. First of all, it has a performance or utility curve that starts very low in the lifecycle beginning of an application, generally called the cold-start problem, because it needs user actions to perform correctly, and those will be absent in the initial phases of the system. When the system begins to be used more intensively, with users, comments, opinions, etc, the algorithm is much more efficient because it has a much wider space of data to use, which invariably contains patterns and trends, and therefore much more probability of its recommendations being correct and fruitful. Collaborative filtering has an open-world component, which means that, unlike the knowledge-based filtering technique, unexpected items may be presented. If we imagine two persons the system believes to be similar, they probably don't have the exactly same viewing history and contain some personality traits that may differ from one another and therefore may result in some different viewed items. With that said, the system may actually recommend items that technically don't result from the similarity between the users themselves, but may help to open the world to the current user into choosing new items and evolving his tastes (a subject which will be more discussed ahead).

Knowledge-based filtering: this kind of approach is almost inevitable to use, because it means using any form of domain knowledge in a RS. This was the conclusion found to be more correct with the technique's name. Here, the focus is put into the items and their properties or attributes, a kind of information totally unavailable in the content-based filtering, which are domain-dependant and represent domain knowledge. For example, knowing that a movie genre can only be one of eight pre-defined genres; or even more simpler: the actual existence of a genre attribute, which already distinguishes that item from other domains. Therefore, similarity functions are here performed between items, using those attributes as the basis of comparison, resulting in recommendations of items most similar to items already used by the user, which the system believes to be certainly enjoyed by the latter. Therefore, apart from similarities between items, items themselves will have to be matched against the user model, by mixing both kinds of semantics, as discussed in (Burke, 1999), (Ghani, et al., 2001) and (Towle, et al., 1999). For example: if a system detects that the user has selected a lot of horror movies, when trying to recommend new movies for him to watch, it'll search for movies of the same genre, possibly with similar titles, actors, etc. The main difference between this technique and the last one is that no focus is put on other users whatsoever, but rather on the items themselves. Knowledge-filtering is much more objective and sustained. This technique's theories and assumptions have stronger theoretical background than the last one's. This means that every recommended item using knowledge-filtering has a greater probability of being accepted. On the other hand, this technique has a lesser chance of going out of the expected recommendations, which may result in too strict results and ultimately in a poor performance in giving its users refreshing results. A good idea is to loosen up the item similarity functions so that items can be considered similar if, for example, only one or two attributes match. This keeps the items loosely similar and at the same time brings in items

with new attributes and consequently completely new items, in a dynamic and cyclic evolution that it is believed to be beneficial both for the user, the system and ultimately the domain.

Hybrid filtering: authors generally call hybrid systems to any system whose recommender component is made of more than one of the filtering techniques described above, or eventually other techniques that result from other different perspectives when trying to enumerate them all. For example, a hybrid system may be the result of an application that uses a content-based filtering but at the same time searches for domain or semantic keywords amongst those contents.

About the last two individual filtering techniques, collaborative and knowledge-based, there seems to be a relatively hot discussion about which is the best one. Despite that, the majority of authors have come up with the same conclusion as this project: except in some cases where, for example, applications lack a social component, or for some other reasons, it is absolutely clear that a system that merges knowledge-based filtering and collaborative-filtering has a higher accuracy than a system that only uses one of them (Berka, et al., 2003) (Felfernig, et al., 2007). We've seen that each technique has its own set of disadvantages, which curiously may be overcome by adopting the other one; a system that recommends items based on a mixture of the two techniques (therefore forming a hybrid system) is without doubt the best solution for an application that uses lots of different sources of information, is user-centered and demands the most confident and correct results and at the same time refreshes its users with new items in order to help evolve their interests and tastes.

2.3.4 Evaluation

RSs evaluation section will be presented in the same manner as UM techniques. Evaluations done within the scope of this state of the art try to be at the same time informal and objective. A casual point of view in analyzing techniques was respected in order to please the reader, while at the same time creating other kinds of structured comparisons. The evaluation considers all techniques with the same value, although any human being unconsciously develops a taste for one or other method. With that said, the three recommender filtering techniques will now be put against each other. The following table presents the several presented filtering techniques' advantages and disadvantages.

Technique	Advantages	Disadvantages
Content-based filtering	✓ A general and "easy" way to recommend items ✓ Doesn't require structured or organized data	× An overall poor technique, can't deal with complex domain knowledge requirements
Collaborative filtering	✓ Simulates real-world "human" recommendations ✓ Makes use of a wide social base currently present in the web, the social web or web 2.0	× Requires a substantial amount of data to function properly (cold-start problem)
Knowledge-based filtering	✓ Makes use of complex domain knowledge ✓ Makes use of the ever important user preferences / interests ✓ Can be used anytime	× Requires a great deal of building effort to come up with good item models vs. user models

Table 7 - Recommender Filtering Technique's Advantages and Disadvantages

As similarly made within the UM techniques, a comprehensive table was built in order for the reader to easily assimilate the main differences between the different recommender techniques.

Technique	Start-up quality	System usage independent	Good Performance	Easy-building	Refreshing Results
Content-based filtering	Yes	Yes	No	Yes	Yes
Collaborative filtering	No	No	No	Yes	Yes
Knowledge-based filtering	Yes	Yes	Yes	No	No

Table 8 - Recommending Filtering Technique's Feature Comparison

Start-up quality: in this feature, techniques are evaluated in relation with the quality of results right at startup. Some techniques might require a significant amount of system use in order to be more accurate.

System usage independent: tells us how the overall performance of the RS changes with the evolution of the system, namely more users, more items, more actions done by the users, and so on.

Good performance: this feature evaluates each technique by the resources they require, namely time and space, for their execution, in relation with the other ones, not only at startup, but also throughout the system use.

Easy-building: just like in the UM techniques, this feature evaluates the initial difficulty in applying each recommender technique.

Refreshing Results: just like has been said in the last paragraphs, it is believed that RSs, besides giving out results that correctly relate with the user model so far, must also endorse new and refreshing results that slowly are to evolve user tastes and make him profit the most out of the system. We acknowledge that some people might find this feature not useful, but this project sticks with it.

2.4 Author's Pick

In this sub-chapter it is presented a model architecture that was found most interesting in structuring user information that a system needs in order to represent correctly the data it possesses about users. This model has a conceptual nature, which means that the described components are higher level than the UM techniques presented before, despite those being used internally. At the same time, and besides the user model structure, several user information components will also be explained.

In 1993, Benyon proposed a "Student User Modeling Architecture" which divides several user information components into a comprehensive and meaningful information hierarchy (Martins, et al., 2008). Although in that example the architecture was used with the purpose of modeling Educational Adaptive Hypermedia (EAH) users, it was found to be perfectly usable in a variety of other situations, such as tourism. The architecture dictates that user model data should be divided, at its root, in two modules: Domain Independent Data (DID) and Domain Dependent Data (DDD):

Domain Independent Data: this component is responsible for hosting user information that is not expected to change with system interaction. Some elements may eventually change, but generally that evolution is user-controlled. We can say, at some extent, that this component is also domain dependent, because only information remotely useful for the system will be stored here, i.e., information that is not relevant to the system will not be used. Such mistake would be a waste of space, time and user effort, since much of the information that exists in this module comes directly and explicitly from the user.

Domain Dependent Data: this module contains information about the user which relates somehow with the domain at hand, generally harvested implicitly or resulting from knowledge retrieval mechanisms in the system. This type of information may be represented in the form of some of the UM techniques that were described earlier in this chapter.

The model will not be extensively described because as we go deeper in it, the more we find domain-dependent structures associated with EHS. Nonetheless, within the DID, the model is subsequently divided into two general-purpose tourism-compatible categories: the Generic Profile and the Psychological Profile. The **Generic Profile** contains pieces of information like personal information, demographic data, academics background, qualifications, background knowledge and handicaps. The **Psychological Profile** deals with information such as student learning style, cognitive capabilities and traces of personality. The DDD contains information gathered throughout the student use of the system, such as current learning objectives, knowledge acquired, assessments' results, aptitudes, interests, deadlines and contextual and environmental variables.

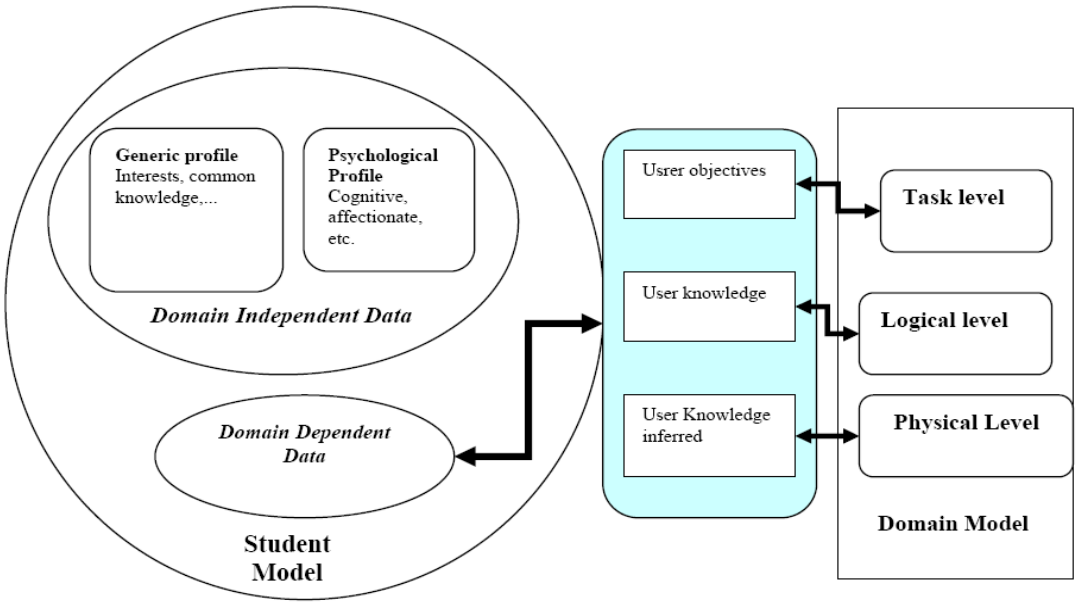

Figure 3 - Benyon's Student User Modeling Architecture (Martins, et al., 2008)

This work does not concur with all the data structures that Benyon devised (Martins, et al., 2008), not just because the domains are different, but also because some DDD components were found worth moving to the DID instead. Specific features of the study, such as the Overlay and

Perturbation models, are not usable in the tourism domain. We also believe that further component organization (merging and dividing information pieces) could lead to an even more comprehensive and organized user model. Despite that, this architecture, along with some ideas that come from the other systems (such as stereotypes) helped forming a concrete idea of how the user model accounted for in the proposed work would be like. This choice also greatly helps the UM technique choice process, because some techniques are automatically excluded, along with the already described fact that some of them are not precisely compatible with the tourism domain.

The described user model architecture, despite being applied to a different domain, presents good basics for any user model, in our point of view. However, we still have to study what will be the domain-dependent information pieces that user models intended to be used in tourism systems must adopt, since Benyon's model accounts for educational-targeted information. The next list presents some information components that tourism-based user models have been dealing with in recent years, according to Fink and Kobsa (Fink, et al., 2002):

I. Past interactions with the system, including: past trips, past visited POIs, and other kinds of past events;
II. Actions performed by the user within the system, such as: rated and commented items, data that comes from purely-social processes, such as travel friends, system navigation's patterns and click stream, and so on;
III. Information about an eventual current trip that the user may be involved in, which may involve knowing the amount of money the tourist possesses and where is he staying at, amongst others;
IV. A comprehensive representation of the system's beliefs about the interests or preferences of the user, regarding the various types of POIs. This information may be represented in different formats, ranging from probabilities to like and dislike ratios. They may even be stored in different kinds of models at the same time in order to increase the assumptions' confidence;
V. Personal information about users that might be important in the tourism domain such as the type and nationality of the food they mostly enjoy eating and religion beliefs;
VI. The grouping of users into pre-defined profiles that allow for easier assumptions to be made regarding their interests. This technique is generally materialized in the form of stereotypes or personas.

It must be noted that not all the previously stated types of information have been equally used in tourism applications, much less quality-wise. This leads to an action space that still can be positively used in proposing innovative tourism systems. This fact, along with the understanding that both UM and RSs still have a long way to go in exploring all possibilities for giving the user an intelligent decision-making experience, grants us the idea that something better can still be done, and the hope that we'll help the world in making that idea a reality.

In summary, this state of the art chapter has presented a brief history of both UM and RSs, also presenting our point of view on the subject by pinpointing the idea of “user modeling as a process”. The second part of the section has divided the state of the art analysis between UM and RSs. In each one of them, current application domains and use cases were presented, as well as the most common techniques that have been applied over the years. One of the most important parts of the chapter is the comprehensive comparison and evaluation of each of those techniques, which tries, in a very simple but effective way, to demonstrate the benefits and drawbacks of each approach. The chapter ends by presenting two concrete conceptual architectures (one for UM and another one for tourism) in which the developed work was based on.

Regarding the conclusions that may be taken upon reading this state of the art, it is clear that, concerning UM techniques, they are very dependent on the domain area of application and the type of user harvesting that needs to be performed. Following that important filter, it is a matter of choosing the technique we feel most comfortable working with, along with the time available to do so. In relation to filtering techniques, the choice is a little easier. With no doubt, hybrid techniques successfully overpower the others, by allowing several theories to be applied simultaneously.

The next two chapters will take the reader throughout the actual work performed in this thesis. The following chapter, Proposed Model, presents the described work in a conceptual and theoretical point of view, while the Implementation section concerns technical specifics of every developed component.

3 Proposed Model

The main corpus of this thesis is composed by this chapter and the next one, which present all the developed work in two different perspectives. This chapter is intended to provide a high-level, theoretical and conceptual approach regarding all developed theories and components. Although this chapter is responsible for the thorough analysis of UM and RS components, how they relate to each other and how they work, no technical specifics will ever be given, as they are destined to chapter 4. This way, readers interested in knowing about the system in a conceptual point of view will find this chapter highly interesting and suitable, with no technical explanations whatsoever, which may sometimes be too inclusive. This chapter will follow the logic behind the project's perspective of the UM process. With that said, it will now begin by explaining the first part of that architecture, i.e., the user model, the basis for all other systems.

3.1 User Model Overview

The user model itself is the root of the UM process and pertains to the broader user architecture that it features. In a certain point in time, the user model photogram dictates the user image as perceived by the system. Dynamic mechanisms will, in turn, use this information in order to reason. In the scope of this work, this chapter will introduce the user model to the reader in first hand, providing a brief access to its constitution and different components. The most important, intelligent and complex UM components, namely the ones which contain an inference nature, will be further explained within the section 3.3. The user model is a detailed view of several information elements the system knows about the user. The model that we created got part of its influence from the "User Modeling Architecture" proposed by Benyon, as was already discussed in the state of the art chapter. As it was referred earlier, we don't entirely agree with all information hierarchies that the referred model devises, so we made a few changes, in order to better represent our point of view on this issue and also to adapt it to the tourism domain. The following diagram, Figure 4, presents the information hierarchy that's comprehended in our user model:

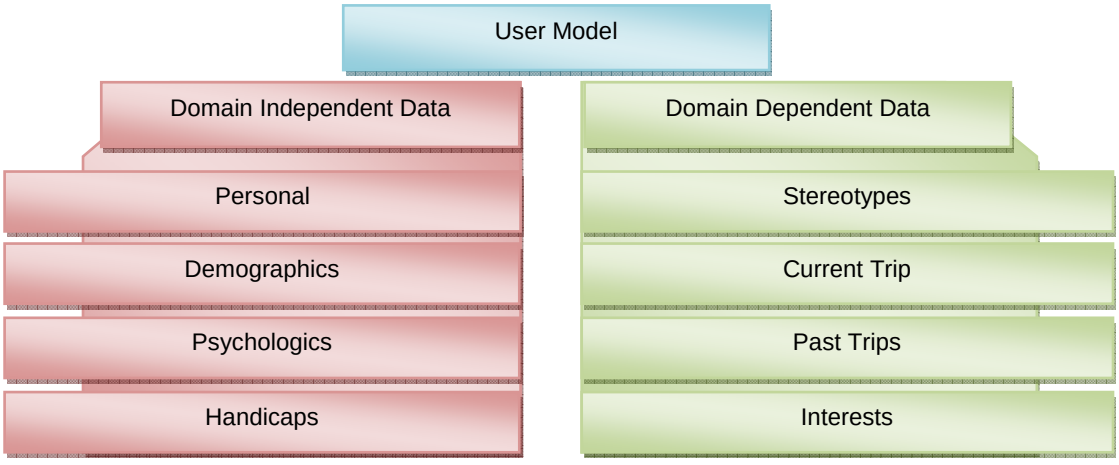

Figure 4 - User Model

The previous diagram represents a “single user” user model. As we will see later in this chapter, there will be other kinds of representation formalisms that attempt to gather information relating to the entire user community. Although some categories displayed in Figure 4 might seem simple and obvious, most of them are not, and others are also further divided into more categories. For those reasons and for the sake of completeness, all categories will be explained.

3.1.1 Domain Independent Data

The first division within the user model, DID, attends to information pieces internally related to the user. It comprises the following components.

Personal: as the name gives away, this component holds user personal information such as name, e-mail and system’s username. This component will not have any effects on the UM processes that will be described later. That is due to the fact that this kind of information does not have an important degree of interest, meaning or semantics associated.

Demographics: this is a slightly more important piece of information. This module is responsible for storing user demographic data such as gender, marital status, age, religion or origin. This kind of information will be used by knowledge discovery techniques, such as Data Mining, in order to find usage patterns which may be useful for the system as well as for the entities or organizations related with the application’s POIs. For example, the repetitive pattern of a certain kind of users to visit certain POIs may trigger a special discount for such visit. Demographic data may also be used as part of the RS, in a longer term decision.

Psychologics: this module is responsible for keeping certain psychological-based information about the user. The user psychological model will be one of the greatest building blocks of the second and third phases of the UM process, namely because it is one of the RSs data sources and represents a new approach in such scope. This module will be further explained in 3.3.5.

Handicaps: this important group of user physical information is responsible for avoiding recommendations that don’t fit a certain user handicap. Handicap functionality, as it is applied in this work, is not popularly used in commercial applications, due to the hard process of acquiring related information from all POIs. We also take this time to remember the reader that handicap attributes are not related with the existence, or not, of accessibility facilities within the POI, a kind of features already successfully proposed by some other systems. Three types of handicaps were found to be reasonably interesting for this domain and were therefore selected. Their value representation is made by a floating number ranging from 0 to 1, in where 0 pertains to no handicap and 1 pertains to full disability:

I. Mobility: this attribute will dictate that only places with special mobility facilities may be elected for recommendation, as well as transportation means that support wheelchairs, amongst others;
II. Blindness: this indicator will remove from recommendation items whose visual appeal is very important or demanding, such as landscapes or movies. Still elected for user approval might be sculpture-based art and music-related items, for example;

III. Deafness: this feature measures user hearing disability and may be important for avoiding music-related items, certain theatrical plays and movies, and so on.

3.1.2 Domain Dependent Data

The categories contained in the DDD, which, as was already explained (2.4), deal with information directly related with the domain, are presented next.

Stereotypes: although stereotype-related mechanisms will be further explained in an appropriate section, we can say that this part of the user model represents the degree by which the user pertains to different stereotype representations contained in the system. Obviously, only stereotypes that the user is minimally related with will be present, using mechanisms that will also be explained ahead.

Current Trip: this module contains information about the current trip the user might be involved in. Important trip elements that were found to be of extreme importance in our system are starting and ending dates, accommodation geographical data, other system user's that might be accompanying and finally all tours within that trip that the user has participated in. Tours, in turn, treat other kinds of contextual information such as money spent, time spent, number of days required, transportation means used, etc. All these data elements will mainly serve as heuristics parameters and restrictions for the final recommendations: for example, routes will be planned with basic directions in mind if a car is present; otherwise, public transportation means will have to be used, which will therefore trigger other kinds of optimization mechanics.

Past Trips: the representation of user's past trips is exactly the same as the current trip, as described previously. In this case, however, the purpose of having this kind of information is different, but equally important. Past trips might be important for executing several kinds of analysis such as user travelling interests, and associate those with all interests' and preferences' knowledge representation formalisms, user trip's patterns and habits, most important travelling buddies, and therefore automatically try to impose consensual recommendations, and most used transportation means.

Interests: this module is further composed by two representation formalisms:

1. Keywords: this type of user interest representation presents a network of keywords associated with the POIs the user has selected. The keywords the system uses, which will be explained in 3.3.6, are very general in nature, are not strongly related to each item and represent a form of uncontrolled knowledge, comparing to other representations. This kind of knowledge presents additional value to the user model because it represents user interests in a slightly different approach than classical methods, like the next one;
2. Likelihood Matrix: this is by far one of the most important user model components and represents the system's assumptions about the user likelihood in relation with the various types of POIs present in the system (Fink, et al., 2002). Because this component is further divided in a complex taxonomy of POIs types which represents an important aspect of the system, it was chosen for that hierarchy to be independently analyzed in section 3.2.

3.2 Points of Interest Taxonomy

Just as it was explained, the user model requires a link between it and the different kinds of POIs present in the system. Given that need, and to present the UM process in a controlled manner, since the other components are much more dynamic and free, it was decided to create a taxonomic component to encompass that aspect of the system. The creation of this taxonomy was too much of a complex semantic effort not to be importantly referenced within a particular section, not only because of that but also because it pertains to one of the most important content basis of the overall system. We believe that a reasonably extensive and complete taxonomy was created, having in consideration the following important properties.

Space-independent: The taxonomy takes into consideration not only the physical domain targeted by the system, Porto and peripheral cities, but also any other tourism-related places. The taxonomy can therefore be deployed in any other system or geographical area.

Customization: The created taxonomy suits the system and the physical domain more accurately than any other already existent. In fact, when the real-world database was included in the developed prototype (see 3.5), POIs have fit almost perfectly within the modeled concepts, with the exception of a few changes that were then found necessary to encompass.

Diversity: There was an effort in order to build an equally rich taxonomy, not only for POIs in the physical form, the classical ones, but also for events, still usually forsaken in this kind of systems. Events provide functionality that, until now, was reserved for other kind of applications.

Organization: There was an effort into considering a mutually exclusive taxonomy, by merging or dividing certain concepts that could otherwise be treated differently: the more we divide concepts, the more difficult it may be to classify them. At the same time, some extension concepts were created which try to cope with unusual situations where other concepts may be found insufficient.

Evolution: The taxonomy will be in permanent evolution. For example, if a certain type of items that until now was placed under one of the extension concepts explained earlier is found to be regularly used, therefore increasing its overall importance in the system, it may be proposed for an independent category. The idea, obviously, is to contain the least number of items in those extension categories.

In the Figure 5, it can be observed all the categories created from the root division, between physical places and events. Only the taxonomy's first level of both places and events might be observed, though. The next explanations serve the purpose of detailing the meaning of each of the already listed concepts, as well as presenting the next omitted levels of the hierarchy; whenever necessary, examples will also be presented in order for the reader to understand the different points-of-view that fundament all concept choices.

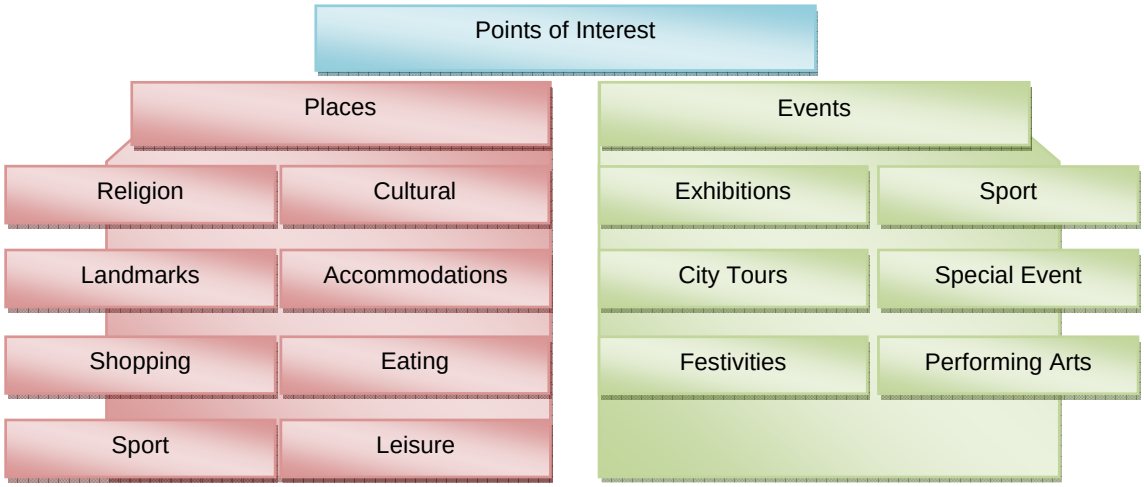

Figure 5 - Points of Interest Taxonomy

3.2.1 Places

Places (in the physical form), as we see it, are fixed, timeless and self-contained POIs. This concept was divided in the following categories.

Religion: places related somehow to religion form a great source of visited places in tourism. Religion places generally present us with great historical buildings or architecturally innovative recent accomplishments. In our model, we further divide this category into four concepts. The first three concepts basically refer to the size of the building. The last one is more general, and can, for example, host religion-specific buildings, like synagogues. There was effort in avoiding choosing religion specific buildings, so instead the taxonomy was kept apart from those kinds of concepts. The chosen concepts are:

I. Churches: the most common type;
II. Cathedrals: great religion buildings, which generally have stand the test of time for several centuries;
III. Chapels: smaller churches, but more numerous than any of the other types;
IV. Temples: it either defines temples (in the true meaning of the word) or it can also represent an escape route from the other concepts, as temple is also a general term that contemplates any religion-based building.

Cultural: cultural-based places were thought of any place which can somehow be the source of human intellectual growth. Religion places might eventually be thought as included in cultural-based buildings, but those were found to be a reasonable unique and important kind of buildings to be elected for having their own category. Cultural places were divided into four concepts:

I. Museums: all kinds of museums, except those containing living beings, which are referred in the next concept;

II. Natural Parks: parks which contain living beings, further constituted by animal preserves and botanic gardens. The first one exhibits animals and may also exhibit plants. This concept was found to be slightly higher-level than zoos, which therefore opens up opportunities for other types of items. The second exhibits only plant species;
III. Monuments: the most usual type of places visited by tourists anywhere in the world, monuments represent the main reason why many people can pinpoint world locations just by watching a picture. For instance, Paris - Eiffel Tower, New York – Liberty Statue, amongst others;
IV. Buildings: this kind of places is an escape route for any building which is not a landmark neither a monument. For example, the Maputo Train Station, considered one of the most beautiful in the world, is classified as a Building, as it is not a landmark, it's more official than that, neither a monument, as it pertains to a clear purpose.

Landmarks: this category may be slightly mistaken for monuments, but that confusion may be eliminated if we analyze the word literally. Therefore, landmarks mean any place inserted into the world's topography that wasn't made just for an historical and aesthetical purpose, like monuments, and aren't as formal and effective as buildings. Landmarks are composed by:

I. Natural landmarks: any landmark which nature created by itself, like mountains, rivers, and others;
II. Human landmarks: enjoyable human-made landmarks, like bridges.

Accommodations: this kind of places is probably the one most studied in current tourist applications. We did not want to reinvent the wheel, as there are already some rich taxonomies for accommodation facilities. The effort was just to encounter the most important kinds of accommodations, from our point of view, which were:

I. Hotels: from the cheaper to the most luxurious, hotels are one of the biggest tourist information sources worldwide, and represent one of the most important type of items that tourists are interested in. Hotels are ultimately divided into low comfort and high comfort;
II. Hostels: hostels are youth-oriented, generally cheap accommodation facilities very popular in Europe. In the proposed taxonomy, though, they mean something more broad: any low-cost accommodation facility targeted for youths which is not a camping park;
III. Camping Parks: this nature-oriented accommodation kind is generally used by the youngsters, because it emphasizes community activities and is generally very cheap;
IV. Pensions: pensions are city-within accommodation facilities with very low comfort, maintenance and cost. They are also much more numerous than hotels in several countries, like Portugal;
V. Other: acting like an alternative for the other concepts, this category can contemplate any other accommodation facilities such as: guest houses, bungalows, apartments, summer houses, etc. Bed & breakfasts, very popular in the United States of America, may also be part of this category.

Shopping: relates to areas (not single stores) with a clear commercial background. If further divided into stores, this concept would become a type of PON (see below the taxonomy). This category was also created having in mind a relatively new type of tourist that has been emerging in the last years: the “shopping tourist”. It was divided into:

I. Shopping Centers: Americanized as “malls”, these gigantic commercial structures host several dozens of different kinds of stores, usually having eating areas, movie theaters, leisure spaces and also high-scale supermarkets like IKEA and such. City markets might be included here too if they pertain to a single structure, which is generally the case;
II. Traditional Commerce: means any area that does not pertain to a single shopping structure, like shopping centers, but do pertain to an area with a clear shopping purpose, generally constituted by many single shops in a given plaza, square, street, or such; for example, Melrose Avenue, in Los Angeles.

Eating: people might not shop or even not visit buildings; but, they have to eat. Although eating is a biological need, it also presents one of the most important travelling factors and reasons of interest. Eating places were divided the following way:

I. Fast Food: generally mostly used by youths, this category means any type of eating place that does not serve a "full-grown meal", like cafeterias, fast-food restaurants, ice cream shops, pizzerias, etc. We acknowledge that this concept might be arguable in relation with plain restaurants, mainly because fast food restaurants have been widening their offer in the last years. Despite that, this division will reveal itself to be very important in the RS, and unarguably gives us more knowledge about the user than with no division whatsoever;
II. Restaurants: just like hotels, restaurants are one of the most developed tourism types of POIs. Restaurants were further divided into regular cuisine and exotic cuisine;
III. Vegetarian: promoting a lifestyle that has been growing in the last years, this category will contain any eating location whose vegetarianism is a key concept.

Sport: whenever one thinks about sport buildings, the idea of a stadium generally arises. Nevertheless, stadiums were thought of a too limited concept to be solely included in this category. On the other hand, this category is not that wide so that it has lots of different kinds of structures; therefore, further divisions were not made within sport buildings.

Leisure: leisure-related places are one of the most difficult categories to classify, but the existence of events rather than only places makes that job a little easier, since lots of leisure activities are events rather than places. Nevertheless, this category’s further divisions contain an extension concept for other unique types of leisure places and activities:

I. Parks: used by all people, parks are a nice place to rest from all the visiting endeavors;
II. Nightlife: any place which can be thought of being primarily night-themed place will be included here. Examples might be nightclubs, pubs, amongst others;

III. Beaches: any named beaches along a coastal area. Also includes riverside beaches;
IV. Swimming Pools: indoor or outdoor swimming pools, either with natural sea water of regular pool water;
V. Other: contemplates other leisure places that, for the sake of importance, were not included in a self-titled concept, such as go-kart rides, theme parks, bowling tracks and ski resorts.

3.2.2 Events

Events represent POIs that do have a time limit, like movies and shows in general, and / or are not contained within a specific spot, like city tours.

Exhibitions: this category represents various types of exhibitions that can be presented. Since this area could create lots of sub-categories, it was decided to summon them up in the following concepts:

I. Industrial: this category deals with commercial and industrial fairs or exhibitions, like product presentations, corporate shows, etc;
II. Cultural: any exhibitions which have a cultural background, like sculpture shows, painting shows and art shows in general.

Sport: just like in the places section, sport events also weren't further divided. The reason here is different, though, as the single possible modeling theory would be to divide this category into the several sports or even competitions that could eventually take place, which would be an excessive effort.

City Tours: although city tours might seem like a too particular kind of events to be addressed singularly, they are also very distinct from all other types of events. Moreover, they are also incredibly tourism-targeted and there are a variety of distinctions that can be observed within city tours, such as historical foot tours, bus tours, boat tours, chopper tours, and so on.

Other: just like the name suggests, this category is to be used whenever the other ones don't completely suit a particular case. Examples of special events may be fashion shows, a pope visit, etc.

Festivities: this category includes all kinds of parties, celebrations, socializations, local costumes and traditions that can exist.

Performing Arts: this concept was one of the most difficult ones to describe because a neutral and meaningful term had to be created in order to contemplate all divisions that follow. Performing arts therefore represent all forms of art that can be performed by a human being and are presented in several kinds of different shows, which includes:

I. Movies: this category represents all movies that may be in exhibition in all the movie theatres present in the system's area of action. Since this domain is a classical application case for RSs, it also presents an opportunity for knowing a little bit more about the user. Here, a decision was made not to pre-define the genres that a movie can have (action, comedy, documentary, etc), as that is always reason for discussion. Instead, we decided that kind of information to be dynamically stored in user-controlled representation formalisms, like keywords, so that it can be later used in the RS. This way we have the ability of correctly model user interests and at the same time maintain a system opened to new developments in the domain;

II. Theatrical Plays: this category, although maybe less problematically categorized than movies, was decided to be treated the same way as the previous category. Therefore, eventual genres of theatrical plays (musical, drama, etc) were left aside to be dealt by other mechanisms in the UM system;

III. Music: if there's a category that less consensus would create if further divided, music would certainly be the greatest. The chosen approach was just to divide music events into single shows (codenamed concerts) versus festivals. Regarding the genres of music, it was mandatory that the abstraction level would stop here, as was already done in the other two performing arts' categories. For instance, metal genre only can be further categorized in several dozens of sub-categories;

IV. Other: this concept was created in order to cope with the never-ending discussion that does exist around what is and what isn't a performing art. From our point of view, this category can therefore be used, for instance, to classify circus shows, magic shows, amongst others.

The described taxonomy, as well as the less referred means of transportation, will be present in the user model because they are required in order to help several UM processes to infer valuable new information about the user, as well as to deliver more intelligent recommendations. However, this does not mean that, in the final system, only these kinds of items will be eligible for selection or navigation. If the final system is to be classified as an ultimate tourism application, with extensive and complete content at all levels, other kinds of items must also be present. Those kinds of items will not be the subject of analysis within this work, because they're not part of the UM process. They mainly consist of several administrative services like hospitals, drugstores and pharmacies, post-offices, news agents, etc. Despite that, in future developments, and since the real world database acquired features many items of this type, a new concept was created to encompass this kind of items, points of necessity (PONs). PONs might be the subject of an independent taxonomy in order for systems to deliver other types of services and features.

3.2.3 Points of Interest Characterization

To characterize POIs, a coherent collection of data elements that pertain to the generality of categories had to be encountered, in order to ease the computational power and complexity of the

algorithms that would work with that information. Nevertheless, a data model devised to cope with such a heterogenic taxonomy could not be made without having in account all the nuances that each category presents. Therefore, our system has two levels of POIs features:

1. Category-independent features: this kind of features will be independent of the type of POIs used and will contain information such as average money spent, schedules, handicap facilities and multimedia elements;
2. Category-dependent features: features like these represent category-specific data elements that need to be addressed if fully-intelligent recommendations and system capabilities are to be present in the application. This kind of features will have a very dynamic and interconnected existence. First, POI features are related with POIs classes whose instances (POIs) will fill those feature values; then, POIs will, or not, be related with one of those features, by applying heritance properties.

3.3 User Modeling Mechanisms

As it was earlier referred, a powerful UM process is endorsed by several components that maximize the accuracy of the necessary user information in a variety of ways. It is believed that, since user information can't all be retrieved in the same manner, our UM technology must therefore be a collaborative effort of several sub-systems, each of them responsible for the retrieval of part of user data (Kobsa, 1994). This is one of the reasons why so many applications don't completely explore their systems: because they think about a UM as a closed process, using a single inference technique or highly associated with a certain methodology. Furthermore, it is also believed that knowing a certain user information space by using more than one method simultaneously successfully increases confidence in existent assumptions and divides responsibility amongst various techniques, which ultimately results in a more backed-up system with more solutions and contingence plans, as noticed in (Fink, et al., 2002), (Kobsa, 2001) and (Kobsa, 1994).

This section, besides specifying the manner by which every one of the previously illustrated user model building blocks contributes for the intelligent nature of the platform, also presents yet other developed techniques. This knowledge retrieval spirit makes up for the second phase of the UM process. We believe that this layer, most of all, represents true evolution of our system against current applications, within our vision of the current state the art.

The knowledge discovery components that constitute our system can be observed in Figure 6. It can be easily seen that tourism applications' upper-level core functions gather information throughout all sub-systems and merge that data into a coherent user profile, in order to generate new information. These forms of representation formalisms represent an advanced view throughout the system's user image and allow for the generation of value added, possibly new, knowledge. The development or addition of new components is forcefully a dynamic subject as well. Although the RS is probably the most important component of a web tourism-related application, the social effort of all components may be used by several other mechanisms.

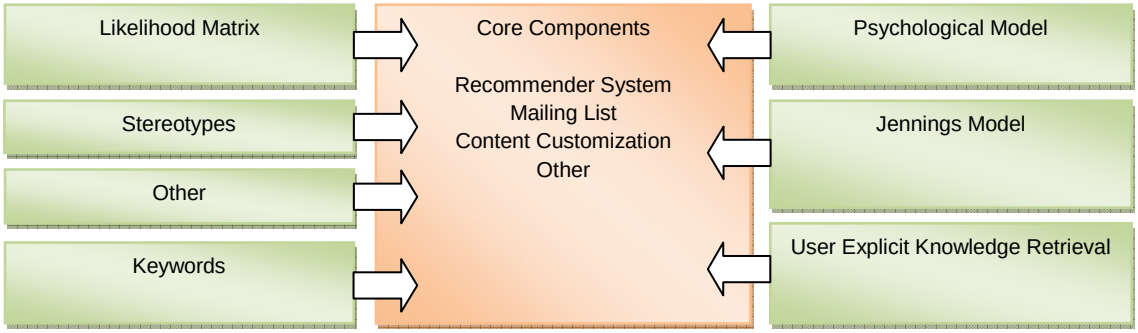

Figure 6 - Reasoning Components Architecture

The core components of the devised project, as illustrated above, are presented in the next section. Self-Organizing Maps, because operate in a community-targeted fashion and therefore different from the other components, will be the first to be analyzed.

3.3.1 Jennings Models

Our UM architecture makes use of a group of two user community Jennings Models (JMs) with very specific purposes. JMs are an adaptation of both neural networks and SOMs which operate in a very simple manner when comparing to regular NNs and have the final purpose of generating two-dimensional representations of the data previously fed into its mechanisms (Jennings, et al., 1991). JMs' practical type of output is mirrored by other techniques, such as dynamic neural networks and dynamic Bayesian networks (both functioning with dynamic number of inner nodes). However, the amount of increased work necessary to deploy those kinds of techniques compared to the value-added knowledge they provide, as opposed to these JMs, was not profitable (Zukerman, et al., 2000). JMs' simplicity at all levels (getting and outputting information) reveals to be the quicker method to be used, even if it hasn't still be used for analyzing purposes. JMs, when properly fed, can give us interesting patterns and associations between items that were previously invisible. Both JMs represent POIs in their nodes, while the relations between nodes represent POIs co-occurrence. The first JM is about system navigation. The node's power measures the frequency of an item being accessed in the system (either searched or visited, not used or selected), and the links between them represent that both POIs were analyzed in the same user session. The second JM has a very similar working method, but the meanings of both nodes and links are different. Here, values represent application commitments, in which the more frequent example is the presence in a visit route of certain POIs. Links represent the co-occurrence of POIs in committing system elements. Figure 7 displays a fictional portion of one of the JMs. In this example, POIs C and D have a very strong connection between them, which means that, in relation to the overall system use, those two POIs have been positively interconnected in chosen routes, in the case of the second JM. The color of each node is just a proper visual reference to their energy value, which means that, again, POIs C and D have also been the ones most visited by users. It's fair to say that high-powered links will most probably be associated with high-powered nodes (Jennings, et al., 1991).

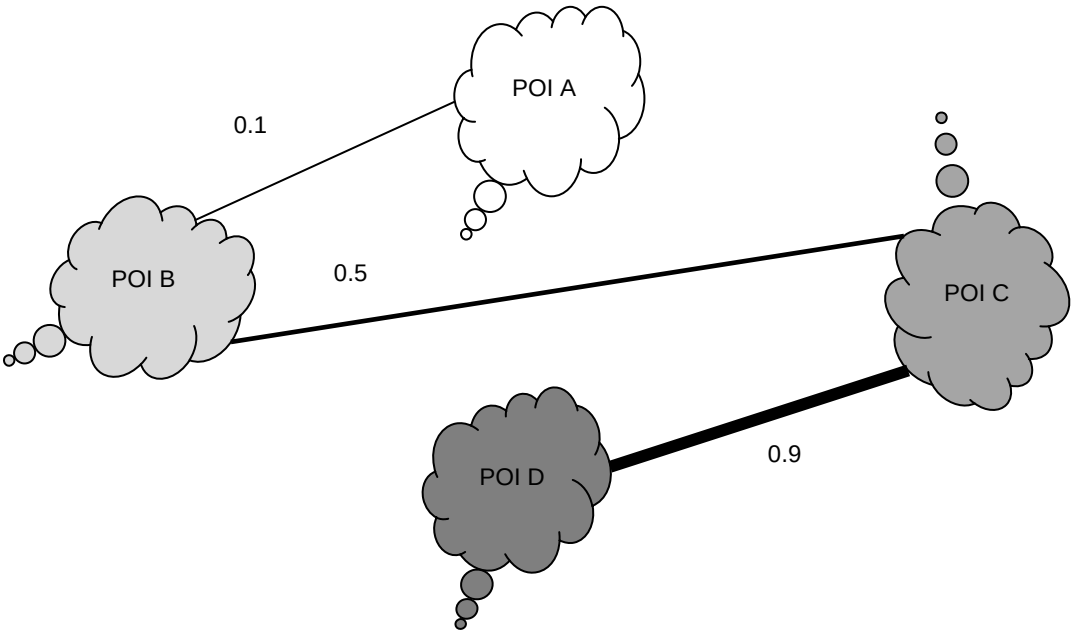

Figure 7 - User Community Jennings Model for Selected POIs Example

Knowledge inference opportunities of the referred JMs are numerous, and can be extrapolated and analyzed in a variety of ways. It was found that the most intuitive way to present its advantages would be to divide them into different analysis perspectives:

I. By type: used / selected POIs versus searched or viewed POIs;

II. By power: although the used information is generally that of the most powerful nodes and links, important analysis can also be made in respect to the least used items;

III. By item granularity: the main representation level of the JMs is the POI, because lots of analysis will be made within that level of specialization. However, analysis may also be made in upward levels of the hierarchy, like, for example, by POIs categories, by places versus events, and so on;

Although much of the previously presented neural networks' purposes are already reasonably thought and attended in current tourism applications, some of them are innovative and do not have precedents in currently seen tourism applications:

- To discover associations between POIs, by analyzing patterns contained within selected groups of items. Discovered associations will, apart from triggering important strategic decisions by related authorities, become an effective help in understanding user personalities and also improving the RS's efficiency. For example, if the system has to complete a certain route to suggest to the user, and all choices seem inappropriate by using all default approaches, the system can ultimately suggest an item which has the pattern of being chosen aside with an already suggested POI, assuming that item hasn't yet been visited by the user, of course. This type of analysis is a very simple but effective

kind of association rules' analysis, which is one of the knowledge discovery algorithms that data mining deals with, as explained in the state of the art chapter;

- Another important analysis that can be made is by relating both JMs. If, ideally, the most used POIs would be the same that were previously viewed, that might not always be the case. In fact, by searching for abnormalities between viewed versus used items, certain interesting patterns may indeed be discovered. For example, mechanisms can be triggered which may include an increase in system visibility of some least selected items, an intelligent detection as to why certain items might be having decreased system use (might lack information, for instance), and so on.

Our JMs share the following properties concerning the development of higher-level neural network systems: dynamism, bi-directionality and computational-containment.

Dynamism: the number of nodes in the network will not be the same every time, as only POIs that have been chosen (either selected or viewed) and relate to each other will be added. That means that the JM will be (in principle) only positively growing with system use. That might not be the case if some special heuristic is decided to be used within this component, such as periodically deleting the nodes and links that don't represent much knowledge in respect with the overall network: for example, links below 0,05. Another important factor that can be changed within the network is the neuron's activation function. Several functions might be available for the nodes to use, depending on the occasion's network objective. For example, if the idea is to make a top N of the most selected items, the JM will be analyzed using an activation function X, while if the objective is to detect relevant item associations, another activation function Y must be used.

Bi-directionality: the network has no clear activation direction, as the nodes are all represented into a single item space; therefore, certain node activations might trigger other activations in any direction, which is actually an intimate and inherent feature in JMs' technology.

Computational-containment: although the size of the maps will most probably grow with time, the data size required by some algorithms that work with the JMs results will not change much. That is due to the chosen representation format of the energies, both from the nodes and the links. Since that information is relative, the system can always filter the JMs the way it wants in order to get only the most important elements, which will end up giving the algorithms only and always a contained part of the entire data space.

3.3.2 Likelihood Matrix

The likelihood matrix is responsible for linking the user with each one of the categories created in the POI taxonomy. It assumes the form of a number from -1 to 1, where -1 means total unlikelihood and 1 represents complete interest (Fink, et al., 2002). The use of a likelihood between types of POIs and the user is not new, as it has already been used in other systems and using different formats (Fink, et al., 2002). However, the techniques employed here allow rarer analysis such as positive / negative likelihoods, thus the choice for the -1 to 1 floating number. Plus, the underlying taxonomy is

much richer and structured than the majority of other tourism systems (WAYN, 2009). This mechanism is the basis for the stereotype module (see 3.3.3) and therefore both components work together in order to provide an over-confident representation of user interests. Although one technique is based on the other, their abstraction level is different, therefore triggering different results by both components. Apart from the usual POIs categories interest analysis, this component also keeps track of user's interests in relation with different types of transportation means, such as car, walk, public transportation, and so on, which are gathered whenever users accept tours containing such moving apparel. The likelihood matrix is an approach that has been reasonably used in recent systems, although manifesting itself in different kinds of representations. The proposed approach coherently represents both user likes and dislikes, by maintaining a negative and positive action space in which system's assumptions can diverge within. Moreover, by defining optimal thresholds, it is possible to identify the most important POI categories within users and therefore trigger adequate response. In several evaluations done using recommending features, the level at which POI classes were considered positively detected within users was 0.50, i.e., halfway into the positive value space. Those tests have considered the faceoff between the amount of recommended material and the regular speed by which likelihoods reach extreme values. Figure 8 shows the overall value space for the likelihood matrix, while Figure 9 shows an example of a portion of a user instantiated likelihood matrix. It must also be referred that the likelihood matrix doesn't relate with the taxonomy hierarchy: the value of a parent node doesn't mean the sum or the average of the child nodes, although the evolution dictates that similar values will most probably be present. This enables users to be applied to situations such as a having a positive likelihood for Religion buildings, while at the same time disliking, for any particular reason, Chapels.

Figure 8 - Likelihood Matrix Representation

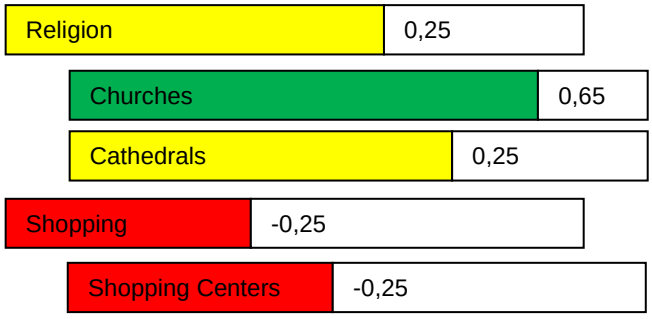

Figure 9 - Likelihood Matrix User Example

The likelihood matrix makes up for a great part of the RS, in two ways, as explained in section 3.3.8:

I. Recommending POIs which relate to POI classes above the defined optimal threshold. This way, positive POIs are recommended with proven theoretical background, while negative POIs are successfully avoided.

II. Recommending POIs which relate to POI classes below the defined optimal threshold but above the neutral value (0). This method relieves the formality and effectiveness of the previous technique and allows for POIs less sustained, but still positive, to be recommended, therefore contributing for the open-world theory, as seen in 2.3.

The likelihood matrix is fed by Application Interaction Triggers (AITs), which monitor important and relevant actions executed by users within activity sessions. AITs are events within navigation sessions that enable the system to acquire hints (with different degrees of certainty, as will be seen in 4.6.3) about the user profile. Examples of AITs might be the application of a search filter for a certain POI category, visiting the individual page of a POI, etc. Depending on the importance of the action, the likelihood matrix is fed by a point system, as explained later in section 4.6.3. The propagation of the referred points follows an inheritance property: by adding points to a leaf-category, its parent-category is also fed. The conversion between that rating system (from 1 to 4) and the confidence ratio (from -1 to 1) is executed by converting the amount of points to be added to the current total of points ever added to that POI class. For example, if Churches have a total of 500 points within a certain user, adding another 4 is one thing; on the other hand, if, for another user, the same class has only 10 overall points, adding 4 represents a great leveling up in the confidence value of that POI category. As will be seen ahead, the user has the ability of changing these confidence values directly, therefore having complete control over results, if desirable.

3.3.3 Stereotypes

The use of stereotypes was mandatory since the beginning of the project, as this component was the initial main focus of the thesis. Stereotypes have been successfully applied in several other areas, and there was the opportunity to include them here as well, since the tourism domain can be analyzed in such a manner. Stereotypes within the tourism domain are not inexistent, but they do not provide evolutionary capabilities as those that were developed and employed here, although they are described in other interesting manners, such as demographic attributes (Rich, 1979). Stereotypes represent a widely used information abstraction mechanism used to group users into categories. The work done in (Rich, 1979) was fundamental in our stereotype component definition, as several ideas were applied in our system. Our stereotype system can be easily explained though a set of development guidelines which originated it. First of all, the POIs taxonomy was **re-conceptualized** and re-organized into hierarchical terms that would better serve as the basis for the stereotype construction. This was done because the original POI taxonomy, which comprises 55 POI classes total, is too large and would force a greater deal of theoretical study regarding the initial stereotypes that would be created. The choice was therefore to create an abstraction of those POI classes into

higher level **POI concepts**. Then, an initial set of **stereotypes** was created, each of them being fully described using the previous concepts, which will form the comparison basis for that stereotype to be linked to a user. Finally, **mechanisms** were created that compensate for an eventual insufficiency that might describe the initial set of stereotypes, as well as the suitability of their terms. To achieve a system that is able to autonomously group its users in an intelligent and sensed fashion is incredibly difficult, mainly because the resulting groups can almost exclusively be achieved using probabilistic data. In our point of view, if reasonable knowledge can be applied to the system, even if little and in the beginning, that will always be better than letting the system make sense of user patterns completely by itself. Indeed, we acknowledge that any initially applied domain knowledge may become obsolete with time, and therefore mechanisms that need to deal with the inherent dynamism of such data were envisioned.

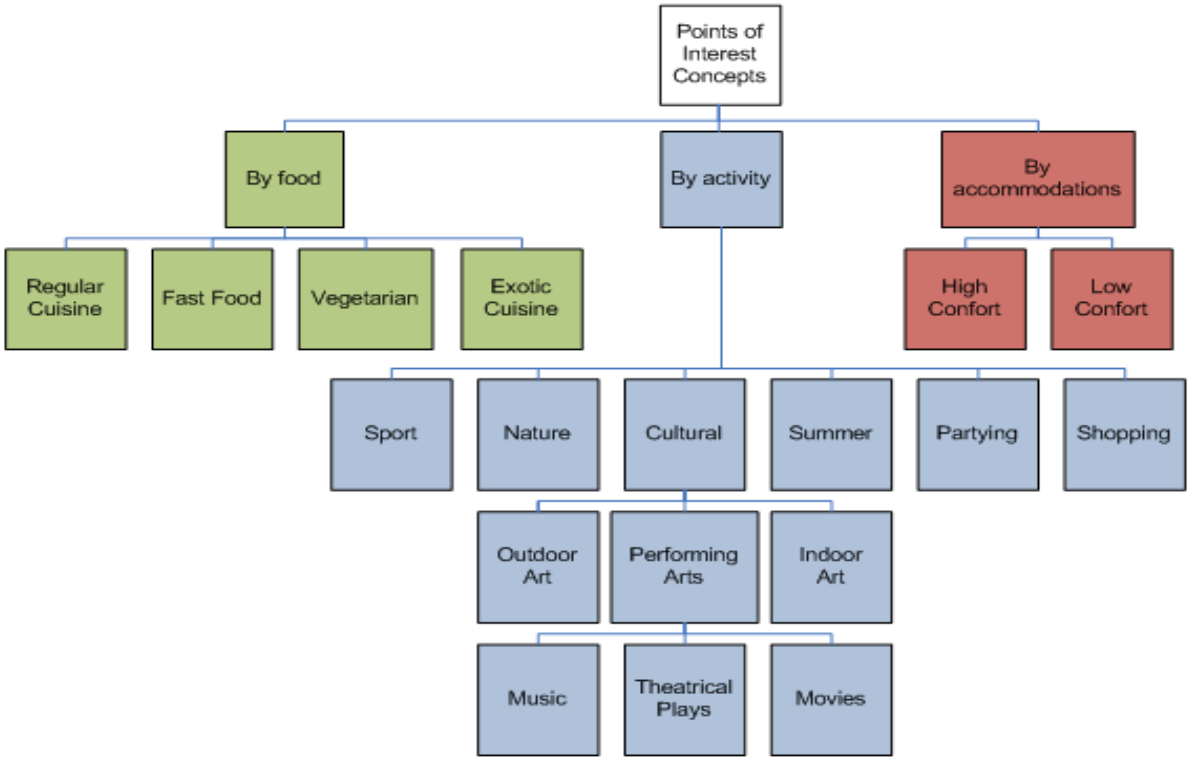

Figure 10 - Points of Interest Taxonomy Re-Conceptualization

As it was said, the first step within the stereotype system development effort was to re-organize the POIs taxonomy into a more comprehensible and stereotype-enabling term hierarchy. This effort implied the merging of conceptually related types of POIs, the relation between those and human psychological cognition and finally a correct mapping between the original taxonomy and the new concept hierarchy, represented in the Figure 10. This process resulted in meaningful and adequate terms to be used in the stereotype creation. The created hierarchy is backed up by the inheritance property, which means that, for example, POIs mapped into the Performing Arts concept will also be mapped into the concept Cultural. The hierarchy was logically divided into three perspectives:

I. Eating: concepts related to eating habits and styles, which encompass all taxonomy Eating categories;
II. Accommodation: concepts related to accommodation facilities, which encompass all taxonomy Accommodation categories and dictate different comfort lifestyles;
III. Activity: the most important of all, concepts related to different activities a tourist may experience when on vacation.

Following is a table containing all conversions between the original taxonomy and these new concepts, or, as we call it, conceptualizations:

New Concept	Original Taxonomy
Regular Cuisine	Regular Cuisine Restaurants
Fast Food	Fast Food Eating Places
Vegetarian	Vegetarian Eating Places
Exotic Cuisine	Foreign Cuisine Restaurants
High Comfort	High Comfort Hotels
Low Comfort	Low Comfort Hotels Hostels Camping Parks, Pensions
Sport	Sport Places, Sport Events
Nature	Natural Parks Natural Landmarks Parks
Cultural	Religion Places, Museums Buildings Cultural Exhibitions, City Tours Human Landmarks, Performing Arts
Summer	Beaches, Swimming Pools
Partying	Nightlife, Festivities
Shopping	Shopping Places
Outdoor Art	Religion Places Buildings, Monuments Human Landmarks, City Tours
Performing Arts	Performing Arts
Indoor Art	Religion Places Monuments, City Tours, Museums Cultural Exhibitions
Music	Music Events
Theatrical Plays	Theatrical Plays
Movies	Movies

Table 9 - Mappings between original Points of Interest Taxonomy and Stereotype Concepts

As the system achieved a richer semantic network of terms that could be analyzed in order to build an initial set of stereotypes, that process was transformed into a simpler task. The big problem would still be to meaningfully group tourists into coherent stereotypes. Although, as it was said, created mechanisms would deal with the eventual poorness of the created initial stereotypes, there was still a reasonably needed effort in grouping tourists into stereotypes that really made sense. After a reasonable but contained research on the subject, six initial stereotypes were, therefore, created, along with the activation conditions that trigger the link between a user and that particular stereotype. Effort was put into avoid over specification of stereotypes, which would increase the time users had to spent choosing one; generality and majority were the terms preferred, creating only stereotypes which overall users feel most comfortable with. Whenever a new user enters the system, the registration form will be responsible for matching him to one of the existing stereotypes within the system (there's more information about that method in a later section). From then on, different user actions will put him into different stereotypes, which will trigger different approaches in several system components, like the RS, the interface itself, etc (Rich, 1979). Users are not constrained to be classified according to a single stereotype, being able to represent a small group of consistent tourist profiles. This can be achieved because the link between a user and a stereotype is not absolute. It is an adequacy value; therefore, the activated stereotypes for a given user in a certain point in time will always be those whose value surpasses a certain threshold, called Stereotype Activation Threshold. If, for example, many stereotypes are activated at the same time, the system can in turn choose only the top-N stereotypes with the highest scores, for simplicity and performance purposes. The input of the described stereotype reasoning system will be the Likelihood Matrix described before (see section 3.3.2), and the result will be a group of the most important stereotypes encountered for a certain user, along with the features that describe those stereotypes, which will determine the nature of recommendations given to that user. As will be detailed in 4.6, stereotypes represent the only UM component which was decided to be evolved at the end of user sessions and not on-the-fly, due to performance issues.

At first, it was thought that correctly naming each stereotype would probably lead to a difficult and unneeded task, besides being impossible to achieve when dealing with automatic or semi-automatic stereotypes (explained later). However, within user surveys and testing phases, it was discovered that users, both in the initial form as well as in the User Area Interface (see 3.3.4 and 4.6), have found the need for a textual description of each stereotype, besides the respective picture. Therefore, and from that point on, it was tried for stereotypes to have some kind of naming attribute to better please and help users in the decision making process. Stereotypes which shall be created in a semi-automatic or automatic fashion, due to obvious reasons, will not be gifted with a name or picture. However, in a later phase, stereotypes themselves might also be source of validation from system operators to ensure a more complete universe of choice in the presentation layer of the application. With that said, the following Table 9, besides describing each of the pre-defined six stereotypes, also presents the reader with a possible name and a description for all of them to make sure that the underlying ideas are correctly passed:

Stereotype	Description	Activation Conditions
Stereotype 1	Youth / Teenager: doesn't really represent a specific age, but rather a youthful, unstressed and uncompromised spirit	Shopping, Partying, Summer, Fast Food, Music
Stereotype 2	High-Cultural: someone with high cultural standards, generally wealthy	Indoor Art, Theatrical Plays, High Comfort, Regular-Cuisine
Stereotype 3	Sport: someone which makes sport and an healthy life an important part of its existence	Sport, Vegetarian, Nature, Summer, Regular-Cuisine, High Comfort
Stereotype 4	Naturalistic: a person very in touch with nature, a free mind, a wild traveler	Vegetarian, Nature, Summer, Low Comfort
Stereotype 5	Family: a family chieftain who has to cope with different tastes, namely its children	Outdoor Art, High Comfort, Summer, Shopping, Foreign Cuisine
Stereotype 6	Low-Cultural: someone who enjoys art but not in an extreme manner like the High-Cultural person	Outdoor Art, Performing-Arts

Table 10 - Initial Stereotype Set

Stereotypes will be activated if one (or both) of the following conditions are met:

1. Stereotype Completeness: the user complies more than 50% (this threshold may be eventually modified over time) of the conditions required for that stereotype to be activated; matching a condition also means surpassing another likelihood threshold. This avoids simple cases such as when a user who only watched a movie one time, for example, to be automatically interpreted as a Low-Cultural stereotype (Stereotype 6). This condition will not be evaluated for stereotypes which exceptionally only have one activation condition, currently inexistent but eventually available due to the autonomous nature of the running system;
2. Stereotype Importance: the user conditions required for certain stereotypes activation represent a great percentage over the total user history. Following the previous example, the user may indeed have only seen a movie once, but that unique occasion may represent the total user available history, which technically means that everything the user has seen are, indeed, movies. Therefore, this condition check has precedence over the previous one, and allows for the cold start problem decrease.

The primary and obvious usefulness for the association of users with stereotypes is that the system may recommend types of items that pertain to the activated stereotype condition group that weren't part of those stereotypes' activations, therefore resulting in refreshing pieces of recommended items. In simpler terms, stereotypes need few data to work with but reveal a much higher degree of possible valuable knowledge in its outputs. This reason has caused the stereotype construction

process to be difficult in the sense that, the more conditions a stereotype has, the more it will be powerful and useful in the recommending effort, but at the same time it increases the possibility for it to be divided into even more stereotypes. A topic related with this one is, as it was said before, the group of several mechanisms for coping with stereotype decay that were defined. These mechanisms will, in a first phase, be treated in a semi-automatic fashion, probably periodically, in order to analyze and control results. The referred mechanisms are:

1. Searching for stereotypes which are not being used or used with very low levels of certainty, and propose them for removal. If any user might be absolutely dependent on any of these stereotypes, actions might be taken so that users in that condition do not end up without a suitable stereotype, for example, by using the mechanism number 4;
2. Searching for conditions not being applied within stereotypes, therefore proposing them for removal from that stereotype. If these removals cause those stereotypes to be composed of only one condition, precautions may be taken, which may include the deletion of those stereotypes, therefore triggering the previous mechanism;
3. Searching for patterns or conditions not initially described for certain stereotypes but which are indeed present in the system. This causes them to be proposed for inclusion within that stereotype activation condition group, therefore enhancing that stereotype's and the whole system's usefulness;
4. Creation of a new stereotype based on the currently available patterns of a user who is constantly in a situation where no stereotype is suitable for him. This mechanism is probably the most important one, because, if correctly handled, will lead to a stereotype network transformation, whose speed will depend on the system's proliferation of users and the overall system's amount of use, which will increasingly better shape the user space into correct stereotypes.

It's in this process that stereotypes can obviously be inserted into the category of clustering and classification techniques. Clustering theories are applied when transforming the stereotype universe, either when new ones are tried to be discovered or when unused cases are detected. Classification theories are used in the process where users are tried to put into the pre-existent set of stereotypes to choose from.

3.3.4 User Explicit Knowledge Retrieval

In this section we'll discuss the UM component most close to the system interface, which deals with what information should the user explicitly provide to the system, and, maybe most important, how should that information retrieval be processed. This subject is studied in different areas, ranging from interface design to software user psychology, but the most important ideas that come out of such studies (Fleming M., 2003) (E., 1999) are the following:

- Users don't want to waste unnecessary time with the software. They want to do whatever objectives he has in mind and immediately quit the system;

- Users are more willing to help the system if they has confidence in its efficiency and are clearly aware of the benefits it brings to them;
- Users don't like long and boring forms (the initial visual impact of them is enough for having a user leave such pages) and prefer to enter information in a fun and intuitive manner, if possible, aided by the system itself;
- In an overall way, users don't want to give information to the system other than the one that they subconsciously think that may be enough and reasonable to give.

In a more practical way, this means that, no matter how many information components we want the user to give, the number of actually requested pieces cannot be above a certain usability threshold, in order to positively attract the user and not bore him. From this issue arises the need of creating intuitive ways to ask the user for information. Some approaches that might be used to deal with this issue are:

1. **Ask only a certain number of items**, while the others will have to be inferred using other mechanisms. Implicitly inferring knowledge about the user is always less trustful, and much more difficult, than the classical way, but is a very common technique that successfully avoids the problem of boring the user, which, in this context is the most important criteria;
2. **Ask certain pieces of information only when they are technically needed**, which basically causes the boredom of the process to be broken down into smaller quantities along the system use. This technique, although asking for smaller pieces of information each time, may also lead to frustration from the user, who might think that every time he gives information to the system, that'll be the last time, which might not be the case. Moreover, information might be asked in situations where the user is not entirely expecting it, causing disappointment.
3. **Try to group information pieces into short versions**, and with those, infer the complete information space. This technique requires coherent modularization of data into smaller but sensed and intuitive information pieces and also implies the existence of knowledge mechanisms in order to convert the short versions back to the complete spaces, therefore adding some degree of uncertainty to the process.

In the proposed system, techniques 1 and 3 will be used; technique 2 will only be used in tours, due to their circumstantial / contextual nature and because each tour represents a different situation every time. The user model proposed in this work is certainly too long to be completely requested by the user implicitly. Therefore, each user model information module had to be thought in relation with what would be the technique to be applied for its acquisition. Table 10 shows the most important information pieces present in our user model (the first level of both the DID and DDD), and for each one, the technique chosen to be used:

User Model Component	Acquisition Technique
Personal	Form
Demographics	Form
Psychologics	Psychology Test and Form
Handicaps	Form
Interests	Inferred and Form
Current Trip	Inferred and Form
Past Trips	Inferred and Form
Stereotypes	Image Association and Form

Table 11 - User Model Components' Acquisition Techniques

Following is a description of each of the used techniques:

I. **Form**: information retrieval through a traditional manner, like a web page form, which includes textboxes, checkboxes, etc;

II. **Psychology Test**: frequently used in psychology tests and games, such as in The Elder Scrolls 3 – Morrowind (Strategy Wiki, 2009), this technique attempts to acquire psychological information about the user by asking easy personality questions. For example, a question for the liveliness psychological feature could be: “Would you prefer to watch a movie all by yourself or play a party game with your friends?”. It must also be referred that the answer to this questions doesn’t mean the placement of the respective feature’s bar on any end of it, but rather halfway of that idea (0.25 or 0.75). This action significantly distinguishes the answer from the midpoint (0.50) and at the same time maintains an action space for that value to eventually change with time;

III. **Inferred**: this type of knowledge will be given by the user implicitly or delivered by one of the user inference mechanisms existing in the system, which in turn uses either implicitly or explicitly given information as their source data;

IV. **Image Association**: this technique pertains to presenting the user with an image that represents an abstract concept that we want to harvest from the user, a concept that also represents a summary of various types of ideas that otherwise would have to be acquired in a very tedious fashion. In the proposed system, image association is used for the user to select its initial stereotype, therefore granting user inference mechanisms with a powerful confidence startup. The following images are some examples that can be used for each of our initial six stereotypes (not necessarily the ones to be used in the final version of the application).

Stereotype 1:

Stereotype 2:

Stereotype 3:

Stereotype 4:

Stereotype 5:

Stereotype 6:

Figure 11 - Image Association Stereotype Examples

The proposed user information retrieval will be unified into a single process, the registration form. We believe that the amount of data asked to the user, along with the information abstraction mechanisms that we developed, will not constitute a tedious task for the user to execute. The inexistence of later needed information is also an advantage for the user, who only has to spend time in that initial effort. We are also vivid adepts of allowing the user to see through the system like a glass, i.e., the user having the ability, if he wishes to, and is able to, of course, of knowing what the system believes about him, along with the ability to modify that information (Cramer, 2008). Indeed, all proposed UM mechanisms were created having in mind that the user would never have access to that kind of knowledge and therefore would be entirely responsible for UM actions. Sometimes, though, the simplest method is left apart: to let the user do the job, if all conditions are met so that can be achieved. Therefore, our system has two possibilities:

Aided Reasoning (not supposed to be the most occurring case) - if the user has the needed knowledge and understanding, initiative, desire and time to review what the system believes about him, and eventually change it, he has the power and ability to do so (later ahead this explicit user profile evolution will be better explained). Needless to say, UM processes are always online and active in the system and will be working with whatever information exists about the user, being inferred or changed by him. This is the reason why the Form is present in the acquisition technique's list of every UM component. Moreover, the user is also noticed that the "on-the-fly" profile evolution may overwrite (in a slightly fashion) explicitly given information. For example, when a user operates in the Aided Reasoning Mode, by making use of the User Area interface, he specifically sets preferences or interests in an effective and absolute manner, pending RS's results towards the changed values. Basically, using the interface means significantly changing old values, therefore provoking significant

changes in the underlying model. Certainly a degree of change that is not experienced and is generally not compared with those of the “on-the-fly” profile evolution. When the on-the-fly profile evolution kicks in, it will again update the user model, but in a much smaller quantity, therefore still evidencing the previously changed information.

Autonomous Reasoning (the most common case) - if the user, because he has no time or knowledge or just because he has a very practical and intuitive way to deal with the system, doesn’t want to be bothered with that kind of issues, the UM process within the system compensates for the user inaction and attempts to build a solid and correct image of him, which indeed represents its fundamental objective and reason for existence.

3.3.5 Psychological Model

The user psychological model, in opposition to demographical models, for example, will be in constant evolution, as the user interacts with the system and gives it traces of his personality evolution. It represents the system’s assumptions about which psychological model best characterizes the user, a model which is not related to the tourism domain itself. This information might also be initially given by the user, as was shown previously in section 3.3.4, to propel the system with a more coherent start. The features selected to be part of this module were based on several psychological models devised by authors along the years (Cattell, 1962), (Jung, 1971) and (Oliver, 1999). Of course that not all concepts devised by those models were selected, as the user model’s focus is not to represent complex psychological features and their combinations. Only four “psychological measures” were elected, ranging from 0 to 1, meaning the two extremes of each feature:

1. **Liveliness**: user's personality in respect to being introvert or extrovert. Close to 1 means extrovert personality, finds happiness in socialization and puts him below others. Close to 0 means introvert, selfish, egocentric, is happy when alone and his interests are above all the others;
2. **Perfectionism**: user's personality in respect to order, control and perfectness. Close to 1 means perfectionist, organized, self-disciplined, precise and control-freak. Close to 0 means imprecise, flexible, undisciplined, impulsive and uncontrolled;
3. **Outdoorness**: user's personality in respect to environmental pleasure. Close to 1 means likelihood for outdoor activities, country-side, sport and nature-related places; close to 0 means likelihood for indoor activities, city-buzz adept, indoor art, etc;
4. **Creativity**; close to 1 means very creative, emotional, artistic, abstract, imaginative and radical. Close to 0 means objective, practical, conventional and conservative.

User psychological models evolve as described next. Each POI category is labeled by one or more of the four psychological attributes. Then, interactions with POIs or POI categories (AITs, see section 3.5.3) will trigger analysis between psychological models of both user and the corresponding POI classes. Analysis results will feed and evolve the user behavioral model, therefore adjusting it and changing the input of all components depending on it, such as the RS. Currently, and by analyzing several psychological profile evolutions within test environments, it was decided that the amount of

change to be made in the user model regarding AITs is of 1 to 20. This means that a certain psychological attribute will vary 5% of the global universe action space towards the new value, in comparison with the old one. Addition or removal of psychological features is very easy to achieve and the values by which POI categories are rated might also be subject of changes along the application lifecycle.

The effective and outputted use of psychological data is very rare in the current computational scene in general, let alone in the tourism or RS specific areas themselves. The user psychological model, along with the use of stereotypes, represents a new approach - **behavioral-based** - in modeling and recommending items to users and contributes to the innovative nature of this work. Regarding the RS, the contribution of the psychological model is made in a very simple fashion. The system finds POI classes which have a positive match against the current user psychological model. This matching process acts by applying a maximum degree of difference (at the present time 0.25) between the two models, one from the POI class and one from the user. The value of each model to be compared is the maximum difference between a single psychological attribute. In practice, this means that a POI class will be positively matched against a user only if any of its psychological features do not differ more than 0.25 of the user one's (see picture 12 below).

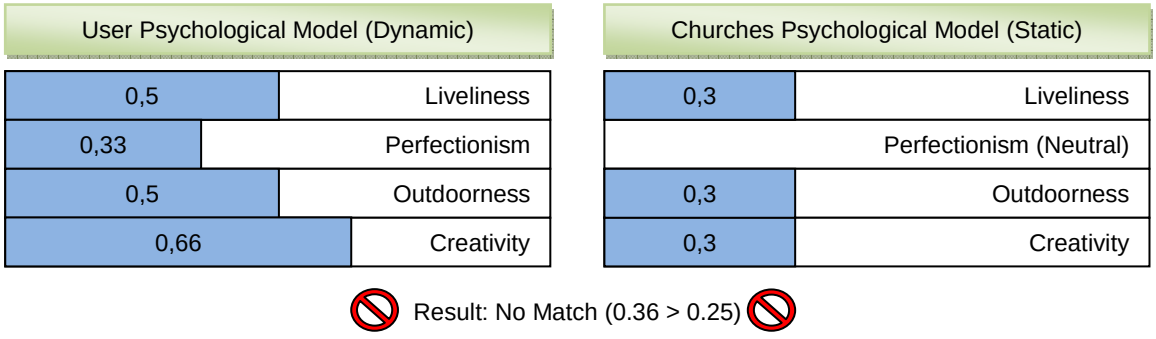

Figure 12 - Psychological Models Comparison Example

3.3.6 Keywords

The concept behind keywords or tags is socially very powerful, because it addresses knowledge about items in a fashion most intimate to the user. Therefore, as we said in the state of the art chapter, it's a mechanism that must be addressed if a system is to be truly user-targeted. The problem with this kind of technology is that it requires significant user participation in order to be fully profitable. The effort in tagging all items currently online was, and still is, an enormous endeavor, if not an impossible one. Only in recent years, due to the web explosion, namely the social web appearance (Web 2.0), the idea got more doable. Users now represent a very important part of every web application. They are spending more time online, they collaborate more, they go to the same website several times a day, and so on (Mathes, 2004). Therefore, some effort of that process was thought of being moved to the users, who could, if they wanted, tag items as they saw or created them. This functionality has to be

extremely well applied, because the user must have the feeling that such effort is for his own profit as well as the whole community. The advantages of using tags or keywords are numerous, and this technology is increasingly being the subject of many studies who relate socialization, culture and ontologies into a relatively new knowledge paradigm. Following are some of the most important benefits of keywords:

- Relate items, using abstractions not contemplated in the information elements addressed in the other components. The meaning and power of keywords exceeds any other kind of controlled information representation, like POIs classes or concepts, addressed in the POIs taxonomy, providing excellent value-added knowledge;
- Featuring of items from the user point of view, which enhances the real world sense of the application and brings it closer to the user; therefore, the user feels more pleased working with the system and builds up that confidence by further collaboration, which in turn benefits the system;
- Keywords are an excellent means of searching for information in a variety of ways. The pure and natural proliferation of keywords can evolve system's information retrieval mechanisms into an extremely dynamic and organic form, which is a top-requisite of large domains such as tourism;
- From the previous two points, slowly and partially remove the importance of system-defined concepts and approaches, which are more strict and controlled, putting the power of the application in the users hands themselves. This advantage is clearly longer-term one;
- By storing and relating keywords with users, tags can propose themselves as yet another means of recommending items to the user, along with all of other methods.

In the proposed system, several approaches for initially inserted tags within the items were devised, without having to take an intensive cognitive process of cataloging them. This way, one of the few disadvantages of tags, which is, along with other technologies, the cold-start problem, could be slightly diminished. The items present in our system are initially and automatically gifted with the following kinds of keywords:

- Keywords that relate to the POI class in which the item is inserted into. Not only the class itself will be added, but all the classes included in the path that goes all the way from the places or events division (excluding those) all the way into the actual item's class. For example, any zoo would have the tags: **cultural**, **natural_parks** and **animal_preserves**;
- Keywords that pertain to all features related with that POI, which were previously added to the respective POI class, as discussed in section 3.2.3. Only category-dependent features will be available to be used in this technique. For example, a certain restaurant might have the keywords **spicy** and **Mexican**, if, for instance, food spiciness and cuisine nationality are included in the Restaurant class features, as those are two traditional kinds of information generally available in current systems;

- Keywords that pertain to special words found within the name and description of the item. This will require a reasonable amount of text mining effort, explained in the next section, in order to extract the required keywords. For example, the bridge “D.Luis I” might have the keywords **bridge**, which is not a POI category but might be a good searching criteria, **Gustave_Eiffel,** a world-wide known name which was involved in the construction of the item and **iron,** potentially a highly referred word among the item description that may cause iron items to be searched all at a time.

Keywords may also be user-added (or else the great advantage of the concept would be lost anyway), therefore creating the need for suitable validation of added tags. Although tags are, in their purest form, uncontrolled and unpredictable, this can cause malicious activities to take place within systems. This way, tags submitted will be kept under that state until application operators can confirm the veracity and coherence of entered concepts and commit the respective changes. Another idea might also be to elaborate an evolving thesaurus of expressions which are to be avoided within keywords and therefore simply apply an ethical grammar check to ensure coherence of proposed keywords.

Tags, besides appearing within POIs they relate to, are mostly presented in the form that people most recognize them: a tag cloud (see next figure). The user community tag cloud appears within the application master page, giving quick access to multi-purpose search criteria. In fact, the keyword component itself is one of the least innovative techniques within the UM architecture. However, it’s the referred keyword extraction mechanisms, whose text-mining algorithm will be decribed next, that bring interesting new features to the tourism big picture (Felfernig, et al., 2007).

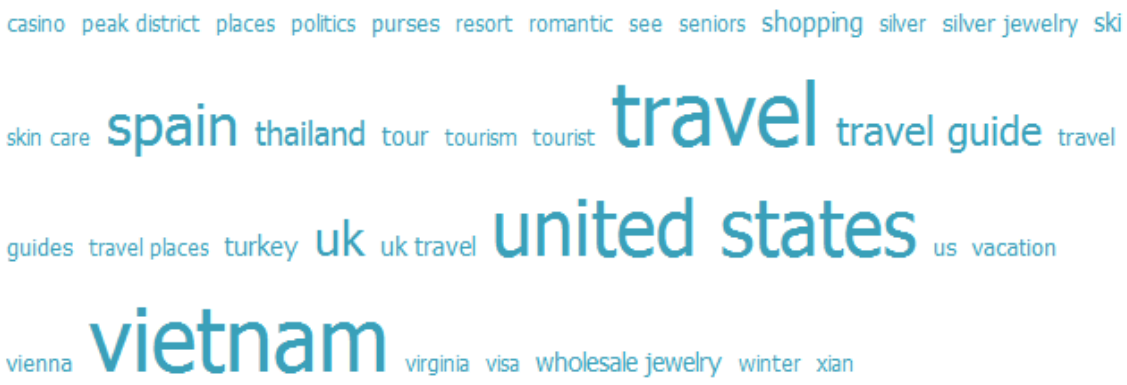

Figure 13 - Example of a Tag Cloud (Travel-Articles, 2009)

3.3.7 Text Mining Algorithm

Text mining was included in the devised system as a means of processing, in a semi-automatic fashion, important keywords included in textual descriptions of the POIs received within the real world database acquired. As text mining represents just by itself a very large domain area for further theories and developments, it was not decided, in a first phase, to apply significant project effort into

this component. However, after the first prototype was developed, it was surprisingly validated and quality-assured by an expert from CENTRIA. Since then, the relevance and importance of this component was substantially enhanced and therefore new and more intelligent forms of text mining techniques were later added.

The text mining process present in the system is a compact but reasonably efficient component that doesn't really innovate in its mechanics, comparing with what have been the current methodologies in the area. The studies made in (Pazienza, 2005) state the difference between linguistic and statistical approaches in the text mining effort, as well as the general assessment that hybrid architectures, which mix both styles, represent a much more capable system. The devised system follows the idea behind that study but divides operations in three comprehensive parts:

I. A **linguistic** filter, which removes English-grammar stopwords (words with no meaning within the tag extraction effort), allowing for tags with no length limit to be successfully extracted, contrary to a seemingly consensus of many systems that restrict the number of grams of extracted terms. In this work, it was decided from the first moment that such heuristic would not be applied for the sake of tag richness, as the tourism domain demands for high-quality and high-complexity tags;

II. A **domain** filter, which searches the previous filtered tags for domain-related knowledge, such as upper-case proper nouns, which tourism-speaking is a very valuable resource, tags already present in the system, matching taxonomy concepts, important numerical tags, and so on. This detection cause those tags to increase their value in output representations, therefore catching the reader's eye;

III. A **statistical** filter, which follows traditional methodologies to rank extracted tags mainly based on the number of tag occurrences in the source text, apart from the already explained domain rank. At the last level, tags are yet ranked by their length.

Since traditional approaches do not use domain filters, not due to rejection itself but mostly because it goes out of the scope of text mining exact purposes, the significance of that component must be endorsed. In the devised project's point of view, the domain filter signifies an incredible increase in the usefulness of the text mining algorithm, allowing result analysis to take place with an increased degree of pace. For the reader to better understand, let's look at an example: in the next figure, a traditional domain-less tag cloud is generated by the project's text mining algorithm applied to a Cathedral of Santa Eulalia corpus (Wikipedia, 2009):

Figure 14 - Domain-less Tag Cloud

As it can be seen, by not applying domain knowledge, the main purpose of text-mining is to select keywords which might have some meaning and therefore avoid unnecessary words, such as stopwords and verbs. All tags acquired throughout this method will be at the same level, as is can be perceived by the regular tag cloud. However, by activating the domain filter, results are scanned for an extra degree of meaning regarding the current domain (in this case tourism) by using several kinds / groups of eligible tags in different sources, as explained earlier. The domain version of the same corpus is displayed next in Figure 15:

Figure 15 - Domain Tag Cloud

The algorithm is composed by a stopword and verb lists created throughout many executions of the process using different input description texts. Other small components were also included in the process as it was found necessary, such as a small stemming technique that deals with regular verbs, singular and plural detection, and many more. Results from this algorithm have shown that, despite the simplicity and even the ad-hoc construction of many of its components, since text mining was not entitled to be the main focus of this project, extracted tags are interestingly user-attractive and may form a very reasonable starting base for new added POIs. Apart from these initial positive results, we acknowledge that this algorithm must evolve into a more complex and sustained form, possibly using more intelligent linguistic techniques such as morphological detection: that is one of the most powerful ways that this project can wide up to.

3.4 Recommender System

The RS is probably the user reasoning component with the most impact on the overall system, because its function is clearly the most important one within the tourism domain: to propose POIs adjusted, adapted and optimized for each user. Although the RS uses almost all components that have been presented so far, it represents a technique and an area of expertise on its own, and must therefore be treated individually. After all, UM techniques can exist without a RS operating in the end of the application pipeline. RSs have been the subject of many studies in the last years, and most of them end up by dividing RS techniques in three major groups: knowledge-based filtering, collaborative filtering and hybrid filtering, as stated in (Berka, et al., 2003), (Burke, 1999) and (Felfernig, et al.,

2007), as was already explained in the state of the art chapter. Content-based filtering is, as already discussed, grammatically misunderstood, and doesn't represent an important margin when compared to the other techniques. All studies point out to the fact that hybrid systems are the best course of action because they incorporate several different techniques, leading to a decrease in certain techniques disadvantage, like the collaborative filtering cold-start issue, and an overall more supported and coherent RS.

In the proposed RS, we acknowledge and concur with those statements, while at the same time organize the different filtering techniques according to a very specific paradigm. In the developed system, instead of specifically thinking about each filtering technique in terms of collaborative or knowledge-based, we made a cognitive effort in relating every designed filtering method with the user itself. Therefore, although our system is clearly a **hybrid RS** and is gifted with all the previously stated filtering techniques, they are, most of all, grouped by their relationship with the user. Our methodology is currently made of seven specific filtering techniques, ordered by the degree of cognitive association between their theories and assumptions with the user itself, which we believe to be the best criteria to address. Those techniques are listed next, ranging from the most to the least important ones, a criterion which is also better perceived in Figure 16.

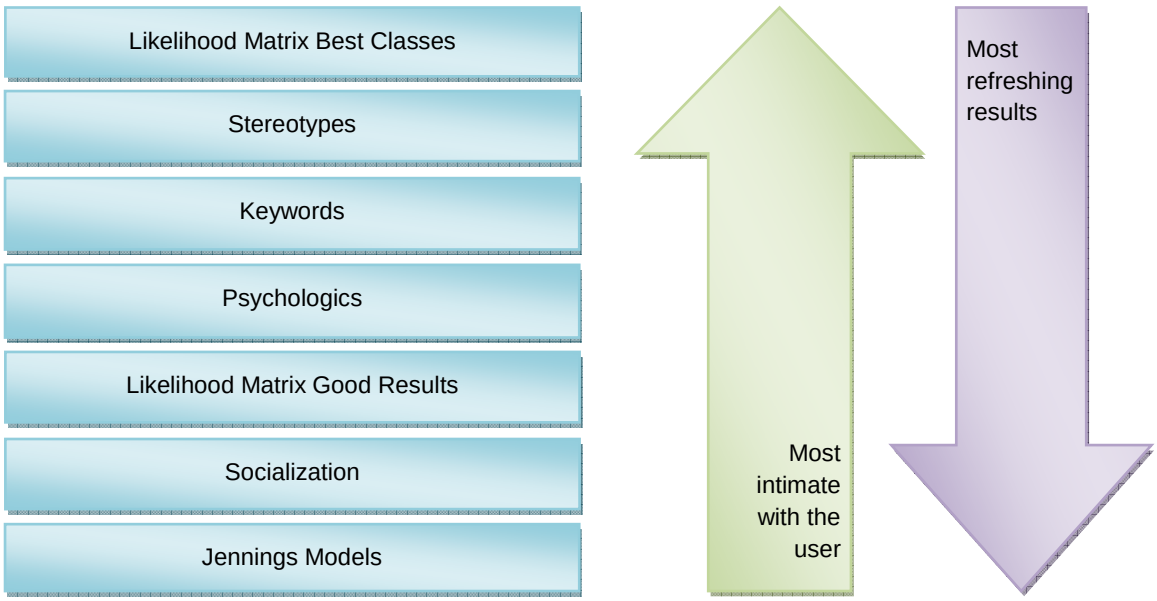

Figure 16 - Recommender System Components

Likelihood Matrix Best Classes: this method returns POIs that pertain to the best classes that match the user likelihood matrix, given the optimal threshold. In the classical topology, this is a kind of knowledge-based filtering;

Stereotypes: this method returns POIs that pertain to the classes featured by the most confident stereotypes set for a particular user;

Keywords: this method returns POIs that feature keywords highly related with the user. This represents a classical case of content-based filtering;

Psychologics: this method returns POIs that pertain to classes that best match a user's psychological model;

Likelihood Matrix Good Results: this method returns POIs that pertain to classes contained in the user likelihood matrix which, although are not as optimal as the best ones, are still found to be positive and interesting when compared to the default and neutral likelihood value;

Socialization: this method returns POIs that relate to the most similar users found against the main user. It clearly represents an example of the classical collaborative filtering approach;

Jennings Models: this method returns the overall most successful POIs, and represents a last effort in recommending items when all the above techniques fail, as this method is not particularly single user-related but rather community-related.

All the previous methods are reasonably straightforward in their execution, with the exception of the special Socialization component, which has to, first, compute the similarity between users. That similarity is in turn based on the first four components of the RS (likelihood matrix, stereotypes, psychological model and keywords). Those components are then compared between the main user and all the other ones, making this component the most intensive and therefore demanding for a lighter solution. There are a number of solutions which may be adopted to diminish the complexity of this social-filtering: (1) reduction of similarity metrics between users; (2) the storing of some static data about similar users, in a periodic fashion, in order to avoid computing them every time and (3) try to limit the universe in which to search for similar users, for example, by allowing only buddies to be analyzed. In the proposed system, it was opted for option (3), as it allows for a better sense of socialization within this domain. Still, the user can parameterize which travel buddies are in fact source of recommendation material for the RS.

It can easily be seen that, besides the classical content-based and collaborative approaches, another kind of techniques seem to arise: techniques related most intimately and cognitively with the user. We may call these techniques cognition-based or psychological-based, but the chosen concept was actually behavioral-based. We believe that, in this field of application, a significant innovation is therefore suggested and contributes for more compelling and personal recommendations. It can also be seen that the more we lose the connection with the user, the more we might find or expect refreshing results from the RS. This subject is not, in our point of view, very well addressed in current researches, which only focus on outputting items mostly related to the user (over specialization). In this work, it is believed that small but existent portions of refreshing results may end up benefiting the final user by evolving his tastes. Therefore, our RS may be user refined / parameterized in order to increase or decrease the amount and importance of these "out of the scope" POIs.

The sequential steps of the RS are as following:

I. First, the RS is filtered regarding the POI classes that are not supposed to be found in outputted results, either when using the POI class filtered RS or by avoiding unnecessary POIs such as Eating and Accommodation places;

II. In the second task, every component gathers its data using its own techniques and POIs are ranked autonomously within each method, setting the value for the **global rating criteria**, which is the most important one;

III. Then, all the previous results are merged and redundancies are erased. Throughout this process, the existence of the same POI in more than one component increases its importance in a criteria called **completeness rating**;

IV. Finally, results are filtered by repetition in relation with the user history and by avoiding those that do not fit certain handicaps that the user might have. For untying purposes, a third rating is also added, to ensure a final contingence plan: the **user rating**.

The output of the RS is therefore a list of POIs ranked by their global rating, followed by their completeness rating and ending with the user rating, for untying purposes. For all upper-level features which do not require further reasoning, the process stops here. However, in the case of the route-generation functionality, other tasks are yet performed. By using complex machine learning and optimization algorithms, the RS results will be once more filtered against the heuristics contained in the current trip and tour building blocks. In this phase, the order by which results were aligned will once more be important, because not all selected POIs might be used. The system will therefore sequentially eliminate all items that are not eligible for choice (for example, a POI might be too far away from the other ones and be too expensive, creating a bad cost-benefit tradeoff) and “pass the turn” to the next items.

Two RS-related issues that are still being researched at the present time are the current underuse of user ratings and how repetitive POIs should be treated. We acknowledge that user ratings must be used in a more productive way within the system and are studying possibilities in order to achieve such results. One of the troubleshooting matters might be, for example, that if by disliking certain POIs, should the user-POI class ratio also be decreased (for now we deny such premise). Another important subject is the question about whether certain POIs should or should not be visited more than once; while typical or immediate theories (which are not necessarily wrong) may dictate that eating or leisure related POIs may pose the main source of repeatable items, a definite outcome of this subject is yet to be achieved. A temporary and always valid approach is also to let the user specifically assign these constraints.

This section ends with the stating that RS results, although majorly and most importantly used by a route planning system component or POI retrieval system, may also be used for the selection of homepage items, navigation suggestions and alerts, mailing lists and other kinds of item selection processes.

This chapter had the main purpose of detailing the functioning of all major components of the devised UM architecture, the advantages and disadvantages of all of them and the decisions that led to such choices, according to the previous state of the art analysis. Despite many conclusions have already been necessarily taken upon explaining all processes, it’s mostly chapter 5 that will be

responsible for a much more thorough analysis of all outputs and results that can be perceived after all work done, along with the overall project's conclusions. The following chapter can be seen as a technical, lower-level version of this one, including also the subsections related with the real world database deployed, as well as the whole developed prototype.

4 Implementation

This chapter will present the developed work by using the second point of view proposed, one with a clear technical nature; plus, it will also describe the prototype created in order to put into practice the proposed model, as well as the underlying real world database. It is felt that this technical view throughout the system will be useful for all readers who wish to know certain specifics about the work done, namely regarding technical documentation, algorithm definition, amongst others. All explicit code versions of the presented algorithms are presented in Attachment I. With that said, this chapter appears as a low-level explanation of all components already explained in section 3, and will, thus, follow the same structure and order when detailing those techniques.

4.1 User Model Overview

Since this is a technical chapter, the user model overview will now be presented with a little more level of formality. In the previous chapter, the user model was presented according to the model proposed by Benyon (Martins, et al., 2008), which was followed superficially; here, though, it is presented in a semi-E-R fashion, for presentation pleasure. The specific E-R models will be divided along each containing component throughout the rest of the chapter, while the complete one is presented in Attachment II.

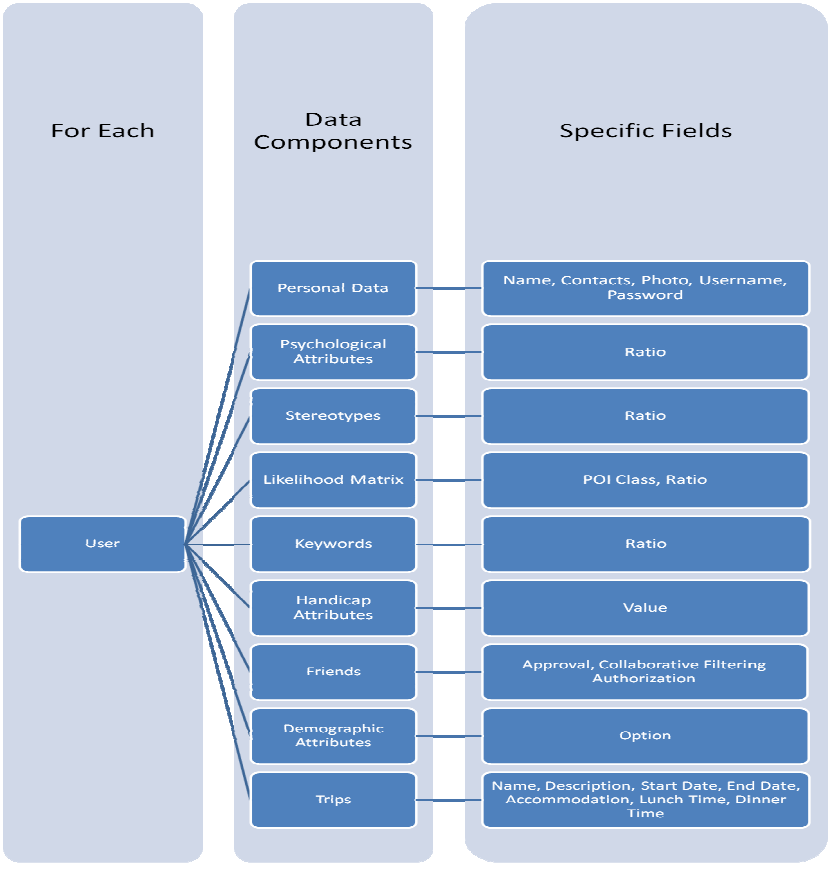

Figure 17 - User Model

As was already done in the last chapter, the more complex UM mechanisms will be described in a more appropriate section further ahead, as they contain several processes that need explaining in a more technical approach. With that said, following are the technical specifics of the least important user profile components. However, these kinds of components, although with less expressiveness in the importance of the system, have been developed with support for later further evolution, mainly regarding their extensibility. Personal Data, Handicap Attributes, Friends and Demographic Attributes models will now be described.

4.1.1 Personal Data

Personal data about users is stored in the table **users**. This table is the bridge for all user-related data components and mechanisms, as it contains the identification of all users in the system, in the attribute *user_id*. The password is submitted a strength test before being accepted, within the registration form of the prototype. The table also contains user contacts; currently, the system doesn't have the need for more, but subsequent information might be available in the future, like, for instance, addresses. We also remember the reader that certain attributes generally stored here, such as the country, are considered demographic attributes within this thesis, and are dynamically stored in other tables, presented in 4.1.4.

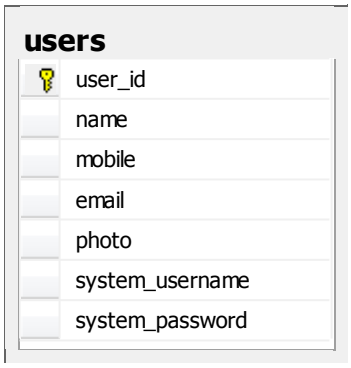

Figure 18 - Personal Information Data Model

4.1.2 Handicap Attributes

Handicap representation is very similar to that of demographics. However, as handicaps require only a value per user association, they only require two tables to make that happen (**handicap_attributes** and **user_handicap_attributes**). On the other hand, since POIs themselves might also be described by handicap facilities, in order for recommendations to be filtered in such way, a third table must also be addressed (**poi_handicap_attributes**). As was already said in the previous chapter, handicap attributes have a very static existence, therefore not having any technical specifics regarding its evolution; however, the user can change them anytime he wants, an operation which is done directly.

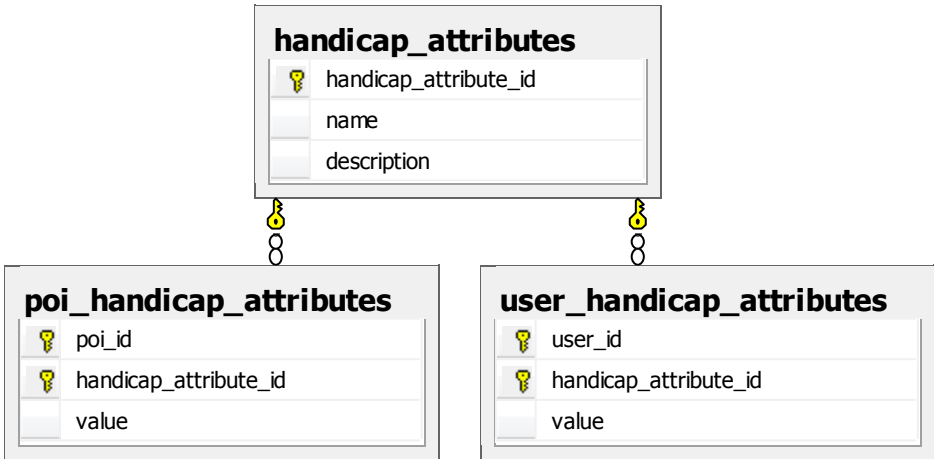

Figure 19 - Handicaps Data Model

Even if handicap attributes are not considered a great UM mechanism, they are further used as part of the RS, in order to avoid recommendations physically unfit with the user. The semantics of a handicap attribute are explained next, with an example regarding mobility:

- POI has 0 mobility: it has no special requirements regarding mobility;
- POI has > 0 mobility: it has special requirements regarding mobility;
- User has 0 mobility: he has no handicap regarding mobility;
- User has > 0 mobility: he has a handicap regarding that mobility.

Given those semantics, it's easy to infer that a recommendation will not occur when the value of a single user handicap surpasses the minimum requirement of the POI, as implemented next.

```
/// <summary>
/// computes the fitness of a set of poi handicap attributes with those of the user
/// </summary>
/// <param name="poi_handicap_attributes">the poi handicap attributes</param>
/// <param name="user_handicap_attributes">the user handicap attributes</param>
/// <returns>the fitness</returns>

public static bool CheckSuitabilityBetweenPOIAndUser(List<poi_handicap_attribute>
poi_handicap_attributes, List<user_handicap_attribute> user_handicap_attributes)
{
        foreach (poi_handicap_attribute poi_handicap_attribute in poi_handicap_attributes)
        {
                foreach (user_handicap_attribute user_handicap_attribute in
user_handicap_attributes)
                {
                        if (poi_handicap_attribute.handicap_attribute_id ==
user_handicap_attribute.handicap_attribute_id && user_handicap_attribute.value)
                        {
                                if (user_handicap_attribute.value > (1 -
poi_handicap_attribute.value))
                                {
                                        return false;
                                }
                        }
                }
        }
        return true;
}
```

4.1.3 Friends

Friends are a small user profile component which keeps user travelling buddies kept for recommendation purposes (as explained in 4.3.8), at this state of the project. However, as will be explained in chapter 5's Future Work section, future Web 2.0 mechanisms will provide a much wider collection of processes using user friends as important data elements. The data model is self-explanatory, with the special attention to the fact that the relationship starting subject is differentiated from the target one to ensure privacy and consensus amongst the relation. Also, when the user accepts a friend as a source of collaboration filtering, the vice-versa operation doesn't occur.

Figure 20 - Friends Data Model

4.1.4 Demographic Attributes

As it was explained, demographics are a user model component with a promising future, as it will be possibly used in further RS refinement and data mining processes. Therefore, demographics have been built with native support so that new attributes might be easily added, as well as the different range of options of each one. The data model is presented next. The table **user_demographic_attributes** links uses to the different demographic attributes, while the content about the latter is stored in **demographic_attributes** (the attribute's text) and **demographic_attribute_options** (the attribute's different options). Initial prototype demographic attributes include gender, age, marital status, country and religion.

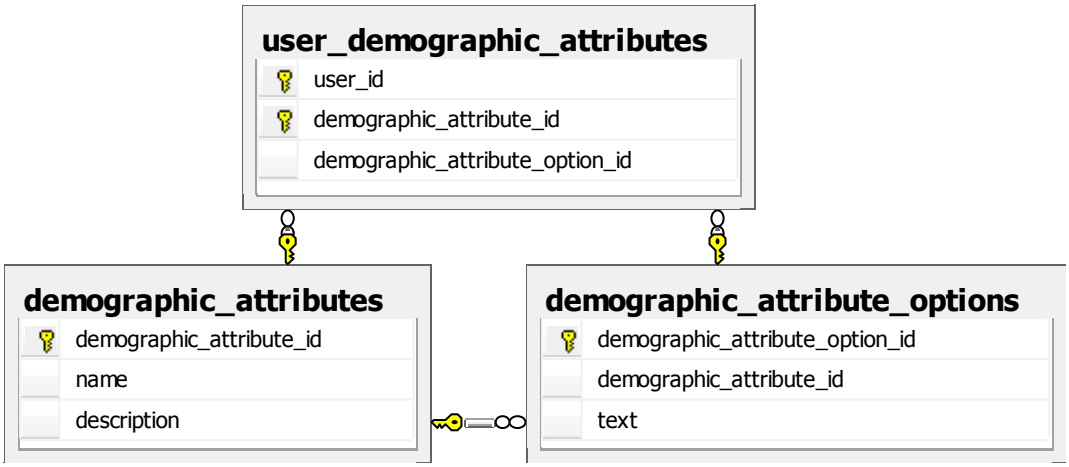

Figure 21 - Demographics Data Model

4.1.5 Trip

The representation of the user's both current and past trips is the same. The scope of this component, as well as the analysis capabilities it may provide in the future regarding user trends, is clearly more targeted to route generation algorithms, which are not an objective of this thesis. However, there was an effort in structuring and organizing trips and tours in a very detailed and future-supporting manner, as is presented next. In a very basis sense, trips are assigned one or more users (**trip_users**) and those (**trips**) have visiting tours associated with them (**tours**). Each tour has, in turn, several segments comprising the different POIs visited (**tour_segments**). However, due to real world restrictions such as traffic and transportation means, more tables are required, which encompass how each of those parts is done regarding mobility facilities (**tour_segment_transportation_segments**, **transportation_segments** and **transportation_means**).

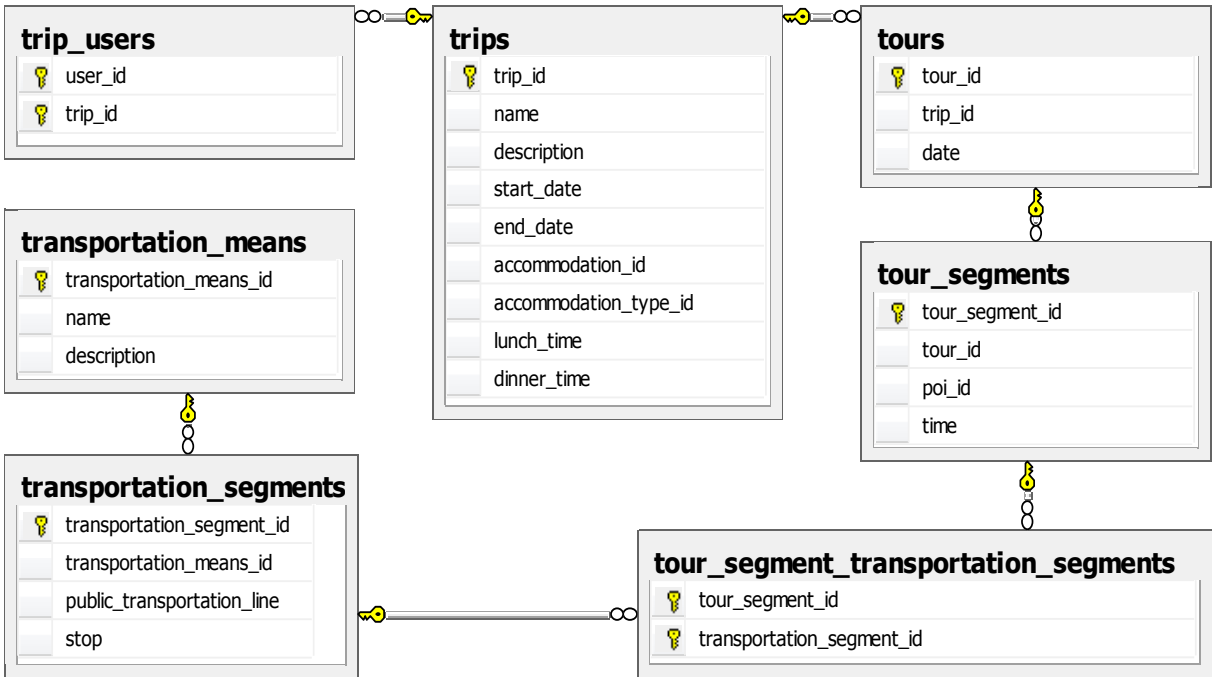

Figure 22 - Trips and Past Trips Data Model

4.2 Points of Interest Taxonomy

Regarding the developed taxonomy, despite the fact that it's a very important aspect of the system, its model is actually very simple. Generally speaking, the taxonomy is just a hierarchy of concepts internally related, therefore creating parent and child nodes. The semi-E-R model of the POI Classes that make up the taxonomy is presented next. Other information elements that relate to POI Classes are psychological attributes (which will be explained in the self-titled section) and features, explained ahead.

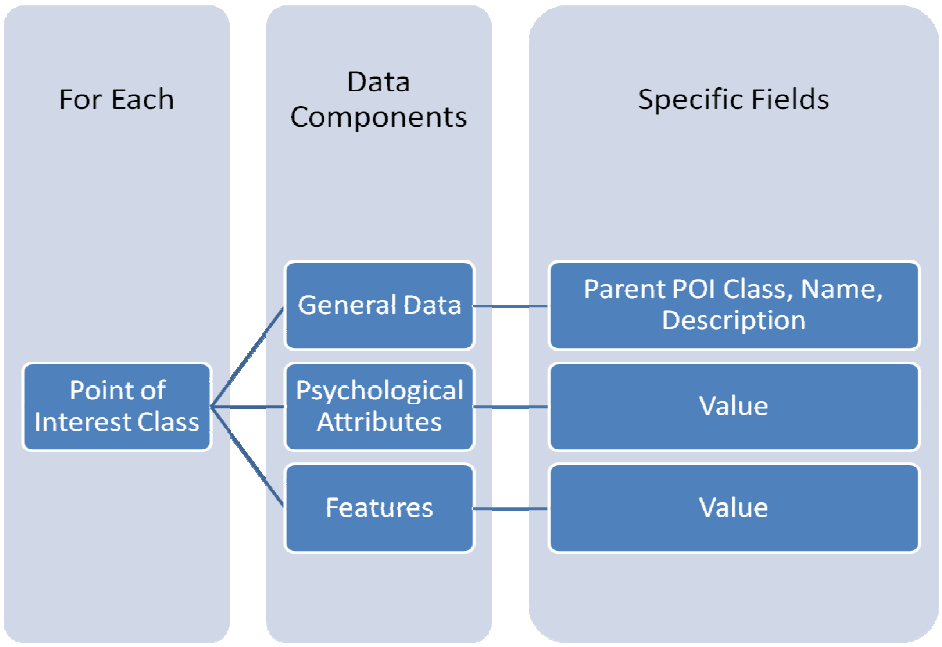

Figure 23 - POI Class Model

With that said, the taxonomy concrete data model is composed of only one self-related table, **poi_classes**. The difference between places and events is not technically visible but rather conceptual, as both kinds of POIs are treated in the same manner to ensure consistency and simplicity of reasoning. As it will be seen afterwards, the differences between places and events will instead be represented in the POI Class features and POI features.

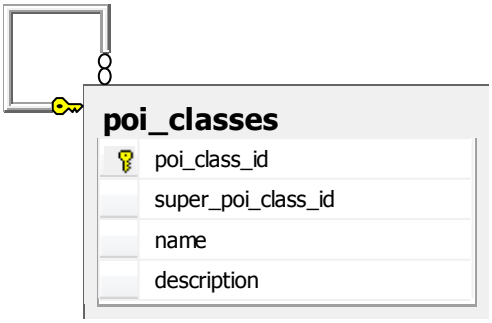

Figure 24 - POI Classes Data Model

We remind the reader that the initial prototype proposal consists of 55 hierarchical POI classes, 39 of them leaf (final) classes.

4.2.1 Points of Interest Characterization

The characterization of the different POI classes is relatively more complex than the taxonomy representation. Features represent the true richness of the taxonomy and what distinguishes classes from one another, as well as particular POIs. Features are stored in the table **features**; then, they are firstly associated with POI classes (**poi_class_features**) and, from then on, they are allowed to be

linked to the respective POIs (**poi_features**). With that said, the difference between places and events, technically speaking, is that events may have a feature called Starting Date, Ending Date or Duration, for example. The inheritance automation ensures that all downward POI classes are automatically gifted with the same features of their parents, if expressed that way.

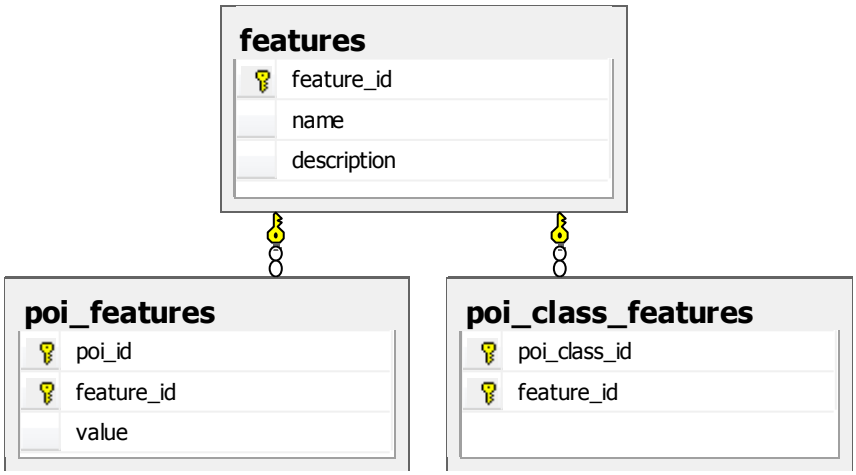

Figure 25 - POI Category-Dependent Characterization Data Model

Apart from the latter kind of features, which are category-dependent, POIs are also gifted with category-independent attributes, which are stored in their main table, **pois**. These attributes are needed for all kinds of POIs, no matter which POI class they belong to; much of their usefulness is only obtained in the area of route generation algorithms. We will also take this opportunity to present the reader with another semi-E-R model that, instead of describing POI classes, describes POIs themselves, which is the system entity which really contains all relevant domain data (see next page). Much of the conceptual map's POI components are not used within the scope of this thesis, such as Distance From Other POIs, Routes From Other POIs, Schedules and Schedule Exceptions. URL's are used for one of the keyword extraction processes, while User Ratings and Multimedia are above all used for prototype purposes. Handicap Attributes and Keywords will be explained in the respective sections later ahead.

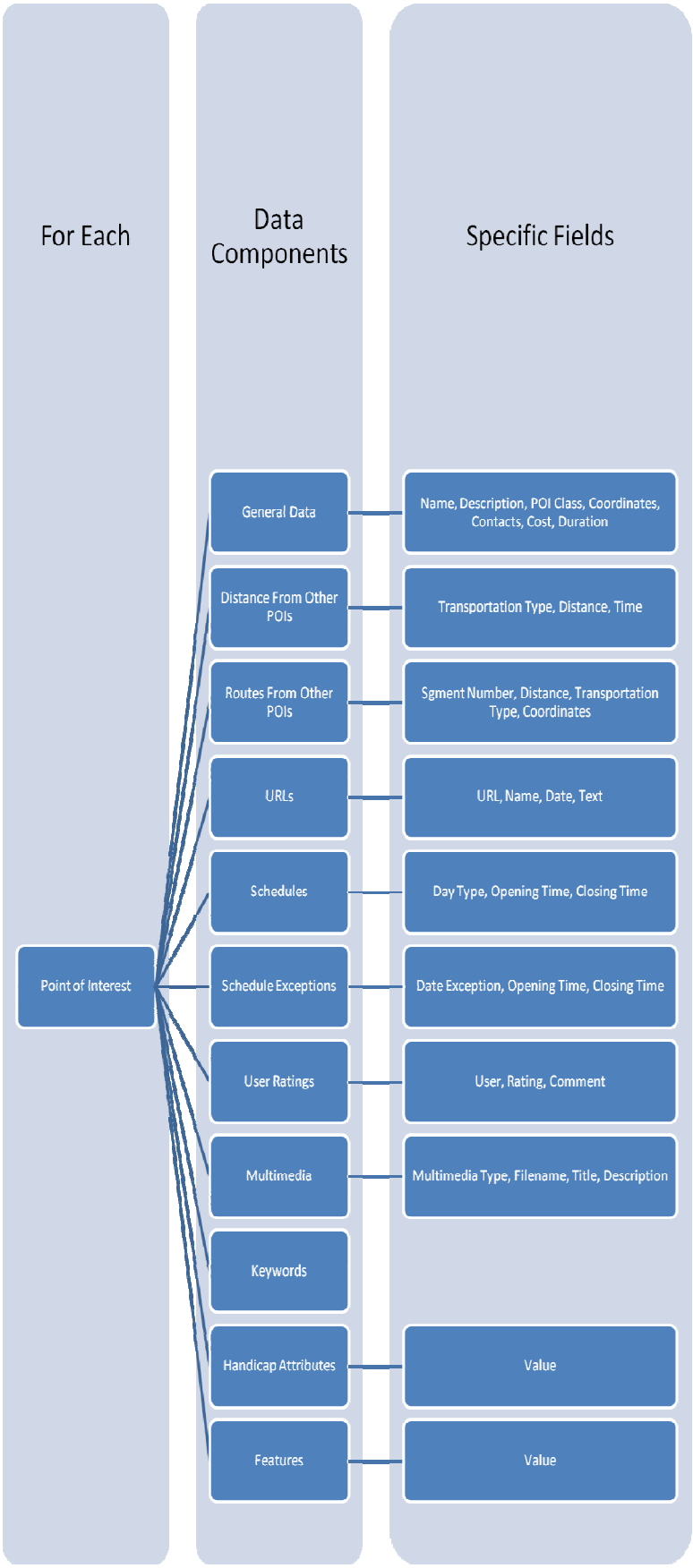

Figure 26 - POIs Model

For completeness purposes, the referred pois table is pictured next:

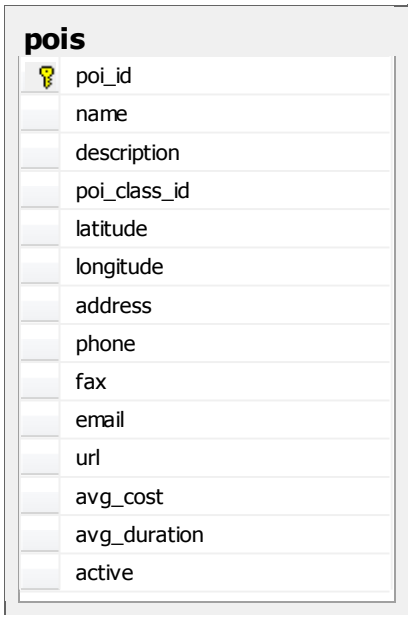

Figure 27 - POI Classes Category-Independent Characterization Data Model

4.3 User Modeling Mechanisms

As the main part of this thesis, UM mechanisms are the soul of the work done and represent the most complex components developed. They ensure a more complete and sustained image of the user, by representing or generating knowledge in diverse forms and through diverse mechanics. Those techniques will be technically described next.

4.3.1 Jennings Models

JMs are fed by the prototype Application Interaction Triggers (AITs) (see 4.6.3). As the current state of deployment of the prototype (see 4.6) excludes the possibility of analyzing the results out of the JMs, the only current process that exists within this component is the information feeding through those same AITs, described ahead. The data model of the JMs is very easy to understand, as it is composed by a nodes and links table, **jm_nodes** and **jm_links** respectively.

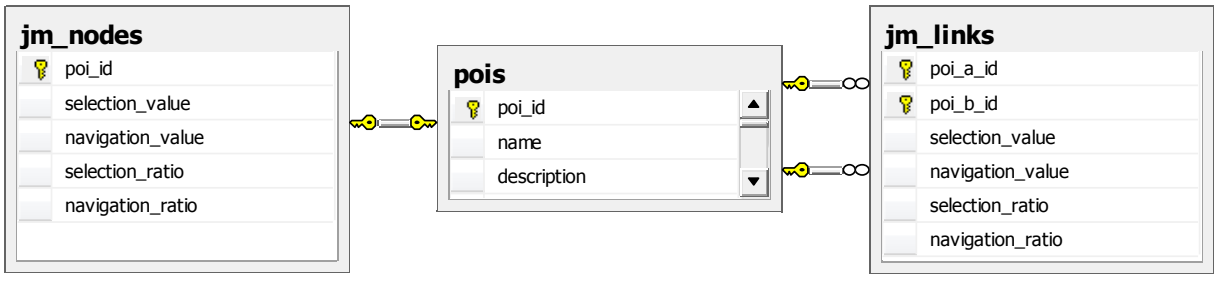

Figure 28 - JMs Data Model

Regarding the process of information feeding, as the JMs deal with data from entire user sessions, such action is done upon session completion, either logging out or session expiration. Then, the list of session POIs visited / searched versus used / committed are sent to the JMs for the respective energies and links to be propagated. The next algorithm shows such propagation regarding the nodes (links propagation is similar).

1. For each POI contained in the session history:
   1.1. Check if the POI is already represented within the JM
      1.1.1. If false, create the node with the parameterized values
      1.1.2. If true, increase node energy
2. Propagate updates into the entire JM, namely the ratio attribute

After the propagation of the node's values, the system will ensure that the ratios represent those same updates in the same ammount of energy. The **value** represents the absolute energy of a node (an integer), while its **ratio** (floating number) represents the relative energy in respect with the entire map, therefore allowing, in the future, different kinds of analysis to be made regarding JMs.

4.3.2 Likelihood Matrix

The likelihood matrix can be considered a bottom-line component, as it serves as the basis for other components, like Stereotypes and the Psychological Model. Nevertheless, its data model is very simple, as it just represents the link between a POI class and the user, in the table **user_likelihoods**. Again, we have an absolute value (value) and a relative one (ratio).

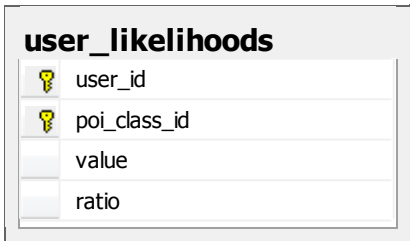

Figure 29 - Likelihood Matrix Data Model

Unlike in the JMs, the propagation of likelihoods into the system, through AITs, is not as simple. That is because the ratio is not performed here in relation with the overall matrix, but instead in relation to the existing data evolution, as was already explained in the example given in chapter 3 regarding Churches. Here, the ratio update means how the current value increase (or decrease) relates with the old value. Therefore, for example, if the current value is 50 and the update will be 5, then the ratio update will be 0.10. This ensures that the edges of the matrix (-1 and 1, respectively) are not as easy to achieve. Also, the existence of hierarchy in the likelihood structure makes it more difficult to apply relative ratios. After the main propagation has been done, the update data will be fed upwards the POI taxonomy, through the method PropagateLikelihoodMatrix. This method will perform that propagation and once again call the main method, operating in a cyclic manner untl the top of the

tree is found. Updates were decided not to be executed top-down, as the underlying assumptions are less trusful: for example, likelihood for Religion-based places might not be equal to likelihood of every and all kinds of Religion-based places. For example, someone with extreme personal motivations for one certain religion might will very doubtly be eager to visit religion-based places other than those relating to his own beliefs. However, in the upward propagation process, we assume the opposite, i.e., enjoying Chapels means enjoying Religion-based places in general, which seems like an assumption with a small degree os risk associated. The main method for likelihood matrix evolution is presented next.

1. Check if the POI class-user likelihood already exists
   1.1. If false, creates it with the parameterized values
   1.2. If true:
      1.2.1. Update the current value
      1.2.2. Perform a 3-simple rule to calculate the adequate matching ratio

$$new\_value = \frac{new\_ratio \times old\_value}{old\_ratio} \quad (1)$$

      1.2.3. The new ratio is conformed to the -1:1 universe
   1.3. Propagation is triggered upwards
      1.3.1. Check if the POI class is 0-level
      1.3.2. If false, keep updating upwards

4.3.3 Stereotypes

As was already explained, the description of stereotypes could not be made with the main POI taxonomy, as there are many classes involved and the reasoning would be substantial. Instead, it was decided to create an abstraction level of those POI classes into more high level POI concepts (**poi_concepts**) and afterwards relate those with the stereotypes (**stereotype_conditions**). The self-targeted key within the table poi_concepts means that, just like POI classes, POI concepts also have a hierarchy of their own, to increase richness and descriptiveness of stereotypes. Therefore, several POI classes gave birth to a POI concept (**poi_conceptualizations**); several POI concepts then gave birth to the actual stereotypes, stored in the table **stereotypes**. The relation between those and the users is then kept in **user_stereotypes**, remembering the reader that users can be assigned more than one stereotype.

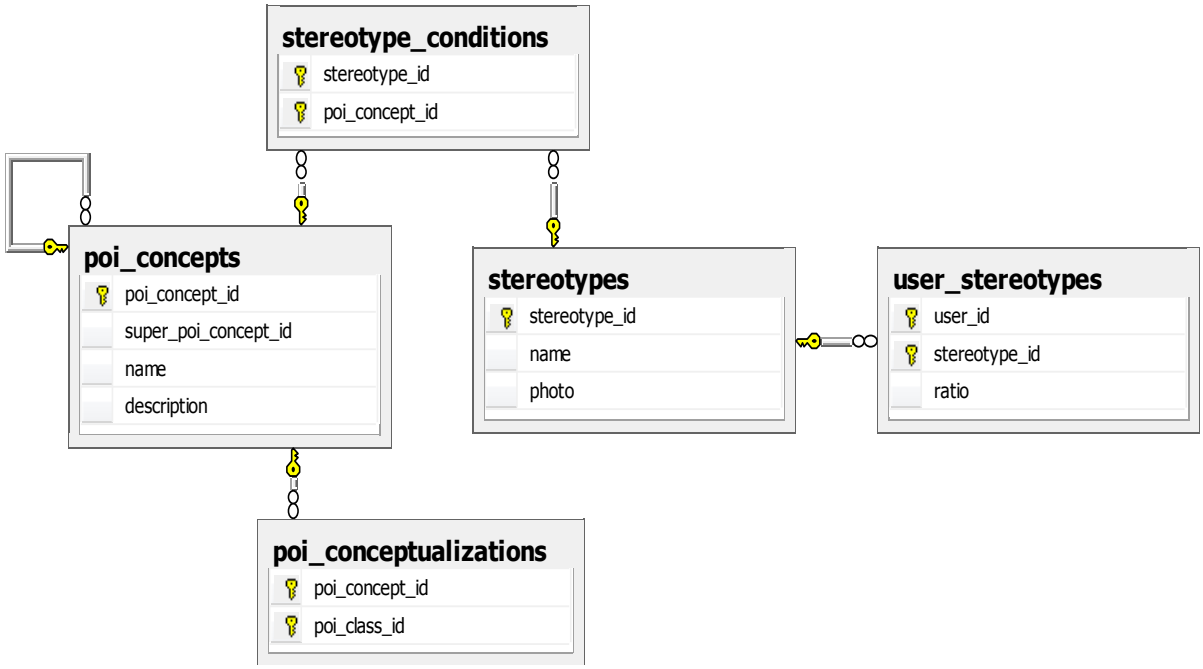

Figure 30 - Stereotypes Data Model

Regarding the mechanics that drive stereotype reasoning, two of the most important processes will now be technically described: (1) the suitability comparison between a stereotype and a user and (2) one the semi-automatic techniques for evolving stereotypes. The first process compares the suitability of a given stereotype in relation with a given user and is therefore performed for all stereotypes currently operating in the system.

1. Collect user's optimal concepts
   1.1. Collect user's optimal classes
   1.2. Check each concept completeness (threshold = 0.75)
   1.3. All those who match the threshold are considered optimal
2. Compare stereotype's conditions with user's optimal concepts
   2.1. Save the number of successful matches
3. For each of the optimal concepts, check for importance (threshold = 0.5)
   3.1. If there's a match, return minimal stereotype requirement (threshold = 0.5)
4. Return percentage of successful matches over all stereotype conditions

It was explained in the theoretical chapter, the stereotype is first checked for completeness, i.e., how many of its conditions are matched in the user profile, by initially gathering the user's best concepts (which in turn gathers the user's best classes). However, after this comparison, the concept importance is analysed and given priority. This concept importance checks if any of the previous completeness matches is especially important in an absolute manner within the current user profile. If any of those concepts is considered extremely important, the stereotype is automatically activated and the suitability is equal to the minimum requirement (or **stereotype activation threshold**). If none

concept is found particularly important, then the previously calculated completeness ratio is returned, which, again, may trigger stereotype activation if equal or greater than the given threshold.

The following algorithm is one of the semi-automatic evolutionary techniques that ensure a correct evolution of stereotypes in the furure. This particular technique is the second one previously presented, which accounts for a process which searches for new conditions that should be added to stereotypes therefore enhancing their usefulness.

1. Make a collection of all conditions that might be successfully added
   1.1. It is equal to all conditions minus the ones already possessed by the stereotype
2. For each of those concepts
   2.1. For each user currently associated with the stereotype
      2.1.1. Checks if one of the user's optimal concepts is the one being matched, using the explained algorithm; if true, counts that match
   2.2. Check if number of matches is significant (threshold = 0.75)
      2.2.1. If true, propose concept for adition to the stereotype

Again, this method works for one stereotype, and should be repeated for all of them as necessary. The algorithm starts by acquiring all conditions that could be new to the stereotype, i.e., all excepting the current ones. Then, it checks the current stereotype user universe for the optimal detection of such concept. If a significant ammout of users within that universe successfully reveal a pattern for using that concept, an ammount called **stereotype conditions proposal threshold** (0.75 at the moment), then that concept is proposed for adition to the current stereotype. Other methods, such as the one that detects low use of certain concepts within stereotypes, act very similarly as this algorithm.

4.3.4 User Explicit Knowledge Retrieval

The first attempt at explicit knowledge retrieval performed by the system is in the registration form, and the science behind some of the information pieces requested was already explained in the Personal Data, Demographic Attributes and Handicap Attributes sections, respectively. However, the most intelligent forms of information gathering are performed by the psychological model and stereotypes, as was already explained in 3.3.4. Those forms are innovative in this field and were chosen using the speed of startup criteria, allowing the user to quickly start using the application. The subsequent processes that allow explicit knowledge retrieval from the user are within the User Area, when users can specifically set ratios for both keywords and the likelihood matrix. Allowing the user to directly set his desired value is more complex and tricky than the automatic profile evolution carried out by the system, as it assumes a linear curve of information fed. When users directly state a value, they can, for example, be assigning something that goes completely in the opposite direction of the current profile, if they wish to; understandingly, this might never happen if the the user follows a logical

use of the system. The next algorithm, for instance, explains how users can direcly assign a ratio to a POI class likelihood (assigning keyword ratios is done in a very similar fashion).

1. Checks if the current likelihood exists in the system
   1.1. If not, it is created with the desired values and the conversion is trivial
   1.2. If it already exists and the opposing ratios are different
      1.2.1. The new value is obtained by a 3-simple rule:

$$new\_value = \frac{new\_ratio \times old\_value}{old\_ratio} \quad (1)$$

The trickyness is in the fact that what is now fed into the system is the new ratio rather than the value, as happens in the normal likelihood evolution, presented in 2.3.2. Between asking users for the absolute value or the ratio, the choice is logical: users can only state a ratio, because of the fact that it is limited, from -1 to 1. Therefore, the 3-simple rule is here applied backwards, not to find the new ratio, but to find the new value that correlates to the ammount of difference injected into the system by the user directly.

4.3.5 Psychological Model

The psychological model conceptual representation is divided by that of users and that of POIs, as presented in the next picture. The main table stores the attributes themselves (**psychological_attributes**), while the relations between those and the project entities are stored in **poi_class_psychological_attributes** and **user_psychological_attributes** respectively. We remind the reader that POI classes might lack certain attributes due to neutrality: for example, it was found meaningless to gift High Comfort Hotels with the Outdoorness attribute, since ideally all Hotels are indoors by default.

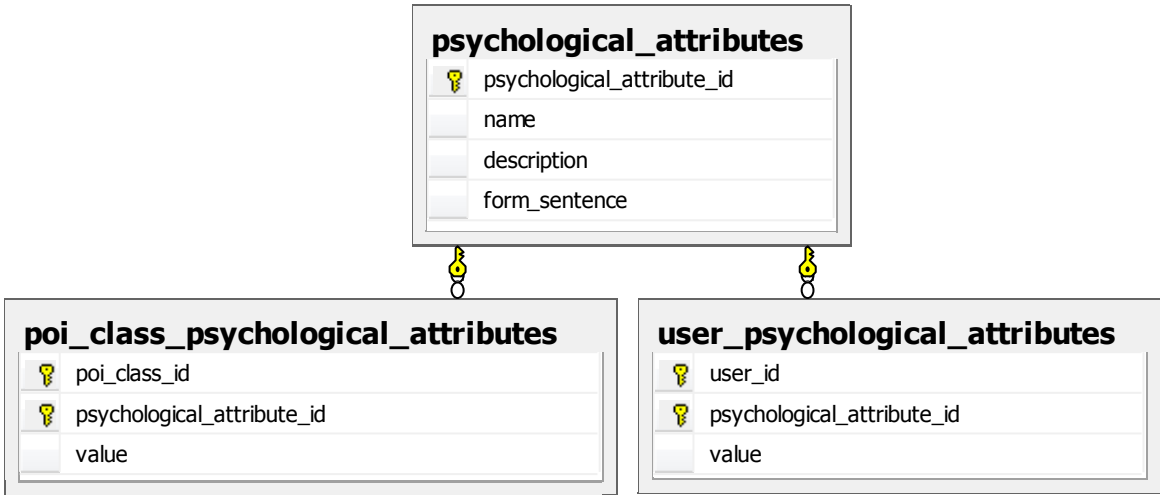

Figure 31 - Psychological Model Data Model

To exemplify how the psychological model of the user evolves (the one from POIs is static), the method that does exactly that will now be described. Its code can be analyzed in attachment I.

1. For each of the psychological attributes in operation in the system
   1.1. The system gathers the user historical value regarding that attribute
      1.1.1. By multiplying each POI class which is defined by that attribute with the user's respective likelihood
   1.2. The system gathers the user current value regarding that attribute
   1.3. The reasoned value is obtained by

$$new\_value = \frac{history\_value + (current\_value \times 20)}{21} \quad (2)$$

This evolution is done after the respective POI classes likelihoods are evolved as well, to ensure fresh data is used. So, when this methods kicks in, it will get the user overall value (which includes the previously updated data and the user entire history) and the most recent value for that attribute. Then, the update is made 5% towards the overall value, therefore slightly changing the user psychological value, while at the same time greatly preserving the last value. The overall value is reasoned by relating the current likelihoods of the user with the respective POI class psychological models: therefore, the more a user enjoys a particular POI class, the more that class psychological model will define him.

4.3.6 Keywords

Keywords are represented in the same 3-table scheme such as psychological models, due to the fact that both users and POIs are described by tags. Therefore, apart from the main table, **keywords**, tables **user_keywords** and **poi_keywords** relate tags to the respective entities in the system, as shown in the following picture.

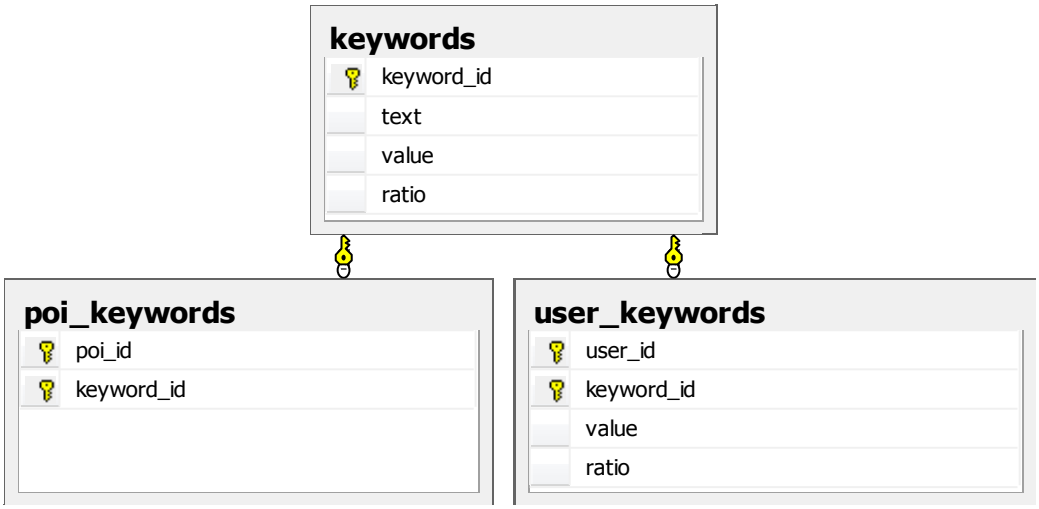

Figure 32 - Keywords Data Model

Regarding their internal mechanisms, keywords operate in a manner much similar to the likelihood matrix. After the user interacts with a particular POI, his profile is fed with all keywords that define that POI, and the relation between him and the keywords is augmented or decreased accordingly. The following method describes just that.

1. Checks if the current user-keyword relation exists in the system
   1.1. If not, it is created with the current parametrized values
2. If it already exists
   2.1. The new value is obtained by a 3-simple rule:

$$new\_ratio = \frac{new\_value \times old\_ratio}{old\_value} \quad (3)$$

   2.2. The new ratio is conformed to the -1:1 universe
3. Propagates the same keyword updates in the community point of view

As it can be seen, the algorithm is very similar to that of the likelihood matrix. The main difference is that, as the likelihood matrix is then propagated up the taxonomy chain, in the keywords what happens is that the personal update will be mirrored as a community update, i.e., the value of the keyword for the entire community will be given the same ammount of update as the personal one, through the method Update. Thus, this evolution evolves the classical tag clouds existent in nowadays social networks (Mathes, 2004).

4.3.7 Text-Mining Algorithm

The text-mining algorithm, as the section name says it, was already superficially approached in the previous chapter. However, the specific flow of the extraction process will be even more described here. Starting with the necessary conceptual model, the only tables required for the execution of the terminology extraction process are those that contain grammatical content, namely **text_mining_stopword_lists**, **text_mining_replacements** and **text_mining_stems**. The first one contains language-dependent stopwords, the second language-dependent verb stems that are to be ignored and the third one contains specific language particularities that go against the algorithm theories and must be replaced by signaling characters (for instance, replacing the period contained in the expression St. Francis).

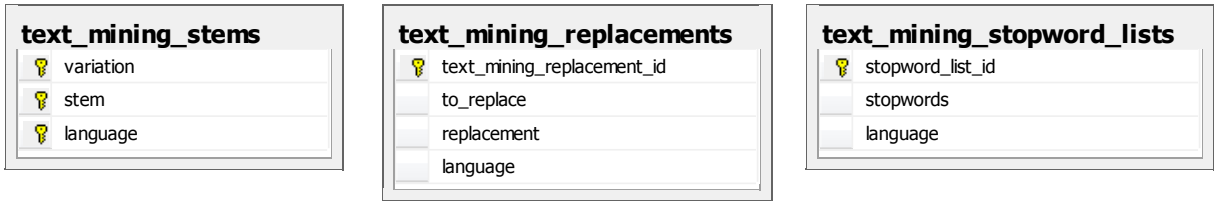

Figure 33 - Text-Mining Algorithm Data Model

The main terminology extraction method will now be presented, which extracts keywords from plain text corpus.

1. Performs language-specific special character substitution
2. Splits source text into sentences
3. Removes ponctuation
4. For each sentence
   4.1. Split it into words and send the set to be analyzed for tags
5. For each keyword obtained
   5.1. Calculate its frequency
   5.2. Check if any of its tuples includes domain knowledge
      5.2.1. If true, rate keyword by 3
   5.3. Check if any of its tuples includes number or proper nouns
      5.3.1. If true, rate keyword by 3
   5.4. Computer final keyword score by:

$$final\_score = (0.90 \times rating\_type) + (0.05 \times frequency) + (0.05 \times length) \quad (4)$$

   5.5. Remove keywords with richer versions
      5.5.1. Analyze all other keywords and check for the complete version of the current plus more meaning

Let's start with the beginninng. As stated earlier, the first step is to perform the replacements that will clean the input text for impurities. Then, the text is divided into sentences and the ponctuation is removed. After that, each sentence is sent to be analysed for important tags in the method ParseSentence, which is explained next. This method basically allows all words to be accepted excepting verbs, stopwords and meaningless numbers, also making sure that the longer a tag is, the better, ensuring tag richness. The next step is to rate every tag, by detecting domain knowledge (3 points) and proper nouns or numbers (2 points). This represents the innovative domain filter applied to the tourism domain, discussed in the last chapter. Tags are also rated (for untying purposes) by their frequency (5%) and length (5%), while the previous rating still makes up for the almost entire rate space (90%). This values were achieved after many tests using rich texts from the Web. The last step is to remove tags which have richer versions, such as removing "herbs" when there is another tag called "red herbs".

1. For each word in the sentence
   1.1. If a word is not a stopword or a verb, but the last one was, then it is a new tag
   1.2. If a word is not a stopword or a verb and the last one was neither, then both tags are merged togrether
2. Sends the words to be checked by structured numbers, which were kept in the last step

2.1. Here, meaninful numbers are kept as tags, and worthless numbers erased
3. Perform residual corrections

It's clear that tag constitution is only halted by the detection of stopwords and verbs (which internally use the previously stated stems). The last keyword is always saved in case the current tag can be added to that one, therefore proposing multi-word keywords. If not, then the last tag is definitely saved and the new one is reset to be proposed as the possible next multi-word keyword.

Finally, the system was untimately gifted with similar methods for extracting keywords automatically from the taxonomy, from URL's previously associated with POIs and from TXT/2 formatted documents (Marques, 2008), this latter one still in initial state.

4.4 Recommender System

The RS is the reason for all other components to exist, and will now be technically described. It uses data that is represented or generated by almost all of the UM reasoning mechanisms. Each one of the data sources that the RS uses basically filters POIs according to the current user profile, depending on the inner mechanics of the corresponding technique. The next example shows how, for instance, keywords contribute their POIs to the RS.

1. Collect user's optimal keywords
   1.1. Collect all keywords whose likelihood is considered optimal (threshold = 0.50)
2. For each of those keywords
   2.1. For each of the POIs defined by that keyword
      2.1.1. Check if the POI type (class) is allowed in this recommendation
         2.1.1.1. If true, add the POI to suggestions, with the likelihood as rating
         2.1.1.2. If POI was already added, check if likelihood might be superior

The method is very straightforward. It collects the best keywords found within a user and recommends the respective POIs which are defined by those keywords, while at the same time rating them by the same value as the corresponding user-keyword relation (in case of a tie, it favours the highest one). This methodology of internally rating POIs with the same rating as the UM mechanism which suggested them is mantained throughout all the other RS data sources, using the **component_rating**. The other RS data sources operate exactly in the same way as this example, given the proper differentes: for example, the likelihood matrix bases suggested POIs on the optimal POI classes detected, while the psychological model bases suggested POIs on the most psychologically similar POI classes found, and so on. Next is the main method of the RS.

1. Filter allowed POI classes
   1.1. If there's no filter, remove just Accommodation and Eating places

1.2. If the filter is high-level, such as Places or Events, perform 1.1 and remove unnecessary POI classes
1.3. If the filter is low-level, leave only specific classes
2. Collect suggestions from all data sources
3. Merge results, accumulating the **completeness_rating** and creating the **global_rating**
4. Corrects the final POI list
4.1. Remove already visited POIs
4.2. Remove physically unfit POIs
5. Order by the **global_rating**, **completeness_rating** and **user_rating**

The first part of the algorithm allows the RS to be used for particular POI classes and not in a general manner, which is the default. This way, users can, for example, request recommendations regarding places only. Plus, accommodation facilities and eating places are left appart of default RS modes, because they represent repetible POIs and only one of them is usually chosen (for example, users have no need of seeing ten Hotels when being defaultly recommended). However, they can be recommended along those kinds of POIs if explicitly chosen in the interface, already developed in the prototype. This step ensures that the following parts of the method don't waste resources looking for unnecessary types of POIs.

The next step is to gather all information from the different RS data sources, with self-explanatory method interfaces. Its this step that shows how clearly hybrid the developed RS is, by joining the different kinds of filtering techiques together (knowledge-based, content-based, collaborative and the new behaviour-based) (see (Berka, et al., 2003), (Luz, 2009) and (Pazzani, et al., 2007)). The merging process unites all gathered POIs, promoting more power to POIs suggested by more than one technique, using the **completeness_rating**. The method still performs the important task of building the main rating factor, **global_rating**, which takes into account the importance of the technique itself. The next step is to remove inconsistent POIs from the almost final list, namely already visited POIs and physically unfit POIs. Lastly, the **user_rating** is introduced as yet another means of untying the resulting list. The RS has no database model as recommendations are not currently saved. However, as the complexity of the system increases and the processes spend more resources, database archival of suggestions for later usage might be a future step to be taken.

The idea of this chapter was to keep the study of this thesis modular. By separating theoretical from technical presentation, different kinds of users can beneficiate from the most suitable document for them. Of course, it is assumed that whoever reads the technical presentation has also read the theoretical one before, to ensure a more proper welcoming into the project. We hope that the data models described here, along with the most important algorithms presented, can shine a light into everyone desiring a more technical approach into the proposed work. Still, as was already explained, all algorithms actual code is presented in attachment I. The next part of this chapter will introduce the reader to the prototype using all the components described until now, plus the real world database that was acquired in order to propel the application with credible content.

4.5 Points of Interest Database

In this brief subchapter, a very important part of the project will be explained, which is generally left apart in such endeavors. However, due to the extreme and utmost importance of POIs and POIs information within tourism, this subject deserves a special spot.

As POIs represent the biggest slice of content information present in the system, with a size degree which tends to increase in such a way to offer its users a very wide space of choice items, it was decided from the beginning of the project that real source data was to be used. Apart from misusing valuable resources, the effort in creating testing data was too great when all information components regarding POIs are thought of, basic data, special features, physical requirements, multimedia, schedules, tags, etc. Therefore, and since this project has a very effective and sustained purpose of being deployed into the real world when all components become available, it was decided that the UM and RS components were to be developed with immediate access to a concrete and real world database concerning the Porto city area..

To accomplish such objective a negotiation with Porto Digital (Porto Digital, 2009) was made, in order to make available to the devised project a substantial portion of the updated database behind their most important tourism-related website, Porto Turismo (Porto Turismo, 2009). Porto Digital is an entity which hosts several official web applications concerning the Municipal Council of Porto. One of the reasons why this very subchapter was necessary is because the acquired database had a very long way to come before being simply attached to the developed components. Since the UM and RS were the most important components of the devised project and their functioning required certain conditions to be met from the database point of view, the data model had to be changed. The original data source received consisted of 26 tables which were scripted from a MySQL database. Apart from the effort in deploying that same data into SQL Server, the chosen Database Management System (DBMS) for the project, and correcting a number of existent errors, there was significant work to be done concerning the decision about what tables would in fact serve the project, as well as integrating that same data. The following tables were selected to be integrated within the project:

- **eventos_fixos** (fixed_events): this table hosts all events which are fixed (repetitive from time to time) or current (are in occurrence or are still to occur) and are therefore elected to be part of the system. All other events are finished and therefore useless;
- **galeria** (gallery): this table hosts the path for all images concerning the application POIs, which were also licensed to be used within the project;
- **locais** (places): this table, eventually the most important one, contains basic information for all POIs; much of the integration effort, which will be detailed ahead, was done having this table as target;
- **restaurantes** (restaurants): this table stores several restaurant-related POIs, therefore acting the same way as the previous table, since in the proposed model all POIs are treated alike;
- **tipos** (types): this table contains all types of POIs which constitute the application's very own taxonomy and are related with the respective POIs;

- **subtipos** (subtypes): this table contains all subtypes of POIs which constitute the application's very own taxonomy and are related with the respective POIs.

Regarding categories and subcategories tables, the task to map those concepts against those of the devised taxonomy was quite simple. This was also the moment where the taxonomy was slightly changed and gifted with concepts that were not devised at startup but were later found important, such as Chapels and Buildings. The picture treatment process was executed without utmost troubles, after harvesting the pictures using a web extraction tool, every picture was associated with the respective POI and the task was complete. The tables which required most effort in integrating data was those relating the POIs, apart from having injected fields with HTML , several kinds of data which to us, are of extreme importance were merged and disorganized into general-purpose fields. The following data elements were therefore treated, sometimes manually, to ensure consistency and adequate and elegant visibility upon presentation layer deployment: addresses, phone numbers, fax numbers, emails, website addresses and average costs. The main reason why all this information integration process was exhausted is that the POIs number of is very reasonable, around 480's, and accounts for a very large data space, mainly in the case of manual adjustments that had to be made. To finish this subchapter, it must be also referred that much of the acquired information was either in Portuguese, English or Spanish, in which not all languages were available. Therefore, and since language-related mechanisms are only reserved to be treated at later phases of the project, it was given precedence to English textual descriptions first and only then Portuguese.

4.6 Prototype

In this subchapter, the portal prototype that was built in order to demonstrate the lower level mechanisms and processes in the UM architecture, as well as the RS, is presented. The application started to be created by the construction of an extremely extensive, complete and functional back-office that encompasses all content that is to be managed by the application. Next, the prototype will be described regarding its technological platform and the choices behind it.

4.6.1 Technological Platform

Starting with the Data Access Layer (DAL), it was decided to use the latest **.NET Framework** offered by Microsoft (3.5 version). At the same time, available features for the 3.0 version of **C#** language were also used, such as lambda expressions. Still, one of the most interesting techniques used in the DAL was the use of **LINQ to SQL**. Language Integrated Query (LINQ) is a relational to Object Oriented (OO) mapping technology that takes a database and automatically creates an OO paradigm to be used instead of SQL. Therefore, no SQL is ever necessary to be used, as everything is dealt with, using classes, objects, properties and instances. Furthermore, LINQ addresses more complex operations such as database concurrent access optimization, transactions, etc. The development of a DAL by using LINQ is incredibly fastened, and operations that required large portions of code (such as 1-to-N relationships) are now made into very simple tasks. The DAL was

also divided in classes representing different portions of the UM reasoning, such as stereotypes, psychological models, etc. The fact of DAL was developed in such platform will allow other project components to be developed much easily, such as the Windows Mobile version of the system. Moreover, Microsoft offers a wide variety of methods for serializing (enabling data structures to be sent over WebServices) information, therefore assuring data interoperability.

Regarding the web application, the same technologies were used, with the logical assumption that LINQ is not used, since no database access is ever made directly within the presentation layer, in order to provide a consistent division of functionality across the system layers. Other technologies also used for this layer were Cascading Style Sheets (**CSS**), for coherent and consistent visual description of elements, Asynchronous Javascript and XML (**AJAX**), to ensure friendly and smooth interaction across the content of pages, Google Maps, allowing geographical information of POIs and finally WebServices, at this time enabling interoperability between rather different technologies, such as .NET and Prolog, used for route generation algorithms by other project components).

SQL Server 2008 was the natural choice to be used regarding DBMSs since it has a very easy connection with Microsoft technologies, such as LINQ, much like MySQL relates to Hypertext Preprocessor (PHP). Finally, the Integrated Development Environment (IDE) chosen to develop the system was mainly Microsoft Visual Studio 2008, along with Microsoft SQL Server 2008 Management Studio.

4.6.2 Portal Areas

The portal itself (user interface of the prototype) was divided by functional areas in order to benefit from a faster deployment. It must also be said that, given the beta spirit of the current state of the prototype, lacking many functionalities that will be endorsed by the other team members, the portal is still very far to be considered at the same level of the current tourism applications, visually, usability and design speaking. That new level will have to be achieved in other project checkpoints. Still, it is felt that the most important and intelligent aspects of the system are already in a very healthy state and those are the system facets that will undoubtedly make the difference upon public release. Such functional areas are presented next.

Homepage: the first page the user sees upon system entering is the homepage. This page presents the project and the application itself. Future improvements of this area may include the following kinds of information: new user additions, new buddies' actions, new uploaded multimedia items, new inserted events, special items of interest, advertising, etc.

Fixed Widgets: fixed widgets are specific content panels which are common to the portal master page and therefore appear throughout all navigation sessions. At the moment, there two developed widgets: (1) community tags panel, which presents a tag cloud relating to the entire user community and (2) a list of the mostly visited POIs, by accessing the respective JM.

New User Page: the registration form makes use of all techniques already discussed in section 3.3.4. The complete size of the form barely surpasses the page height and is very easy to fill in, to ensure that users are compelled to proceed. As was already mentioned, the required elements are: personal data, demographics, psychologics, handicaps and finally stereotypes. All these information pieces are dynamically built and therefore are ready for further completion; for example, current demographic attributes are religion, gender, marital status, age and country.

User Area: one of the most important areas of the application, the user area is a modular place for users to manage all their information. Apart from reviewing data pieces already entered in the registration form, users can also manage friends, trips and the friends-trips relationships. The friend management area contains a very interesting feature called friend recommender (FR). The FR is another practical use for the UM platform, besides the main RS, and proves what has been said throughout this thesis: UM technology is not solely applied to RSs. The FR recommends friends based on four of the main UM building blocks (the same ones that are used in the RS collaborative filtering technique): likelihood matrix, stereotypes, psychological model and keywords. Moreover, the User Area is also the place for the user to review current assumptions sustained by the system, namely the user interests and preferences. UM components that can be managed within this zone are the likelihood matrix and the importance of each keyword in respect to the user. Upon modification of each of the previous components, changes are instantly propagated into the user profile. However, as it was explained before in section 3.3.4, further system interactions may change those manual modifications if other user profile updates happen (user profile updates are explained ahead). Still, significant manual changes will remain powerful for a reasonable amount of time.

POI Area: the POI area is another important modular area that deals with all kinds of information regarding POIs. The main information corpus of a single POI presents its category-independent features, along with the category-dependent characteristics, as mentioned in section 3.2.3. Besides those, special physical requirements, as well as keywords, are also viewed. In the same context, the user may immediately suggest another tag for that POI, which will remain in a temporary state until validation. The other sections of this area contain a list of multimedia items that users can use to visually get to know the POI (at the present time this section regards the pictures what were mentioned before), along with user comments, which may also be introduced, and are not, at the present time, validated. The last POI Area component accounts for geo-referenced information about the POI, by making use of Google Maps technology.

POIs Directory: this area unites in a single page all POI searching modes. The current methods available are:

I. Category Search: search POIs by their taxonomy class;
II. Tag Search: search POIs that relate to a certain tag existent in the system. Only the top community tags will here be available for selection, to avoid an exponential increase in the page size;

III. Recommender Search: the function that activates the RS, this search method returns the list of recommended items for the current user; therefore, it's only available for logged users. Results are viewed either by the default general list or sorted by taxonomy class, to ensure a quick access to each one of the POI classes. Another extremely useful feature concerning the RS is the option to get results relating only to a certain POI class, or by specifying a certain POI class parent node and going through all child nodes as well. As was already explained in section 3.3.8, this is an ideal feature for those who already know what POI classes they want to be recommended against. It also greatly decreases the amount of time required by the system, in comparison to the default RS mode of operation;

IV. Free Search: in this type of search, users can fill any search term that they desire. The system will search for matches along POIs's names and descriptions, as well as tags, features (of the category-dependent type, as explained in section 3.2.3) and taxonomy concepts, either classes or stereotype-related concepts themselves.

Tour Basket: this page allows the user to manage his current tour basket. Apart from deleting items, the user can also change the order of the visit, therefore explicitly telling the tour generation algorithms that the order component of the tour is already taken care of. However, route generating and optimizing algorithms might later change this order while aiming to deliver a better plan.

Tours: in this area the user can request the generation of user tours. The internal mechanisms of such techniques will not be explained here because they are authored by other project researchers and do not constitute an objective within this work. The two main request methods are:

I. **Generate a tour based on the current tour basket items**: by using the items and the order already set in the tour basket, this method will take less time and resources from the optimization algorithms, which will still treat formal aspects like schedules, transportation means, money, amongst others;

II. **Generate an automatic tour based on the user profile**: by profiting from the RS results, tour generating algorithms will undertake a complex task of creating the most effective and perfect tour possible for the current user. This is the time when the complete POI list returned by the RS will be important, in order for the items to be sequentially analyzed, in case of exceptions happen. This is also the method that will employ all decision making constraints available in the system, which are, at the present time: starting and ending times, number of days to fill out tours, money available, walking disposition, lunch options, dinner options, people to account for and finally transportation means options. The meal options concern the decision about whether return to the current lodging facility to execute it or include that in the tour itself.

Figure 34 - Prototype Screenshot

4.6.3 Application Interaction Triggers

Application interaction triggers are events within user interaction sessions which activate subsequent lower-level components. In the proposed system, those processes relate to the evolution of the user profile. Our user profile, as presented before, apart from being extensive and detailed, is composed of complex machine learning mechanisms which are generally very intensive resource-speaking. This way, some UM components are updated in real-time (the majority of them), while some other processes are only triggered upon session termination. That division was decided to be performed as follows:

I. Real-time user profile evolution: likelihood matrix, psychological model and keywords (user-relative and community-relative). The previous components were found perfectly smooth and quick to operate and therefore are instantly updated as the user works with the system. As they constitute a great part of importance of the UM components and RS building blocks, recommendations are instantly customized and user-targeted, leading to a peaceful “on-the-fly” operation of the system.

II. Offline user profile evolution: stereotypes and JMs. While JMs are clearly to be treated upon session completion, or upon tour commitment, because they deal with whole session items, stereotypes were found to be the bottleneck of the profile evolution, taking too much time to execute through several kinds of comparisons between users, POIs classes, POIs concepts and stereotypes themselves. Therefore, its computation was also moved to the end of each session.

In the next list, the actual application triggers and their way of functioning will be presented. This list is highly dependent on current system characteristics like usability, navigability and design. These system features, as was already explained, are, due to the prototype nature of the application, and since they are not part of the thesis objectives, still not as evolved as they will eventually be in the future. Still, the next group of system events is still very reasonable and capable of keeping up with normal user session interactions:

I. **Clicking on a tag**: tags exist in the master page, on search methods and on a single POI personal page. Clicking them increases by one point user-tag relation;

II. **Entering a single POI Area**: by entering or visiting the page of a particular POI, that item is propagated into the user profile by 2 points, as explained: the respective POI class likelihood increases 2 points, the psychological model of the POI is fed into the user psychological model, and all relating keywords are also increased by 2 points, user-speaking and community-speaking;

III. **Searching for a particular POI class**: searching for a certain POI category increases the respective class likelihood in 1 point;

IV. **Free searching**: by free searching and encountering specific and exact POIs, those are propagated into the user profile, using the same methods explained earlier. The value here, as it's only a search action, is only 1 point;

V. **Adding a POI to the tour cart**: adding items to the cart propagates the user model in respect to that POI, using the explained mechanisms, in 3 points;

VI. **Removing POIs out of the tour cart**: removing POIs from the cart propagates the user model with a decrease in 3 points. At the present time, this is the only negative AIT present in the system, and its functioning is performed exactly the same way as a regular positive AIT. In a very broad sense, negative AITs must be dealt with a significant increase in sensibility and caution, to ensure a healthy system evolution, and therefore propose themselves as one of the areas of long-time future development;

VII. **Committing a tour**: by committing a tour, the system assumes the ultimate user likelihood in respect to the requested POIs, as well as the actual and effective visit of the user into those POIs. The propagation value is 4 points.

The following table summarizes the point system that is described in the previously explained application triggers:

Type of Action	Points
Searching	1
Visiting	2
Tour-basket managing	3
Committing	4

Table 12 - Application Triggers Point System

One related issue that is still being thoroughly addressed is the comments submitted by users to POIs. Since there is not, in the present state of the system, a direct relation between users and particular POIs, the information about a positive or negative reaction before a single POI cannot be used directly. The respective POI class can still be used, but, for example, if a person who visited a church didn't like it, it doesn't mean that the likelihood for churches has decreased, or vice-versa. Regarding this problem, some studies may yet need to be studied.

This chapter has ended with the presentation of all work developed within the application prototype, starting from the necessary information integration process. In fact, the core technologies implied within this thesis do not have a visual instantiation. However, to put into practice all theories, and most of all, for users to test them, a real world example had to be created, which has brought even more tasks into the project; those were presented in this chapter. In the previous part of the chapter, the reader was presented with a technical point of view throughout all those low-level components. To finish this thesis, a comprehensive conclusion chapter will now follow, summarizing the project and stating the most important aspects that it features, both positively and negatively.

5 Conclusions

This chapter will take the reader throughout a set of conclusions that may be taken after the development, evaluation and analysis of all the work developed. The aspects that will be, here, discussed are a discussion about the advantages, innovations and strengths versus disadvantages and weaknesses that co-exist in the application, future work that may yet be developed within other phases of the project and finally a much more general summary of all the previous content and the overall document.

5.1 Advantages / Disadvantages

In this subchapter, some informal evaluation guidelines will be presented in order to demonstrate capabilities and weaknesses of the devised system. An analysis process which objectively evidences the advantages and disadvantages of the system against the current state of the art may be of extreme interest, but such an analysis process only can be carried out when the portal has a significant amount of users, which is not the case yet. Anyway, an overview of the system's strengths and drawback can still be performed. Starting with a positive nature, the following system features are highlighted in order to demonstrate the current state of the project capabilities.

5.1.1 Advantages

The following advantages are aspects of the system believed to make the difference against current applications. While total innovation is a welcomed feature in many advantages, sometimes the quality factor concerns to proved theories applied in a different manner, such as the conjugation and effort of many systems that until now have been applied independently and separately or quite simply basic but effective techniques.

Startup quality: the devised system delivers a very interesting and increased startup quality of response concerning filtering features and personalization mechanism. By making use of a clever and abstracting model for initial asked information, as explained in 3.3.4, the application can automatically achieve a coherent user profile. It can, technically, fill around 50% of the user model information within the initial form. This kind of startup quality is not performed by other systems, at least at the same level and asking a small amount of information such as the current model. This way, users can, instantly, get highly customized and sustained suggestions regarding POIs.

Transparency: besides profiting from an automatic UM platform which does everything in turn of the user and lets him free to really appreciate the system, the user can also be invited into viewing, in a transparent manner, everything that the system believes about him. By making use of a friendly interface (the User Area interface, as explained in 4.6.2), users can, in an optimal manner, increase confidence of such critical information and enhance RS results immediately. This way, the user has a privileged view about the system working methods, which we believe to end up enhancing his / her

belief and confidence in the system, in opposition to the majority of the systems which hide its inner mechanisms.

Powerful Recommender System: the RS state of the art has reached a very critical state regarding innovation (see 3.3.8), (Ghani, et al., 2001), (Luz, 2009) and (Pazzani, et al., 2007). The UM platform here presented forms a very diverse basis for RS computation and introduces a new way of filtering complex-domain items, behavior-based. This technique merging causes RS results to be retrieved by using other theories than depleted ones, outputting items with diverse sources and assumptions, increasing likelihood for item commitment. Moreover, by making use of several filtering techniques within the RS, the platform has the ability to overcome eventual problems that certain techniques might have, such as under-confidence and the famous cold-start issue, with results from the other ones.

On-the-fly profile evolution: as almost all of the UM building blocks are propagated immediately (excepting stereotypes, due to performance reasons, as stated in 4.6.3), much of the consequent RS components are also updated with new and current results. Therefore, without damaging any interaction flow during normal user sessions, the user is immediately gifted with up-to-date responses from any application-level process, without even noticing it and without having to confirm such changes. As it was explained in 2.3.2, such instant adequacy of system results is not performed, for example, by TripAdvisor.

Diverse knowledge: the UM components that comprise the system use knowledge representation formalisms of diverse sources. One of the most important ways of analyzing these different sources is through their degree of control. While controlled knowledge (for example in the likelihood matrix and in stereotypes) allows for expected, coherent, sustained and guaranteed outputs, uncontrolled knowledge, existent in keywords and semi-automatic stereotypes, grants users freedom and control in exploring, managing and evolving the way the system works. This balanced nature of knowledge existent in the system accounts for an exceptionally important equilibrium regarding content which benefits all intervenient.

5.1.2 Disadvantages

All the work done within this thesis accounts for a set of applied theories and developed components which are thought of being of utmost importance to the current big picture concerning tourism applications, some of them even bringing a very interesting degree of innovation. However, it cannot be left unnoticed that, as everything, some disadvantages can indeed be pinpointed and detected when some situations might happen (more or less frequently). It must also be referred that many features contained, for example, within future work development, are not considered disadvantages as their development is expected within project growth.

Recommender system classical issues: as was already explained in 3.3.8, some classical downsides of filtering techniques, such as the cold start problem, are successfully avoided and surpassed by the social nature of all RS building blocks. However, this does not mean that such problems do not actually occur. They may, in fact, exist within the system; they are just overcome and overwhelmed by the RS itself. Therefore, even if the RS manages to output a consistent degree of optimal results, those will be even better in the absence of these issues, which may be present.

Stereotype and keyword validation: both stereotypes and keywords contain a validation component in their inner mechanisms that halts system evolution and true power. Concerning stereotypes, the semi-automatic evolution mechanisms explained in 3.3.3 are the most important part of their dynamic nature. However, they require supervision in order to be properly evolved. This situation might change in the future, when the system feels that such evolution is indeed being well executed, a situation which will only become obvious with the real world deployment of the application. Despite that, the issue concerning the correct naming and picturing of each stereotype cannot, ever, be treated in an automatic fashion. Keywords are yet another system feature, which requires validation concerning user proposed tags. However, as they are the most truly free-form and evolutionary component, they force the need for automatic validation techniques, still not existent.

User Area elitism: at the moment, User Area presents user interests and preferences, namely the likelihood matrix and keywords, exactly as they exist within the system. However, that representation format might not be too attractive or adequate for the majority of potential system users (values from -1 to 1). It is believed that developing some kind of component which intelligently correlates that kind of values to a more user-fit representation might be necessary. Moreover, such optimization might end up causing the User Area to be used in a friendlier manner and therefore more frequently, enhancing some UM components usefulness, as discussed in 3.3.4.

Stereotype rigid conditions: stereotypes represent a user grouping mechanism which is well applied within the current system. However, stereotypes possess the power to be used even more intelligently. At the top of techniques that could be used to enhance stereotypes' usefulness is the conditions diversity that forms the stereotype reasoning basis. Currently, the conditions that link a stereotype to a user are contained within the POI concept taxonomy, which in turn uses the original POI class taxonomy, as explained in 3.3.3. However, if stereotypes were yet to be described by other knowledge forms, such as demographic data or the psychological model, stereotypes would have the power to be used in a manner with more sustainment, completeness and richness.

POI classes' psychological model lack of theoretical assurance: as it was explained in 3.3.5, the psychological evolution of the user is made by relating his actions to the respective POIs classes, each one of them was previously gifted with a psychological model of their own. The psychological models concerning POI classes were developed without any sustained or proved theoretical background or assurance, but rather by a common sense approach of how every concept

relates with the created psychological attributes. The main idea behind this technique was the correct appliance of a psychological component within UM and the RS rather than a deep cognitive analysis of both the user and POIs classes.

Following is a summarizing table which puts together all the strengths and drawbacks of the system, just presented.

Strengths	Weaknesses
Startup Quality	Recommender System Classical Issues
Transparency	Stereotype and Keyword Validation
Powerful Recommender System	User Area Elitism
On-the-fly Profile Evolution	Stereotype Rigid Conditions
Diverse Knowledge	POI Classes' Psychological Model Lack of Theoretical Assurance

Table 13 - System's Strengths and Weaknesses

5.2 Future Work

This subchapter accounts for a broader view of new developments, enhancement of current components and other innovative theories that can, in a future-level basis, be undertaken in order to yet evolve this system into a higher degree of quality. Due to lack of time, excessive distance from the thesis scope, those same tasks could not be developed throughout the duration of this thesis development. In this subchapter, some logical tasks, such as the elimination of the previously stated disadvantages, will not be presented due to obvious reasons. It must also be noted that this section accounts for tasks intended to be executed within the project in general, since UM and RS-related features were, obviously, all tried to be successfully and timely deployed within this very thesis.

Further database integration: although there was significant effort in integrating all data from the acquired real world database, there are still some related tasks undone. On one hand, the table regarding restaurants couldn't be integrated in time, on the other hand, long fields, such as textual descriptions, were not parsed. Finally, some interesting content information, such as city news and themed tours, which could propose themselves as a nice addition to the prototype, were also not explored.

Text mining morphological analysis: as it was explained in 3.3.7, the text mining algorithm acts in a very simple manner and gathers keywords by applying the following rules: the preference for many words keywords and the removal of stopwords. However, the execution of a more complete analysis of results in order to produce a set of syntax forms to search for in the algorithm could provide the component with a little more theoretical background to work from. Another alternative would be to get those same forms from another kind of official source.

Accessibility: as the system features a handicap component which, at this point, is only used to avoid recommendations that cannot be used by limited users, they can also be used for several other tasks. The most important of those is that all visual layers of the application must be gifted with accessibility features in order to be effectively used by people with physical disorders. Examples of such nature of the project might be audio descriptions of POIs, aided navigation of the portal, amongst others.

Multi-language: given the multi-cultural and international nature of the project, it would just be logical to ensure that, again, all visual layers of the application were gifted with multi-language support. As opposed to a common practice within this domain, a limit for the number of supported languages must not be applied. iIf there is a domain which demands for extreme options concerning languages that can be chosen, tourism certainly tops it. Also, this feature can and should be, associated with the previous issue, resulting, for example, in multi-language POIs descriptions.

Map exploitation: the use of virtual interactive maps, as well as other kinds of advanced multimedia, can be explored in many ways with the final purpose of increasing user experience and providing a more intuitive and fun manner of interaction with the system. Maps can be used, amongst others, to: (1) pinpoint user current lodging facilities; (2) change system outputted planned routes, in order to contemplate personal interests regarding, for example, landscapes or other aesthetics; (3) deploy of a map community where users can create notes, pictures and other kinds of shared information and (4) assist the other shapes of the project, which shall be developed and will be explained ahead, in features such as real-time adjustments, GPS-mode, etc.

Make use of demographic data: as with several components of the system which cannot be practically tested and evaluated with few or none real world use, demographic data could not, yet, be analyzed. However, upon reasonable testing phase, demographic data can be used by any data mining technique in order to analyze user profile space and to harvest some important information. Such effort may also be included as part of a much bigger data warehouse subproject within this one, executed to perform a more professional long-term analysis of how the system is being deployed. Moreover, as was already stated, this analysis can also be linked with stereotypes to ensure a more professional and sustained use of those, for example, by associating age and gender to stereotype condition lists.

Increase applicable physical area: one of the logical and most obvious paths in which the application can wide up its importance, regards the geographical applicable area concerning POIs and content in general. Apart from the possibility of the system being deployed with defined content to specific areas, therefore not needing much improvement, the application can also benefit greatly by the integration of other information sources. A system which manages content from more than a simple city, such as an entire region or country, must evolve its mechanisms in order to assimilate

other decision components regarding trips, tours, transportation means and route generation, amongst others.

Platform support and ubiquity: given the dynamic and real time nature of the project and tourism in general, the system demands for mobile support for a variety of reasons. First of all, and simply for completeness reasons, offering access to more platforms ensures more users connect and use the system. Then, as typical web-based applications can only assist in pre and post route phases, adding real-time support will significantly increase the overall system usefulness. Adjusting routes given last hour changes, helping users get back on track in case of being lost and offering on-the-fly commentaries on POIs are just some features that can be offered by gifting this project with ubiquity technology.

Taxonomy chain-free: probably using theories from the ontology domain, and as was already discussed in 3.3.6, slightly remove rigidness and strictness from the taxonomy, allowing keywords to be absorbed in the process. The objective, basically, is to create a middle ground between both components (taxonomy and keywords) so that POI classes and concepts can be evolved and maintained in a more pure manner. For example, using such approaches, a recently user-added keyword can automatically be linked with the system in order to provide more options regarding information retrieval, such as stating that movies and films are the same thing.

Social and Web 2.0 technology: apart from some already existent Web 2.0-related features such as keywords, comments and user friend management, there is much that can still be done in order to empower system's social capabilities. Namely:

I. Creation of a visual user profile to be viewed by the other users (at the present time profiles are private). This feature itself, is reasonably extensive, and, apart from the mentioned excellent use case of WAYN (see 2.6), can also be influenced by other social-based systems such as Hi5, Facebook, etc. Regarding the domain at hand (tourism), and besides all information that the referred systems already deal with, the profile can still show the following data: (1) all information regarding the past trips of the user, such as the trip data itself, comments, tours, multimedia, etc; (2) personal and social related information which are not yet contained in the current data model (such as weight, height, eye color, etc.); (3) a friend system with some more complex features such as friends met while on a trip or POI, groups of friends, etc and (4) the inclusion of several kinds of interactions between users, such as messages, comments, moods, Hi5's, forums, blogs, and so on;
II. General information about the last site interactions, such as new members, new added multimedia, new comments, new trips, new friend interactions and so on. This kind of information is generally posted on the application homepage and is generally called "Update Centre";

III. The user power to add several types of multimedia to POIs, with respective validation; these kinds of multimedia don't have to be necessarily linked with the user profile (such as in WAYN), but rather represent a communitarian help from users in order to increase the richness of POIs profiles;

IV. Gift the system with a much more complete and complex geographical platform, which requires the logical widening of the system geographical borders, including continent, country, region, city, etc. After that, several system components can behave differently depending on the respective kind of physical scope. For example, content filtering and recommendations might be based on region basis;

V. Include past information regarding tourism in the registration form in a very intuitive way, therefore increasing the profile completeness and user knowledge right at startup. WAYN achieves this with a very successful and simple method, although it does not extract any knowledge from that information, which would be extremely necessary;

VI. Add a range of techniques for searching for people, namely by text, personal attributes, geography, current activity or trip, personal tastes, etc. In a very straightforward and direct manner, Web 2.0 can be seen as a people RS, where the human factor plays an important role throughout all application processes;

VII. Add a variety of other techniques for recommending friends, based on personal tastes, travelling trends, tours taken, etc. As it was already explained in 3.5.2, there are already some developed theories regarding friend recommendation based on the UM deployed platform.

5.3 Summary

This research starts up from the realization that AI and machine learning reasoning haven't been fully explored in the tourism domain, particularly in what the tourist model is concerned. Tourism applications are indeed in a widespread state and very present and used in our current online big picture, but they still haven't taken the evolution step of intelligence, as well as usability. Users still have to perform several actions to achieve what they really want, and a substantial amount of smart reasoning from the system doesn't seem still present (Felfernig, *et al*., 2007) (Hannes, 2006). Moreover, and maybe more significantly, user information has been misused and falsely utilized. On one hand, current systems ask more information than they really use, which is a downside both usability-wise (users spend time submitting that information) and resource-wise (information is uselessly stored). On the other hand, UM and user adaptive-related mechanisms are used in a very poor manner, based on very simple and weak assumptions. It must also be noted that cases where more than one kind of UM technique is used are very rare, along with user reasoning using complex and heterogeneous data sources.

The final purpose of this kind of systems (tourism, movies, music, etc.) is, besides presenting information, filtering it and give recommendations, thus the need for a revolution concerning RSs and

UM, if true innovation is to be achieved. However, and although both technologies form the clear main content of this thesis, one of the most important acknowledgements was to notice that UM potential can be used in a variety of other situations, recommending-related or not.

When the user model presented here was starting to be built , it was clear that some specific kind of information had to be presented, which account for different perspectives of tourism and human context. First of all, knowledge representation formalisms that had been successfully used in tourism domain were smoothly accepted. However, in current systems, there is a lack of relation between those and the user itself, therefore decreasing UM theories in today's endeavors. On the other hand, and given the extremely social and personality related nature of tourism, it was also evident that using such means of user featuring would gift the system with an innovative knowledge paradigm which would make the difference. Those two kinds of mechanisms alone, plus some community-oriented components, started to compose a very strong basis for this project to ensure quality when it concerns adaptive reasoning and recommending facilities.

Although the UM architecture was very interesting and sustained, it was still unsure how it would evolve with time, how components would relate with each other and how would the user, the main profiting agent of the tourism context, see its experience really enhanced, customized and fastened up using the developed theories. It was in this phase that secondary tasks, contrasting to the core scope of thesis, were first envisioned to be developed. In order to verify the actual usefulness of all so far theoretical fundaments, a prototype portal was built. The main objective of having such kind of system tested was to control RS execution, UM evolutionary nature, on-the-fly application response and performance issues. However, certain assumptions would never be guaranteed given the unreal deployment of such superficial platform, such as stereotypes evolution and JMs visual representation. Plus, there was also not enough time to guarantee a stable user-compatible version of the prototype to be evaluated, since other project dependencies were at an earlier stage, such as route generation features and the portal itself. We remember the reader that the core work done in this thesis corresponds to an application layer very user unfriendly, and all upper layers were not developed in time. Therefore, regarding the thesis tasks presented in 1.1, the Evaluation step was the only one that could not be performed in time. Still, the RS was successfully applied with the chosen UM application method, to perform a social, inter-related and merged profit of the UM platform using of all user profile building blocks.

Another issue that also had to be tackled was the harvesting of user information featuring a minimal amount of effort. In the proposed system, a modeling platform that can be deployed in a very quick start way was successfully thought of, the result is a reasonably complex and intelligent user model acquired in a few seconds. The rest of the UM architecture functioning is entirely made automatically by the system, excepting when performed and perfected by the user himself (see 3.3.4). As a means of quickly presenting the major benefits of the devised project, next is a brief list of some reasons why all the work accounted in the devised project is thought of being of utmost importance to the current state of the art scene:

- Advanced and innovative UM;
- Hot-start results quality;
- Transparent UM functioning;
- On-the-fly user profile update;
- Behavioral-filtering introduction;
- Text-mining value-added algorithm;
- Multi-technique and heterogeneous RS;
- Controlled and uncontrolled knowledge.

Apart from all the previous advantages, all the future work contemplated and intended to be done in the near future (some of it already undergoing) will, with no doubt, elevate and increase interest and richness in the devised proposal.

From our point of view, the UM deployed platform, along with its constituent complex knowledge inference mechanisms are excellent basic elements for any tourism-focused system. In a certain point of view, and by making some obvious but minor adjustments, it can also be assumed, with no error, that this approach can be used in other domains as well. We acknowledge that every created representation or inference mechanism requires evolution and therefore artifacts are being created which try to cope with significant changes that may happen on the system, the user community and even in the tourism domain *modus operandi* itself. One of our most important goals is to create a decreasing user effort curve that finally will result in nothing but a few clicks for the user to achieve whatever he wanted in a tourist-based application. Plus, further negotiations are under way in order to increase data sources regarding, for example, public transportation means, as well as more serious kinds of deployments of the platform in formal entities such as government cultural systems, town hall representation, amongst others.

References

2009. *Amazon.* [Online] 2009. http://www.amazon.com/

2009. *CyberGuide Project Page*. [Online] 2009. http://www.cc.gatech.edu/fce/cyberguide/

2009. *FilmTrust.* [Online] 2009. http://trust.mindswap.org/FilmTrust/

2009. *Wikipedia.* [Online] 2009. http://en.wikipedia.org/wiki/Cathedral_of_Santa_Eulalia

2009. *Travel-Articles.* [Online] 2009. http://top-travel-articles.info/cloud.php?range=4

2009. *Porto Digital.* [Online] 2009. http://www.portodigital.pt/

2009. *Porto Turismo.* [Online] 2009. http://www.portoturismo.pt/

2009. *TripAdvisor.* [Online] 2009. http://www.tripadvisor.com/

2009. *DieToRecs.* [Online] 2009. http://dietorecs.itc.it/

2009. *TIP - A Mobile Tourism Information Provider.* [Online] 2009. http://isdb.cs.waikato.ac.nz/TIP

2009. *Heracles - Constraint Integration.* [Online] 2009. http://www.isi.edu/integration/Heracles/

2009. *WAYN.* [Online] 2009. http://www.wayn.com

2009. *Strategy Wiki.* [Online] 2009. http://strategywiki.org/wiki/The_Elder_Scrolls_III:_Morrowind/Character_creation

Agrawal, R. And Srikant, R. *Fast Algorithms for Mining Association Rules.* IBM Almaden Research Center. 1994

Berka, T. and Plößenig, M.. *Designing Recommender Systems for Tourism.* Salzburg Research. 2003

Burke, Robin. *Knowledge-based Recommender Systems.* To Appear in the Encyclopedia of Library and Information Science. Department of Information and Computer Science, University of California, Irvine. 1999

Casali, A., Godo, L. and Sierra, C. *Modeling Travel Assistant Agents: a graded BDI approach.* IFIP - International Federation for Information Processing, Vol. 217/2006, Springer Boston. 2005

Cattell, Raymond B. *Handbook for the sixteen personality factor questionnaire, "The 16 P.F. Test".* Institute for Personality and Ability Testing, 1962

Coelho, B. and Pereira, I. *Relatório do Trabalho Prático de Usabilidade de Sistemas Inteligentes - USAINTech.* 2008

Coelho, B. *Relatório de Projecto / Estágio - WebMeeting.* 2007

Cramer, H., Evers, V., Ramlal, S., Someren, M., Rutledge, L., Stash, N., Aroyo, L. & Wielinga, B. *The Effects of Transparency on Trust in and Acceptance of a Content-based Art Recommender*. User Modeling and User-Adapted Interaction. 5, Vol. 18. 455-496. 2008

Cunningham, P., Bergmann, R., Schmitt, S., Traphöner, R., Breen, S. and Smuth, B. *WEBSELL: Intelligent Sales Assistants for the World Wide Web.* Trinity College Dublin, Compuer Science, Technical Report. 2000

E., Horvitz. *Principles of Mixed-Initiative User Interfaces.* Microsoft Research. Conference on Human Factors in Computing Systems. Pages 159-166. 1999

Felfernig, A., et al. *A Short Survey of Recommendation Technologies in Travel and Tourism.* OEGAI Journal, Vol. 25, Number 7, Oesterreichische Gesellschaft fuer Artificial Intelligence. Pages 17-22. 2007

Fink, J. and Kobsa, A. *User Modeling for Personalized City Tours.* Artificial Intelligence Review. Vol. 18, Issue 1. Pages 33-74. 2002

Fleming M., Choen R. *Reasoning About Interaction in Mixed-Initiative AI Systems.* PhD Thesis. University of Waterloo. 2003

Gay, Geri and Hembrooke, Helene. *Activity-Centered Design: An Ecological Approach to Designing Smart Tools and Usable Systems*. MIT Press, 2004

Ghani, Rayid and Fano, Andrew. *Building Recommender Systems using a Knowledge Base of Product Semantics.* Accenture Technology Labs. 2nd International Conference on Adaptive Hypermedia and Adaptive Web Based Systems. 2001

Hannes, W. *e-Tourism: Impact of New Technologies.* Vienna University of Technology & Electronic Commerce Competence Center. 2006

Jennings, Andrew and Higuchi, Hideyuki. *A Personal News Service Based on a User Model Neural Network.* User Modeling and User-Adapted Interaction. Vol. 3, Number 1. Pages 1-25. 1991

Jung, Carl G. *Psychological Types.* Princeton University Press, 1971

Kobsa, A. *Generic User Modeling Systems.* User Modeling and User-Adapted Interaction. Vol. 11, Issue 1-2. Pages 49-63. Kluwer Academic Publishers. 2001

Kobsa, A. *User Modeling: Recent Work, Prospects and Hazards.* WG Knowledge-Based Information Systems. Department of Information Science. University of Konstanz. 1994

Kohonen, T. *Self-Organizing Maps.* Springer Series in Information Sciences. Vol. 30. Springer Berlin / Heidelberg. 2001

Luz, N., Anacleto R. *Tourism Mobile and Recommendation Systems.* 2009

Marques, N., Monteiro, A. and Barbas, J. *Utilização da Programação Declarativa para processamento do CETEMPúblico.* CENTRIA - DI FCT/UNL. 2008

Marreiros, M. *Um Sistema de Apoio à Tomada de Decisão em Grupo.* Dissertação de Mestrado em Gestão de Informação. Faculdade de Engenharia da Universidade do Porto. 2002

Martins, A. C., et al. *User Modeling in Adaptive Hypermedia Educational Systems.* Educational Technology & Society. Vol. 11, Number 1. 2008

Mathes, A. *Folksonomies - Cooperative Classification and Communication Through Shared Metadata.* Computer Mediated Communication. Graduate School of Library and Information Science. University of Illinois Urbana-Champaign. 2004

Oliver, P. J., Srivastava S. *The Big-Five Trait Taxonomy: History, Measurement, and Theoretical Perspectives.* Guilford Press, 1999

Pazienza, M. T. & Pennacchiotti, M. & Zanzotto, F. M. *Terminology Extraction: An Analysis of Linguistic and Statistic Approaches, Studies in Fuzziness and Soft Computing.* Springer Berlin / Heidelberg, 2005

Pazzani, M. J. and Billsus, D. *Content-based Recommendation Systems.* Lecture Notes in Computer Science. Vol. 4321. The Adaptive Web. Pages 325-341. Springer Berlin / Heidelberg. 2007

Porter, J. Watch and Learn: How Recommendation Systems are Redefining the Web. [Online] 2006. http://www.uie.com/articles/recommendation_systems/

Ramos, C. *Documentação de Inteligência Artificial e Sistemas de Apoio à Decisão.* 2007

Rich, E. *User Modeling via Stereotypes.* Readings in Intelligent User Interfaces. Pages 329-342. Morgan Kaufmann Publishers Inc. 1979

Schafer, J. Ben, Konstan, Joseph and Riedl, John. *Recommender Systems in E-Commerce.* Proceedings of the 1st ACM Conference on Electronic Commerce. Pages 158-166. 1999

Simcock, T., Hillenbrand, S. P., Thomas, B. H. *Developing a location based tourist guide application.* Conferences in Research and Practice in Information Technology Series. Vol.34. 2003

Tartaglione, A. *Hemisphere Asymmetry in Decision Making Abilities: An Experimental Study in Unilateral Brain Damage*. Guarantors of Brain, Vol. 114, Number 3. 1991

Tedlow, Richard S. *Exploring Marketing with Delta Airlines as a Case Study*. 2000

Towle, Brendon and Quinn, Clark. *Knowledge Based Recommender Systems Using Explicit User Models.* Knowledge Planet.com. 1999

Zhang, Yi and Koren, Jonathan. *Efficient Bayesian Hierarchical User Modeling for Recommendation Systems.* Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Pages 47-54. 2006

Zukerman, I. and Albrecht, David W. *Predictive Statistical Models for User Modeling.* User Modeling and User-Adapted Interaction. Vol. 11, Numbers 1-2. Pages 5-18. 2000

Attachment I - Presented Algorithms Code

Related to 4.3.1 - JM Propagation

```
/// <summary>
/// updates the jennings model's nodes, given an initial set of points of interest; changes
are then propagated
/// </summary>
/// <param name="poi_ids">all poi ids that were viewed / selected by the user</param>
/// <param name="action_type">the type of update to be done</param>
/// <returns>the outcome of the operation</returns>

static public string UpdateNodes(ArrayList poi_ids, string action_type)
{
        TOURSPLANDataContext database = new TOURSPLANDataContext();

        try
        {
                foreach (int poi_id in poi_ids)
                {
                        som_node current_poi_node = database.jn_nodes.SingleOrDefault(n =>
n.poi_id == poi_id);

                        if (current_poi_node == null)
                        {
                                CreateNode(poi_id, action_type);
                        }
                        else
                        {
                                jn_node jn_node = database.jn_nodes.SingleOrDefault(n =>
n.poi_id == poi_id);

                                if (action_type == "selection")
                                {
                                        jn_node.selection_value++;
                                }
                                else if (action_type == "navigation")
                                {
                                        jn_node.navigation_value++;
                                }
                        }
                }

                Propagate(action_type);

                database.SubmitChanges();

                return "1";
        }
        catch (Exception exception)
        {
                return exception.Message;
        }
}
```

Related to 4.3.2 - Likelihood Matrix Propagation

```
/// <summary>
/// updates a user likelihood matrix (increase or decrease values), given a poi class
/// </summary>
/// <param name="user_id">the user id to search for</param>
/// <param name="poi_class_id">the poi_class_id</param>
/// <param name="update_value">the amount of update to be done</param>
/// <returns>the outcome of the operation</returns>

static public string UpdateLikelihoodMatrix(int user_id, int poi_class_id, double
update_value)
{
        TOURSPLANDataContext database = new TOURSPLANDataContext();

        try
        {
                user_likelihood current_likelihood =
database.user_likelihoods.SingleOrDefault(l => l.user_id == user_id && l.poi_class_id ==
poi_class_id);

                if (current_likelihood == null)
                {
                        CreateLikelihoodByValue(user_id, poi_class_id, update_value);
                }
                else
                {
                        //save old value for equation

                         double old_value = current_likelihood.value;

                        //perform the update

                        current_likelihood.value = current_likelihood.value + update_value;

                        //perform a 3-simple rule to achieve the new ratio value, submitting
the maximum value to -1 and 1

                        double new_ratio = Math.Round((current_likelihood.value *
current_likelihood.ratio.Value) / old_value, 2);

                        if (new_ratio > 1)
                        {
                                new_ratio = 1;
                        }
                        else if (new_ratio < -1)
                        {
                                new_ratio = -1;
                        }

                        current_likelihood.ratio = new_ratio;

                        database.SubmitChanges();
                }

                PropagateLikelihoodMatrix(user_id, poi_class_id, update_value);

                return "1";
        }
        catch (Exception exception)
        {
                return exception.Message;
        }
}
```

Related to 4.3.3 - User <> Stereotype Suitability Checker and Stereotype Conditions Proposal

```
/// <summary>
/// gets the level of a stereotype suitability in relation to a user
/// </summary>
/// <param name="user_id">the user id to be matched against</param>
/// <param name="stereotype_id">the stereotype id</param>
/// <returns>the level of suitability</returns>

static public double GetSuitability(int user_id, int stereotype_id)
{
	List<stereotype_condition> stereotype_conditions = GetConditions(stereotype_id);

	List<system_parameter> system_parameters = Miscellaneous.GetParameters();

	ArrayList best_concepts = POIConcepts.GetBestByUser(user_id);

	ArrayList condition_matches = new ArrayList();

	int partial_matches = 0;

	//check for stereotype completeness

	foreach (stereotype_condition stereotype_condition in stereotype_conditions)
	{
		for (int i = 0; i < best_concepts.Count; i = i + 2)
		{
			if (stereotype_condition.poi_concept_id ==
int.Parse(best_concepts[i].ToString()))
			{
				partial_matches++;

				condition_matches.Add(stereotype_condition.poi_concept_id);
			}
		}
	}

	//get chosen concepts absolute importance

	foreach (int condition_match in condition_matches)
	{
		ArrayList current_condition_match_values =
POIConcepts.GetValueByUser(user_id, condition_match);

		//if a certain stereotype condition (which was already tested for best
likelihood) surpasses the importance threshold, that'll be enough for returning the minimum
ratio for accepting the stereotype, in order to avoid the cold-start problem, for example

		if ((double)current_condition_match_values[0] >= system_parameters.Single(p
=> p.name == "likelihood_matrix_threshold").value)
		{
			if ((double)current_condition_match_values[0] >= partial_matches /
stereotype_conditions.Count())
			{
				return system_parameters.Single(p => p.name ==
"stereotype_activation_threshold").value;
			}
		}
	}

	//if none of the found conditions successfully achieved the importance condition,
then a completeness ratio is returned

	return partial_matches / stereotype_conditions.Count();
}
```

```
/// <summary>
/// gets unforessen conditions that are found to be existent within a certain stereotype
/// </summary>
/// <param name="stereotype_id">the stereotype to be analysed</param>
/// <returns>the retrieved conditions</returns>

static public ArrayList GetProposedConditions(int stereotype_id)
{
        ArrayList proposed_conditions = new ArrayList();

        stereotype stereotype = GetByID(stereotype_id);

        List<system_parameter> system_parameters = Miscellaneous.GetParameters();

        ArrayList current_conditions = new ArrayList();

        foreach (stereotype_condition stereotype_condition in
stereotype.stereotype_conditions)
        {
                current_conditions.Add(stereotype_condition.poi_concept_id);
        }

        ArrayList available_conditions = POIConcepts.GetOpposite(current_conditions);

        foreach (int available_condition in available_conditions)
        {
                int total = 0;

                foreach (user_stereotype user_stereotype in stereotype.user_stereotypes)
                {
                        ArrayList best_concepts =
POIConcepts.GetBestByUser(user_stereotype.user_id);

                        for (int i = 0; i < best_concepts.Count; i = i + 2)
                        {
                                if (available_condition ==
int.Parse(best_concepts[i].ToString()))
                                {
                                        total++;
                                }
                        }
                }

                //if the condition surpasses the proposal threshold, it is proposed for
adition within that stereotype

                if ((double)total / stereotype.user_stereotypes.Count() >=
system_parameters.Single(p => p.name == "stereotype_conditions_proposal_threshold").value)
                {
                     proposed_conditions.Add(int.Parse(available_condition.ToString()));
                     proposed_conditions.Add((double)total /
stereotype.user_stereotypes.Count());
                }
        }

        return proposed_conditions;
}
```

Related to 4.3.4 - Likelihood Matrix Manual Update

```
/// <summary>
/// directly updates a poi class likelihood row using a new ratio
/// </summary>
/// <param name="user_id">the user id to be updated</param>
/// <param name="poi_class_id">the poi class id</param>
/// <param name="new_ratio">the new ratio to be set</param>
/// <returns>the retrieved records</returns>

static public string UpdateLikelihoodDirect(int user_id, int poi_class_id, double
new_ratio)
{
        TOURSPLANDataContext database = new TOURSPLANDataContext();

        try
        {
                user_likelihood user_likelihood =
database.user_likelihoods.SingleOrDefault(l => l.user_id == user_id && l.poi_class_id ==
poi_class_id);

                if (user_likelihood == null)
                {
                        CreateLikelihoodByRatio(user_id, poi_class_id, new_ratio);
                }
                else
                {
                        if (new_ratio != user_likelihood.ratio)
                        {
                                //perform a 3-simple rule to achieve the new "value" value

                                double new_value = Math.Round((new_ratio *
user_likelihood.value) / user_likelihood.ratio.Value, 2);

                                user_likelihood.ratio = new_ratio;
                                user_likelihood.value = new_value;

                                database.SubmitChanges();
                        }
                }

                return "1";
        }
        catch (Exception exception)
        {
                return exception.Message;
        }
}
```

Related to 4.3.5 - Psychological Model Evolution

```
/// <summary>
/// updates user psychological features by analysing the user patterns and the previous
value
/// </summary>
/// <param name="user_id">the user id to be updated</param>
/// <returns>the retrieved records</returns>

static public string UpdateAll(int user_id)
{
        TOURSPLANDataContext database = new TOURSPLANDataContext();

        List<psychological_attribute> psychological_attributes = (from p in
database.psychological_attributes select p).ToList<psychological_attribute>();

        foreach (psychological_attribute psychological_attribute in
psychological_attributes)
        {
                double overall_value = GetUserReasonedValue(user_id,
psychological_attribute.psychological_attribute_id);

                double current_value = GetUserValue(user_id,
psychological_attribute.psychological_attribute_id);

                double new_value = (double)((overall_value + (current_value * 20)) / 21);

                SetUserValue(user_id, psychological_attribute.psychological_attribute_id,
new_value);
        }

        return "1";
}
```

Related to 4.3.6 - Keyword Propagation

```
/// <summary>
/// updates a keyword value in relation to a certain user
/// </summary>
/// <param name="keyword_ids">the keywords to be updated</param>
/// <param name="user_id">the user id to be used</param>
/// <param name="update_value">the update value to be applied</param>
/// <returns>the success of the operation</returns>

static public string UpdateByUser(int keyword_id, int user_id, int update_value)
{
        TOURSPLANDataContext database = new TOURSPLANDataContext();

        try
        {
                user_keyword user_keyword = database.user_keywords.SingleOrDefault(k =>
k.user_id == user_id && k.keyword_id == keyword_id);

                if (user_keyword == null)
                {
                        CreateRelationWithUser(user_id, keyword_id, update_value);
                }
                else
                {
                        //save old value for equation

                        double old_value = user_keyword.value;

                        //perform the update

                        user_keyword.value = user_keyword.value + update_value;

                        //perform a 3-simple rule to achieve the new ratio value, submitting
the maximum value to -1 and 1

                        double new_ratio = Math.Round((user_keyword.value *
user_keyword.ratio) / old_value, 2);

                        if (new_ratio > 1)
                        {
                                new_ratio = 1;
                        }
                        else if (new_ratio < -1)
                        {
                                new_ratio = -1;
                        }

                        user_keyword.ratio = new_ratio;

                        database.SubmitChanges();
                }

                Update(keyword_id, update_value);

                return "1";
        }
        catch (Exception exception)
        {
                return exception.Message;
        }
}
```

Related to 4.3.7 - Keyword Extraction From Text and Sentence Parser

```
/// <summary>
/// the main class method, extracts keywords using textmining methodologies
/// </summary>
/// <param name="text">the text to be parsed</param>
/// <param name="language">the language</param>
/// <returns>the retrieved words</returns>

static public DataTable ExtractKeywordsByTextMiningAnalysis(string text, string language)
{
        //corrects grammar expressions that collide with some methods or facilitate parsing

        text = PerformReplacements(text, language);

        //divide the next into sentences

        string[] sentences = text.Split(GetPonctuation(),
StringSplitOptions.RemoveEmptyEntries);

        //removes ponctuation so that the ratings can look for the pattern correctly

        text = RemovePonctuation(text);

        //creates the final datatable

        DataTable results = new DataTable();

        results.Columns.Add(new DataColumn("keyword",
System.Type.GetType("System.String")));
        results.Columns.Add(new DataColumn("rating",
System.Type.GetType("System.Double")));
        results.Columns.Add(new DataColumn("rating_type",
System.Type.GetType("System.Int32")));
        results.Columns.Add(new DataColumn("count", System.Type.GetType("System.Int32")));
        results.Columns.Add(new DataColumn("frequency",
System.Type.GetType("System.Double")));
        results.Columns.Add(new DataColumn("length", System.Type.GetType("System.Int32")));

        //sends each sentence to be parsed by the keyword searcher dividing it by words

        char[] word_delimiters = { ' ' };

        foreach (string sentence in sentences)
        {
                AddResults(results,
ParseSentence(RemovePonctuation(sentence).Split(word_delimiters,
StringSplitOptions.RemoveEmptyEntries), language));
        }

        //sets each keyword rating

        foreach (DataRow current_keyword in results.Rows)
        {
                int rating = 1;

                string[] count = Regex.Split(text.ToLower(), " " +
current_keyword["keyword"].ToString().ToLower() + " |^" +
current_keyword["keyword"].ToString().ToLower() + " | " +
current_keyword["keyword"].ToString().ToLower() + "$|^" +
current_keyword["keyword"].ToString().ToLower() + "$");
```

```
                string[] parsed_keyword =
current_keyword["keyword"].ToString().Split(word_delimiters);

                foreach (string keyword_part in parsed_keyword)
                {
                        if (IsDomainKnowledge(keyword_part))
                        {
                                rating = 3;
                        }

                        if ((IsProperNoun(keyword_part) || HasNumber(keyword_part)) &&
rating < 3)
                        {
                                rating = 2;
                        }
                }

                current_keyword["count"] = count.Count() - 1;
                current_keyword["rating_type"] = rating;
                current_keyword["rating"] =
(int.Parse(current_keyword["rating_type"].ToString()) * 0.90) +
(int.Parse(current_keyword["count"].ToString()) * 0.05) +
(int.Parse(current_keyword["length"].ToString()) * 0.05);
        }

        //removes redundancies, i.e., keywords whose semantic value is surpassed by other
keywords

        ArrayList surpassed_value_keywords = new ArrayList();

        foreach (DataRow current_keyword_1 in results.Rows)
        {
                foreach (DataRow current_keyword_2 in results.Rows)
                {
                        if (current_keyword_1 != current_keyword_2)
                        {
                                string word_1 =
current_keyword_1["keyword"].ToString().ToLower();
                                string word_2 =
current_keyword_2["keyword"].ToString().ToLower();

                                if (word_2.Contains(word_1) &&
int.Parse(current_keyword_1["rating_type"].ToString()) == 1)
                                {
                                        if
(!surpassed_value_keywords.Contains(current_keyword_1))
                                        {

        surpassed_value_keywords.Add(current_keyword_1);
                                        }
                                }
                        }
                }
        }

        for (int i = 0; i < surpassed_value_keywords.Count; i++)
        {
                results.Rows.Remove((DataRow)surpassed_value_keywords[i]);
        }

        return results;
}
```

```
/// <summary>
/// parses a sentence for valuable keywords
/// </summary>
/// <param name="words">the words to be parsed</param>
/// <param name="language">the language</param>
/// <returns>the retrieved words</returns>

static private ArrayList ParseSentence(string[] words, string language)
{
        ArrayList results = new ArrayList();

        bool last_added = false;

        foreach (string word in words)
        {
                //lets all words be added, excluding stopwords and verb forms

                if (!IsStopWord(word, language) && !IsVerb(word))
                {
                        if (last_added == false)
                        {
                                //adds a new keyword

                                results.Add(word);

                                last_added = true;
                        }
                        else
                        {
                                //treats multi-word keywords, by merging strings

                                results[results.Count - 1] = results[results.Count -
1].ToString() + " " + word;

                                last_added = true;
                        }
                }
                else
                {
                        //resets multi-keyword forming

                        last_added = false;
                }
        }

        //performs number validations

        ParseNumbers(results);

        //performs residue validations

        ParseResidues(results);

        return results;
}
```

Related to 4.3.8 - Keyword Recommender System Results and Main Recommender System

```
/// <summary>
/// returns the group of pois related with user's keywords, by analysing most important
ones
/// </summary>
/// <param name="user_id">the user id to search for</param>
/// <returns>the found results</returns>

static public DataTable GetResultsByKeywords(int user_id, ArrayList poi_classes)
{
        //gets user's best keywords

        List<user_keyword> user_keywords = Keywords.GetBestByUser(user_id);

        //search for respective pois and rates each of them

        DataTable total_pois = new DataTable();

        total_pois.Columns.Add(new DataColumn("poi_id"));
        total_pois.Columns.Add(new DataColumn("component_rating",
System.Type.GetType("System.Double")));

        foreach(user_keyword user_keyword in user_keywords)
        {
                foreach (poi_keyword poi_keyword in user_keyword.keyword.poi_keywords)
                {
                        if (poi_classes.Contains(poi_keyword.poi.poi_class_id))
                        {
                                bool already_added = false;

                                foreach (DataRow current_poi in total_pois.Rows)
                                {
                                        if (int.Parse(current_poi["poi_id"].ToString()) ==
poi_keyword.poi_id)
                                        {
                                                already_added = true;

                                                if (user_keyword.ratio >
double.Parse(current_poi["component_rating"].ToString()))
                                                {
                                                        current_poi["component_rating"] =
user_keyword.ratio;
                                                }

                                                break;
                                        }
                                }

                                if (already_added == false)
                                {
                                        DataRow current_poi = total_pois.NewRow();

                                        current_poi["poi_id"] = poi_keyword.poi_id;
                                        current_poi["component_rating"] = user_keyword.ratio;

                                        total_pois.Rows.Add(current_poi);
                                }
                        }
                }
        }

        return total_pois;
}
```

```
/// <summary>
/// returns recommended pois based on a single user UM components
/// </summary>
/// <param name="user_id">the user id to search for</param>
/// <param name="order_by_poi_class">if the results are to be ordered by poi class</param>
/// <param name="poi_class_id">the poi class that the recommendation must be filterred
against</param>
/// <returns>the generated results</returns>

static public DataTable RecommendByUser(int user_id, bool order_by_poi_class, int
poi_class_id)
{
	//pre-filters poi categories to ensure more efficiency

	ArrayList poi_classes = new ArrayList();

	ArrayList accommodation_poi_classes =
POIClasses.BuildTopDownList(POIClasses.GetByName("Accommodations").poi_class_id);

	ArrayList eating_poi_classes =
POIClasses.BuildTopDownList(POIClasses.GetByName("Eating").poi_class_id);

	List<poi_class> all_poi_classes = POIClasses.GetAll();

	if (poi_class_id == 0)
	{
		//remove accommodation and eating poi classes

		foreach (poi_class poi_class in all_poi_classes)
		{
			if (!accommodation_poi_classes.Contains(poi_class.poi_class_id) &&
!eating_poi_classes.Contains(poi_class.poi_class_id))
			{
				poi_classes.Add(poi_class.poi_class_id);
			}
		}
	}
      else if (poi_class_id == POIClasses.GetByName("Places").poi_class_id || poi_class_id
== POIClasses.GetByName("Events").poi_class_id)
      {
		//remove accommodation and eating poi classes, and leaves only applicable
poi classes

		ArrayList applicable_poi_classes =
POIClasses.BuildTopDownList(poi_class_id);

		foreach (poi_class poi_class in all_poi_classes)
		{
			if (!accommodation_poi_classes.Contains(poi_class.poi_class_id) &&
!eating_poi_classes.Contains(poi_class.poi_class_id) &&
applicable_poi_classes.Contains(poi_class.poi_class_id))
			{
				poi_classes.Add(poi_class.poi_class_id);
			}
		}
	}
	else
	{
		//leave only applicable poi classes

		ArrayList applicable_poi_classes =
POIClasses.BuildTopDownList(poi_class_id);
```

```
                foreach (poi_class poi_class in all_poi_classes)
                {
                        if (applicable_poi_classes.Contains(poi_class.poi_class_id))
                        {
                                poi_classes.Add(poi_class.poi_class_id);
                        }
                }
        }

        //collects pois from all "data sources"

        DataTable likelihood_matrix_best_results =
GetResultsByLikelihoodMatrixBestClasses(user_id, poi_classes);

        DataTable stereotype_results = GetResultsByStereotypes(user_id, poi_classes);

        DataTable keyword_results = GetResultsByKeywords(user_id, poi_classes);

        DataTable psychological_results = GetResultsByPsychologicalAttributes(user_id,
poi_classes);

        DataTable likelihood_matrix_good_results =
GetResultsByLikelihoodMatrixGoodClasses(user_id, poi_classes);

        DataTable collaborative_filtering_results =
GetResultsByCollaborativeFiltering(user_id, poi_classes);

        DataTable neural_network_results = GetResultsBySelfOrganizingMap(user_id,
poi_classes);

        //merges results, cumulating component ratings and completeness ratings (the same
poi being recommended by different components)

        DataTable final_results = new DataTable();

        final_results.Columns.Add(new DataColumn("poi_id"));
        final_results.Columns.Add(new DataColumn("global_rating",
System.Type.GetType("System.Double")));
        final_results.Columns.Add(new DataColumn("completeness_rating"));
        final_results.Columns.Add(new DataColumn("user_rating",
System.Type.GetType("System.Double")));

        AddResultsByRating(final_results, likelihood_matrix_best_results, 4);

        AddResultsByRating(final_results, stereotype_results, 4);

        AddResultsByRating(final_results, keyword_results, 4);

        AddResultsByRating(final_results, psychological_results, 4);

        AddResultsByRating(final_results, likelihood_matrix_good_results, 3);

        AddResultsByRating(final_results, collaborative_filtering_results, 2);

        AddResultsByRating(final_results, neural_network_results, 1);

        //correct poi list for inconsistencies and handicaps and fills the user_rating
collumn, which will only be necessary next

        final_results = CorrectPOIList(final_results, user_id, poi_class_id);
```

```
        //sorts results firstly by the global rating which encompasses all components,
following by the number of components that positively suggested a poi and finally by the
user rating of that poi

        if (order_by_poi_class == true)
        {
                final_results.Columns.Add(new DataColumn("poi_class_id"));

                foreach (DataRow data_row in final_results.Rows)
                {
                        poi poi = POIs.GetByID(int.Parse(data_row["poi_id"].ToString()));

                        data_row["poi_class_id"] = poi.poi_class_id;
                }

                final_results.DefaultView.Sort = "poi_class_id DESC,global_rating
DESC,completeness_rating DESC,user_rating DESC";
        }
        else
        {
                final_results.DefaultView.Sort = "global_rating DESC,completeness_rating
DESC,user_rating DESC";
        }

        return final_results;
}
```

Attachment II - Overall Data Model